Pontifícia Universidade Católica
do Rio de Janeiro

Ney Barchilon

Enriquecimento de Dados com Base em Estatísticas de Grafo de Similaridade para Melhorar o Desempenho em Modelos de ML Supervisionados de Classificação

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre pelo Programa de Pós–graduação em Informática, do Departamento de Informática da PUC-Rio .

Orientador: Prof. Hélio Côrtes Vieira Lopes

Rio de Janeiro
Abril de 2024

Pontifícia Universidade Católica
do Rio de Janeiro

Ney Barchilon

Enriquecimento de Dados com Base em Estatísticas de Grafo de Similaridade para Melhorar o Desempenho em Modelos de ML Supervisionados de Classificação

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre pelo Programa de Pós–graduação em Informática da PUC-Rio . Aprovada pela Comissão Examinadora abaixo:

Prof. Hélio Côrtes Vieira Lopes
Orientador
Departamento de Informática – PUC-Rio

Prof. Marcos Kalinowski
PUC-Rio

Dr. Jefry Sastre Perez
PUC-Rio

Rio de Janeiro, 11 de Abril de 2024

Ney Barchilon

Bacharel em Ciências Estatísticas pela Escola Nacional de Ciências Estatísticas (ENCE).

Ficha Catalográfica

Barchilon, Ney

Enriquecimento de Dados com Base em Estatísticas de Grafo de Similaridade para Melhorar o Desempenho em Modelos de ML Supervisionados de Classificação / Ney Barchilon; orientador: Hélio Côrtes Vieira Lopes. – 2024.

246 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2024.

Inclui bibliografia

1. Informática – Teses. 2. Aprendizado de Máquina. 3. Grafo. 4. Similaridade. 5. Grafo por Similaridade. 6. Predição. 7. Enriquecimento de Dados. 8. Características Topológicas Grafo. 9. Redes Complexas. I. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. II. Título.

CDD: 004

Aos meus pais, irmãos e família
pelo apoio e encorajamento.

Agradecimentos

Ao meu orientador Professor Hélio Côrtes Vieira Lopes pelo estímulo e parceria para a realização deste trabalho.

À PUC-Rio, pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

Aos meus pais, pela educação, atenção e carinho de todas as horas.

Ao meu irmão, por todo estímulo, apoio, paciência e compreensão.

À Professora Tatiana Escovedo, pelo incentivo e apoio que me impulsionaram ao mestrado.

Aos meus colegas de disciplinas da PUC-Rio.

Aos professores que participaram da Comissão Examinadora pelas valiosas contribuições a esse trabalho.

A todos os professores e funcionários do Departamento pelos ensinamentos e pela ajuda.

A todos os amigos e familiares que de uma forma ou de outra me estimularam ou me ajudaram.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) Código de Financiamento 001.

Resumo

Barchilon, Ney; . **Enriquecimento de Dados com Base em Estatísticas de Grafo de Similaridade para Melhorar o Desempenho em Modelos de ML Supervisionados de Classificação**. Rio de Janeiro, 2024. 246p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A otimização do desempenho dos modelos de aprendizado de máquina supervisionados representa um desafio constante, especialmente em contextos com conjuntos de dados de alta dimensionalidade ou com numerosos atributos correlacionados. Neste estudo, é proposto um método para o enriquecimento de conjuntos de dados tabulares, fundamentado na utilização de estatísticas provenientes de um grafo construído a partir da similaridade entre as instâncias presentes neste conjunto de dados, buscando capturar correlações estruturais entre esses dados. As instâncias assumem o papel de vértices no grafo, enquanto as conexões entre elas refletem sua similaridade. O conjunto de características originais (FO) é enriquecido com as estatísticas extraídas do grafo (FG) na busca pela melhora do poder preditivo dos modelos de aprendizado de máquina. O método foi avaliado em dez conjuntos de dados públicos de distintas áreas de conhecimento, em dois cenários distintos, sobre sete modelos de aprendizado de máquina, comparando a predição sobre o conjunto de dados inicial (FO) com o conjunto de dados enriquecido com as estatísticas extraídas do seu grafo (FO+FG). Os resultados revelaram melhorias significativas na métrica de acurácia, com um aprimoramento médio de aproximadamente 4,9%. Além de sua flexibilidade para integração com outras técnicas de enriquecimento existentes, o método se apresenta como uma alternativa eficaz, sobretudo em situações em que os conjuntos de dados originais carecem das características necessárias para as abordagens tradicionais de enriquecimento com a utilização de grafo.

Palavras-chave

Aprendizado de Máquina; Grafo; Similaridade; Grafo por Similaridade; Predição; Enriquecimento de Dados; Características Topológicas Grafo; Redes Complexas.

Abstract

Barchilon, Ney; (Advisor). **Data Enrichment Based on Similarity Graph Statistics to Improve Performance in Classification Supervised ML Models**. Rio de Janeiro, 2024. 246p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The optimization of supervised machine learning models' performance represents a constant challenge, especially in contexts with high-dimensional datasets or numerous correlated attributes. In this study, we propose a method for enriching tabular datasets, based on the use of statistics derived from a graph constructed from the similarity between instances in the dataset, aiming to capture structural correlations among the data. Instances take on the role of vertices in the graph, while connections between them reflect their similarity. The original feature set (FO) is enriched with statistics extracted from the graph (FG) to enhance the predictive power of machine learning models. The method was evaluated on ten public datasets from different domains, in two distinct scenarios, across seven machine learning models, comparing prediction on the initial dataset (FO) with the dataset enriched with statistics extracted from its graph (FO+FG). The results revealed significant improvements in accuracy metrics, with an average enhancement of approximately 4.9%. In addition to its flexibility for integration with existing enrichment techniques, the method presents itself as a effective alternative, particularly in situations where original datasets lack the necessary characteristics for traditional graph-based enrichment approaches.

Keywords

Machine Learning; Graph; Similarity; Similarity Graph; Prediction; Data Enrichment; Topological Features Graph; Complex Networks.

Sumário

Lista de figuras

Lista de tabelas

Lista de Abreviaturas

AN – *Average Neighbor Degree*

BAGG – *Bagging: bootstrap aggregating*

BC – *Betweenness Centrality*

BCC – *Breast Cancer Coimbra Dataset*

CAR – *CAR Evaluation Dataset*

CC — *Closeness Centrality*

CL – *Clustering*

DC – *Degree Centrality*

DG – *Degree*

DM – *Dermatology Dataset*

DT – Árvore de Decisão

EC – *Eigenvector Centrality*

ETREE – Árvores Extras

FO – *Features* Originais

FG – *Features* do Grafo

HDH – *Heart Disease Hungarian Dataset*

HDHNI – *Heart Disease Hungarian Non-Invasive Dataset*

HT – *Hits*

IP – *Internet Protocol*

KNN – K-vizinhos mais próximos

LC – *Load Centrality*

LPA – *Label Propagation*

ML – *Machine Learning*

MLP – *Multi-Layer Perceptron*

NBG – *Naïve Bayes Gaussian*

PIMA – *Pima Indians Diabetes Dataset*

PR – *Pagerank*

RFOR – Florestas Aleatórias

RLOG – Regressão Logística

SVM – Máquina de Vetores de Suporte

TR – *Triangles*

VCB – *Vertebral Column Binary Dataset*

VCM – *Vertebral Column Multiclass Dataset*

WM – *Wine Multiclass Dataset*

WQ – *Wine Quality Dataset*

XGB - *Extreme Gradient Boosting*

Data is a precious thing and will last longer than the systems themselves.

Tim Berners-Lee, *A Framework for Web Science.*

1
Introdução

1.1
Contexto

Este estudo se dedica à proposição de um método para enriquecimento de conjuntos de dados tabulares, visando otimizar a eficácia de modelos de aprendizado de máquina supervisionados de classificação. A abordagem proposta baseia-se na utilização de estatísticas extraídas de um grafo (alguns autores se referem como características topológicas do grafo) cuja estrutura é construída a partir das instâncias do conjunto de dados, sendo as instâncias os vértices deste grafo e as arestas reflexo da similaridade (contendo características numérica e/ou categórica) entre essas instâncias.

As estatísticas derivadas do grafo de conhecimento representam uma ferramenta para aprimorar o desempenho de algoritmos de classificação e previsão, ao integrarem informações fundamentais sobre os dados em análise. Essas estatísticas se concentram primordialmente na análise da correlação entre os dados, destacando aspectos topológicos que oferecem perspectivas elucidativas sobre as inter-relações entre as instâncias de dados.

Trabalhos como os de Abdelmageed, N. (ABDELMAGEED, 2020) e Dong e Oyamada (DONG; OYAMADA, 2022), enriquecem conjuntos de dados tabulares por meio do acesso a diversas fontes, incluindo fontes externas. Esse enriquecimento de dados permite a obtenção de informações adicionais que tornam possível a construção do grafo de conhecimento a partir desses dados ampliados.

Outros pesquisadores, como Gulum, M. A. (GULUM, 2018) e Alharbi, A. e Alsubhi, K. (ALHARBI; ALSUBHI, 2021) empregam conjuntos de dados tabulares que já contêm informações sobre entidades e relacionamentos para construir seus grafos de conhecimento. Eles então extraem estatísticas desses grafos para enriquecer o conjunto de dados original, preparando-o para ser utilizado em modelos de aprendizado de máquina.

Os estudos de Zaki, N., Mohamed, E. e Habuza, T. (ZAKI; MOHAMED; HABUZA, 2021) e Albreiki, B., Habuza, T. e Zaki, N. (ALBREIKI; HABUZA; ZAKI, 2023) adotam uma abordagem que envolve a utilização de estatísticas derivadas do grafo para enriquecer conjuntos de dados tabulares, preparando-os para a aplicação em modelos de aprendizado de máquina (ML). Esses pesquisadores recorrem a métodos alternativos para estabelecer conexões

(arestas) na construção do grafo. Essas técnicas variam desde a captura de correlações estruturais entre os dados até a definição de proximidade entre as instâncias, estabelecendo a conexão dos vértices do grafo com base nessas técnicas e mediante um limite arbitrário.

O objetivo geral deste trabalho é propor e avaliar uma solução flexível e eficaz para o enriquecimento de conjuntos de dados tabulares baseado em estatísticas extraídas do grafo de similaridade, considerando distintamente as características categóricas e numéricas, visando melhorar a performance de modelos de aprendizado de máquina supervisionados de classificação.

A indagação central que norteia esta pesquisa é delineada pela seguinte pergunta: "É possível enriquecer conjuntos de dados tabulares de forma eficaz e flexível, empregando estatísticas derivadas de um grafo de similaridade entre suas instâncias, com o intuito de potencializar modelos de aprendizado de máquina supervisionados de classificação, mesmo que esse conjunto de dados não possua as características que possibilitam a construção direta do grafo sem acesso a outras fontes ?"

O método desenvolvido representa uma alternativa, particularmente em cenários onde os conjuntos de dados originais carecem das características necessárias para implementações tradicionais, as quais demandam identificação plena de entidades e/ou relacionamentos nestes. Ademais, a flexibilidade do método permite sua integração a outras técnicas de enriquecimento existentes. Para validar sua eficácia, foram empregados dez conjuntos de dados abrangendo diversas áreas do conhecimento, contemplando desde problemas com classes binárias até aqueles com múltiplas classes, número e tipos de atributos, e levando em consideração diferentes níveis de balanceamento de classe.

A avaliação do método envolveu a submissão dos conjuntos de dados enriquecidos a sete modelos de aprendizado de máquina em dois cenários distintos. O primeiro cenário contemplou avaliações com hiperparâmetros padrão, antes e após o enriquecimento, permitindo a análise da eficácia do método mediante apenas a inclusão das novas características ao conjunto de dados original. No segundo cenário, buscou-se avaliar os limites de performance do método em um espaço de hiperparâmetros ampliado com ajuste (*tunning*) dos modelos a esse novo espaço. A comparação com artigos selecionados da literatura científica enriqueceu ainda mais a análise, evidenciando a competitividade do método proposto.

A contribuição científica deste trabalho materializa-se na apresentação de um método que se destaca como alternativa para o enriquecimento de dados tabulares, visando aprimorar a performance de modelos de aprendizado de máquina supervisionados de classificação. A possibilidade de integração em

pipelines de predição, aliada à combinação de técnicas estabelecidas, confere-lhe a característica de ferramenta acessível para pesquisadores e cientistas de dados.

Os resultados das avaliações sugerem melhorias importantes na performance, dos modelos de aprendizado selecionados sobre os conjuntos de dados trabalhados, com incrementos de 4,9% na média da métrica de acurácia em ambos os cenários avaliados. A variabilidade observada entre modelos e conjuntos de dados, nos dois cenários, destaca a sensibilidade do método às características específicas de cada contexto. Da mesma forma, o consumo de tempo para o cálculo das estatísticas do grafo *betweenness centrality*, *load centrality* e *clustering*, deve ser um ponto de atenção, pois aumentaram significativamente o tempo total do *pipeline* do método.

A comparação com outros estudos anteriores, que utilizaram diferentes técnicas de melhoria de performance sobre os mesmos conjuntos de dados, revelou resultados positivos na maioria dos casos, reforçando a competividade do método proposto.

Em síntese, este estudo confirma que o método de enriquecimento apresentado é eficaz e competitivo em relação a outras técnicas, evidenciando o potencial como mais uma alternativa para impulsionar a performance de modelos de aprendizado de máquina supervisionados de classificação em diferentes domínios.

1.2 Questões de Pesquisa

O presente estudo procura investigar o impacto do enriquecimento de dados utilizando métricas de grafo de similaridade de suas instâncias na performance de modelos de predição em diferentes conjuntos de dados.

São delineadas as seguintes questões de pesquisa:

Q1. Como a utilização de métricas de grafo pode ser empregada para enriquecer conjuntos de dados que não possuam características explícitas que possam ser utilizadas para relacionar as instâncias para construção do grafo ?

Q2. Em que medida a performance dos modelos de predição é afetada pela adição de características derivadas de estatísticas extraídas do grafo de similaridade do conjunto de dados, considerando a utilização de conjuntos de dados de diferentes domínios de conhecimento, perfis de classe (binária ou multiclasse), balanceamento, quantidade de atributos, entre outros ?

Q3. Como a performance dos modelos de predição, após o enriquecimento de características derivadas de estatísticas extraídas do grafo de similaridade do conjunto de dados, se compara à performance de modelos reportados em es-

tudos anteriores que utilizaram diferentes técnicas de melhoria de performance sobre os mesmos conjuntos de dados ?

Essa pesquisa visa responder a essas questões por meio da construção de grafo com base nesses conjuntos de dados tabulares e sua(s) matriz(es) de similaridade entre as instâncias, extração de estatísticas relevantes, a integração dessas novas características ao conjunto de dados original e a submissão destes aos modelos de aprendizados de máquina para predição.

A avaliação dos modelos de predição será conduzida em um conjunto diversificado de dados tabulares com diferentes características estruturais, permitindo uma análise abrangente da eficácia do enriquecimento com métricas de grafo em diversos cenários de predição.

Com a investigação dessas questões, este estudo visa contribuir para o avanço do conhecimento em enriquecimento de dados utilizando estatísticas de grafo e fornecer *insights* relevantes para a comunidade de pesquisadores e profissionais de ciência de dados na aplicação dessa abordagem em problemas reais de predição em diferentes domínios de conhecimento.

1.3 Justificativa

A crescente demanda por aprimoramento de modelos de aprendizado de máquina supervisionados tem impulsionado a busca por métodos eficazes de enriquecimento de conjuntos de dados tabulares. Os métodos atuais de enriquecimento utilizando grafo, em grande parte, dependem da identificação clara de entidades no conjunto de dados original para proporcionar resultados significativos, isto é, muitas abordagens existentes pressupõem que as entidades (por exemplo, objetos, eventos) no conjunto de dados são claramente identificáveis e relacionáveis. No entanto, essa abordagem apresenta limitações, especialmente quando lidamos com conjuntos de dados que não possuem características ideais para a identificação precisa de entidades e relacionamentos entre suas características.

O enriquecimento de dados por meio de grafos, embora muito promissor, também enfrenta desafios relacionados à necessidade de identificação de entidades e à clareza nas características do conjunto de dados original que podem estabelecer relações significativas entre essas entidades. Muitas vezes, esses métodos exigem uma estruturação específica do conjunto de dados, o que pode não ser possível em situações em que as características não são explicitamente vinculadas a entidades ou não possuem características que os relacionem.

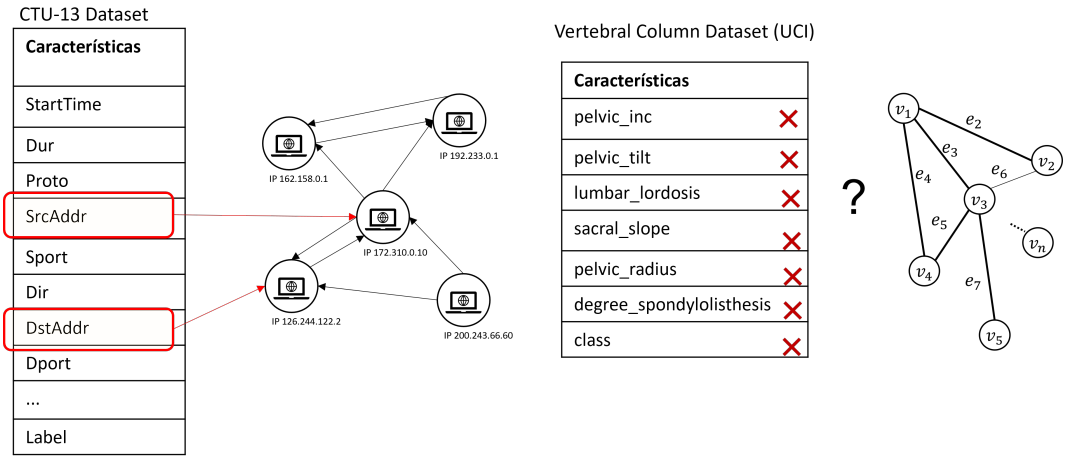
Por exemplo, observando os conjuntos de dados “CTU-13” (GARCíA et al., 2014), que contém fluxos reais de tráfego de rede, utilizado, entre outros, no

trabalho “Botnet Detection Approach Using Graph-Based Machine Learning,” de Alharbi, A. e Alsubhi (ALHARBI; ALSUBHI, 2021) e o conjunto de dados Vertebral Column (BARRETO; NETO, 2011) que contém características obtidas de radiografias da coluna vertebral, também muito utilizado em diversas pesquisas na área de aprendizado de máquina, com seis características numéricas.

No trabalho de Alharbi, A. e Alsubhi, K. foi possível criar o grafo a partir do conjunto de dados CTU-13, pois este possui características que permitem a composição dos vértices e arestas do grafo, conforme a Figura 1.1. Nesse caso, as entidades são os equipamentos que geram fluxos de rede, representados pelo endereço IP como vértices do grafo e as arestas ligariam o par de vértices pelas carcaterísticas “SrcAddr” e “DstAddr”, que representam o endereço IP de Origem e Destino, respectivamente.

Por outro lado, o conjunto "Vertebral Column"não possui características que permitam a criação direta do grafo através da identificação das entidades e relaciomentos entre essas entidades, logo, não teria acesso a possibilidade de enriquecimento do conjunto de dados com características extraídas do grafo.

Figura 1.1: Conjuntos de Dados CTU-13 e Vertebral Column



Alguns autores desenvolveram trabalhos que buscam suprir essa lacuna com o enriquecimento das características nesse conjunto de dados através do acesso a bases externas, enriquecendo as semânticas, metadados, incrementando características, instâncias e até descobrindo relações entre essas características e instâncias, mas esse é um procedimento que demanda alto conhecimento e esforço.

Neste contexto, a justificativa para o desenvolvimento do método proposto nesta pesquisa torna-se relevante. O método em questão oferece uma abordagem ao contornar a necessidade estrita de identificação clara de entida-

des nas características do conjunto de dados original e/ou do relacionamento entre essas entidades e só utilizando os dados que já existem no conjunto de dados, embora não impeça o enriquecimento prévio por outros métodos. Ao extrair estatísticas de um grafo construído com base na similaridade entre instâncias, o método proposto abre novas perspectivas para o enriquecimento de dados tabulares.

Portanto, algumas das vantagens distintivas do método proposto é sua independência da identificação explícita de entidades e/ou relacionamentos no conjunto de dados original e o fato de que só se utiliza de informações que já constam desse conjunto de dados. Isso não apenas simplifica o processo de aplicação, mas também amplia significativamente o escopo de conjuntos de dados nos quais o enriquecimento pode ser efetivamente realizado.

Além disso, o método proposto se destaca por sua capacidade de ser empregado em conjunto com outras soluções de enriquecimento existentes, oferecendo uma abordagem complementar e flexível para melhorar a qualidade dos dados e, por conseguinte, a performance dos modelos de aprendizado supervisionado de classificação.

Dessa forma, esta pesquisa busca endereçar lacunas importantes nos métodos atuais de enriquecimento de dados tabulares baseados em estatísticas do grafo, promovendo uma solução de amplo alcance para pesquisadores na academia, assim como para cientistas de dados. Ao remover as restrições impostas pela necessidade de identificação explícita de entidades e relacionamentos para a construção do grafo e a utilização de características que já existem nesse conjunto de dados para tanto, a pesquisa visa proporcionar um método alternativo acessível, aplicável em uma variedade de cenários, e capaz de contribuir para o avanço no campo do enriquecimento de dados tabulares para aprendizado de máquina.

1.4 Objetivos

Este estudo tem como objetivo geral desenvolver, aplicar e avaliar um método para o enriquecimento de conjuntos de dados tabulares, por meio da incorporação de estatísticas extraídas de um grafo de similaridade construído a partir de características do próprio conjunto de dados. A proposta busca aprimorar a performance de modelos de aprendizado de máquina supervisionados de classificação, oferecendo uma abordagem flexível e eficaz para lidar com conjuntos de dados tabulares que não apresentam características ideais para métodos convencionais de enriquecimento utilizando grafo.

1.4.1 Objetivos Específicos

1. Aplicar Métricas de Similaridade: Propor e testar métricas de similaridade na busca por capturar a relação entre as instâncias do conjunto de dados, considerando diferentes tipos de atributos e suas interrelações

2. Construir um Grafo de Similaridade: Desenvolver e implementar uma técnica para construção de um grafo de similaridade, tendo as instâncias do conjunto de dados original como vértices e as arestas refletindo a similaridade entre essas instâncias.

3. Integrar Estatísticas no Conjunto de Dados: Estabelecer e implementar um método eficiente para a extração e seleção de estatísticas a partir do grafo construído, incorporando-as de forma coerente ao conjunto de dados tabular original.

4. Avaliar a Eficácia do Método Proposto: Realizar experimentos extensivos com dez conjuntos de dados de diversas áreas de conhecimento, contemplando diferentes características, tais como número de atributos, tipos de atributos, e níveis de balanceamento.

5. Avaliar a Aplicabilidade em Diferentes Cenários: Submeter e avaliar os conjuntos de dados enriquecidos a sete modelos de aprendizado de máquina supervisionados em dois cenários distintos. O primeiro cenário com hiperparâmetros padrão, antes e após o enriquecimento, buscando a análise da eficácia do método mediante apenas a inclusão das novas características ao conjunto de dados original. No segundo cenário, com um espaço de hiperparâmetros ampliado com ajuste (*tunning*) dos modelos a esse novo espaço, buscando avaliar os limites de performance do método em cada conjunto de dados e modelo.

6. Analisar a Variabilidade do Método: Investigar a variabilidade do desempenho do método em diferentes contextos, identificando as condições em que se destaca e aquelas em que pode ser aprimorado.

7. Comparar com Métodos Existentes: Efetuar comparações dos resultados alcançados entre o método proposto e outras técnicas de melhoria de performance presentes na literatura com os mesmos conjuntos de dados, considerando métricas de desempenho e limites de performance.

Ao alcançar esses objetivos específicos, busca-se consolidar uma contribuição significativa para o campo científico, oferecendo aos profissionais de ciência de dados e academia uma alternativa de método eficaz e flexível para aprimorar a performance de modelos de aprendizado de máquina supervisionados de classificação em uma variedade de cenários.

Embora existam várias aplicações para o enriquecimento de conjuntos tabulares pela inclusão das estatísticas de grafo de similaridade em modelos

não supervisionados ou em modelos supervisionados de regressão, essa pesquisa se concentra nos modelos supervisionados de classificação para investigar a eficácia das estatísticas de grafo de similaridade em melhorar o desempenho de modelos de aprendizado de máquina (ML).

Essa escolha é justificada pela amplitude de aplicação desses modelos em problemas reais. Focar nesses modelos permite uma análise mais detalhada da integração e otimização das estatísticas de grafo, impulsionando o avanço do conhecimento nesse campo de estudo. Essa abordagem garante uma investigação coesa e detalhada, evitando dispersão e permitindo uma análise aprofundada das implicações das estatísticas de grafo de similaridade em modelos supervisionados de classificação.

1.5 Estrutura do Documento

Esse trabalho está organizado de forma a abordar de maneira abrangente o tema do enriquecimento de dados tabulares com estatísticas extraídas de grafos de similaridade e sua contribuição para melhora na performance de modelos de ML supervisionados de classificação em diferentes domínios e cenários.

O primeiro capítulo introduz o contexto da pesquisa, delineando o problema, justificativa e objetivos, tanto geral quanto específicos.

O segundo capítulo constitui a fundamentação teórica, explorando desde a engenharia de características até os modelos de aprendizado de máquina supervisionados, passando por conceitos importantes como teoria dos grafos, redes complexas, medidas de similaridade e distância, estatísticas do grafo, seleção de características e avaliação de modelos. Inclui também uma revisão da literatura, apresentando os trabalhos relacionados, onde foi realizada uma revisão da literatura para situar a presente pesquisa no contexto mais amplo do enriquecimento de dados tabulares.

O terceiro capítulo detalha o método proposto, incluindo visão geral, fluxo básico e etapas do processo, enquanto o quarto capítulo descreve os experimentos, abrangendo conjuntos de dados, cenários de avaliação, parâmetros, métricas de desempenho e configurações do ambiente de experimento.

Os resultados dos experimentos são apresentados no quinto capítulo, subdivididos pelos dois cenários propostos e comparados com pesquisas anteriores. O trabalho conclui-se no sexto capítulo, destacando as principais descobertas e apontando direções para trabalhos futuros. Essa estrutura permite uma compreensão gradual e aprofundada do desenvolvimento, experimentação e resultados do método proposto.

2
Fundamentação Teórica

Este capítulo aborda conceitos fundamentais relacionados aos temas que servem como alicerce teórico deste estudo. Esses tópicos são essenciais para a compreensão das técnicas empregadas na formulação da metodologia proposta. São discutidos conceitos ligados à teoria dos grafos, grafos de similaridade, métricas de similaridade e distância, bem como estatísticas relacionadas a grafos, modelos de aprendizado de máquina e medidas de avaliação. Portanto, essa exposição tem como objetivo fornecer um conjunto de conceitos essenciais para uma compreensão sólida das abordagens propostas neste trabalho.

2.1
Enriquecimento de Dados Tabulares

O enriquecimento de dados é um procedimento que envolve a adição de informações contextualmente pertinentes a conjuntos de dados já coletados, resultando em uma melhoria na qualidade e utilidade desses dados. Esse processo frequentemente implica na incorporação de dados provenientes de fontes suplementares, com o objetivo de enriquecer e aprimorar as informações disponíveis (KNAPP; LANGILL, 2015, p. 323). Em outras palavras, o enriquecimento de dados é o processo de aprimorar um conjunto de dados existente com informações adicionais de fontes internas ou externas.

Esse processo abrange diversas etapas, desde a validação e limpeza de informações até a expansão do conjunto de dados com elementos contextuais relevantes, permitindo uma análise mais completa e informada. Por meio do enriquecimento de dados, procura-se aprimorar a precisão, a confiabilidade e o valor dos dados.

O enriquecimento de dados tabulares são um caso particular desse domínio.

Os dados tabulares são o formato mais diversificado de representação de dados, abrangendo uma ampla variedade de domínios (HARARI; KATZ, 2022, p. 1577).

No entanto, esse formato de dados geralmente carece de informações contextuais que poderiam tornar sua análise mais eficaz. Para superar essa limitação, cientistas de dados utilizam a engenharia de características (do inglês *feature engineering*). Isso envolve a aplicação de transformações nas características (ou atributos) de dados já existentes para criar representações adicionais dos dados (HARARI; KATZ, 2022, p. 1577).

2.2 Engenharia de Características

O campo do aprendizado de máquina tem como objetivo ajustar modelos matemáticos aos dados para extrair *insights* e realizar previsões. Nesse contexto, os modelos utilizam características como entradas, onde uma característica é definida como uma representação numérica de um aspecto específico dos dados brutos. Essas características desempenham um papel intermediário entre os dados brutos e os modelos no processo de aprendizado de máquina. A Engenharia de Características é fundamental nesse domínio, envolvendo a extração e transformação de características a partir dos dados brutos, tornando-as formatos adequados para alimentar os modelos de aprendizado de máquina (ZHENG; CASARI, 2018, p. 3).

A qualidade dessas características desempenha um papel fundamental na melhoria do desempenho geral do modelo de aprendizado de máquina. Além disso, a escolha e criação de características dependem significativamente do problema específico que está sendo abordado (SARKAR; BALI; SHARMA, 2018, p. 182).

Portanto, as características ideais variam de acordo com a natureza e os requisitos do problema subjacente. Dessa forma, a engenharia de características envolve a seleção ou criação cuidadosa de características relevantes e informativas, adaptadas ao contexto do problema, a fim de otimizar o desempenho do modelo de aprendizado de máquina.

Os profissionais de ciência de dados frequentemente confrontam o desafio de aprimorar a acurácia das previsões quando confrontados com dados tabulares que se mostram insuficientes para tal finalidade. A escassez de características (recursos ou atributos) é uma constante em cenários práticos de análise de dados (DONG; OYAMADA, 2022, p. 1).

O processo de extração e engenharia de característica (ou recurso ou atributo) é uma etapa fundamental em todo o ciclo de aprendizado de máquina. A qualidade das características que descrevem representações adequadas dos dados, desempenha um papel central na construção de modelos eficazes. Em muitos casos, a eficácia do modelo é determinada não tanto pelos algoritmos, mas pela qualidade das características contidas neste. Simplificando, boas características conduzem à construção de bons modelos. Os cientistas de dados geralmente dedicam a maior parte do tempo, entre 70% e 80%, ao processamento de dados, organização e engenharia de características (ou recursos ou atributos) para desenvolver modelos de aprendizado de máquina (SARKAR; BALI; SHARMA, 2018, p. 181).

Segundo Geron, A. (GERON, 2019, p. 27) o processo de engenharia

de característica envolve a Seleção de Características (*feature selection*, em inglês) que é a seleção das características mais úteis para treinar entre as características existentes; Extração de Características (*feature extraction*, em inglês) que procura combinar características existentes para produzir uma característica mais útil; e a Criação de Novas Características através da coleta de novos dados.

Da mesma forma Sarkar, D *et al.* (SARKAR; BALI; SHARMA, 2018, p. 177) acrescenta mais uma etapa no processo de engenharia de característica que é o escalonamento de características (*feature scaling*, em inglês) quando define três áreas principais no fluxo de trabalho de engenharia de características: Extração e engenharia de características, escalonamento de características (*feature scaling*, em inglês) e seleção de características.

O escalonamento de características e a seleção de características serão abordados em Seções mais a frente nesse trabalho. Nesse momento, vamos nos concentrar na Criação de Novas Características. Antes, porém, vamos definir o que são conjuntos de dados tabulares.

O leitor que desejar se aprofundar no tema de engenharia de características e todas as suas técnicas e variações, sugerimos as referências aqui citadas, em especial, as obras "Practical Machine Learning with Python: A Problem-Solver's Guide to Building Real-World Intelligent Systems" de Sarkar, D *et al.* (SARKAR; BALI; SHARMA, 2018), bem como o trabalho de Zheng, A. e Casari, A. (ZHENG; CASARI, 2018) "Feature engineering for machine learning: principles and techniques for data scientists.".

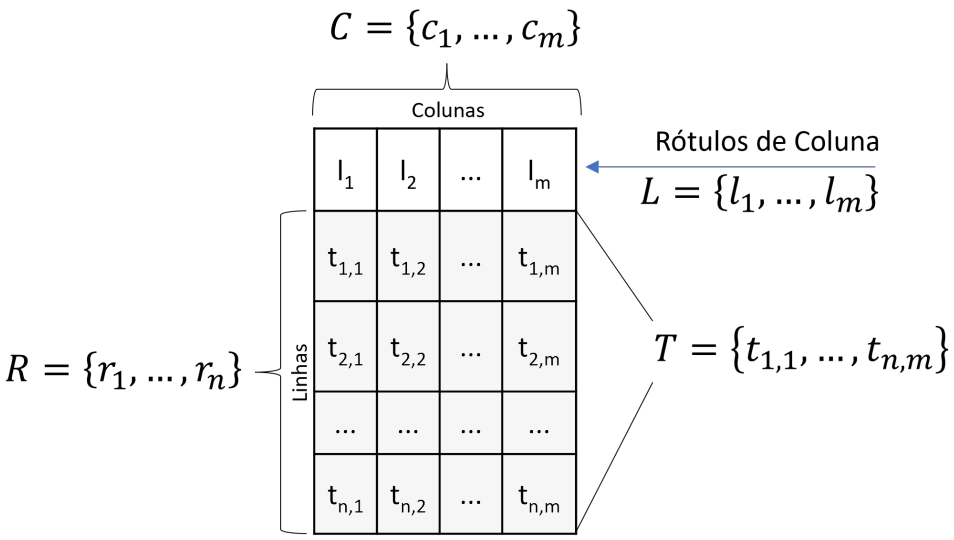
A definição formal de dados implica em uma coleção de variáveis qualitativas e/ou quantitativas que contêm valores obtidos a partir de observações. Esses dados são tipicamente medidos e coletados de várias observações, sendo armazenados em sua forma bruta, prontos para processamento e análise posterior. Esses dados podem assumir uma estrutura tabular, com linhas representando observações e colunas representando atributos, ou podem ser não estruturados, como dados textuais não formatados. Um conjunto de dados (*dataset*, em inglês) pode ser definido como uma coleção de dados (SARKAR; BALI; SHARMA, 2018, p. 178).

Formalizando, Jiménez-Ruiz, E., *et al* (JIMÉNEZ-RUIZ et al., 2020, p. 516) propõe que os dados tabulares podem ser vistos como um conjunto de colunas $C = \{c_1, c_2, ..., c_n\}$, um conjunto de linhas $R = \{r_1, r_2, ..., r_n\}$, ou a matriz de células $T = \{t_{1,1}, ..., t_{n,m}\}$, onde uma coluna $c_k = \{t_{1,k}, ..., t_{n,k}\}$ e uma linha $r_k = \{t_{k,1}, ..., t_{k,m}\}$ são tuplas de células, m representa o número de colunas no conjunto de dados e n o número de linhas (ou instâncias ou exemplos ou amostras) do conjunto de dados tabulares. A Figura 2.1 representa

essa estrutura.

Zhang , S e Balog, K. (ZHANG; BALOG, 2017, p. 3) definem que, adicionalmente, essa tabela pode conter uma linha superior especial no topo, onde se encontram os títulos das colunas. Isto posto, a tabela passará a conter n+1 linhas, onde a primeira linha contém os títulos das colunas seguidas por n linhas regulares com o conteúdo e $L = \{l_1, l_2, ..., l_n\}$, é a lista de rótulos de cabeçalho de coluna.

Figura 2.1: Conjunto de Dados Tabular com Rótulo de coluna



Nesse caso, assumimos que todas as colunas e linhas são uniformes em tamanho, admitindo a presença de células $t_{i,j}$ com valores nulos. Em dados tabulares arbitrários, diferentemente do que ocorre nas tabelas relacionais, é admissível a ausência de nomes de colunas e de identificadores de linhas, usualmente representados como chaves primárias. Nas tabelas da Web e nas tabelas relacionais, as linhas tipicamente delineiam uma entidade específica, enquanto, em dados tabulares arbitrários (por exemplo, em arquivos CSV amplamente utilizados na área de ciência de dados), a presença de uma entidade principal em cada linha pode não ser um requisito obrigatório (JIMÉNEZ-RUIZ et al., 2020, p. 516).

A geração de novas características em conjuntos de dados tabulares, resultando em um conjunto de dados enriquecido, é um procedimento que pode ser realizado de diversas maneiras, frequentemente não sendo mutuamente exclusivas. Uma abordagem envolve a combinação de características preexistentes, com o intuito de criar características mais informativas, utilizando os próprios dados do conjunto de dados original. Simultaneamente, o enriquecimento do conjunto de dados pode ocorrer através da incorporação de informações exter-

nas, provenientes de fontes adicionais, expandindo assim o contexto e a riqueza dos dados disponíveis.

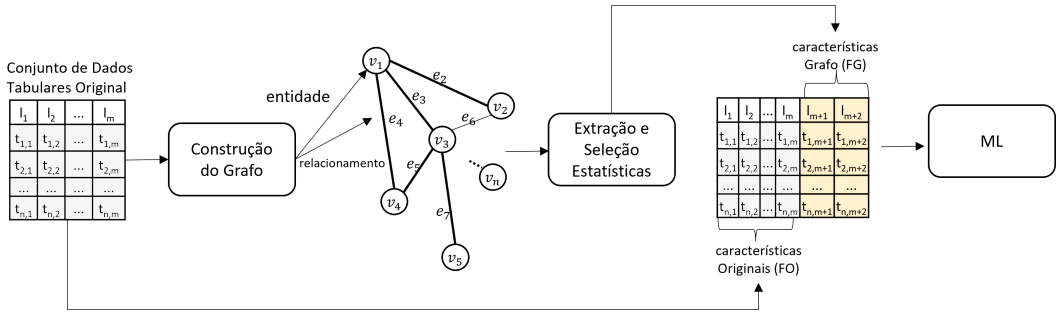
Sarkar, D *et al.* (SARKAR; BALI; SHARMA, 2018, p. 177) apresentam alguns exemplos de características que podem ser criadas pela combinação de características preexistentes para produzir uma característica mais informativa:

- Deduzir a idade de uma pessoa a partir da data de nascimento e da data atual;
- Obter a contagem média e mediana de visualizações de músicas e videoclipes específicos;
- Extração de contagens de ocorrências de palavras e frases de documentos de texto;
- Extração de informações de pixel de imagens brutas;
- Tabulação de ocorrências de diversas notas obtidas pelos alunos.

Outro procedimento de geração de novas características pela utilização de características preexistente é a utilização de estatísticas derivadas de grafos criados a partir desses dados originais. Conforme demonstrado na Figura 2.2 esse processo se baseia na construção de um grafo, direcionado ou não, em que se identifica a entidade e relacionamentos entre as características originais que serão os vértices (nós) e arestas desse grafo, respectivamente.

No contexto de conjuntos de dados tabulares e enriquecimento de dados, o termo 'entidade' refere-se a um objeto ou item específico que está sendo representado pelos dados em uma linha ou observação na tabela (ZHANG; BALOG, 2017, p. 1). Em outras palavras, uma entidade pode ser uma instância única de dados que possui características ou atributos associados a ela. Por exemplo, em um conjunto de dados de clientes de uma loja, cada linha pode representar uma entidade única, ou seja, um cliente específico, com suas características como nome, idade, histórico de compras etc. Portanto, no enriquecimento de dados, a noção de entidade está relacionada à representação e manipulação de objetos individuais ou instâncias dentro do conjunto de dados tabulares. Os relacionamentos são baseados em características no conjunto de dados cujos elementos permitam a conexão entre essas entidades.

Figura 2.2: Criação e Extração Estatísticas de Grafo



Esse método e suas variações têm alcançado resultados em trabalhos aplicados para diferentes domínios, tais como "Botnet Detection Approach Using Graph-Based Machine Learning," de Alharbi, A. e Alsubhi, K. (ALHARBI; ALSUBHI, 2021), "Application of Support Vector Machine Modeling and Graph Theory Metrics for Disease Classification" de Rudd, J. (RUDD, 2017) e "Graph-based Feature Enrichment for Online Intrusion Detection in Virtual Networks" de Sanz, Igor Jochem e Duarte, Otto Carlos M. B. (SANZ; DUARTE, 2019), entre outros. No entanto, como vimos acima, esse método se baseia na premissa da identificação plena, entre as características disponíveis no conjunto de dados tabular original, das entidades e de relacionamento entre elas para construção do grafo. Nem sempre os conjuntos de dados tabulares originais possuem essa informação em seu conteúdo, o que pode dificultar a adoção do método. Principalmente com a adoção da proteção e anonimização dos dados, essa dificuldade se torna presente.

O enriquecimento de dados tabulares a partir de dados externos tem sido bastante pesquisado, como os trabalhos de (SARMA et al., 2012), (LEHMBERG et al., 2015) e (DONG; OYAMADA, 2022), entre outros. Dong, Y. e Oyamada, M. (DONG; OYAMADA, 2022) apresentam esse processo de enriquecimento de dados tabulares em três etapas: recuperação de tabelas, junção de tabelas (join) e avaliação de ML. A recuperação de tabelas é o procedimento de obter tabelas que contenham características que atendam a uma necessidade específica de informação. Essas tabelas podem estar em ambientes como *datalakes* ou na *web* e em formato estruturado ou não. Esses procedimentos podem estar voltados para tarefas como extensão de coluna e linha, correspondência de esquema, preenchimento de tabela e construção de grafo de conhecimento (*knowledge graph*). O procedimento de junção de tabelas (*join*) está ligado a capacidade de efetuar a combinação das tabelas recuperadas que permita a criação de novas características na tabela original. A submissão dos dados enriquecidos por modelos de ML vem no sentido de

avaliar se as novas características se traduziram em melhorias de performance na suas predições.

Nesse estudo, estamos interessados no enriquecimento de conjuntos de dados tabulares por meio da geração de novas características derivadas de estatísticas de um grafo construído a partir desse conjunto de dados já enriquecido por meio de outras abordagens ou não. Assim, essa atividade está inserida no âmbito do processo de engenharia de características discutido nesta seção, podendo ser realizada de forma independente ou em conjunto com a criação de novas características provenientes de outros métodos descritos. O método proposto nesse estudo, se apresenta como uma alternativa àqueles que empregam grafos, pois fundamenta-se na similaridade entre as instâncias para estabelecer os relacionamentos (arestas) entre os vértices do grafo, sendo as instâncias do conjunto de dados tabular os vértices (nós) desse grafo.

A definição do método proposto neste estudo, sua arquitetura e pipeline, serão abordadas no Capítulo 3.

2.3 Teoria dos Grafos e Redes Complexas

Nesta seção, delinearemos os conceitos básicos da teoria dos grafos (alguns autores se referem como rede ou *network*, em inglês) e redes complexas que são essenciais para a compreensão desta pesquisa. Iniciaremos nossa explanação estabelecendo uma base ao conceituar um grafo, que é o principal ente matemático empregado na modelagem de redes. Em seguida, procederemos com a apresentação de definições cruciais que desempenharão um papel significativo mais a frente nesse trabalho, abrangendo conceitos como caminhos, caminhos mais curtos, medidas de distância, componentes conectados e redes complexas.

2.3.1 Grafo

A Teoria dos Grafos é um campo da matemática discreta que estuda a representação e análise de relacionamentos entre objetos por meio de estruturas gráficas chamadas grafos que são estrutura de dados e uma linguagem universal para descrever sistemas complexos. Um grafo é uma coleção de vértices (nós) V e um conjunto de arestas E entre estes vértices. (HAMILTON, 2020, p. 1-2).

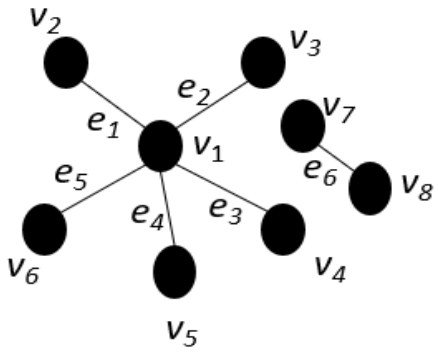
Essa teoria fornece ferramentas matemáticas para a modelagem de uma ampla gama de problemas do mundo real, incluindo redes de computadores, planejamento de rotas, análise de redes sociais, na biologia com interações entre drogas e proteínas e muito mais.

Formalmente, segundo Diestel, R. (DIESTEL, 2017, p. 2) um grafo é um par $G = (V, E)$ de conjuntos tais que $E \subseteq [V]^2$; assim, os elementos de E são subconjuntos de 2 elementos de V, sendo $V \cap E = \emptyset$. Os elementos de V são os vértices (nós) do grafo G, os elementos de E são arestas (ou conexões).

Representamos uma aresta que vai do vértice $u \in V$ ao vértice $v \in V$ como $(u, v) \in E$. No nosso caso, estaremos preocupados apenas com grafos simples não direcionado, onde há no máximo uma aresta entre cada par de vértices (nós), nenhuma aresta entre um nó e ele mesmo, e onde as arestas são todas não direcionadas, ou seja, $(u, v) \in E \leftrightarrow (v, u) \in E$.

Um grafo pode ser representado por um diagrama onde os pontos representam os vértices (nós) e uma linha que liga os pontos representa a aresta (ou conexão). Na Figura 2.3 representamos os vértices $V = \{v_1, v_2, ..., v_8\}$ e $E = \{e_1, e_2, ..., e_6\} = \{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_1, v_5), (v_1, v_6), (v_7, v_8)\}$ as arestas.

Figura 2.3: Grafo simples não direcionado



Dois vértices u e $v \in V$ são considerados adjacentes em G se, e somente se, existe uma aresta $\{u, v\} \in E$ conectando diretamente u e v. Em outras palavras, u e v são adjacentes se a relação de incidência $\{u, v\}$ pertencer ao conjunto de aresta E. Caso contrário, se não houver uma aresta que conecte diretamente u e v, então eles são considerados não adjacentes em G.

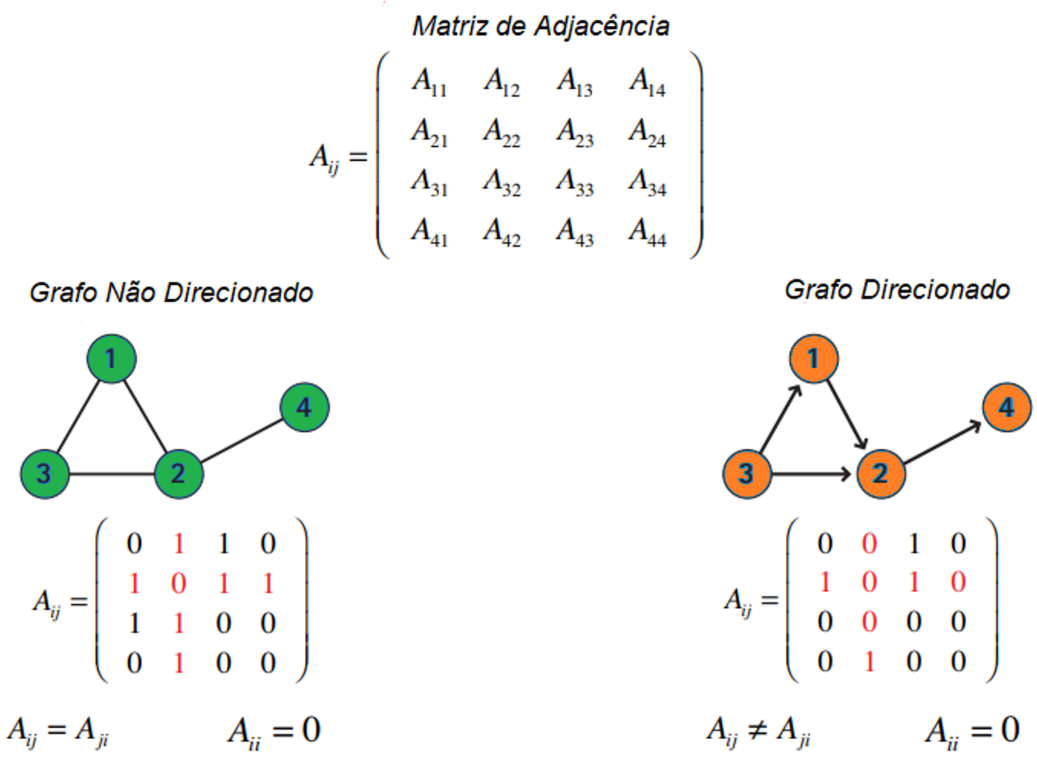
Um grafo completo K_n com n vértices é um tipo especial de grafo que possui a propriedade de que cada par de vértices distintos está diretamente conectado por uma aresta. Em outras palavras, em um grafo completo, não há vértices não adjacentes, e todas as combinações possíveis de pares de vértices estão conectadas por arestas (DIESTEL, 2017, p. 3).

O grau de um vértice é o número de arestas incidentes sobre ele. Se todos os vértices de G têm o mesmo grau k, então G é k-regular, ou simplesmente regular (DIESTEL, 2017, p. 5)

De acordo com Hamilton, W. L. (HAMILTON, 2020, p. 2), uma maneira conveniente de representar grafos é através de uma matriz de adjacência

$A \in \Re^{|V|*|V|}$. Para representar um grafo com uma matriz de adjacência, Figura 2.4, ordenamos os vértices no grafo de forma que cada vértice indexe uma linha e coluna específicas na matriz de adjacência. Podemos então representar a presença de arestas como entradas nesta matriz: $A[u, v] = 1$ se $(u, v) \in E$ e $A[u, v] = 0$ caso contrário. Se o grafo contiver apenas arestas não direcionadas, então A será uma matriz simétrica, mas se o grafo for direcionado (ou seja, a direção das arestas é importante), então A não será necessariamente simétrica.

Figura 2.4: Matriz de Adjacência, baseado em (BARABASI; MARTINO; POSFAI, 2012, p. 32 sec.5)

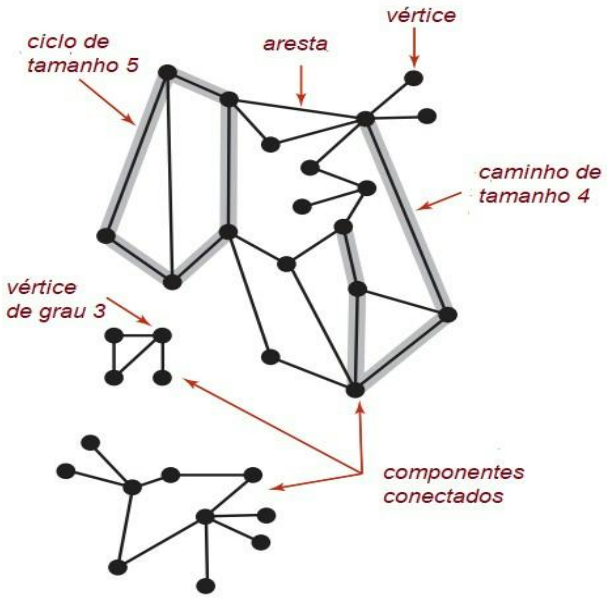


Alguns grafos também podem ter arestas ponderadas, onde as entradas na matriz de adjacência são valores reais arbitrários em vez de $\{0, 1\}$. Por exemplo, uma aresta ponderada num grafo de interação aeroporto-aeroporto pode indicar a força da associação entre esses aeroportos.

A distância $d(u, v)$ em G de dois vértices u e $v \in V$ é o comprimento de uma distância $d(u, v)$ do caminho u-v mais curto em G; se tal caminho não existir, definimos $d(u, v) := \infty$ (DIESTEL, 2017, p. 8). Em outras palavras, a distância entre um par de vértices em um grafo é o número de arestas em um caminho mínimo conectando o par. Se não existe caminho algum conectando dois vértices, então a distância é definida como infinita.

Em um grafo G, um caminho é uma sequência ordenada de vértices que são ligados por arestas, sendo denominado simples quando não há repetições de vértices. Um ciclo, por sua vez, é um tipo especial de caminho que possui pelo menos uma aresta conectando o primeiro e último vértices, enquanto um ciclo simples é caracterizado por não conter vértices ou arestas repetidos, à exceção da necessária repetição do primeiro e último vértices. O comprimento de um caminho ou ciclo é determinado pelo número de arestas que o compõe (SEDGEWICK; WAYNE, 2011, p. 519).

Figura 2.5: Anatomia de um Grafo, baseado em (SEDGEWICK; WAYNE, 2011, p. 519)



Um grafo conectado, Figura 2.5, é aquele no qual existe pelo menos um caminho entre cada par de vértices (SEDGEWICK; WAYNE, 2011, p. 519). Em outras palavras, um grafo conectado é aquele em que você pode viajar de qualquer vértice para qualquer outro vértice no grafo seguindo as arestas. Um grafo conectado não necessariamente precisa ser completo; ele pode ter várias arestas ausentes entre pares de vértices, desde que haja um caminho que os conecte indiretamente.

Vale destacar que enquanto um grafo completo se concentra na presença de arestas entre todos os pares de vértices, um grafo conectado se concentra na existência de caminhos que conectam todos os vértices, direta ou indiretamente. Um grafo conectado não precisa ser completo, mas um grafo completo é sempre conectado

Hamilton, W. L. (HAMILTON, 2020, p. 2) introduz novos conceitos relacionados as arestas do grafo, onde devemos considerar, além da distinção entre arestas não direcionadas, direcionadas e ponderadas, também grafos

que possuem diferentes tipos de arestas. Um exemplo seria em grafos que representam interações medicamentosas, onde podemos querer que diferentes arestas correspondam a diferentes efeitos colaterais que podem ocorrer quando você toma dois medicamentos ao mesmo tempo.

Para formalizar esse conceito, Hamilton, W. L. (HAMILTON, 2020, p. 3) sugere a extensão da notação de aresta para incluir uma aresta ou tipo de relação τ, por exemplo, $(u, \tau, v) \in E$, e podemos definir uma matriz de adjacência A_τ por tipo de aresta. Esses grafos são chamados grafos multi-relacionais, e o grafo inteiro pode ser resumido por um tensor de adjacência $A \in \mathbb{R}^{[V]*[R]*[V]}$, onde R é o conjunto de relações.

Dois subconjuntos importantes desses grafos multi-relacionais são frequentemente conhecidos como grafos heterogêneos e grafos multiplex. De acordo com Hamilton, W. L. (HAMILTON, 2020, p. 3) esses dois subconjuntos podem ser definidos como: grafos heterogêneos, são aqueles em que os vértices (nós) possuem tipos distintos, o que permite particionar o conjunto de vértices (nós) em conjuntos separados, onde cada conjunto representa um tipo específico de nó. As arestas nesses grafos geralmente seguem restrições de acordo com os tipos de vértices (nós), como a restrição de que determinadas arestas só se conectam com de tipos específicos.

Por exemplo, em um grafo biomédico heterogêneo, pode haver tipos de vértices (nós) que representam proteínas, medicamentos e doenças. Arestas que representam 'tratamentos' só ocorreriam entre vértices (nós) de medicamentos e vértices (nós) de doenças, enquanto arestas que representam 'efeitos colaterais de polifarmácia' só ocorreriam entre dois vértices (nós) de medicamentos.

Grafos multipartidos são um caso especial bem conhecido de grafos heterogêneos, onde as arestas só podem conectar vértices (nós) de tipos diferentes; Grafos multiplex: assumimos que o grafo pode ser decomposto em um conjunto de k camadas. Cada nó é considerado pertencente a todas as camadas, e cada camada corresponde a uma relação única, representando o tipo de aresta intracamada específico para aquela camada. Também assumimos que tipos de arestas intercamadas podem existir, conectando o mesmo nó em diferentes camadas.

Por exemplo, em uma rede de transporte multiplex, cada nó pode representar uma cidade e cada camada pode representar um modo de transporte diferente (por exemplo, viagem aérea ou viagem de trem). As arestas intracamadas representariam cidades conectadas por diferentes modos de transporte, enquanto as arestas intercamadas representariam a possibilidade de trocar modos de transporte dentro de uma cidade específica.

2.3.2 Redes Complexas

Redes complexas referem-se a grafos que possuem propriedades topológicas diferenciadas, as quais não são observadas em grafos mais simples (METZ et al., 2007, p. 3).

A pesquisa das redes complexas pode ser compreendida como situada entre a teoria dos grafos e a mecânica estatística, resultando em uma abordagem genuinamente multidisciplinar (COSTA et al., 2007, p. 3). Ela não apenas expande o formalismo da teoria dos grafos, mas, também, introduz medidas e métodos estatísticos fundamentais para se caracterizar a estrutura complexa de conexões da rede, assim como a análise das propriedades do sistema no mundo real.

Uma das principais razões para a ampla adoção das redes complexas reside em sua versatilidade e capacidade abrangente de representar virtualmente qualquer estrutura encontrada na natureza (COSTA et al., 2007, p. 3).

Propriedades das Redes Complexas

Nem todo grafo pode ser categorizado como uma rede complexa, uma vez que essa classificação somente se aplica quando o grafo exibe certas características (propriedades) topológicas ausentes em grafos mais simples (METZ et al., 2007, p. 4)

Apresentamos a seguir, de forma resumida, algumas dessas propriedades que, segundo Metz, J, et al (METZ et al., 2007, p. 4-6), têm recebido muita atenção na literatura:

(1) O coeficiente de aglomeração, às vezes chamado de fenômeno de transitividade, quantifica os agrupamentos naturais presentes em redes. Esse fenômeno se manifesta quando um vértice A está ligado a um vértice B, e o vértice B está ligado a um vértice C, aumentando a probabilidade de que o vértice A também esteja conectado ao vértice C. (2) A distribuição de graus, que é uma função probabilística que descreve as chances de um vértice ter um grau particular. O grau de um vértice, por sua vez, representa o número de arestas que se conectam a esse vértice. (3) Resistência, que se refere à capacidade da rede de suportar a remoção de alguns vértices sem comprometer sua funcionalidade. Isso está ligado à distribuição de graus dos vértices, já que a remoção de vértices pode levar à desconexão entre vértices ou aumentar a distância entre eles. (4) Misturas de Padrões, onde casos em que certos tipos de redes exibem uma combinação de padrões diversos, onde os vértices podem representar tipos distintos de entidades. (5) Correlação de Graus, que indica se as arestas em uma rede tendem a conectar vértices com graus semelhantes, ou seja, se há uma associação entre vértices que possuem graus aproximados.

Tipo de Redes Complexas

Diversos modelos (ou tipos) de redes foram desenvolvidos para analisar as propriedades topológicas de redes reais. Alguns desses modelos populares incluem grafos aleatórios, o modelo de mundo pequeno, o grafo aleatório generalizado e as redes de Barabási-Albert (ou redes livres de escala). Além disso, há modelos específicos para redes geográficas e redes com estrutura de comunidades (COSTA et al., 2007, p. 9).

Abordaremos somente aqueles modelos considerados, de acordo com Metz, J., et al (METZ et al., 2007, p. 6), os principais modelos (ou tipos) de redes complexas (redes aleatórias, redes pequeno-mundo e redes livres de escala).

Redes Aleatórias (COSTA et al., 2007, p. 10): Neste modelo, ocorre a aleatória conexão de arestas entre um conjunto fixo de N vértices. Cada aresta é acrescentada independentemente com uma probabilidade p. O grau de cada vértice na rede, indicando quantas arestas estão ligadas a ele, segue uma distribuição de Poisson com um limite máximo de N. O grau esperado de um vértice qualquer é dado por:

$$\langle k \rangle = p\,(N-1),$$

onde p denota a probabilidade de que um vértice estabeleça uma conexão com qualquer outro vértice na rede, enquanto N representa a quantidade total de vértices na rede e k refere-se ao número total de arestas que se conectam a um vértice específico.

Redes de Mundo Pequeno: Neste modelo, a média da distância entre quaisquer dois vértices em uma rede consideravelmente grande é limitada a um número reduzido de vértices (METZ et al., 2007, p. 7). Em outras palavras, são redes nas quais a distância média entre quaisquer dois vértices na rede é notavelmente curta, mesmo em redes muito grandes. Isso implica que, em termos de números de vértices e arestas, as redes de mundo pequeno exibem uma grande eficiência na comunicação entre seus membros, permitindo a rápida transmissão de informações através de apenas alguns intermediários, independentemente do tamanho da rede. Um exemplo de demonstração direta do efeito pequeno-mundo é sugerido por Metz, J, et al (METZ et al., 2007, p. 7) com o experimento conduzido por Stanley Milgram em 1960, se uma carta fosse entregue a um indivíduo, que não fosse o destinatário, e ele a repassasse a um outro e, assim, por diante, em aproximadamente seis passagens ela chegaria ao destinatário. Nesse exemplo o caminho percorrido pela carta, partindo de um indivíduo qualquer até o destinatário, é mínimo. Metz, J, et al (METZ et al., 2007, p. 7) ainda acrescenta a definição de comprimento do caminho

mínimo médio $CM\,(l)$ entre pares de vértices em um grafo não direcionado e com um componente é dada por:

$$l = \frac{1}{\frac{1}{2}n(n+1)} \sum_{i>=j} d_{ij},$$

onde d_{ij} é a distância geodésica do vértice i até o vértice j.

Redes Livres de Escala: São redes complexas que exibem uma distribuição de grau livre, indicando que parte significativa dos seus vértices tem um grau relativamente baixo (poucas conexões-arestas), enquanto poucos vértices (os chamados "*hubs*") possuem graus excepcionalmente altos (muitas conexões-arestas) (BARABASI; MARTINO; POSFAI, 2012, p. 4 sec.4). Segundo (COSTA et al., 2007, p. 13) a distribuição dos graus segue a lei de potência (ou *power law* em inglês):

$$P\,(k) \sim k^{-\gamma},$$

onde $P\,(k)$ se refere à probabilidade de um vértice na rede ter um grau igual a k (ou seja, estar conectado a k outros vértices), γ é um parâmetro que descreve a inclinação da distribuição quando γ é positivo significa que há uma probabilidade significativa de encontrar vértices com graus muito altos (*hubs*) e quando γ é baixo (ou negativo) a probabilidade de vértices altamente conectados é menor.

Ainda de acordo com Costa, L. F. et al (COSTA et al., 2007, p. 13-14), as redes livres de escala baseiam-se em duas regras básicas: crescimento e adesão preferencial. A rede começa com um conjunto inicial de vértices, e a cada passo, novos vértices são adicionados (crescimento). Cada novo vértice estabelece conexões com vértices existentes. A seleção dos vértices existentes com os quais o novo vértice se conecta é feita com base na preferência por vértices de maior grau (adesão preferencial), ou seja, a probabilidade de um novo vértice se conectar a um vértice existente é proporcional ao grau desse vértice.

Formalmente, Costa, L. F. et al (COSTA et al., 2007, p. 14) definiu que a probabilidade do novo vértice i se conectar com um vértice j existente é proporcional ao grau de j resultando em:

$$\mathcal{P}\,(i \to j) = \frac{k_j}{\sum_u k_u}$$

Newman, M. (NEWMAN, 2018, p.317-323) apresenta exemplos de sistemas que exibem distribuições que seguem de maneira aproximada a “lei de potência”, como a Internet, a *World Wide Web* e em redes de citações de artigos científicos.

2.3.3 Grafo de Similaridade

Segundo Liu, J. e Han, J. (LIU; HAN, 2014, p.178), um grafo de similaridade (ou *similarity graph*, em inglês) é um de três passos da família da técnica de agrupamento espectral (*spectral clustering* em inglês), técnica cujos resultados obtidos muitas vezes superam as abordagens tradicionais (LUXBURG, 2004, p.1).

O agrupamento espectral, resumidamente, é uma abordagem analítica de dados que se fundamenta na teoria dos grafos e na álgebra linear com a finalidade de agrupar conjuntos de dados, considerando as suas relações de similaridade. Esse processo envolve a construção de um grafo de similaridade (passo 1) no qual os pontos de dados são representados como vértices e as conexões (arestas) entre eles são ponderadas com base na medida de similaridade entre esses pontos. Posteriormente (passos 2 e 3), um algoritmo específico utiliza a decomposição espectral da matriz de Laplace do grafo para identificar e extrair os grupos de dados subjacentes (LIU; HAN, 2014, p.178).

Dentro desse contexto, de acordo com Luxburg, U. (LUXBURG, 2004, p.2-3), o grafo de similaridade é uma representação gráfica de dados onde os vértices (nós) do grafo correspondem a pontos de dados e as arestas (ligações) entre esses vértices são efetuadas com base em uma medida de similaridade entre os pontos de dados correspondentes e o peso dessas arestas é uma função dessa similaridade. Em outras palavras, esse tipo de grafo é construído para destacar as relações de similaridade entre os elementos do conjunto de dados. A ideia-chave é que pontos de dados mais semelhantes estejam mais fortemente conectados no grafo, enquanto pontos menos semelhantes tenham conexões mais fracas ou inexistentes.

De acordo com Liu, J. e Han, J. (LIU; HAN, 2014, p.179), a definição de grafo de similaridade como um dos passos do agrupamento espectral é dada por: Seja o conjunto de pontos que se deseja particionar em k subconjuntos, denotados por $X = \{x_1, ..., x_n\}$ em $\mathcal{R}^m$. Para realizar o agrupamento espectral, primeiro representamos esses dados na forma de um grafo de similaridade não direcionado $G = (V, E)$. Aqui cada ponto x_i é representado pelo vértice v_i e E se refere as arestas entre esses vértices. Note que x_i é um vetor que representa um ponto de dados enquanto v_i um vértice sem qualquer atributo. Nesse caso, podemos utilizar uma matriz de adjacência W de peso não negativo, n por n, para descrever G, onde $W = \{w_{ij}\}_{ij=1,...,n}$. Note que w_{ij} é igual a 0 quando os vértices v_i e v_j não são conectados. Como os algoritmos de agrupamento espectral visam particionar os vértices para permitir que aqueles no mesmo cluster tenham alta similaridade e aqueles em diferentes clusters têm baixa

similaridade.

Existem diversas abordagens para transformar um conjunto de pontos de dados em um grafo. A construção de grafos de similaridade visa, primordialmente, modelar as relações de vizinhança local entre os pontos de dados em questão. Esses grafos são cruciais para a análise e representação de informações, permitindo a exploração das relações inerentes à proximidade ou similaridade entre os dados, aspecto fundamental em diversas aplicações analíticas e científicas (LUXBURG, 2004, p. 3).

De acordo com (LUXBURG, 2004, p.3), existem três tipos principais de grafos usados na análise de dados com base na similaridade entre pontos. Primeiro tipo, o chamado grafo de vizinhança ε, que conecta pontos cujas distâncias mútuas estejam abaixo de um valor ε pré-definido e geralmente é considerado não ponderado. Segundo tipo, os chamados grafos dos K-vizinhos mais próximos, que buscam conectar pontos quando um deles está entre os K-vizinhos mais próximos do outro, resultando em um grafo direcionado que pode ser transformado em um grafo não direcionado de três maneiras diferentes. A primeira abordagem consiste em desconsiderar o sentido das ligações. Ou seja, conectamos v_i e v_j com uma aresta não direcionada se v_i estiver entre os k vizinhos mais próximos de v_j, ou vice-versa. O grafo resultante é denominado grafo k-vizinhos mais próximos. Uma segunda alternativa é conectar os vértices v_i e v_j somente se v_i estiver entre os k vizinhos mais próximos de v_j e v_j estiver entre os k vizinhos mais próximos de v_i. O grafo obtido por essa construção é chamado de grafo mútuo k-vizinhos mais próximos. Nas duas abordagens, após a conexão dos vértices, as arestas são ponderadas pela similaridade de seus pontos finais. O terceiro tipo, o grafo completamente conectado, que liga todos os pontos com similaridade positiva entre si e atribui pesos às arestas com base na similaridade, o que é útil se a função de similaridade já modelar relacionamentos locais entre os pontos.

Os grafos de similaridade são frequentemente utilizados em tarefas de análise de dados, mineração de dados e aprendizado de máquina, onde a estrutura das conexões pode revelar informações sobre a estrutura subjacente dos dados, como agrupamentos naturais ou padrões de similaridade.

Como já citado, o presente estudo propõe uma abordagem cujo foco primordial reside não na busca exaustiva pelas métricas de similaridade consideradas ideais, mas, sobretudo, na validação da influência exercida pela incorporação do conceito de grafo de similaridade e das métricas subsequentemente derivadas desse grafo, na melhoria do desempenho dos modelos destinados à tarefa de predição. Nesse contexto, o enfoque aqui recai sobre a utilização das técnicas de construção e formalização de grafos de similaridade. Estas técni-

cas serão empregadas para estabelecer um procedimento sistemático e bem definido na criação do referido grafo, que representa a peça fundamental da solução em análise.

Nessa direção, utilizaremos os conceitos de grafo de similaridade para construir o grafo que vai suportar a solução com algumas adaptações. Podemos definir o grafo de similaridade utilizado nesse estudo como: Seja um conjunto de pontos (objetos) de dados $X = \{x_{1k}, \cdots, x_{rk}\}$ (no nosso estudo, um conjunto de dados tabular onde x_{ik} é uma das r instâncias deste e que possui k atributos e alguma medida de similaridade s_{ij} entre os pares de dados de pontos (objetos) x_i, x_j e o grafo $G = (V, E)$ com conjuntos de vértices $v_i \in V$ neste grafo G representa um ponto de dados x_i sem atributos, E se refere as arestas entre esses vértices, além disso, este grafo G é não direcionado e ponderado, ou seja, cada aresta entre dois vértices, v_i e v_j, leva consigo um peso positivo, w_{ij}. Dois vértices são conectados (arestas) se a similaridade s_{ij} entre os pontos de dados correspondentes x_i, x_j for positivo e maior que um determinado limite (*threshold*, em inglês). Esse grafo conectado $G = (V, E)$ será a base da extração de dados para nosso estudo. A definição desse limite (*threshold*) será abordada mais adiante no detalhamento do método proposto no Capiítulo 3.

2.4 Medidas de Similaridade e Distância

Medidas de Similaridade e Distância (ou Dissimilaridade) entre dois objetos desempenham um papel importante e são amplamente aplicadas em várias técnicas que envolvem cálculo de distância, notadamente no campo da mineração de dados, incluindo tarefas como agrupamento, classificação de vizinho mais próximo e detecção de anomalias. Uma vez que essas medidas de semelhança ou diferença tenham sido calculadas, em determinados cenários, a necessidade do conjunto de dados original pode ser reduzida ou até mesmo eliminada. Nesse contexto, essas abordagens podem ser consideradas como uma transformação dos dados em um espaço de similaridade (ou distância), seguida pela condução da análise com base nesse novo espaço transformado (TAN et al., 2019, p.91).

Ao lidar com um conjunto de dados, determinar a similaridade entre seus elementos é uma questão de grande relevância e desafio. A heterogeneidade inerente aos dados, proveniente de diferentes tipos de atributos, torna essencial encontrar uma abordagem que permita realizar comparações significativas entre eles. Uma estratégia comumente adotada é a conversão dos atributos em valores numéricos, o que possibilita representar cada característica como uma dimensão em um espaço multidimensional. Nesse contexto, os pontos

de dados são interpretados como coordenadas nesse espaço, e a medida de distância entre eles é utilizada como uma métrica de similaridade apropriada. Essa abordagem visa estabelecer uma base sólida para a análise comparativa de dados heterogêneos, permitindo a tomada de decisões informadas quanto à sua similaridade.

No contexto desta pesquisa, a proposta é utilizar algumas dessas medidas como fundamento para estabelecer as conexões (arestas) dentro do grafo gerado a partir das instâncias de um conjunto de dados tabular original como vértices (nós) deste grafo. Para atingir esse objetivo, é imperativo adquirir uma maior compreensão das medidas de similaridade e distância, discernir suas distinções fundamentais e examinar suas aplicações para orientar de maneira mais precisa os esforços voltados para uma implementação eficaz no âmbito da nossa aplicação. Cabe ressaltar que estamos voltados não na busca exaustiva pelas medidas de similaridade consideradas ideais, mas, sobretudo, na validação da influência exercida nos modelos de aprendizado de máquina supervisionados de classificação pela incorporação do conceito de grafo de similaridade e algumas de suas estatísticas como novas características no conjunto de dados tabular utilizado por estes como fonte de dados.

Portanto, nesta seção definiremos alguns dos conceitos básicos de similaridade e distâncias, assim como algumas das suas aplicações e como vamos utilizá-las em nossa pesquisa. Para uma explicação completa da teoria dessas medidas, sugerimos, dentre outras fontes, a consulta de obras como "Introduction to Data Mining"de Tan, P.-N., et al (TAN et al., 2019), "Data Mining: Concepts and Techniques"de Han, J., et al (HAN; KAMBER; PEI, 2012), bem como o estudo do artigo de Boriah, S., et al, (BORIAH; CHANDOLA; KUMAR, 2008), intitulado "Similarity Measures for Categorical Data: A Comparative Evaluation".

Definições Básicas

A similaridade entre dois objetos é uma medida quantitativa que avalia o grau de correspondência ou concordância entre eles. Em outras palavras, é uma maneira de avaliar até que ponto os objetos são semelhantes ou diferentes em comparação um com o outro. Como resultado, os valores de similaridade tendem a ser mais elevados para pares de objetos que compartilham maior semelhança (TAN et al., 2019, p.91) e (HAN; KAMBER; PEI, 2012, p.65)

Han, J. et al (HAN; KAMBER; PEI, 2012, p.65) fornece um exemplo: uma loja pode querer procurar grupos de clientes (objetos), resultando em grupos de clientes com características semelhantes (renda semelhante, área de residência, idade etc.).

É importante notar que as medidas de similaridade, em geral, assumem

valores não negativos e tipicamente variam em uma escala de 0 (indicando ausência de similaridade) a 1 (indicando similaridade completa) (TAN et al., 2019, p.91). Por outro lado, uma medida de dissimilaridade (distância) funciona de maneira oposta, podendo, em alguns casos, variar de 0 a infinito. Ela retorna um valor 0 se os objetos forem iguais (e, portanto, longe de serem diferentes) e quanto maior o valor de dissimilaridade (distância), mais diferentes são os dois objetos (HAN; KAMBER; PEI, 2012, p.67).

Em resumo, a similaridade se refere à medida numérica do grau em que dois objetos ou pontos são semelhantes em um espaço multidimensional. Valores de similaridade mais altos indicam maior semelhança. A distância se refere à medida numérica que quantifica a diferença ou separação entre dois objetos ou pontos em um espaço multidimensional. Valores de distância mais baixos indicam maior semelhança.

Alguns autores, vide (HAN; KAMBER; PEI, 2012, p.91) por exemplo, utilizam o termo proximidade para se referir à ideia de quão próximos ou semelhantes dois objetos ou pontos são em um espaço multidimensional. Esse termo é frequentemente usado de forma mais ampla e abstrata para descrever a relação entre objetos, independentemente de serem semelhantes ou diferentes. É um termo mais genérico que pode abranger tanto a similaridade quanto a distância (dissimilaridade).

Objeto, Atributo e Tipos de Atributo

Para medir a similaridade e distância, antes precisamos efetuar algumas definições básicas que serão importantes para nos suportar mais a frente quando formos detalhar as medidas de similaridade e distância que adotamos nesse trabalho.

Começando pela definição de objeto de dados que, doravante, vamos nos referir simplesmente como objeto, que representa uma entidade em um conjunto de dados. Por exemplo, uma pessoa seria um objeto em um conjunto de dados que contém os dados de uma pessoa física. Outro exemplo seria o objeto estudante em um conjunto de dados acadêmico etc. Em outros palavras, podemos encarar um objeto como uma instância de um conjunto de dados. Esse objeto é composto de atributos. Quando armazenado em um banco de dados, esse objeto é chamado de linhas (ou tuplas) e seus atributos colunas (ou característica), sendo que cada coluna tem um tipo. Um atributo é um campo de dados que representa uma característica do objeto. Chamamos de vetor de atributos um conjunto de atributos utilizado para descrever um objeto. Se um objeto possui um atributo ele é chamado de univariado e multivariado se possuir múltiplos atributos. Um tipo de atributo é determinado pelo conjunto de possíveis valores que esse atributo pode aceitar (HAN; KAMBER; PEI,

2012, p.40)

Um tipo de atributo pode ser classificado em dois grupos e esses em outros grupos, a saber: Tipos Categóricos que normalmente expressam qualidades e os tipos Numéricos que expressam quantidades. Os Categóricos podem ser do tipo nominal ou ordinário, mesmo que expressos em números, e os Numéricos podem ser do tipo intervalo e proporção (*ratio*) (TAN et al., 2019, p.49-50).

Os valores associados a um atributo nominal ou ordinal consistem em símbolos ou denominações de entidades. Cada um desses valores representa uma categoria, código ou estado específico. Os valores em atributos nominais não exibem uma ordem intrínseca ou significativa e podem estar expressos em números. Os valores em um atributo ordinal possuem uma ordem significativa ou classificação entre eles, mas a magnitude entre valores sucessivos não é conhecida e podem também estar expressos em números. Os valores associados a um atributo intervalo ou proporção (*ratio*, em inglês) são representados por números inteiros ou reais. Os valores em atributos intervalo têm ordem e podem ser positivos, 0 ou negativos. Os valores em um atributo proporção (*ratio*) são expressos como múltiplos ou razões de outro valor. É importante notar que esses valores estão sujeitos a uma ordenação intrínseca, o que implica que podemos realizar operações como a medição das diferenças entre eles e o cálculo de estatísticas como média, mediana, moda, entre outras (HAN; KAMBER; PEI, 2012, p.41-42)

Tan, P.-N. (TAN et al., 2019, p.52), fornece uma abordagem alternativa para categorizar os tipos de atributos que se baseia na quantidade de valores que eles podem assumir, o que resulta na distinção entre atributos discretos e contínuos. Os atributos discretos abarcam um conjunto de valores que é finito ou infinitamente contável. Esses atributos podem ser subdivididos em categorias, como códigos postais ou números de identificação, ou podem assumir valores numéricos, tais como contagens. Frequentemente, os atributos discretos são representados mediante variáveis inteiras, e uma categoria especial destes é constituída pelos atributos binários, que admitem apenas dois valores possíveis, como verdadeiro/falso, sim/não, masculino/feminino ou 0/1.

Por outro lado, os atributos contínuos são definidos pela natureza de seus valores, que não são restritos a um conjunto discreto. Para uma definição mais precisa, podemos caracterizar os atributos contínuos como aqueles que assumem valores numéricos reais, geralmente expressos como variáveis de ponto flutuante. Exemplos típicos englobam grandezas como temperatura, altura, peso, entre muitos outros (TAN et al., 2019, p.52).

Matriz de Dados

Na seção anterior, elucidamos nossa definição do termo "objeto"utilizado

nesse trabalho. Com base nesta, vamos pensar que quando todos os objetos contidos em um conjunto de dados compartilham o mesmo conjunto predeterminado de atributos numéricos, é factível concebê-los como pontos (ou vetores) situados em um espaço multidimensional. Cada dimensão desse espaço corresponde a um atributo específico que caracteriza o objeto. Esse conjunto de objetos pode ser compreendido como uma matriz com dimensões n por m, onde n representa o número de linhas, cada uma representando um objeto, e m denota o número de colunas, cada uma representando um atributo distinto. Essa matriz é comumente denominada matriz de dados (TAN et al., 2019, p.52). Formalmente, seja x_i um vetor de atributos do objeto i com m atributos (ou características): $x_1 = (x_{11}, x_{12}, \cdots, x_{1m})$, $x_2 = (x_{21}, x_{22}, \cdots, x_{2m})$, ..., onde x_{ij} é o valor do j-ésimo atributo do objeto i, sendo que a matriz (n objetos x m atributos) será dada por:

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix}$$

2.4.1 Matriz de Similaridade

Uma matriz de similaridade é uma matriz que quantifica a semelhança ou proximidade entre pares de objetos ou elementos em um conjunto de dados.

Seja *s(i,j)* a medida de similaridade entre os objetos i e j. Algumas das características fundamentais da matriz de similaridade são: a simetria, isto é, os elementos na posição [i, j] são iguais aos elementos na posição [j, i]. Isso ocorre porque a similaridade entre o objeto i e o objeto j é a mesma que a similaridade entre o objeto j e o objeto i; na diagonal principal, isto é, os elementos na posição [i, i], geralmente têm um valor máximo, representando a similaridade de um objeto com ele mesmo, que é igual a 1; os elementos fora da diagonal principal da matriz contêm os valores de similaridade entre pares de objetos. Esses valores de similaridade podem ser calculados usando várias métricas, como veremos mais a frente nessa pesquisa; os valores na matriz *s(i,j)* geralmente estão na faixa de 0 a 1, onde 0 indica ausência de similaridade (ou dissimilaridade completa) entre os objetos e 1 indica similaridade completa. Finalmente, o tamanho da matriz de similaridade depende do número de objetos no conjunto de dados. Se houver n objetos, a matriz será de tamanho n x n ou uma matriz quadrada de ordem n. Nesse caso, a matriz de similaridade *s(i,j)* será dada por:

$$s\left(i,j\right)=\begin{bmatrix}1 & s_{1,2} & \cdots & s_{1,n}\\ \vdots & \vdots & \ddots & \vdots\\ s_{n,1} & s_{n,2} & \cdots & 1\end{bmatrix}$$

2.4.2 Matriz de Dissimilaridade

A matriz de dissimilaridade (ou distâncias, que são dissimilaridades com certas propriedades).armazena uma coleção de proximidades que estão disponíveis para todos os pares de n objetos (TAN et al., 2019, p.94). Esta matriz é complementar à matriz de similaridade, onde valores maiores indicam maior dissimilaridade. Seja d(i,j) a medida de dissimilaridade entre os objetos i e j. Algumas das características da matriz de dissimilaridade são: a simetria, isto é, os elementos na posição [i, j] são iguais aos elementos na posição [j, i]. Isso ocorre porque a dissimilaridade entre o objeto i e o objeto j é a mesma que a dissimilaridade entre o objeto j e o objeto i; na diagonal principal, isto é, os elementos na posição [i, i], geralmente têm um valor mínimo, representando a dissimilaridade de um objeto com ele mesmo, que é igual a 0; os elementos fora da diagonal principal da matriz contêm os valores de dissimilaridade entre pares de objetos. Esses valores de dissimilaridade podem ser calculados usando várias métricas, como veremos mais a frente nessa pesquisa; O tamanho da matriz de dissimilaridade depende do número de objetos no conjunto de dados. Se houver n objetos, a matriz será de tamanho n x n ou uma matriz quadrada de ordem n (HAND; MANNILA; SMYTH, 2001, p.31-32). Nesse caso, a matriz de dissimilaridade d(i,j) será dada conforme abaixo:

$$d\left(i,j\right)=\begin{bmatrix}0 & d_{1,2} & \cdots & d_{1,n}\\ \vdots & \vdots & \ddots & \vdots\\ d_{n,1} & d_{n,2} & \cdots & 0\end{bmatrix}$$

2.4.3 Medindo Similaridade e Dissimilaridade

Existem diversas medidas de valores de similaridade e dissimilaridades (ou distâncias, que são dissimilaridades com certas propriedades). Algumas dessas medidas são voltadas para atributos nominais, binários, numéricos, ordinais e, finalmente, quando todos ou parte destes estão presentes no mesmo objeto (HAN; KAMBER; PEI, 2012, p.68-79). A escolha da melhor métrica deve se basear no contexto de sua aplicação específica e nas características dos dados.

Existem diversos algoritmos para cálculo de medidas que avaliam a similaridade e dissimilaridade entre objetos em análises de dados. As medidas de dissimilaridade, que também podem ser denominadas distâncias quando satisfazem propriedades específicas (TAN et al., 2019, p.96), abrangem uma variedade de tipos, adequadas para diferentes tipos de atributos, tais como nominais, binários, numéricos (discretos e flutuantes), ordinais, e para cenários em que múltiplos tipos de atributos coexistem em um único objeto (HAN; KAMBER; PEI, 2012, p.68-79). Portanto, a diferenciação entre métricas destinadas a atributos categóricos e métricas voltadas para atributos numéricos é essencial devido à natureza intrinsecamente diferente desses tipos de dados. É importante destacar que a escolha da métrica apropriada deve ser embasada no contexto específico de sua aplicação e nas características intrínsecas dos dados em questão.

A escolha criteriosa das medidas de similaridade desempenha um papel importante na validade e abrangência dos resultados obtidos em nossa pesquisa. Considerando a diversidade intrínseca dos conjuntos de dados investigados, optamos por uma abordagem abrangente ao selecionar sete medidas distintas. Para capturar nuances específicas em dados categóricos, incorporamos as métricas *Jaccard, Overlap* e *Eskin*, reconhecendo a sensibilidade desses métodos às características discretas e distintas de diferentes domínios. Simultaneamente, a inclusão de medidas numéricas, como *Euclidean, Cosine, Manhattan* e *Mahalanobis*, visa abranger conjuntos de dados com variáveis contínuas e facilitar a análise em domínios mais complexos.

Essa abordagem multifacetada busca garantir que as medidas selecionadas atendam às particularidades inerentes a conjuntos de dados diversos, capturando efetivamente tanto a semelhança em atributos categóricos quanto a proximidade em atributos numéricos. Portanto, essas medidas foram escolhidas pela sua robustez, abrangência, capacidade de capturar diferentes tipos de similaridade, adequação aos tipos de dados presentes nos conjuntos tabulares selecionados, ampla documentação na literatura acadêmica e disponibilidade de bibliotecas de software especializadas para o cálculo.

No Apendice A detalhamos os fundamentos de cada uma dessas medidas de similaridade e dissimilaridade selecionadas.

No Capítulo 3, abordamos o método proposto e apresentamos detalhes sobre a implementação do procedimento de similaridade/dissimilaridade entre o par de objetos.

2.5 Estatísticas do Grafo

Em redes pequenas é possível visualizar o grafo, efetuar algumas análises e extrair algumas informações relevantes deste. No entanto, na medida em que essas redes crescem e ficam mais complexas, essa análise fica cada vez mais difícil. Uma abordagem para superar esse obstáculo e extrair informações relevantes de grafos representando grandes redes complexas, é definir medidas matemáticas que capturem quantitativamente características interessantes da estrutura da rede, resumindo grandes volumes. de dados estruturais complexos em números simples que são fáceis para as pessoas entender (NEWMAN, 2018).

Para compreender e analisar essas redes complexas, muitas métricas foram concebidas e testadas (NEWMAN, 2018). Nessa pesquisa, vamos investigar e avaliar o impacto da adição ao conjunto de dados tabular original de algumas destas estatísticas (ou medidas) oriundas do grafo na performance das predições dos modelos de ML selecionados. Nesse sentido, nos concentramos em algumas medidas chamadas de Centralidade, sendo as demais medidas de Estrutura Local e Análise de Links. Uma breve introdução a essas métricas e a lógica por trás de aproveitá-las é discutida nessa seção.

Para manter a compatibilidade com a terminologia utilizada por diferentes autores nas referências deste trabalho, optamos por empregar a terminologia "medidas" ou "métricas" quando os autores assim as utilizam. No entanto, é importante ressaltar que, no contexto deste trabalho, estamos seguindo a mesma abordagem adotada por Hamilton, W. L. (HAMILTON, 2020) utilizando a terminologia "estatísticas do grafo" ou "características do grafo". Isso se deve à necessidade de diferenciar claramente essas características daquelas que se referem a "medidas de similaridade ou dissimilaridade" ou "medidas de avaliação de modelos de machine learning". Dessa forma, o uso de "estatísticas do grafo" será mantido para representar as características específicas extraídas da estrutura do grafo, enquanto os termos "medidas" ou "métricas" serão adotados em conformidade com as convenções das fontes de referência.

2.5.1 Classificação de Estatísticas do Grafo

Hamilton, W. L. (HAMILTON, 2020, p.9) classifica em duas grandes categorias as estatísticas extraídas de um grafo com base na sua aplicação: estatísticas de nível global e estatísticas de nível de nós ou vértices (alguns outros autores se referem como locais).

As estatísticas de nível global referem-se às propriedades globais de um grafo e, portanto, consistem em um único número para cada grafo ou rede; estatísticas nodais referem-se a propriedades dos nós (ou vértices) de um grafo e, portanto, consistem em um vetor de números — um para cada nó do grafo.

Hernandez, J. e Mieghem, P. V. (HERNÁNDEZ; MIEGHEM, 2015, p.1) sugerem que a definição completa de uma rede engloba não apenas sua estrutura topológica, mas, também, protocolos, dinâmicas e restrições, resultando na categorização de métricas em topológicas e de serviço, onde as métricas topológicas seriam classificadas em Distância, Conexão e Spectral.

Neste estudo, optamos por concentrar nossa análise exclusivamente em estatísticas de nível nodal (vértices), uma vez que as estatísticas globais são empregadas para a compreensão do grafo como um todo e para comparações entre diferentes grafos ou subgrafos. As estatísticas de nível nodal (vértices) fornecem uma visão da importância e do papel de cada nó (vértice) na estrutura da rede.

Com a utilização dessas estatísticas de nível nodal (vértices), procuramos identificar os nós (vértices) mais importantes e influentes da rede, compreender a estrutura e organização da rede, descobrir comunidades e grupos dentro da rede, analisar o fluxo de informação e comunicação na rede. Portanto, essa abordagem nos permite uma avaliação mais detalhada das características específicas de cada nó (vértice) e sua contribuição para a rede.

Somente na ferramenta que suporta essa pesquisa no trabalho com grafos (Nertworkx), existem dezenas de estatísticas de nível nodal (vértice) disponíveis. O presente estudo propõe uma abordagem que reside não na busca exaustiva pelas estatísticas do grafo consideradas ideais, mas, sobretudo, na validação da influência exercida pela incorporação destas ao conjunto de dados original no desempenho dos modelos destinados à tarefa de predição.

Nesse sentido, selecionamos onze estatísticas de nível nodal (vértices) para análise. Medidas de Centralidade: *Degree* (DG), *Degree Centrality* (DC), (*Average Neighbor Degree* (AN), *Betweenness Centrality* (BC), (*Closeness Centrality* (CC), *Eigenvector Centrality* (EC), *Load Centrality* (LC); Medidas de Estrutura Local: *Clustering* (CL) e *Triangles* (TR); Medidas de Análise de Links: Pagerank (PR) e Hits (HT).

Cabe ressaltar que, em situações normais de uso, é importante escolher as estatísticas mais adequadas para cada caso, levando em consideração o tipo de análise que se deseja realizar, as características e estrutura do grafo, o esforço de tempo de cálculo dessas estatísticas e os objetivos da pesquisa.

No Apêndice B descrevemos os detalhes de cada uma dessas estatísticas do grafo selecionadas e a lógica por trás de aproveitá-las.

As estatísticas do grafo selecionadas nesta pesquisa cobrem diferentes aspectos da importância e conectividade dos vértices na rede. Essa seleção visa alcançar uma variedade que permita uma análise mais completa da estrutura da rede e seus impactos nos modelos de ML.

2.5.2 Seleção de Características

A incorporação de novas características derivadas de um grafo em um conjunto de dados representa uma extensão dos procedimentos tradicionais de preparação de dados para modelos de aprendizado de máquina. Como observado por Kotsiantis (KOTSIANTIS, 2007), essa expansão pode introduzir complexidade adicional ao processo de análise de dados. Portanto, é imperativo explorar abordagens que otimizem o desempenho dos modelos de ML ao considerar a inserção de novas camadas de características. Nesse contexto, a seleção de características relevantes emerge como uma estratégia viável para mitigar os desafios de dimensionamento e aumentar a eficiência computacional.

A seleção de características, de acordo com Guyon e Elisseeff (GUYON; ELISSEEFF, 2003), é um processo no qual um conjunto de atributos ou características relevantes é escolhido a partir de um conjunto maior de atributos. Ainda segundo Guyon e Elisseeff (GUYON; ELISSEEFF, 2003) o principal objetivo é melhorar o desempenho de modelos de aprendizado de máquina, reduzir a dimensionalidade dos dados e eliminar atributos redundantes, irrelevantes ou ruidosos. Isso se alinha com o nosso objetivo de aliviar a carga computacional introduzida pelo enriquecimento do conjunto de dados com informações oriundas do grafo e, dessa forma, melhorar a precisão e o desempenho do modelo.

Existem várias estratégias de seleção de características, cada uma com suas abordagens e métodos específicos. Daya et al. (DAYA et al., 2020) e Saeys (SAEYS; INZA; LARRAñAGA, 2007) apontam que, em geral, as técnicas de seleção de características podem ser divididas com base em três diferentes estratégias ou categorias de seleção: métodos de filtro, *wrapper* e *embedded*.

A técnica de seleção adotada nessa pesquisa foi a de filtro devido a sua independência de um modelo de ML específico, rápido processamento e sua utilização em artigos relacionados na seleção de estatísticas extraídas de grafo.

No Apêndice C detalhamos os fundamentos de cada uma dessas técnicas de seleção de características.

Neste estudo, temos o interesse de investigar o efeito do enriquecimento de conjuntos de dados por meio da inclusão de características derivadas de estatísticas do grafo. Com este propósito, restringimos a aplicação de

técnicas de seleção de características exclusivamente às atribuições originadas do contexto do grafo. Esta abordagem tem por finalidade proporcionar um forte controle sobre os resultados obtidos a partir da técnica de enriquecimento de dados, garantindo que quaisquer melhorias observadas sejam efetivamente atribuídas à incorporação das características derivadas do grafo.

2.6 Aprendizado de Máquina

Aprendizado de Máquina (*Machine Learning*, em inglês ou simplesmente ML) é a ciência (e a arte) da programação de computadores para que eles possam aprender com os dados. Como exemplo o filtro de spam é um programa de Aprendizado de Máquina que pode aprender e assinalar e-mails como spam e como regulares (GERON, 2019, p.4).

Mitchel, T.M. (MITCHELL, 1997), se refere ao aprendizado de máquina como o campo da Inteligência Artificial responsável pelo desenvolvimento de modelos (hipóteses) gerados a partir de dados, e que automaticamente aperfeiçoam-se com a experiência.

O Aprendizado de Máquina pode ser separado em tipos, segundo critérios não exclusivos e que podem ser combinados, classificados em extensas categorias (GERON, 2019, p.8):

- Serem ou não treinados com supervisão humana (supervisionado, não supervisionado, semi-supervisionado e aprendizado por reforço);
- Se podem ou não aprender rapidamente, de forma incremental (aprendizado online versus aprendizado por lotes);
- Se funcionam simplesmente comparando novos pontos de dados com pontos de dados conhecidos, ou se detectam padrões em dados de treinamento e criam um modelo preditivo, como os cientistas (aprendizado baseado em instâncias versus aprendizado baseado em modelo).

Nessa pesquisa, vamos nos concentrar no Aprendizado de Máquina do tipo Supervisionado baseado em modelo.

No aprendizado Supervisionado, os dados de treinamento que você fornece ao algoritmo incluem as soluções que se desejam predizer, chamadas rótulos (GERON, 2019, p.8).

Escovedo, T. e Koshiyama, A. (ESCOVEDO; KOSHIYAMA, 2020) ainda completam que o Aprendizado Supervisionado é geralmente utilizado em problemas de Classificação e Regressão.

O filtro de spam é um bom exemplo de um problema de Classificação, pois ele é treinado com muitos exemplos de e-mails junto às classes (spam ou não spam) e deve aprender a classificar novos e-mails (GERON, 2019, p.4).

Um exemplo de problema de Regressão é a predição do valor estimado das vendas em uma nova filial de uma determinada cadeia de lojas (ESCOVEDO; KOSHIYAMA, 2020).

Alguns dos algoritmos mais importantes do aprendizado supervisionado são: K-Nearest Neighbours, Regressão Linear, Regressão Logística, Máquinas de Vetores de Suporte (SVM), Arvores de Decisão e Florestas Aleatórias, Redes Neurais (GERON, 2019, p.10).

Nesse estudo vamos nos concentrar na tarefa de classificação, que surge quando o propósito do problema de aprendizado é construir um modelo capaz de determinar a classe de uma nova observação dentre um conjunto de classes previamente conhecida. Essa ação se enquadra em um contexto de aprendizado supervisionado em problemas de classificação quando cada observação no conjunto de dados de treinamento possui uma classe previamente identificada associada a ela.

Com exceção das Redes Neurais, todos os algoritmos listados acima, e muitos outros, serão objeto desse estudo na construção de modelos capazes de uma boa generalização, utilizando a técnica proposta nesse estudo.

2.7 Algoritmos de Classificação

Um algoritmo de classificação, no contexto de aprendizado de máquina, refere-se a um procedimento ou método computacional utilizado para criar um modelo capaz de classificar ou categorizar dados em classes predefinidas ou distintas. Esses algoritmos são treinados com conjuntos de dados rotulados, onde os atributos das instâncias estão associados a uma classe conhecida. O objetivo é que o algoritmo aprenda padrões e relações nos dados de treinamento para poder prever a classe de novos dados, com base nas características observadas durante o treinamento.

Os algoritmos de classificação trabalhados nessa pesquisa foram: K-vizinhos mais próximos (KNN), Naïve Bayes (NBG), Naive Bayes Bernoulli (NBB), Máquina de Vetores de Suporte (SVM), Regressão Logística (RLOG), Árvore de Decisão (DT) e Label Propagation (LPA). Pelo lado dos algoritmos *Ensemble* de *Bagging* temos Florestas Aleatórias (RFOR) e Árvores Extras (ETREE). No lado dos algoritmos *Ensemble* de *Boosting* temos *AdaBoost* (ADAB), *Gradiente Boosting* (GB) e *Extreme Gradient Boosting* (XGB). Alguns outros algoritmos foram utilizados mas somente para comparação com

artigos de referência.

No Apêndice D descrevemos os principais conceitos por trás dos algoritmos de aprendizado de máquina utilizados nesse estudo.

2.8 Avaliação de Modelos

Xu Y. e Goodacre R. (XU; GOODACRE, 2018, p.249) consideram que a validação do modelo é a parte mais importante da construção de um modelo supervisionado.

A etapa de validação de um modelo assume um papel importante no desenvolvimento de sistemas de aprendizado supervisionado. A materialização de um modelo que demonstre capacidade de generalização requer a implementação de uma estratégia de particionamento de dados sensata e criteriosa. A adoção de tal estratégia desempenha um papel crucial no processo de validação do modelo, uma vez que, por meio dela, torna-se possível avaliar a robustez e a capacidade de generalização do modelo a novos dados e cenários.

Na primeira subseção, intitulada "Separação dos Dados", serão discutidas as estratégias empregadas na divisão do conjunto de dados, destacando-se a importância desta etapa no contexto da avaliação de desempenho dos modelos. A subsequente subseção, denominada "Dimensionamento dos Conjuntos de Treinamento e Teste", abordará as considerações acerca das proporções ideais de dados a serem alocadas a esses conjuntos, um fator crítico para balancear o poder de generalização e o poder de teste dos modelos. Dando prosseguimento, na terceira subseção, abordamos o “Escalonamento de Características” que é uma das transformações mais importantes que devemos efetuar nas características numéricas do conjunto de dados. Na quarta subseção, será apresentada a definição e o estabelecimento das métricas de desempenho dos modelos de classificação, fundamentais para avaliar a precisão, a sensibilidade, a especificidade e outros aspectos-chave do desempenho dos modelos.

Por fim, será apresentada a definição da otimização de hiperparâmetros, suas vantagens e desvantagens e forma de implementação nesse estudo. Esses procedimentos, em conjunto com os procedimentos já descritos com grafo, similaridade e estatísticas do grafo, constituem a espinha dorsal teórica dessa pesquisa, garantindo a solidez e a relevância das análises conduzidas.

2.8.1 Separação dos Dados

A separação de dados é uma técnica comum de validação de modelos, na qual o conjunto de dados é particionado em conjuntos de treinamento e

teste. Modelos são ajustados com o conjunto de treinamento e testados no conjunto de teste. Essa abordagem permite avaliar o desempenho preditivo de modelos sem risco de sobreajuste (*overfiting*) no treinamento, ajudando a medir a capacidade de generalização do modelo para dados novos e não observados (JOSEPH, 2022, p.531).

Imagine que temos um conjunto de dados Z com N objetos (ou instâncias ou exemplos ou amostras), cada um descrito por vetores de recursos (ou atributos ou características) de n dimensões. O objetivo é usar a maior parte dos dados para treinar o classificador e reservar uma porção para avaliar o desempenho em testes mais detalhados. No entanto, usar os mesmos dados para treinamento e teste pode levar a um excesso de ajuste, tornando o classificador incapaz de lidar com dados desconhecidos (KUNCHEVA, 2004, p.9). Portanto, é fundamental manter um conjunto de dados separado para a avaliação final do modelo.

No Apêndice E detalhamos os fundamentos da separação de dados.

Na presente pesquisa, adotamos o método *hold-out* para a separação dos dados em conjuntos de treinamento e teste. Essa abordagem é fundamental para garantir a integridade e a validade dos resultados obtidos durante a avaliação dos modelos de aprendizado de máquina. Ao realizar a separação dos dados dessa maneira, asseguramos que as estatísticas extraídas do grafo de treinamento não estejam disponíveis durante a fase de teste, evitando assim qualquer viés na avaliação dos modelos.

Tal procedimento é crucial para garantir a generalização e a confiabilidade dos modelos desenvolvidos, uma vez que impede que o desempenho do modelo seja superestimado devido à exposição prévia às informações de teste. Ao manter uma separação estrita entre os conjuntos de treinamento e teste, podemos realizar uma avaliação objetiva e imparcial do desempenho dos modelos, fornecendo resultados mais confiáveis e significativos para a validação do método proposto.

Como será discutido posteriormente, submeteremos o método proposto a análises em dez conjuntos de dados distintos. Com o objetivo de avaliar o desempenho do método proposto, planejamos a comparação com e sem as estatísticas do grafo incorporados ao conjunto de dados original e, também, estabelecer uma comparação direta com um estudo de referência que tenha utilizado o mesmo conjunto de dados para a avaliação de sua própria metodologia, conforme documentado no artigo de referência.

Para assegurar a comparabilidade dos resultados, entre outros procedimentos, é fundamental que adotemos a mesma proporção de divisão dos dados utilizada no estudo de referência. Portanto, a proporção de divisão entre o

conjunto de dados original, destinada ao dimensionamento dos conjuntos de treinamento e teste, será idêntica àquela empregada no artigo de referência associado a este conjunto de dados.

2.8.2 Escalonamento de Características

O procedimento de "escalonamento de características" (ou *feature scale*, em inglês) é uma das transformações mais importantes que devemos efetuar nas características numéricas, uma vez que, em sua maioria, algoritmos de Aprendizado de Máquina apresentam desempenho inadequado diante de atributos numéricos de entrada caracterizados por escalas diferentes (GERON, 2019, p.72). Essa técnica refere-se ao processo de ajustar a escala das características (ou atributos ou variáveis ou features, em inglês) em um conjunto de dados para que todas essas características tenham escalas comparáveis (GERON, 2019, p.72).

Segundo Geron, A. (GERON, 2019, p.72), os dois métodos mais comuns de "escalonamento de características" são a padronização (*standardization*) e a normalização (*min-max scaling*).

Padronização (*Standardization*): Consiste em transformar as características de modo que elas tenham uma média zero e um desvio padrão igual a um. Isso é feito subtraindo a média de cada característica e dividindo pelo desvio padrão. É útil quando as características seguem uma distribuição normal.

Formalmente, Raschka, S. (RASCHKA, 2015, p.41) define a Padronização como: Seja x_j a j-ésima característica do conjunto de dados que se deseja padronizar e $x_j^{'}$ o valor padronizado dessa j-ésima característica. Só precisamos subtrair a sua média μ_j e dividi-la pelo seu desvio padrão σ_j:

$$x_j^{'} = \frac{\mu_j}{\sigma_j}, \tag{2-1}$$

onde j=1,2,...,n e n é o número de instâncias nesse conjunto de dados.

Normalização (*Min-Max Scaling*): Redimensiona as características para um intervalo específico, geralmente entre 0 e 1. Para fazer isso, as características são ajustadas de modo que o valor mínimo se torne 0 e o valor máximo se torne 1. Adotamos neste estudo o método de Padronização (*Standardization*) que se ajusta melhor as características dos conjuntos de dados que vamos utilizar e, também, foi a técnica adotada nos artigos de *benchmark* selecionados para comparação.

Cabe ressaltar que o procedimento de Padronização é ajustado com base no conjunto de dados de treinamento, sendo então apenas a transformação aplicada no conjunto de dados de teste.

O tipo de escalonamento de características adotado deve levar em consideração os dados e modelos de ML a serem utilizados. A adoção de medidas de escalonamento de características especificamente voltadas para as características de cada conjunto de dados e modelo de ML pode influenciar positivamente o desempenho dos modelos de ML. Nesse estudo, por uma questão de simplificação e comparação, adotamos a Padronização como escalonamento de características para todos os conjuntos de dados.

2.8.3 Métricas de Desempenho em Modelos de Classificação

Para avaliar o desempenho dos modelos de classificação de aprendizado, precisamos definir quais são as métricas que desejamos adotar para efetuar a medição de desempenho (ou performance) de modo a avaliar o resultado da predição de um modelo.

É importante escolher a métrica certa ao avaliar modelos de aprendizado de máquina. Nesse sentido, utilizamos as métricas de acurácia, precisão, sensibilidade (*recall*) e F_1 como medidas principais para avaliar o desempenho dos modelos.

É importante destacar que, ao lidar com problemas de classificação com múltiplas classes, optamos por utilizar a média ponderada como padrão para calcular essas métricas, garantindo uma avaliação abrangente e ponderada do desempenho dos modelos em todas as classes. Essas métricas fornecem uma compreensão detalhada da capacidade preditiva dos modelos em diferentes aspectos, permitindo uma análise abrangente e precisa do desempenho em diversos cenários e contextos.

Para informações mais detalhadas sobre as métricas de desempenho utilizados nessa pesquisa, consulte o Apêndice F.

2.8.4 Otimização de Hiperparâmetros

De acordo com Raschka, S. (RASCHKA, 2015, p.185), no contexto do aprendizado de máquina, é importante destacar a existência de dois tipos distintos de parâmetros: aqueles que são iterativamente ajustados com base nos dados de treinamento, como os pesos em modelos de regressão logística, e os parâmetros que não são diretamente determinados a partir dos dados de treinamento, mas sim configurados previamente como parte do processo de otimização do modelo. Esses últimos são frequentemente denominados hiperparâmetros (*hyperparamaters*, em inglês) ou parâmetros de ajuste, e desempenham um papel crítico na conformação do comportamento e desempenho do

modelo. Alguns exemplos incluem o parâmetro de regularização em modelos de regressão logística e o parâmetro de profundidade em árvores de decisão.

Neste trabalho utilizamos uma técnica chamada pesquisa em grade (*grid search* em inglês) para otimizar hiperparâmetros. Essa abordagem pode ajudar a melhorar o desempenho do modelo, pois busca a combinação perfeita de configurações para os hiperparâmetros.

Para implementar essa técnica de pesquisa em grade, estamos utilizando a função GridSearchCV, da biblioteca Scikit-learn (PEDREGOSA et al., 2011), no contexto da análise comparativa dos resultados obtidos pelo método proposto em um cenário específico e, também, em relação a estudos anteriores que adotaram diferentes estratégias para aprimorar o desempenho de modelos de aprendizado de máquina supervisionados de classificação.

Para mais detalhes sobre a otimização de parâmetros e a pesquisa em grade (*grid search*), consulte o Apêndice G.

2.9 Trabalhos Relacionados

Nesta seção exploramos uma revisão da literatura para situar a presente pesquisa no contexto mais amplo do enriquecimento de dados tabulares. Foi realizada uma pesquisa bibliográfica focada no enriquecimento de conjuntos de dados tabulares dos últimos quinze anos, que resultou na seleção de trinta e quatro trabalhos relevantes nesse campo. Esses estudos representam uma amostra significativa da diversidade de abordagens existentes, contribuindo para o desenvolvimento e compreensão neste domínio específico. O levantamento englobou estudos que exploram diferentes abordagens em variadas metodologias, incluindo aquelas centradas na identificação de entidades e outras direcionadas à construção de grafos, enriquecendo assim o panorama de estratégias de enriquecimento de dados.

Para facilitar o entendimento organizamos os trabalhos selecionados em seis categorias principais. São elas:

1. Aumento de tabela, que se refere a tarefa de estender uma tabela inicial com mais dados;

2. pesquisa de tabela, que se refere a tarefa de responder a uma consulta de pesquisa com uma lista classificada de tabelas;

3. Interpretação de tabela (metadados e semânticas), com trabalhos que procuraram desvendar a semântica dos dados contidos em uma tabela, com o objetivo de tornar os dados tabulares processáveis de forma inteligente por máquinas;

4. Construção de grafo de conhecimento (*knowledge graph*), que se utilizam de dados tabulares para explorar, construir e aumentar bases de conhecimento;

5. Extração de estatísticas de grafo de conhecimento (*knowledge graph*), que se utilizam de dados tabulares para construir grafo de conhecimento com o objetivo de extrair estatísticas desse grafo para enriquecimento dos dados tabulares originais;

6. Extração de estatísticas de grafo de similaridade, que se utilizam de dados tabulares para construção do grafo através da similaridade de suas instâncias, com o objetivo de extrair estatísticas desse grafo para enriquecimento dos dados tabulares originais.

(1) Enriquecimento para o aumento de linhas e/ou pelo aumento de colunas

Na área de enriquecimento de conjuntos de dados tabulares, alguns estudos recentes têm explorado métodos para melhorar a predição de modelos de *machine learning* (ML) por meio da incorporação de dados externos. Dong e Oyamada (2022) (DONG; OYAMADA, 2022) propuseram um método automatizado de enriquecimento de dados tabulares com colunas externas provenientes de data lakes, resultando em melhorias significativas na predição de modelos de ML. Cvetkov-Iliev, Allauzen e Varoquaux (CVETKOV-ILIEV; ALLAUZEN; VAROQUAUX, 2023) abordaram a incorporação de dados relacionais para enriquecer tabelas de dados com fontes externas, utilizando representações vetoriais de entidades baseadas em grafos para capturar informações relevantes.

Zhang e Balog (ZHANG; BALOG, 2017) desenvolveram uma abordagem para equipar programas de planilhas com recursos de assistência inteligente, enquanto Zhang e Chakrabarti (ZHANG; CHAKRABARTI, 2013) apresentaram o InfoGather+, um sistema para automatizar a coleta de informações sobre entidades por meio de tabelas web, utilizando modelos probabilísticos generativos e algoritmos eficientes. Lehmberg e Bizer (LEHMBERG; BIZER, 2017) exploraram desafios na correspondência entre tabelas web e bases de dados de conhecimento, propondo uma solução que combina tabelas de diferentes sites antes de realizar a correspondência.

(2) Enriquecimento com pesquisa de tabela

Na área de enriquecimento de dados com pesquisa de tabelas web, vários estudos foram conduzidos. Cafarella, Halevy, e Khoussainova (CAFARELLA; HALEVY; KHOUSSAINOVA, 2009) desenvolveram o WebTables, um sistema que extrai e analisa dados de tabelas HTML na web, permitindo a integração e

análise em grande escala. Pimplikar e Sarawagi (PIMPLIKAR; SARAWAGI, 2012) propuseram um mecanismo de pesquisa estruturado que utiliza tabelas web para fornecer resultados de tabelas em resposta a consultas de palavras-chave. Bhagavatula, Noraset, e Downey (BHAGAVATULA; NORASET; DOWNEY, 2013) desenvolveram o WikiTables, um aplicativo que permite aos usuários explorar o conhecimento tabular extraído da Wikipedia.

Outros estudos incluem Sarma et al. (SARMA et al., 2012), Yakout et al. (YAKOUT et al., 2012), Gupta e Sarawagi (GUPTA; SARAWAGI, 2009), Lehmberg et al. (LEHMBERG et al., 2015), Nargesian et al. (NARGESIAN et al., 2018), e Fetahu et al. (FETAHU; ANAND; KOUTRAKI, 2019), que abordaram diferentes aspectos da pesquisa de tabelas web e sua aplicação em enriquecimento de dados. Outro cenário explorado foi a pesquisa por tabelas em data lakes, com o framework PEXESO proposto por Dong et al. (DONG et al., 2020), que utiliza abordagens baseadas em similaridade de alta dimensão para descobrir tabelas que podem ser unidas de forma eficiente e significativa.

(3) Enriquecimento com interpretação de tabela (metadados e semânticas)

Na área de enriquecimento de metadados de tabelas web, Venetis et al. (VENETIS et al., 2011) desenvolveram um sistema que recupera a semântica de tabelas web, facilitando operações como busca e localização de tabelas relacionadas. Wang et al. (WANG et al., 2021) propuseram o Table Convolutional Network (TCN), um modelo que extrai informações de tabelas relacionais usando informações contextuais intra e inter-tabelas. Steenwinckel et al. (STEENWINCKEL et al., 2019) apresentaram o CSV2KG, um sistema que transforma dados tabulares em conhecimento semântico na DBPedia, permitindo a integração e reutilização de dados em diferentes contextos.

Outras iniciativas incluem Chabot et al. (CHABOT et al., 2019), que desenvolveram o DAGOBAH para a anotação semântica de tabelas com entidades do Wikidata e do DBpedia, e Baazouzi et al. (2022) (BAAZOUZI; KACHROUDI; FAIZ, 2022), que propuseram a FAIRificação de dados tabulares utilizando grafos de conhecimento para melhorar a interoperabilidade e reutilização de dados. Jiomekong and Foko (JIOMEKONG; FOKO, 2022) e Gottschalk and Demidova (GOTTSCHALK; DEMIDOVA, 2022) também exploraram abordagens baseadas em grafos para melhorar a correspondência e interpretação semântica de tabelas.

Tyagi e Jiménez-Ruiz (TYAGI; JIMÉNEZ-RUIZ, 2020) propuseram o LexMa, um sistema que permite a anotação semântica automática de dados tabulares usando um grafo de conhecimento, visando facilitar o acesso semântico às informações em tabelas.

(4) Enriquecimento para construção de grafo de conhecimento (*knowledge graph* - KG)

Trabalhos recentes, como os de Abdelmageed (ABDELMAGEED, 2020), Bach e Badaskar (BACH; BADASKAR, 2007), e Detroja, Bhensdadia e Bhatt (2023) (DETROJA; BHENSDADIA; BHATT, 2023), exploraram métodos de extração de relações em dados tabulares. Enquanto Abdelmageed propõe um framework de aprendizado de máquina para integrar dados tabulares em um grafo de conhecimento, Bach e Badaskar investigaram técnicas de extração de relações sem a necessidade de rotular os dados. Por outro lado, Detroja, Bhensdadia e Bhatt destacaram abordagens baseadas em *Deep Learning* (DL).

O framework de Abdelmageed consiste em três módulos principais para prever conceitos, detectar relações e construir o grafo de conhecimento, visando reduzir o esforço manual na integração de dados relevantes para pesquisas científicas.

(5) Extração de Estatísticas de Grafo de Conhecimento (*knowledge graph*)

Vários estudos recentes abordam a extração de estatísticas de grafos de conhecimento a partir de dados tabulares para enriquecer conjuntos de dados originais. Sanz e Duarte (SANZ; DUARTE, 2019) propõem uma abordagem para detecção de intrusão em redes virtuais, enquanto Baumann et al. (2017) (BAUMANN et al., 2017) exploram a construção de grafos de sessões de usuário na web com base em dados de clickstream. Por sua vez, Gulum (GULUM, 2018) investiga o uso de características extraídas de grafos para prever resultados de corridas de cavalos.

Bryan e Moriano (BRYAN; MORIANO, 2023) apresentam um modelo de aprendizado de máquina baseado em grafos para prever defeitos em desenvolvimento de software em tempo real, enquanto Poenaru-Olaru (POENARU-OLARU, 2018) propõe uma técnica para previsão de pontuações de crédito usando características extraídas de grafos. Alharbi e Alsubhi (ALHARBI; ALSUBHI, 2021) propõem um modelo de detecção de botnets baseado em aprendizado de máquina que incorpora características estatísticas extraídas de grafos construídos a partir de fluxos de rede.

Os trabalhos acima demonstram a aplicabilidade e eficácia da extração de estatísticas de grafos de conhecimento para diversas aplicações, desde segurança de rede até previsão de resultados e detecção de padrões.

(6) Extração de estatísticas de grafo de similaridade

A extração de estatísticas de grafo de similaridade é uma técnica que utiliza dados tabulares para construir grafos com base na similaridade entre suas instâncias, visando enriquecer os dados originais. Em um estudo no

campo da saúde, Zaki, Mohamed e Habuza (ZAKI; MOHAMED; HABUZA, 2021) propõem uma abordagem baseada em correlação para converter dados tabulados em grafos de conhecimento, capturando correlações estruturais entre os dados e melhorando o desempenho de modelos de classificação em conjuntos de dados de saúde.

Por outro lado, em um contexto educacional, Albreiki, Habuza e Zaki (ALBREIKI; HABUZA; ZAKI, 2023) investigam a identificação de estudantes em risco de desempenho acadêmico usando características topológicas extraídas de grafos construídos a partir de dados tabulares. Eles utilizam uma rede convolucional de grafos para prever o desempenho dos alunos, demonstrando a eficácia dessa abordagem na detecção precoce de problemas acadêmicos e no desenvolvimento de intervenções preventivas.

Em ambos os estudos, o grafo e a extração de suas estatísticas é efetuado com base em todo o conjunto de dados tabular e, em sequência, a técnica de validação cruzada foi empregada para o treinamento e teste dos modelos de aprendizado de máquina.

Um detalhamento de todos os trabalhos relacionados em cada uma dessas categorias está disponível no Apêndice H

Na Tabela 2.1 apresentamos uma visão geral dos trabalhos relacionados.

Tabela 2.1: Visão Geral dos Trabalhos Relacionados

#	Referência	Categ.
1	(ABDELMAGEED, 2020)	(4)
2	(ALBREIKI; HABUZA; ZAKI, 2023)	(6)
3	(ALHARBI; ALSUBHI, 2021)	(5)
4	(BAAZOUZI; KACHROUDI; FAIZ, 2022)	(3)
5	(BACH; BADASKAR, 2007)	(4)
6	(BAUMANN et al., 2017)	(5)
7	(BHAGAVATULA; NORASET; DOWNEY, 2013)	(3)
8	(BRYAN; MORIANO, 2023)	(5)
9	(CAFARELLA; HALEVY; KHOUSSAINOVA, 2009)	(2)
10	(CHABOT et al., 2019)	(3)
11	(CVETKOV-ILIEV; ALLAUZEN; VAROQUAUX, 2023)	(1)
12	(SARMA et al., 2012)	(2)
13	(DETROJA; BHENSDADIA; BHATT, 2023)	(4)
14	(DONG et al., 2020)	(2)
15	(DONG; OYAMADA, 2022)	(1)
16	(FETAHU; ANAND; KOUTRAKI, 2019)	(2)
17	(GOTTSCHALK; DEMIDOVA, 2022)	(3)
18	(GULUM, 2018)	(5)
19	(GUPTA; SARAWAGI, 2009)	(2)
20	(JIOMEKONG; FOKO, 2022)	(3)
21	(LEHMBERG; BIZER, 2017)	(1)
22	(LEHMBERG et al., 2015)	(2)
23	(NARGESIAN et al., 2018)	(2)
24	(PIMPLIKAR; SARAWAGI, 2012)	(2)
25	(POENARU-OLARU, 2018)	(5)
26	(SANZ; DUARTE, 2019)	(5)
27	(STEENWINCKEL et al., 2019)	(3)
28	(TYAGI; JIMÉNEZ-RUIZ, 2020)	(3)
29	(VENETIS et al., 2011)	(3)
30	(WANG et al., 2021)	(3)
31	(YAKOUT et al., 2012)	(2)
32	(ZAKI; MOHAMED; HABUZA, 2021)	(6)
33	(ZHANG; CHAKRABARTI, 2013)	(1)
34	(ZHANG; BALOG, 2017)	(1)

3
Método Proposto

Apresentamos neste capítulo nosso método para enriquecer conjuntos de dados tabulares com informações de grafo, visando aprimorar o desempenho de modelos de Aprendizado de Máquina Supervisionado de Classificação.

Primeiro será apresentada uma visão geral e o fluxo básico do método, destacando os seus principais componentes. A seguir, serão exploradas em profundidade as seis etapas distintas que compõem o processo, cada uma desempenhando um papel fundamental na consecução dos objetivos propostos. A compreensão detalhada dessas etapas é essencial para um entendimento abrangente da aplicação prática do método e dos benefícios que podem ser alcançados por meio de sua utilização em projetos de ML.

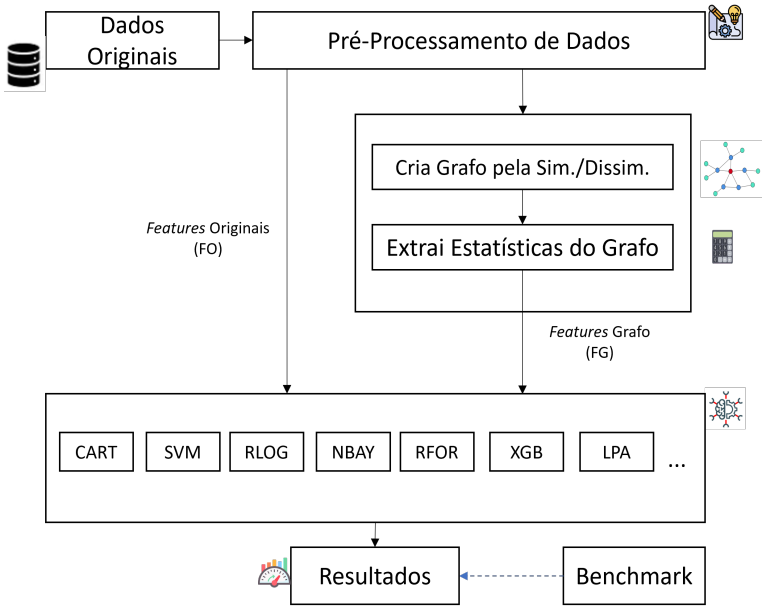
3.1
Visão Geral

Iniciamos com uma descrição geral das etapas do processo, introduzindo a arquitetura que direciona nossa solução de enriquecimento dos dados e destaca sua importância na otimização dos modelos.

Como já citado, nosso objetivo é o de propor um método para enriquecimento de conjuntos de dados tabulares com características extraídas do grafo de similaridade, gerado a partir desse mesmo conjunto dados, visando o aprimoramento do desempenho de modelos de Aprendizado de Máquina Supervisionados de Classificação.

O pipeline utilizado nessa técnica está alinhado ao pipeline padrão para a predição de modelos de aprendizado de máquina supervisionados como aquele apresentado por (ESCOVEDO; KOSHIYAMA, 2020, p.10) para projetos de ciência de dados, mas com algumas particularidades. A Figura 3.1 representa uma visão geral do método proposto. De uma maneira geral, nosso método segue as etapas de captura dos dados originais, pré-processamento, submissão do conjunto de dados original (FO) aos diversos modelos de Aprendizado de Máquina Supervisionados de Classificação, criação do grafo e extração de estatísticas, incorporação dessas novas características ao conjunto de dados original, submissão do novo conjunto de dados enriquecido aos diversos modelos de ML também utilizados para o conjunto de dados original (FO) e Avaliação dos resultados individualmente e contra um *benchmark* selecionado.

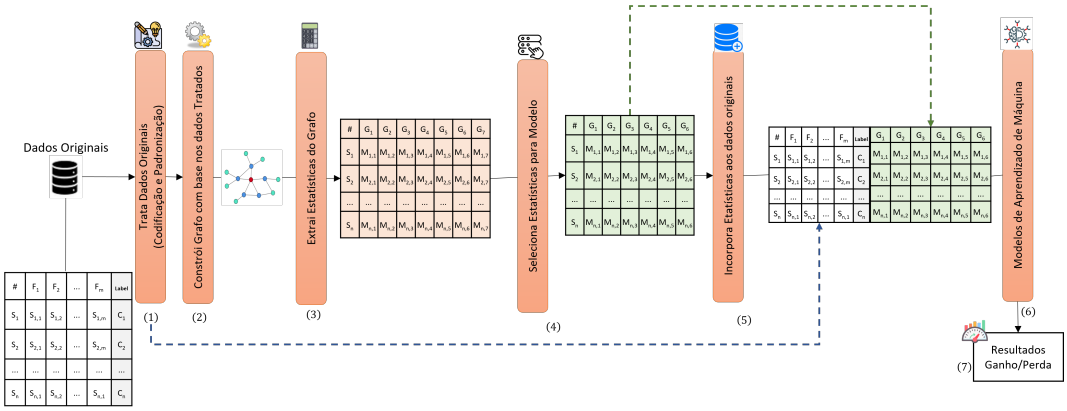
Figura 3.1: Metodologia - Visão Geral



3.2 Fluxo Básico

Avançando um pouco mais no detalhamento com a Figura 3.2, que ainda apresenta uma visão geral do fluxo seguido pelo método proposto, verificamos que o conjunto de dados tabulares rotulados originais são submetidos a um fluxo geral, a saber: Iniciando com um tratamento de padronização e codificação (1), logo após começamos o processo de construção do grafo, com base nesses dados tratados, com a criação pela similaridade/dissimilaridade (2), em seguida vem a extração das estatísticas do grafo criado (3), prosseguindo, procedemos com a seleção das estatísticas para processamento (4), incorporação dessas estatísticas extraídas do grafo como novas características ao conjunto de dados original (5) para, então, efetuar a submissão desse conjunto de dados enriquecido aos modelos de Aprendizado de Máquina Supervisionado (6), concluído com a avaliação dos resultados computando as perdas e ganhos derivados dessa incorporação de novos dados (7).

Figura 3.2: Fluxo Básico das Etapas do Método



Essa visão geral já nos permite visualizar alguns aspectos importantes na técnica, tais como: Os dados submetidos à criação do grafo base serão os dados tratados e esses dados tratados serão aqueles que serão enriquecidos com as estatísticas extraídas do grafo. Também é possível verificar que existe um procedimento para filtrar quais das estatísticas extraídas serão selecionadas, de fato, para participar do enriquecimento do conjunto de dados original para entrada nos modelos.

3.3 Etapas do Processo (*Pipeline*)

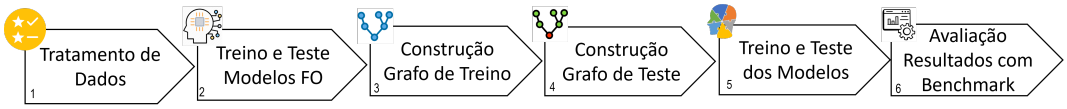
Percebe-se que, até o momento, o método proposto teria quatro grandes etapas, a saber: Tratamento de Dados, Construção do Grafo, Execução dos Modelos de Predição e Avaliação dos Resultados. No entanto, como já vimos no Capítulo 2 Seção 2.8.1, um procedimento importante de todo processo de predição em modelos de Aprendizado de Máquina Supervisionado é a separação dos dados originais em conjuntos de treino e teste. Essa separação recebeu uma atenção especial, pois é importante isolar do treinamento dos modelos os dados que serão utilizados para o teste e avaliação desses modelos. Isto posto, a etapa de criação do grafo será separada em duas etapas, sendo uma para criação do grafo voltado para treino e outro somente para criação do grafo de teste que será base para as avaliações de performance dos modelos e comparações.

Outra etapa que precisou ser separada foi a etapa de Execução dos Modelos de Predição, pois é importante a submissão desse conjunto de dados original (FO) para os modelos de predição sem ainda incorporar as estatísticas extraídas do grafo, de modo a constituir com esses resultados uma base de comparação (*baseline*) com os resultados que serão produzidos com esse mesmo

conjunto de dados enriquecido das características com as estatísticas extraídas do grafo.

Portanto, o *pipeline* do método proposto passa a ter seis grandes etapas, a saber: Tratamento de Dados, Treino e Teste Modelos FO, Construção Grafo de Treino, Construção Grafo de Teste, Treino e Teste Modelos Grafo e Avaliação Resultados com *Benchmark*, conforme consta da Figura 3.3.

Figura 3.3: Metodologia - 6 Etapas do Processo



Existem alguns outros pontos importantes que não constam do Fluxo Básico apresentado na Figura 3.2, tais como os critérios para criação do grafo com base nos dados originais, quais a estatísticas extraídas do grafo etc. Esses detalhes e muitos outros serão detalhados à seguir, quando abordarmos cada uma dessas etapas que compõe o método proposto nesse estudo.

3.3.1 Etapa 1 - Tratamento de Dados

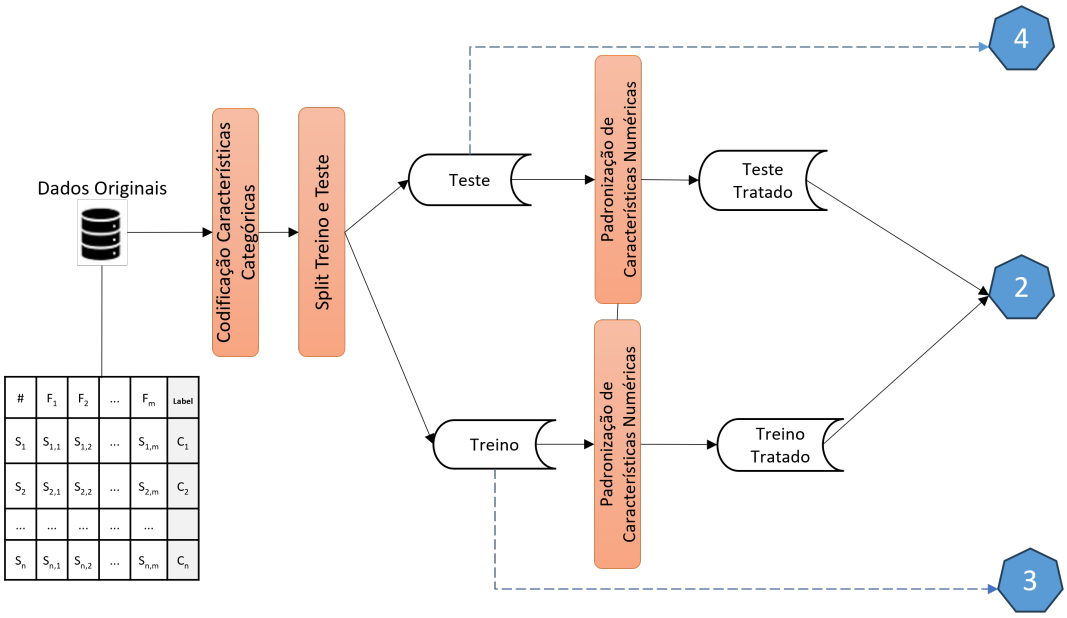
Nesta etapa, nos concentramos em procedimentos essenciais de preparação e tratamento do conjunto de dados tabular, com as características originais (FO), antes de sua inclusão no processo de enriquecimento. Exploramos transformações específicas e procedimentos necessários para aprimorar a qualidade dos dados.

Conforme apresentado na Figura 3.4, inicialmente, as características categóricas, se houverem, passarão por um processo de codificação onde cada categoria identificada será convertida em um número. Se a característica categórica já estiver expressa como um número, então ela será mantida.

Com os dados codificados, efetuamos a separação do conjunto de dados em conjunto de treino e conjunto de teste. Uma vez que vamos comparar os resultados desta proposta com outras técnicas que utilizaram o mesmo conjunto de dados, então, para manter um nível de comparação adequado, a razão de separação do conjunto de treino e teste deve ser a mesma. Isto posto, na separação do conjunto de treino e teste seguiremos a proporção utilizada no artigo de *benchmark* selecionado para comparação.

Separados os dois conjuntos de treino e teste, se houver característica numérica, então será efetuado o chamado procedimento de “escalonamento de

Figura 3.4: Etapa 1 – Tratamento de Dados



características" (ou feature scale, em inglês) com a Padronização dos dados, conforme abordado no Capítulo 2 Seção 2.8.1.

Cabe ressaltar que o procedimento de padronização é ajustado com base no conjunto de dados de treinamento, sendo então apenas a transformação aplicada no conjunto de dados de teste.

É importante destacar que o conjunto de dados será submetido a nenhuma outra transformação além daquelas já citadas para não influenciar no processo de comparação de resultados.

Agora que os conjuntos de dados de treinamento e teste foram tratados, podemos seguir para submeter esses dados aos modelos de predição, o que será abordado na etapa seguinte.

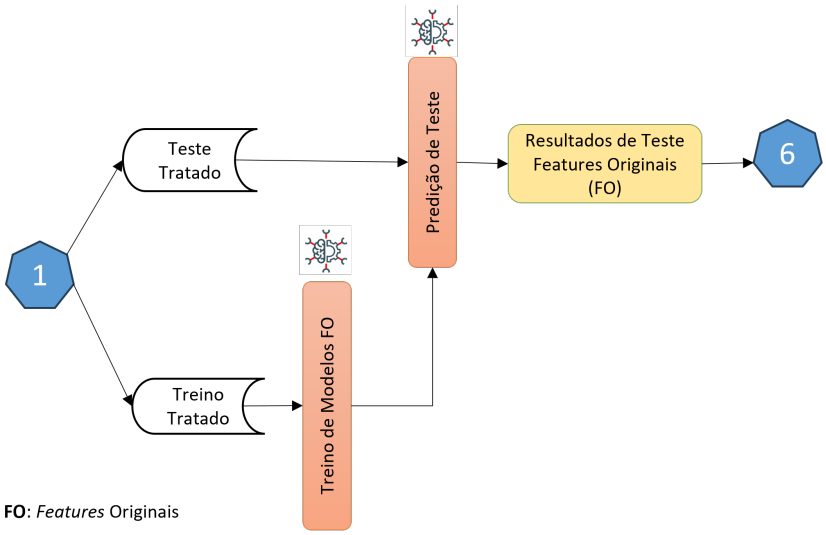
3.3.2 Etapa 2 - Treino e Teste Modelos FO

Com os conjuntos de dados de treinamento e teste tratados, inicia-se o procedimento de treinamento dos modelos de predição selecionados.

A submissão desse conjunto de dados tratado para os modelos de predição, tem como objetivo constituir, com os resultados destes, uma base de comparação (*baseline*) com os resultados que serão produzidos a partir desse mesmo conjunto de dados enriquecido com as estatísticas extraídas do grafo.

A etapa se inicia, conforme apresentado na Figura 3.5, com o treinamento dos modelos com base no conjunto de treinamento tratado na etapa anterior (vide Figura 3.4). Os modelos, assim como os parâmetros (ou hiperpârametros)

Figura 3.5: Etapa 2 – Treino e Teste Modelos FO (*baseline*)



utilizados, serão abordados no Capítulo 4 Seção 4.3 onde verificamos os detalhes do experimento do método proposto.

De modo a evitar uma eventual introdução de tendência (viés) nos resultados, os modelos serão processados com parâmetros fixos, isto é, esses parâmetros não passarão pelo processo cíclico em que os parâmetros (ou hiperpâarametros) são ajustados manualmente na busca pelos melhores resultados (conhecido como *tunning*, em inglês). Essa abordagem se justifica porque não estamos buscando a melhor performance dos modelos, mas, sim, queremos observar o impacto do enriquecimento dos dados na performance dos modelos, então a premissa é que a única mudança entre um teste e outro seja o enriquecimento do conjunto de dados com as estatísticas extraídas do grafo.

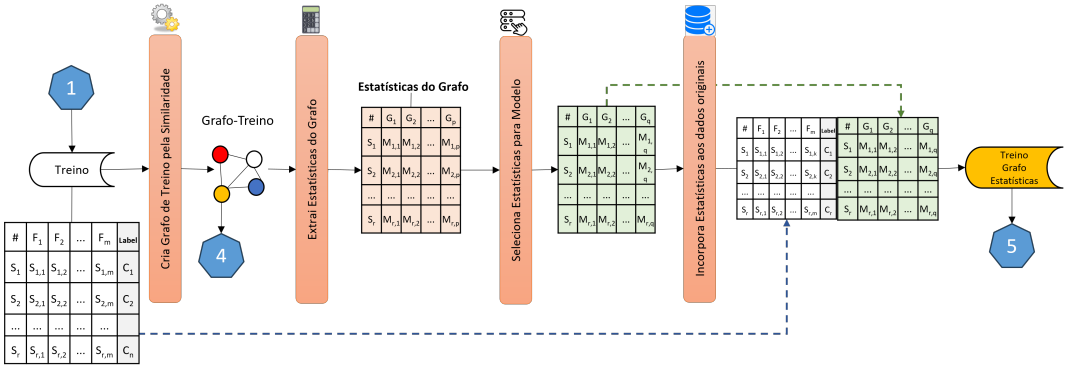
As métricas de performance utilizadas para avaliação da performance da predição dos modelos são aquelas abordadas no Capítulo 2 Seção 2.8.3, isto é, Acurácia, Precisão, Sensibilidade (ou *Recall*) e F_1.

Uma vez que o modelo foi treinado, este segue para ser submetido ao conjunto de teste tratado na etapa anterior e os resultados com as métricas de performance de teste seguem para a etapa final de avaliação de resultados.

3.3.3 Etapa 3 - Construção Grafo de Treino

Nessa etapa efetuaremos os procedimentos para construção do grafo de treino, tendo como base o conjunto de dados de treino pela similaridade de suas instâncias, a extração das estatísticas do grafo de treino, a seleção daquelas estatísticas extraídas que, de fato, farão parte do novo conjunto de treino e a inserção destas estatísticas selecionadas no conjunto de treino para os modelos de predição.

Figura 3.6: Etapa 3 – Construção Grafo de Treino



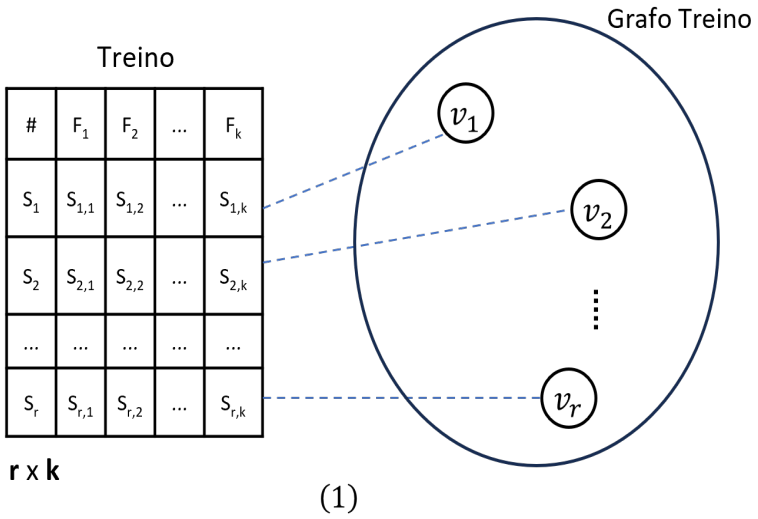
Conforme apresentada na Figura 3.6, a etapa se inicia com a criação do grafo com base nos dados do conjunto treino original tratado. Conforme vimos no Capítulo 2 Seção 2.3, um grafo é composto de duas características básicas: os vértices e a conexão (ou ligação ou relacionamento) entre um vértice e outro vértice, chamada aresta. No nosso caso, os vértices serão representados pelas instâncias do conjunto de dados, isto é, cada vértice do grafo representará uma instância do conjunto de dados de treino. A ligação (aresta) entre cada par de vértices no grafo será efetuada em função da similaridade entre as instâncias do conjunto de dados de treino que representam, isto é, se a medida de similaridade entre um par de instâncias for acima de um valor limite arbitrário (*threshold*, em inglês), então os vértices que representam essas instâncias serão conectados com uma aresta. Vimos também no Capítulo 2 Seção 2.4.3 que essa similaridade tem formas diferentes de medição para características categóricas e numéricas.

Portanto, nosso processo de construção do grafo de treino será efetuado com quatro passos, a saber: (1) criação dos vértices do grafo com base nas instâncias do conjunto de dados de treino; (2) separação das características categóricas e numéricas do conjunto de dados de treino, formando dois subconjuntos distintos, para medição das similaridades em separado; (3) medição das similaridades em cada subconjunto de dados utilizando uma das técnicas de medição de similaridade abordadas no Capítulo 2 Seção 2.4.3 e alinhadas ao tipo que cada subconjunto representa, criando uma matriz de similaridade para cada subconjunto; (4) Estabelecimento das ligações (arestas) dos vértices com base no valor da similaridades entre esses vértices e, também, um limite (*threshold*, em inglês) arbitrário.

O primeiro passo será criar os vértices no grafo com base nas instâncias do conjunto de treino, isto é, cada instância será um vértice no grafo. Portanto, se nosso conjunto de treino tem r instâncias (ou exemplos ou amostras), então

nosso grafo terá r vértices correspondentes, conforme Figura 3.7.

Figura 3.7: Passo de Criação de Vértices Grafo de Treino



Formalmente, do Capítulo 2 Seção 2.3.1, estamos criando o grafo *G=(V,E)* não direcionado, onde os elementos de V são os vértices (ou nós, ou pontos) do grafo G , os elementos de E são as arestas (ou linhas ou conexões) e representamos uma aresta que vai do vértice $u \in V$ ao vértice $v \in V$ como *(u, v)* $\in E$. Se cada instância do conjunto de dados está representada no grafo G como um vértice e o conjunto de dados possui r instâncias (ou linhas), temos $V = \{v_1, v_2, .., v_r\}$ e as arestas $E = e_1, e_2, e_3, ... = (v_1, v_2), (v_1, v_3), (v_2, v_3), ...$ Vamos utilizar uma matriz de adjacência W de peso não negativo, r por r, para descrever G, onde $W = \{w_{ij}\}_{ij=1,2,..,r}$ em que w_{ij} é igual a 0 quando os vértices v_i e v_j não são conectados.

Seja S_i um vetor de características (ou features ou atributos) do objeto i com k características (ou atributos ou features): $s_1 = (s_{11}, s_{12}, ..., s_{1k})$, $s_2 = (s_{21}, s_{22}, ..., s_{2k})$, ..., onde s_{ij} é o valor da j-ésima característica do objeto i, sendo a matriz de dados com r objetos x k características. Portanto, o vértice $v_i \in V$ representa a instância i do conjunto de dados de treino no grafo G.

O segundo passo nessa Etapa, será separar as características (ou features ou atributos) do conjunto de dados tratado de treino em dois subconjuntos distintos, com um subconjunto de características categóricas e um subconjunto de características numéricas, conforme Figura 3.8.

Formalmente, a matriz de dados de treino com r objetos x k características (ou features ou atributos) foi separada em duas matrizes, uma matriz de dados de características categóricas e outra matriz de dados de características numéricas. A matriz de dados de características categóricas com r objetos x c características e a matriz de dados de características numéricas com r objetos

x n características, onde c é a quantidade de características categóricas e n a quantidade características numéricas, sendo $c+n=k$.

Um pilar central de nosso método é a criação de conexões no grafo. Detalhamos a seguir os procedimentos para obter as medidas de similaridade e dissimilaridade, aplicadas aos dados para criar uma representação significativa no grafo de treino.

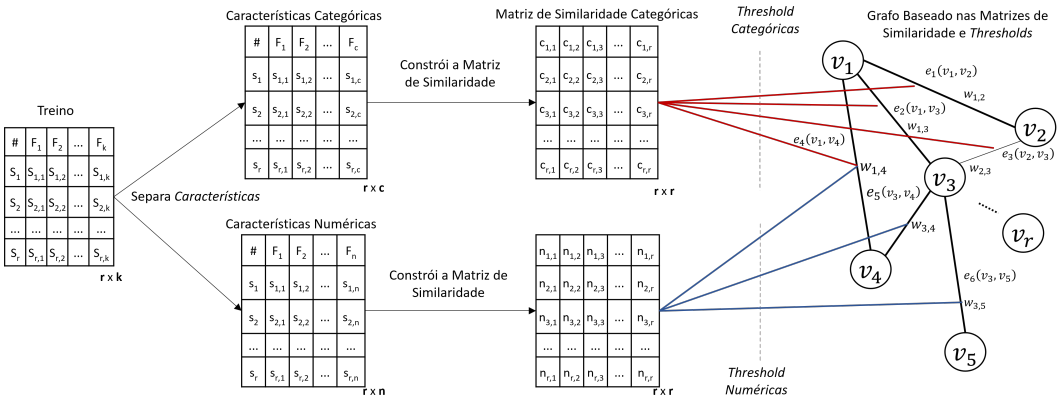
Efetuamos a medição da similaridade, entre os objetos da matriz de dados de características categóricas, utilizando uma medida de similaridade voltada para esse tipo de característica, criando uma matriz quadrada de similaridade de características categóricas, contendo os valores de similaridade entre os objetos da matriz de dados de características categóricas. Conforme abordamos no Capítulo 2 Seção 2.4.3, as medidas de similaridade que estamos trabalhando para características categóricas são: *Jaccard*, *Overlap* e *Eskin*. O mesmo procedimento é efetuado para a medição da similaridade entre os objetos da matriz de dados de características numéricas, criando uma matriz quadrada de similaridade de características numéricas, contendo os valores de similaridade entre os objetos da matriz de dados de características numéricas. As medidas de similaridade que estamos trabalhando para as características numéricas são: *Euclidean*, *Cosine*, *Manhattan* e *Mahalanobis*.

Conforme vimos no Capítulo 2 Seção 2.4.1, uma característica da matriz de similaridade é a simetria, isto é, os elementos na posição *[i, j]* são iguais aos elementos na posição *[j, i]*. Outra característica dessa matriz é que os elementos na diagonal principal são iguais a 1, isto é, os elementos *[i,i]* são iguais a 1, representando a similaridade de um objeto com ele mesmo, e os valores nessa matriz geralmente estão na faixa de 0 a 1, onde 0 indica ausência de similaridade (ou dissimilaridade completa) entre os objetos e 1 indica similaridade completa. Finalmente, a matriz de similaridade é uma matriz quadrada, portanto, no nosso caso, a matriz de similaridade de características categóricas e a matriz de similaridade de características numéricas tem r objetos x r objetos, isto é, uma matriz quadrada de ordem r.

Formalmente as Matrizes de Similaridade Categóricas e Numéricas serão representadas por $C = \{c_{11}, ..., c_{rr}\}$ e $N = \{n_{11}, ..., n_{rr}\}$, respectivamente.

Pela Figura 3.8, a partir das Matrizes de Similaridade estabelecemos a conexão entre os pares de vértices do grafo de treino desde que maior que um limite arbitrário (*threshold*), isto é, dois vértices são conectados (arestas) se a similaridade entre os elementos [*i, j*] entre os pontos de dados correspondentes x_i ,x_j for positivo e maior que um determinado limite. Para determinar esse limite, adotaremos uma abordagem semelhante à utilizada por Albreiki, B., Habuza, T. e Zaki, N. em estudo intitulado "Extracting topological features

Figura 3.8: Passos para construção Arestas Grafo de Treino



to identify at-risk students using machine learning and graph convolutional network models" (ALBREIKI; HABUZA; ZAKI, 2023), com uma pequena modificação.

No referido trabalho, o limite foi estabelecido como a média dos valores de proximidade entre o vértice i e os demais vértices. No entanto, optaremos por utilizar a mediana devido à sua maior resiliência a valores extremos em comparação com a média, sendo mais adequada para conjuntos de dados assimétricos.

Esta escolha pela mediana é particularmente relevante para o nosso contexto, uma vez que estamos lidando com conjuntos de dados provenientes de diferentes domínios de conhecimento, apresentando desafios como desequilíbrio, variação na quantidade de atributos, entre outros. Assim, o limite a ser estabelecido será a mediana dos valores de similaridade entre o vértice i e os demais vértices. Os pesos das arestas serão as similaridades apuradas em ambos os tipos.

Se a aresta tem origem somente de uma das Matrizes de Similaridade, então o peso será a similaridade que deu origem a aresta. Se a aresta tem origem das duas Matrizes de Similaridade, então o peso será a soma dessas similaridades. Por exemplo, verificamos pela Figura 3.7 que a aresta e_4, que conecta o par de vértices (v_1,v_4), tem similaridade com origem na Matriz de Similaridade de atributos categóricos (C) e da Matriz de Similaridade de atributos numéricos (N), portanto o peso $w_{1,4}$ será a soma das similaridades $c_{1,4}$ e $n_{1,4}$.

Construído o grafo de treino pela similaridade, efetuamos a extração das estatísticas derivadas desse grafo de treino, conforme abordado no Capítulo 2 Seção 2.5. Para cada vértice do grafo, serão extraídas as estatísticas Degree (DG), Degree Centrality (DC), Average Neighbor Degree (AN), Eigenvector

Centrality (EC), Closeness Centrality (CC), Betweenness Centrality (BC), Load Centrality (LC), Local Clustering (CL), Triangles (TR), PageRank (PR) e HITS (HT). Portanto, extraímos 11 estatísticas, para cada vértice do grafo, com potencial para serem incorporadas como novas características ao conjunto de dados de treino original tratado.

Antes da incorporação dessas novas características, efetuaremos o procedimento conhecido como seleção de característica, abordado no Capítulo 2 Seção 2.5.2, onde avaliamos quais delas, de fato, devem ser incorporados ao conjunto de dados original de treino tratado. A técnica de seleção adotada foi a de filtro devido a sua independência de um modelo de ML específico, rápido processamento e sua utilização em artigos relacionados na seleção de estatísticas extraídas de grafo. Nesse processo, serão selecionadas até seis estatísticas que alcançarem as maiores pontuações na técnica de seleção adotada.

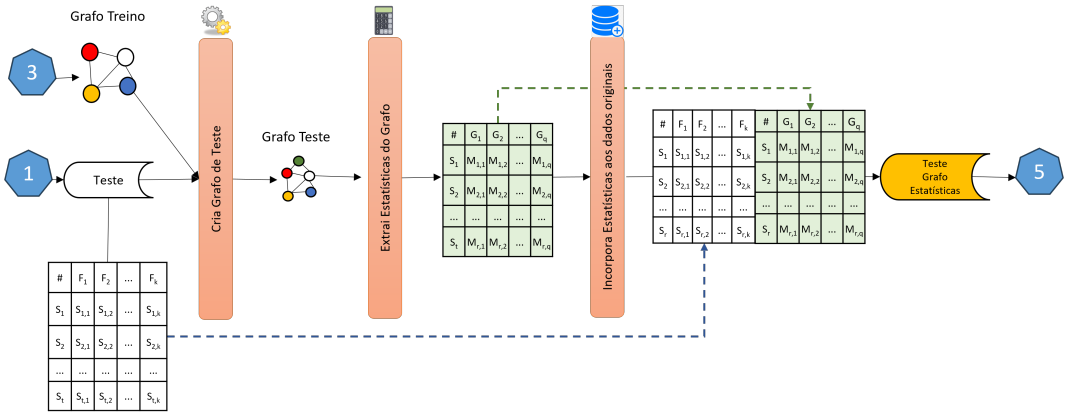
Selecionadas as estatísticas que devem compor o novo conjunto de treino, efetuamos a incorporação dessas estatísticas de cada vértice ao novo conjunto como características novas. Uma vez que cada vértice representa uma instância (ou exemplo ou amostra) do conjunto de dados de treino, então essas estatísticas são acrescentadas a cada instância correspondente no conjunto de dados. Portanto, se o conjunto de dados original tratado possui r instâncias com k atributos cada, o novo conjunto de dados de treino, identificado na Figura 3.6 como Treino Grafo Estatística, ficará com r linhas e até $k+q$ atributos, onde q=1,...,6.

Neste estudo, optamos por explorar uma ampla variedade de medidas de proximidade (3 para características categóricas e 4 para numéricas) e estatísticas do grafo (11 medidas). Essa escolha diversificada reflete a natureza investigativa desta pesquisa acadêmica, na qual buscamos validar um método proposto sob diferentes perspectivas. No entanto, é crucial destacar que, na prática, a escolha de medidas para um conjunto específico de dados pode ser mais focalizada. Compreendemos que a inclusão de múltiplas medidas pode aumentar, e muito, o esforço de tempo no pipeline de treino, e, em cenários aplicados, a seleção cuidadosa de medidas é muitas vezes preferida para otimizar eficiência sem comprometer a qualidade dos resultados.

3.3.4 Etapa 4 - Construção Grafo de Teste

Nessa etapa efetuaremos os procedimentos para construção do grafo de teste, tendo como base o conjunto de dados de teste e o grafo de treino pela similaridade de suas instâncias, a extração das estatísticas do grafo de teste e a inserção destas estatísticas no conjunto de teste para os modelos de predição.

Figura 3.9: Etapa 4 – Construção Grafo de Teste



Os procedimentos para construção do grafo de teste se assemelham a construção do grafo treino, mas, como veremos adiante, existem diferenças importantes para garantir que não exista qualquer relação entre os dados de teste influenciando os dados de treino.

A etapa se inicia, conforme a Figura 3.9, com a construção do grafo de teste se baseando não apenas no conjunto de teste original tratado, mas, também, pelo grafo de treino criado na Etapa anterior. A ideia básica é que as estatísticas para uma instância do conjunto de teste devem refletir a mudança que a introdução desse novo vértice (instância de teste) e suas arestas ocasionarão no grafo de treino.

Para alcançar esse objetivo, foram observados os seguintes passos: (1) Inicialmente o grafo de teste será igual ao grafo de treino; (2) Inclui no grafo de teste os dados do conjunto de teste estabelecendo as similaridades utilizando os mesmos critérios adotados para construção do grafo de treino; (3) Extrai do grafo de teste atualizado as mesmas estatísticas de grafo extraídas no grafo de treino; (4) Inclui no conjunto de dados de teste as estatísticas extraídas do grafo de teste. Ao final desse processo, teremos o conjunto chamado Teste Grafo Estatísticas, que é o conjunto de dados de teste original tratado acrescido das novas características com as estatísticas derivadas do grafo de teste.

Com esse procedimento, procuramos garantir que as instâncias do conjunto de dados de treino, assim como o grafo de treino, não têm nenhum contato com as instâncias do conjunto de teste.

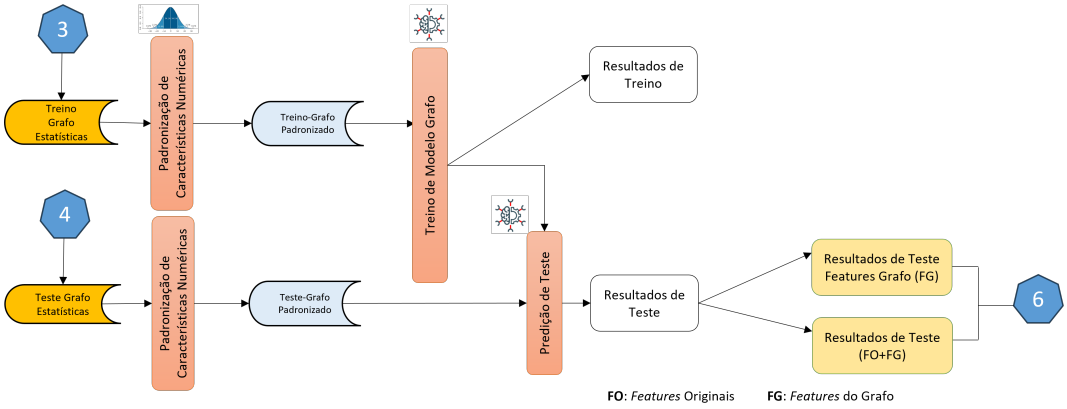
Com os conjuntos de treino e teste tratados e acrescidos das características extraídas dos respectivos grafos, podemos seguir para etapa seguinte de treinamento e teste dos modelos sobre esses conjuntos.

3.3.5
Etapa 5 - Treino e Teste Modelos Grafo

Com os conjuntos de dados de treinamento e teste tratados e com as estatísticas derivadas dos grafos incorporadas como novas características, inicia-se o procedimento de treinamento dos modelos de predição selecionados. Os parâmetros (ou hiperpârametros) utilizados no ajuste desses modelos serão descritos mais a frente quando abordamos os detalhes do experimento. A submissão desses conjuntos de dados enriquecidos com as novas características derivadas das estatísticas do grafo para os modelos de predição, tem como objetivo produzir os resultados que permitem a avaliação do comportamento desses modelos com essas versões dos conjuntos de dados de treino e teste.

A etapa tem início com o procedimento de "escalonamento de características" (Figura 3.10), com as características numéricas de ambos os conjuntos de treinamento e teste (gerados na etapa anterior), separadamente, da mesma forma que efetuamos na Etapa 1 para tratamento dos dados para geração dos resultados do *baseline*. Conforme visto anteriormente, essa técnica refere-se ao processo de ajustar a escala das características (ou atributos ou variáveis ou *features*, em inglês) em um conjunto de dados para que todas essas características tenham escalas comparáveis (GERON, 2019, p.72). Esse procedimento também é conhecido como Padronização dos dados.

Figura 3.10: Etapa 5 – Treino e Teste Modelos Grafo



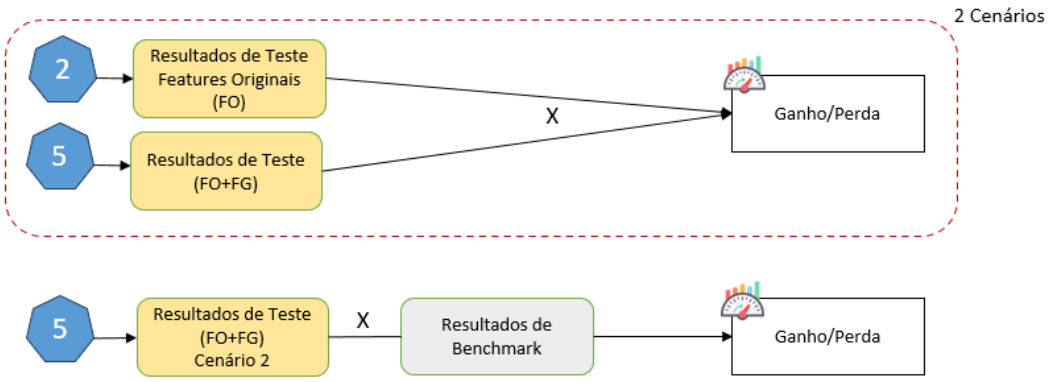
Após a Padronização dos conjuntos de dados de treinamento e teste, inicia-se o procedimento de treinamento dos modelos com base no conjunto de treinamento Padronizado. Naturalmente que os modelos, assim como os parâmetros (ou hiperparâmetros), serão os mesmos que os utilizados na Etapa Etapa 2 (Treino e Teste Modelos FO) Seção 3.3.2), onde foram efetuados o treinamento e teste para geração dos resultados do *baseline*. Da mesma forma, as métricas de desempenho (performane) também serão as mesmas.

Uma vez que o modelo foi treinado, este segue para ser submetido ao conjunto de teste Padronizado e os resultados seguem para a etapa final de avaliação de resultados.

3.3.6 Etapa 6 - Avaliação Resultados *Benchmark*

Nessa etapa (Figura 3.11) efetuamos a avaliação dos resultados dos testes efetuados nas Etapas 2 (Seção 3.3.2) e 5 (Seção 3.3.5), onde teremos os resultados do conjunto de dados de teste com as características originais (*baseline*) sendo confrontadas com os resultados do conjunto de dados de teste enriquecido das características com as estatísticas extraídas do grafo. Essas comparações serão efetuadas em dois Cenários modificando apenas o espaço de parâmetros e a forma como os modelos são tratados. Ao final, os resultados do segundo Cenário são confrontados com a performance de modelos reportados em estudos anteriores que utilizaram diferentes técnicas buscando a melhoria de performance sobre o mesmo conjunto de dados.

Figura 3.11: Etapa 6 – Avaliação Resultados *Benchmark*



Através dessa comparação é que podemos avaliar se a incorporação dessas novas características ajudou a melhorar a performance dos modelos de predição para esse conjunto de dados. No Capítulo 4, efetuamos a descrição da série de experimentos efetuados com diversos conjuntos de dados diferentes aplicando a técnica descrita nessas Etapas em dois Cenários para que possamos avaliar os resultados gerados.

4
Experimentos

Neste capítulo, apresentamos alguns detalhes dos experimentos conduzidos para avaliar o método proposto nesta pesquisa. Começamos discutindo os conjuntos de dados escolhidos para testar o método, destacando os artigos relevantes selecionados como referência para comparação. Em seguida, descrevemos dois cenários de avaliação construídos para testar a eficácia do método em diferentes contextos.

Prosseguindo, detalhamos os parâmetros e hiperparâmetros utilizados nas diversas bibliotecas para implementação do método proposto e dos modelos de aprendizado de máquina, juntamente com as métricas de desempenho selecionadas para avaliar a qualidade dos resultados. Para contextualizar os experimentos, fornecemos informações sobre o ambiente computacional e as bibliotecas de software utilizadas. Este capítulo é fundamental para compreender o rigor e a metodologia aplicados em nossa pesquisa, e como os resultados são avaliados em profundidade.

4.1
Conjuntos de Dados e Referências

No intuito de avaliar a eficácia do método proposto, foram cuidadosamente selecionados dez conjuntos de dados amplamente reconhecidos na comunidade acadêmica. Esses conjuntos de dados abrangem uma diversidade de domínios de conhecimento, variando em dimensões, tamanhos e perfis de balanceamento de classe. A escolha dessa variabilidade estratégica visa proporcionar uma avaliação abrangente e precisa da validade do método proposto. A seleção criteriosa desses conjuntos de dados permite a exploração do desempenho em uma ampla gama de contextos, ampliando a confiabilidade dos resultados obtidos.

Os detalhes de cada conjunto de dados e do artigo de referência associado a cada um desses, estão descritos no Apêndice I

A Tabela 4.1 resume os conjuntos de dados utilizados e algumas de suas características: Dataset: Sigla do Conjunto de Dados; Dom.: Domínio; Linhas: Nr. de linhas; Cols: Numero de Colunas; Cat.: Nr. de Colunas Categoricas; Label: Tipo classe B-Binário M-Multiclasse Num.: Numero de Colunas Numéricas; Bal.: Se a classe é balanceada ou não; Teste: Proporção separada para teste.

Tabela 4.1: Resumo dos Conjuntos de Dados e Características

Dataset	Dom.	Linhas	Cols	Cat.	Num.	Label	Bal.	Teste
BCC	Saúde	116	9	0	9	B	Sim	20%
CAR	Auto	1728	6	6	0	M	Não	30%
DM	Saúde	366	34	33	1	M	Não	23%
HDH	Saúde	294	13	8	5	B	Sim	30%
HDHNI	Saúde	294	2	2	0	B	Sim	30%
PIMA	Saúde	768	8	1	7	B	Não	30%
VCB	Saúde	310	6	0	6	B	Não	50%
VCM	Saúde	310	6	0	6	M	Não	25%
WM	Bebida	178	13	0	13	M	Sim	20%
WQ	Bebida	1599	11	0	11	M	Não	30%

A tabela 4.2 resume os artigos de referência por conjunto de dados e as métricas de resultados contidas nestes que foram utilizadas como *benchmark* nesta pesquisa.

Foram selecionados somente artigos que também utilizam o metodo *holdout* na separação dos conjuntos de treino e teste. A seleção dessa forma visa garantir uma avaliação justa e comparável do desempenho dos modelos de *machine learning* no contexto da avaliação desenvolvida nesse trabalho. A razão de separação dos conjuntos de treino e teste destes artigos foram utilizadas para separação dos respectivos conjuntos de dados desta pesquisa.

Tabela 4.2: Artigos de Referência por Conjuntos de Dados

Dataset	Resultados / Autores
BCC	DT: 66,7%, SVM: 83,3%, RLOG: 75,0%, RFOR: 75,0%, KNN: 89,9%, BAGG_KNN: 100%, BAGG_DT: 100%, BAGG_SVM: 83,3%, BAGG_RLOG: 91,7% e BAGG_RFOR 90,9%. Melhor Modelo BAGG_KNN: 100%.
	(NAVEEN; SHARMA; NAIR, 2019)
CAR	RFOR 93,7%, KNN 93,5% e NBG 69,3%. Melhor Modelo EAC 96%.
	(JALALI; LEAKE; FOROUZANDEHMEHR, 2017)
DM	ANN: 98,0%, SVM: 97%. Melhor Modelo Ensemble ANN/SVM: 99,0%
	(SHARMA; HOTA, 2013)
HDH	DT: 95,5%, RLOG: 98,8%, NBB: 82,8%, RFOR: 99%, GB: 98,8% e MLP: 94,0%. Melhor Modelo RFOR: 99,0%.
	(GARATE-ESCAMILA; Hajjam El Hassani; ANDRES, 2020)
HDHNI	DT: 82,9%, RLOG: 81,9%, NBB: 78,7%, RFOR: 81,3%, GB: 82,3% e MLP: 81,5%. Melhor Modelo: DT 82,9%.
	(GARATE-ESCAMILA; Hajjam El Hassani; ANDRES, 2020)
PIMA	DT: 75,7%, RFOR: 79,6%, NBG: 77,8%. Melhor Modelo RFOR: 79,6%
	(CHANG et al., 2022)
VCB	SVM: 86,9%. Melhor Modelo ANN: 92,1%.
	(ANSARI et al., 2013)
VCM1	RFOR: 98%, ETREE: 99%, RLOG: 85%, ADAB: 95% e SVM: 80%. Melhor Modelo ETREE: 99%.
	(RESHI et al., 2021)
VCM2	RFOR: 86%, ETREE: 85%, RLOG: 93%, ADAB: 86% e SVM: 90%. Melhor Modelo RLOG: 93%.
	(RESHI et al., 2021)
WM	DT: 74,6%, SMV: 71,9%, NBG: 97,3% e MLP: 99,1%. Melhor Modelo MONT: 100%.
	(OJHA; NICOSIA, 2020)
WQ	SVM: 68,6%, NBG: 55,9 e RFOR: 65,5%. Melhor Modelo SVM: 68,6%.
	(KUMAR; AGRAWAL; MANDAN, 2020)

4.2 Cenários de Avaliação

Para atender a algumas questões de pesquisa precisamos montar dois cenários de experimento para avaliação. O primeiro Cenário será voltado para avaliar em que medida a performance dos modelos de predição, com parâmetros (hiperpârametros) fixos e padrão, é afetada pela adição de características derivadas de estatísticas extraídas do grafo de similaridade, considerando conjuntos de dados de diferentes domínios de conhecimento, perfis de classe (binária ou multiclasse), balanceamento, quantidade de atributos, entre outros.

O segundo Cenário se dedica na avaliação de como a performance dos modelos de predição, agora ajustados a um espaço de parâmetros (hiperpârametros) ampliado, após o enriquecimento de características derivadas de estatísticas extraídas do grafo de similaridade, e com uma forma diferente de como os modelos são tratados.

Ambos os cenários sob análise se fundamentam na criação de ambientes experimentais caracterizados pelo paradigma de situação anterior e posterior, a fim de gerar os resultados necessários para fins de avaliação. Essa abordagem metodológica propicia a investigação das mudanças e impactos observados nas métricas de performance dos modelos de aprendizado de máquina selecionados, antes e após a introdução de características derivadas de estatísticas extraídas do grafo de similaridade, permitindo, assim, a análise comparativa e crítica das métricas de avaliação e o estabelecimento de conclusões sólidas e fundamentadas no contexto dessa pesquisa.

Para que os cenários possam avaliar como a adição de características derivadas de estatísticas de grafo afeta a performance de modelos de predição em conjuntos de dados variados, devemos estabelecer alguns critérios sobre os dois elementos que influenciam decisivamente os resultados: O conjunto de dados e os parâmetros (ou hiperparâmetros) utilizados nas diversos etapas ao longo método proposto. Nesse sentido, foram estabelecidos os seguintes critérios a serem seguidos, nos dois momentos (antes e depois), sobre os procedimentos que atuam sobre os conjuntos de dados e parâmetros (e hiperparâmetros):

- Os tratamentos (correções, codificação, separação etc.) efetuados sobre os conjuntos de dados de treino e teste devem ser os mesmos nos dois cenários;
- As transformações efetuadas (escalonamento, similaridade etc.) sobre os conjuntos de dados de treino e teste devem ser as mesmas nos dois cenários;

- Os parâmetros nos procedimentos de transformações (medidas de similaridade, thresholds etc.) efetuadas sobre os conjuntos de dados de treino e teste devem ser as mesmas nos dois cenários;
- Os modelos de aprendizado de máquina devem ser os mesmos nos dois momentos;
- Os parâmetros (e hiperparâmetros) nos modelos de aprendizado de máquina devem ser os mesmos para todos os conjuntos e de acordo com cada Cenário, isto é, no Cenário 1 com parametros padrão e no Cenário 2 com parametros(e hiperparâmetros) em grade;
- As medidas de avaliação da performance dos modelos de aprendizado de máquina devem ser os mesmos nos dois momentos.

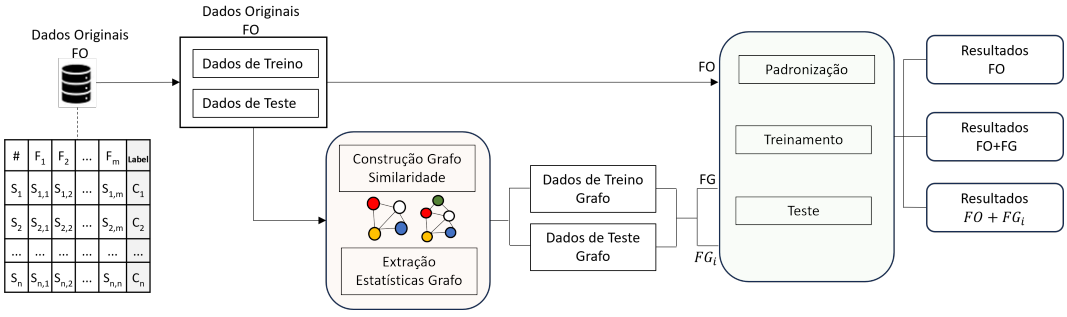
Para cada conjunto de dados, em ambos os cenários, o pipeline do método foi processado dez vezes e a média destes foi adotada como resultado.

Estabelecidos os critérios, veremos a seguir os detalhes dos cenários montados.

4.2.1 Cenário 1

O Cenário 1 (Figura 4.1) é voltado para avaliar em que medida a performance dos modelos de predição é afetada pela adição de características derivadas de estatísticas extraídas do grafo de similaridade. Esse cenário se baseia na fixação dos parâmetros de avaliação dos modelos de aprendizado nos dois momentos (antes e depois da inserção das novas características), assim como os conjuntos de dados de treino e teste. É importante destacar que não estamos querendo identificar ou avaliar o melhor modelo entre os modelos selecionados, mas, sim, avaliar o impacto desse enriquecimento do conjunto de dados na performance da predição destes modelos.

Figura 4.1: Cenário 1 do Experimento com o Método Proposto



Garantir que os conjuntos de dados utilizados em treino e teste serão exatamente os mesmos nos dois momentos (antes e depois), só modificando com o enriquecimento destes, significa dizer que todas as transformações só podem ter sido aplicadas aos conjuntos de treino e teste original até a constituição dos conjuntos de dados utilizados como base de comparação (*baseline*). Após esse momento, a única diferença será a inclusão das novas características extraídas do grafo de similaridade. Em todos esses momentos também utilizando os mesmos parâmetros. No que compete a proporção para separação dos conjuntos de treino e teste, essa será em função da proporção utilizada no artigo de *benchmark* selecionado para comparação.

Para construção do grafo de similaridade, serão adotadas as medidas *Jaccard, Overlap* e *Eskin* para características categóricas, *Euclidean, Cosine, Manhattan* e *Mahalanobis* para características numéricas. Para o estabelecimento de uma aresta no grafo, serão avaliados os limites (*thresholds*) da mediana dos valores de similaridade entre o vértice avaliado e os demais vértices, isto é, com o valor de similaridade do par de vértices acima do limite avaliado, então a aresta entre o par de vértice será estabelecida.

As estatísticas extraídas do grafo de similaridade serão Degree (DG), Degree Centrality (DC), Average Neighbor Degree (AN), Eigenvector Centrality (EC), Closeness Centrality (CC), Betweenness Centrality (BC), Load Centrality (LC), Local Clustering (CL), Triangles (TR), PageRank (PR) e HITS (HT). Serão selecionadas até seis dessas estatísticas para, de fato, serem incorporadas ao conjunto de dados como novas características. O método de seleção das estatísticas será o de filtro.

Ao mesmo tempo, da parte do Modelos de Aprendizado de Máquina, os parâmetros (e hiperparâmetros) também devem ser os mesmos utilizados quando do treinamento e teste com os conjuntos originais tratados (*baseline*). Para garantir essa estabilidade dos parâmetros (e hiperpârametros) nesse cenário, vamos utilizar as definições padrão das bibliotecas utilizadas para implementação, sem modificação nos dois momentos. Dessa forma, garantimos que não houve nenhum ajuste nestes parâmetros (e hiperparâmetros) que venham a influenciar a performance dos modelos.

Nesse primeiro cenário, serão avaliados somente os modelos que compõe o grupo de modelos avaliados em todos os conjuntos de dados, independente do artigo de comparação, são eles: DT, LPA, NBG, RFOR, RLOG, SVM e XGB, com seus parâmetros (e hiperparâmetros) padrão nos dois momentos.

No que compete as métricas de desempenho dos modelos de aprendizado de máquina, serão utilizadas as medidas Acurácia, Precisão, Sensibilidade (*Recall*) e a medida F_1. No caso dos conjuntos de dados com a variável

dependente com múltiplas classes, serão adotadas as versões ponderadas dessas medidas, conforme abordamos no Capítulo 2 Seção 2.8.3.

Serão efetuados três tipos de submissão de conjunto de dado aos modelos de aprendizado de máquina em função de quais características compõe a entrada para esses modelos, conforme descrito abaixo:

1. Conjunto de Dados completo com todas as características originais (FO);

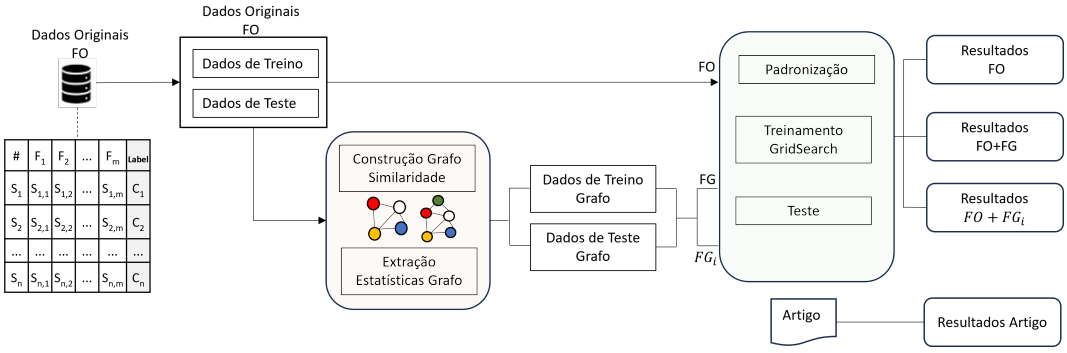
2. Conjunto de Dados completo com todas as características originais (FO) e todas as novas características extraídas e selecionadas do grafo de similaridade (FG com 6 estatísticas selecionadas), isto é, todas as características originais mais todas as características selecionadas do grafo (FO+FG);

3. Um subconjunto contendo todas as características originais (FO) e uma das novas características extraídas e selecionadas do grafo de similaridade (FG_i é uma estatística do grafo selecionada, onde i=1,..,6). Nesse caso, serão construídos e submetidos aos modelos subconjuntos com cada uma das novas características extraídas e selecionadas do grafo de similaridade (FO+FG_i).

4.2.2 Cenário 2

O Cenário 2 (Figura 4.2) é voltado para avaliar como a performance dos modelos de predição, agora ajustados a um espaço de parâmetros (hiper-parâmetros) ampliado, após o enriquecimento de características derivadas de estatísticas extraídas do grafo de similaridade.

Figura 4.2: Cenário 2 do Experimento com o Método Proposto



O Cenário 2 é praticamente igual ao Cenário 1, com apenas uma mudança na ampliação do espaço de parâmetros (hiperparâmetros) e na forma como os modelos são tratados. Como foi abordado anteriormente, no

Cenário 1 são utilizados os hiperparâmetros padrão das bibliotecas utilizadas para implementação, sem modificação nos dois momentos (antes e depois).

No Cenário 2 esses hiperparâmetros serão ajustados dentro de um espaço de possibilidades pré-determinadas, para que sejam utilizados pelos modelos na busca pela melhor combinação de hiperparâmetros que alcance a melhor performance de predição. A esse processo de busca pelo melhor modelo em um espaço de hiperpârametros chamamos de tuning de hiperparâmetros e que abordamos no Capítulo 2 Seção 2.8.4. É sempre bom lembrar que, nesse caso, não estamos buscando qual a melhor performance entre modelos diferentes para selecionar um deles, mas, sim, buscando a melhor predição em um modelo, utilizando a técnica proposta nesse estudo, para comparação, através das métricas de performance, com esse mesmo modelo utilizando técnicas diferentes de melhoria de performance de predição.

Da mesma forma que o Cenário 1, no Cenário 2 serão efetuados três tipos de submissão de conjunto de dado aos modelos de aprendizado de máquina em função de quais características compõe a entrada para esses modelos, sendo que, nesse cenário, estaremos sempre lidando com as métricas alcançadas pelo *tuning* de hiperparâmetros. O espaço de possibilidades pré-determindas para os hiperparâmetros será igual para todos os tipos de submissão.

Os parâmetros (e hiperparâmetros) utilizados nos dois cenários serão vistos na seção seguinte.

4.3 Bibliotecas Python

O experimento dos cenários descritos anteriormente foi conduzido com base na utilização de diversas bibliotecas e suas funções/classes em Python, proporcionando as ferramentas necessárias para o desenvolvimento e análise do projeto. A seguir relacionamos essas bibliotecas: Pandas Numpy, Networkx, Scikit-learn (LabelEncoder, Train_test_split, Metrics, Make_score, StandardScaler, MinMaxScaler, GridSearchCV, Pairwise_distance), Scipy.spatial.distance.

Relacionamos a seguir, as bibliotecas de algoritmos de classificação utilizadas no experimento, sendo todas da biblioteca Scikit-learn. Cada um desses algoritmos possui características específicas, abordadas no Capítulo 4 Seção 2.7, e que foram exploradas no experimento: MLPClassifier (MLP), LogisticRegression (RLOG), ExtraTreesClassifier (ETREE), DecisionTreeClassifier (DT), RandomForestClassifier (RFOR), KNeighborsClassifier (KNN), GaussianNB (NBG), BernoulliNB (NBG), SVC (SVM), GradientBoostingClassifier (GB), AdaBoostClassifier (ADAB), XGBClassifier (XGB), BaggingClassifier

(BAGG), LabelPropagation (LPA).

No Apêndice I.3 descrevemos essas bibliotecas e componentes relevantes utilizados, assim como os parâmetros (e hiperparâmetros) aplicados e que são diferentes daqueles padrões da função utilizada.

Essas bibliotecas e funções desempenharam um papel central na implementação e avaliação dos modelos de aprendizado de máquina no contexto deste experimento. A combinação dessas ferramentas proporcionou um ambiente robusto e flexível para a pesquisa e análise de dados.

4.4 Métricas de Desempenho

A avaliação rigorosa do desempenho de modelos de aprendizado de máquina desempenha um papel fundamental na pesquisa acadêmica e na aplicação prática de algoritmos.

Em todos os cenários e etapas de medição dos modelos de aprendizado de máquina foram apuradas quatro métricas de desempenho, a saber: Acurácia, Precisão, Sensibilidade (*Recall*) e medida F_1.

No Cenário 1, para a etapa de treino do modelo com os dados de origem para constituição da base de comparação dos modelos (*baseline*), os modelos com parâmetros padrão foram submetidos a esses dados e medimos a performance do modelo em treinamento para as métricas de desempenho citadas. O conjunto de teste foi submetido aos mesmos modelos treinados e sua performance foi medida com as mesmas métricas de treino. Essa métricas foram obtidas pelas funções *accuracy_score, precision_score, recall_score e f1_score* da biblioteca Scikit-learn *metrics*.

No Cenário 2, para a etapa de treino do modelo com os dados de origem para constituição da base de comparação dos modelos (*baseline*) foi efetuado o procedimento de *tunning* de parâmetros através da biblioteca *GridSearch* (Capítulo 2 Seção 2.8.4) que realiza a busca em grade (*grid search*), a fim de encontrar a combinação ideal de hiperparâmetros para um modelo de aprendizado de máquina. As medidas de performance do treino no *grid* são as médias dos valores retornados pelo teste do *grid*. O conjunto de teste foi submetido ao melhor modelo treinado no *grid* e sua performance foi medida com as mesmas métricas de treino e também utilizadas no Cenário 1. Para o conjunto de dados com as novas características com estatísticas extraídas do grafo, efetuamos os mesmos procedimentos, utilizando os mesmos modelos, parâmetros e medidas que aqueles utilizados com os dados de origem.

4.5 Configurações do Ambiente Experimento

O desenvolvimento, execução e avaliação de todas as etapas realizadas nos dois cenários propostos neste estudo foram conduzidos em conformidade com configurações específicas de hardware e software. As configurações de hardware incluíram um processador Intel Core i7 - 1065G7 CPU, com uma frequência de 1.30GHz, e uma memória RAM de 16GB, além de um armazenamento em SSD de 256GB.

Para suportar o ambiente de experimentação, as configurações de software foram igualmente essenciais. O sistema operacional utilizado foi o Microsoft Windows 11 Pro, versão 10.0.22621, X64. A linguagem de programação predominante foi o Python 3.11.4, a qual foi amplamente utilizada no desenvolvimento e execução dos modelos de aprendizado de máquina. O ambiente de desenvolvimento integrado (IDE) escolhido para programação foi o Visual Studio Code (VSCode), versão 1.83.1, proporcionando um ambiente flexível e eficiente para codificação e experimentação. Para visualização de dados, avaliação de resultados e na geração de relatórios, foi utilizado o software Microsoft Power BI, versão 2.119.986.0, 64-bit.

Além disso, diversas bibliotecas Python foram utilizadas nas análises e modelagens realizadas no decorrer deste estudo. Algumas das principais bibliotecas incluíram o IPython (versão 8.14.0), Pywin32 (versão 306), Jupyter Client (versão 8.3.0), Jupyter Core (versão 5.3.1), Python-Dateutil (versão 2.8.2), Matplotlib-Inline (versão 0.1.6), Pytz (versão 2023.3), Tzdata (versão 2023.3), Matplotlib (versão 3.7.1), Xlsxwriter (versão 3.1.2), NetworkX (versão 3.1), Category-Encoders (versão 2.6.1), Pandas (versão 2.0.2), Numpy (versão 1.25.0), Scipy (versão 1.10.1), Scikit-Learn (versão 1.2.2), Statsmodels (versão 0.14.0) e Seaborn (versão 0.12.2). Essas bibliotecas desempenharam um papel importante na manipulação, análise e visualização de dados, bem como na implementação e avaliação de modelos de aprendizado de máquina.

As configurações de hardware e software proporcionaram o ambiente ideal para a condução dos experimentos e avaliações de desempenho, garantindo a integridade e confiabilidade dos resultados obtidos, que serão apresentados no Capítulo 5.

5
Resultados

Esse Capítulo concentra-se na avaliação dos resultados obtidos por meio da aplicação do método proposto em dois cenários de experimentação e é importante para a validação e análise da eficácia do método, permitindo uma compreensão aprofundada do desempenho sob diferentes contextos. Inicialmente, apresentamos os resultados obtidos no Cenário 1, no qual as características originais são utilizadas com parâmetros padrão nos modelos de aprendizado de máquina. Em seguida, no Cenário 2, exploramos os benefícios da pesquisa em grade e da ampliação do espaço de parâmetros para otimização do desempenho.

Ao final, nos dedicamos a uma análise comparativa dos resultados com *benchmarks* estabelecidos por meio de artigos científicos relevantes na área de Aprendizado de Máquina que utilizaram o mesmo conjunto de dados em suas avaliações. Através dessa análise, objetivamos não apenas avaliar a eficácia do método em diferentes configurações, mas também situá-lo em relação às melhores práticas e resultados previamente reportados na literatura acadêmica. O enfoque é fornecer uma visão abrangente do desempenho do método proposto, identificando suas vantagens, limitações e contribuições para o campo de pesquisa em questão.

5.1
Cenário 1: Parâmetros Padrão

Neste cenário, os modelos de aprendizado são utilizados com parâmetros (hiperparâmetros) padrão. Efetuamos a apresentação da performance dos modelos e sua avaliação nos diversos conjuntos de dados. Ao final, efetuamos uma avaliação do cenário como um todo. Na análise desse cenário apresentaremos informações agregadas, mas todas as informações de resultados detalhadas estão disponíveis no Apêndice J.1 desse documento.

É importante lembrar que a única transformação que esses dados receberam foram a sua codificação, no caso de características categóricas, e a padronização, no caso de características numéricas, além, é claro, do enriquecimento com as características extraídas de estatísticas do grafo de similaridade.

Os resultados obtidos a partir do uso de diferentes modelos de aprendizado de máquina nos conjuntos de dados apresentam variações importantes sobre os resultados pelo enriquecimento das novas características oriundas do grafo de similaridade.

A Tabela 5.1 e o gráfico na Figura 5.1 apresentam um resumo dos resultados de ganhos obtidos ao comparar o desempenho dos modelos de aprendizado de máquina quando submetidos a subconjuntos de características originais (FO) em contraposição aos subconjuntos enriquecidos com características derivadas das estatísticas de grafos, obtidas por meio de métricas de similaridade (FG e FG_i). As colunas de medida são a média aritmética, Desvio Padrão (DP) e Mediana dos ganhos na medida Acurácia nos melhores resultados dos modelos em cada conjunto de dados.

Tabela 5.1: Cenário 1: Resumo de Ganhos na Acurácia (%) por Dataset

Dataset	**Média**	**DP**	**Mediana**
BCC	14,3	6,6	12,5
CAR	1,0	3,4	0,0
DM	1,3	1,0	1,2
HDH	1,5	1,0	1,2
HDHNI	0,6	0,8	0,0
PIMA	3,3	1,6	2,6
VCB	12,3	2,3	11,0
VCM	11,7	4,1	12,8
WM	1,6	1,4	2,8
WQ	1,8	0,7	1,9
Geral	**4,9**	**6,0**	**2,3**

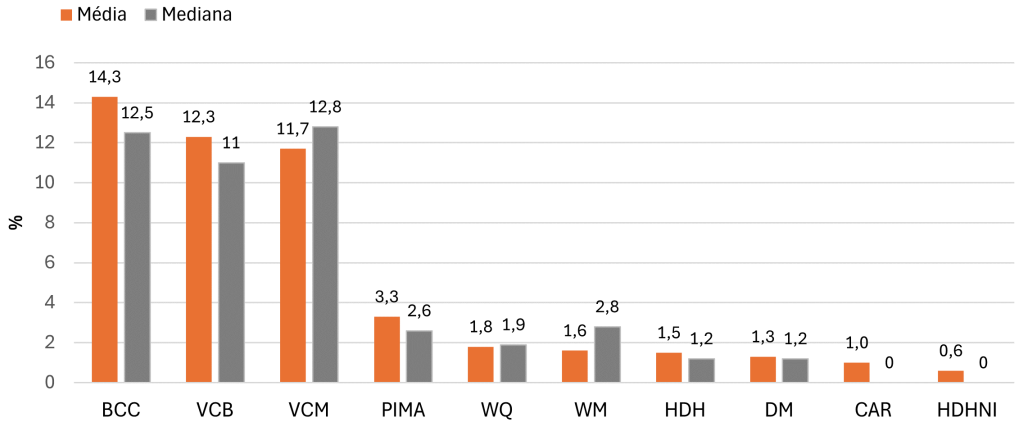


Figura 5.1: Cenário 1: Resumo de Ganhos Média e Mediana na Acurácia (%) por Dataset. Ordenação: Média (desc)

Inicialmente já se percebe uma alta variação nas médias de ganhos por conjunto de dados, o que já era esperado pela natureza diversificada do perfil de cada um destes em relação a quantidade de características, tipo de características, tipo de label e balanceamento das classes. Outro fator que também ajuda a explicar essa variação foi o fato de utizarmos as mesmas

medidas de similaridade com o mesmo perfil de tipo de carcaterísticas e as mesmas estatísticas do grafo para todos eles. Isto posto, sempre que possível, devemos analisar os resultados observando a sua variabilidade e medidas de média e mediana. De uma forma geral, a média de ganhos nesse cenário, quando da inclusão das características extraídas do grafo, ficou em 4,9% com alta variabilidade 6,0% e mediana 2,3%.

A Tabela 5.2 apresenta um resumo dos resultados obtidos ao comparar o desempenho dos modelos de aprendizado de máquina nesse cenário. A análise desses resultados não apenas evidencia um incremento nas métricas em todos os conjuntos de dados, mas também releva variações significativas entre as medidas dos conjuntos de dados.

Tabela 5.2: Cenário 1: Resultados (%) com o Método por Dataset

Dataset	**Medida**	**Acurácia**	**Precisão**	**Recall**	F_1
BCC	Média	14,3	13,5	14,3	14,4
	Mediana	12,5	12,2	12,5	12,5
CAR	Média	1,0	0,2	1,0	0,7
	Mediana	0,0	0,0	0,0	0,0
DM	Média	1,3	1,1	1,3	1,4
	Mediana	1,2	1,5	1,2	1,3
HDH	Média	1,5	1,4	1,5	1,5
	Mediana	1,2	1,1	1,2	1,1
HDHNI	Média	0,6	0,7	0,6	0,7
	Mediana	0,0	0,0	0,0	0,0
PIMA	Média	3,3	3,7	3,3	3,6
	Mediana	2,6	3,3	2,6	3,3
VCB	Média	12,3	11,0	12,3	12,1
	Mediana	11,0	10,9	11,0	10,8
VCM	Média	11,7	12,0	11,7	12,0
	Mediana	12,8	12,2	12,8	13,0
WM	Média	1,6	1,5	1,6	1,6
	Mediana	2,8	2,6	2,8	2,8
WQ	Média	1,8	1,3	1,8	1,5
	Mediana	1,9	1,1	1,9	1,5
Geral	**Média**	**4,9**	**4,6**	**4,9**	**4,8**
	Mediana	**2,3**	**2,4**	**2,3**	**2,4**

Uma observação interessante que surge da análise da Tabela 5.2 é a correspondência entre os valores nas métricas com a adição das características

do grafo, onde o aumento ou a diminuição em uma métrica (por exemplo, acurácia) é concomitante com mudanças proporcionais nas outras métricas (Precisão, *Recall* e F_1). Esse padrão sugere que a inclusão das características do grafo não é seletiva, afetando de maneira abrangente o desempenho global dos modelos de aprendizado de máquina.

É relevante notar que a média/mediana de incremento de cerca de 4,9%/2,3% nas métricas denota que a inclusão das características extraídas do grafo teve um impacto positivo nas métricas de desempenho dos modelos para os diversos conjuntos de dados analisados, embora com algumas variações importantes. Alinhado a essa observação, nossa análise, daqui para frente, se concentrará na métrica de acurácia, dado que as demais métricas seguiram uma direção e magnitude consistentes nas variações com essa medida.

Avançando no detalhamento dos resultados, a Tabela 5.3 resume os resultados de ganhos e perdas na métrica Acurácia ao comparar a performance dos modelos de aprendizado de máquina ao utilizarem subconjuntos de características originais (FO) e subconjuntos enriquecidos com características extraídas das estatísticas do grafo por meio de métricas de similaridade (FG e FG_i). A análise desses resultados revela variações importantes entre os conjuntos de dados e modelos avaliados.

Tabela 5.3: Cenário 1: Ganho/Perda por Dataset e Modelo – Acurácia (%)

Dataset	**DT**	**LPA**	**NBG**	**RFOR**	**RLOG**	**SVM**	**XGB**
BCC	20,8	8,4	25,0	12,5	4,1	16,7	12,5
CAR	0,0	0,8	9,1	-1,7	0,0	-0,8	-0,5
DM	0,0	2,3	1,1	2,3	2,3	0,0	1,2
HDH	0,0	1,2	1,2	3,4	1,1	1,1	2,3
HDHNI	0,0	1,1	0,0	2,2	0,0	1,1	0,0
PIMA	4,3	6,5	2,6	3,5	2,6	2,1	1,3
VCB	16,2	9,6	10,3	14,8	11,0	10,9	13,5
VCM	17,9	7,7	12,8	15,4	5,1	10,2	12,8
WM	0,0	0,0	2,8	0,0	2,8	2,8	2,8
WQ	0,6	2,1	2,1	2,9	1,7	1,0	1,9
Média	**6,0**	**4,0**	**6,7**	**5,5**	**3,1**	**4,5**	**4,8**
Mediana	**0,3**	**2,2**	**2,7**	**3,2**	**2,5**	**1,6**	**2,1**

Em geral, pelo gráfico na Figura 5.2, observa-se que a maioria dos conjuntos de dados e modelos experimentou melhorias nas métricas após a inclusão das características de grafo, com ganhos médios na faixa de 3,1% a 6,7%, com destaque para os modelos LPA, RFOR e RLOG, que alcançaram, na

média, as maiores melhorias com menor dispersão. Isso sugere que a introdução dessas características de grafo tem o potencial de aprimorar a capacidade preditiva dos modelos de forma considerável. No entanto, é importante notar que esses ganhos não são uniformes, variando substancialmente entre os diferentes conjuntos de dados e modelos, inclusive com valores negativos.

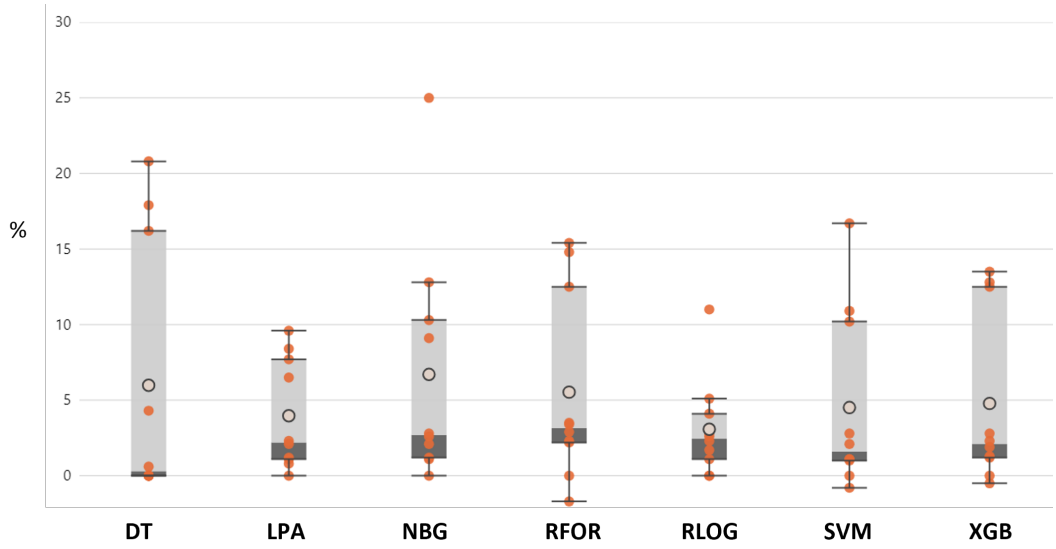


Figura 5.2: Cenário 1: Média (%) de Ganho/Perda Acurácia por Modelo

Em particular, o conjunto de dados BCC demonstrou um importante ganho na média 14,3% e mediana 12,5%, para métrica Acurácia, com a inclusão das características de grafo em vários modelos, destacando a eficácia dessas características para aprimorar a capacidade de predição nesse cenário específico. Outros conjuntos de dados, como CAR e HDHNI, não apresentaram ganhos substanciais, indicando que a relevância das características de grafo pode depender do contexto e das particularidades do conjunto de dados.

Além disso, as médias indicam um aumento geral de 4,9% e 2,3% na mediana para métrica Acurácia, demonstrando a tendência positiva da inclusão de características de grafo nos modelos de aprendizado de máquina. Esses resultados reforçam a importância de considerar a inclusão de informações de grafo para aprimorar a qualidade das previsões em várias aplicações de aprendizado de máquina supervisionada de classificação, com implicações importantes, especialmente em domínios onde pequenos aumentos na precisão podem ter um impacto substancial.

Cabe ressaltar que, em alguns casos, os modelos experimentaram uma média de acréscimo em torno de 1%. Este acréscimo, apesar de modesto em magnitude, reveste-se de significativa relevância, especialmente em cenários onde os modelos pré-existentes tenham exaurido suas margens de otimização através de abordagens convencionais. Este ganho de 1% ou mais na Acurácia

do modelo pode representar uma distinção decisiva, por exemplo, em domínios onde a precisão na predição tenha repercussões diretas na vida e bem-estar de muitos indivíduos.

Avançando na análise dos resultados, agora em relação a esses e os perfis dos conjuntos de dados, revelam tendências que fornecem *insights* para a aplicação de características de grafo em modelos de aprendizado de máquina.

Primeiramente, algumas evidências sugerem que o número de características em um conjunto de dados desempenha um papel fundamental na resposta dos modelos a essa técnica. Conjuntos de dados com menos características originais exibem uma variação mais ampla nos ganhos/perdas de desempenho ao adicionar características de grafo. Essa evidência pode sugerir que a inclusão de características de grafo tende a ter um impacto maior em conjuntos de dados com uma menor riqueza de características originais. A exceção dessa evidência fica com os conjuntos de dados HDHNI e CAR que possuem detalhes específicos, como somente características categóricas e desbalanceamento da classe, que serão objeto de análise a seguir.

A natureza das características no conjunto de dados também é um ponto que merece atenção. Pelo gráfico na Figura 5.3, a presença de um maior número de características categóricas pode limitar os ganhos da métrica Acurácia com a inclusão de características de grafo, como evidenciado nos conjuntos de dados HDHNI e CAR. Isso pode sugerir que características categóricas originais já fornecem informações distintas o suficiente para atender às demandas do modelo ou que as novas características extraídas do grafo podem não ser informativas o suficiente para influenciar o modelo, fazendo com que esses passem a não as considerar como importantes na tomada de decisões.

Nessa última hipótese, existe também a possibilidade de que as medidas de similaridade adotadas para características categóricas (*Jaccard, Overlap e Eskin*) não estejam adequadas no estabelecimento das arestas do grafo e que essa inadequação se reflete nas estatísticas extraídas do grafo. Em contraste, conjuntos de dados com um número considerável de características numéricas, como BCC, mostram ganhos mais notáveis, sugerindo que as características de grafo podem ser particularmente eficazes nesse contexto.

Outro fator de influência observado é o tipo de rótulo do conjunto de dados. Pelo gráfico na Figura 5.4, conjuntos de dados com rótulos binários, como BCC e VCB, tendem a exibir ganhos na métrica Acurácia. Isso pode ser atribuído à natureza binária do problema, onde as características de grafo podem ter um papel mais relevante na diferenciação entre as classes. Em contraste, conjuntos de dados com rótulos multiclasse apresentam ganhos menores, como DM e CAR, embora, em alguns casos, como VCM, os resultados

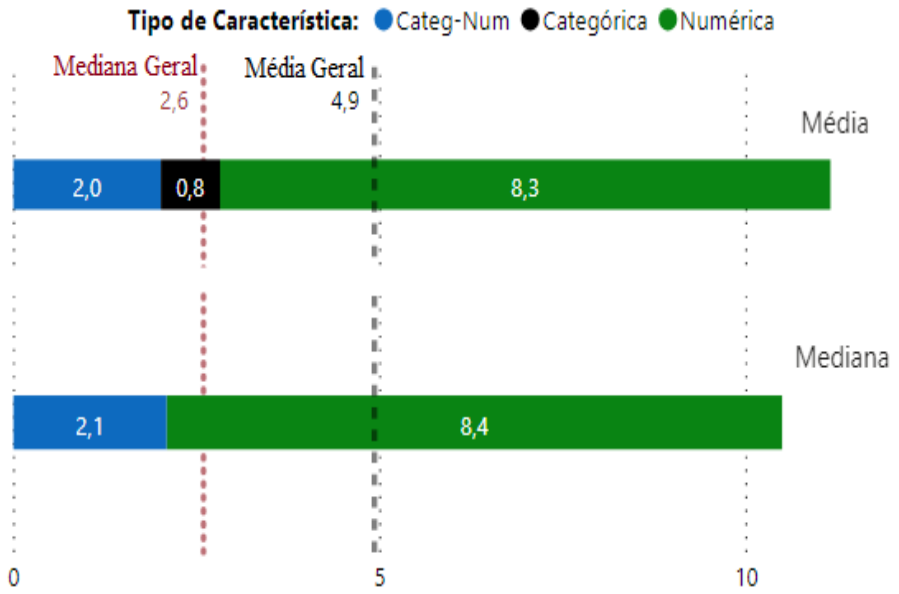


Figura 5.3: Cenário 1: Ganho/Perda Acurácia (%) por Tipo Característica

tenham sido expressivos.

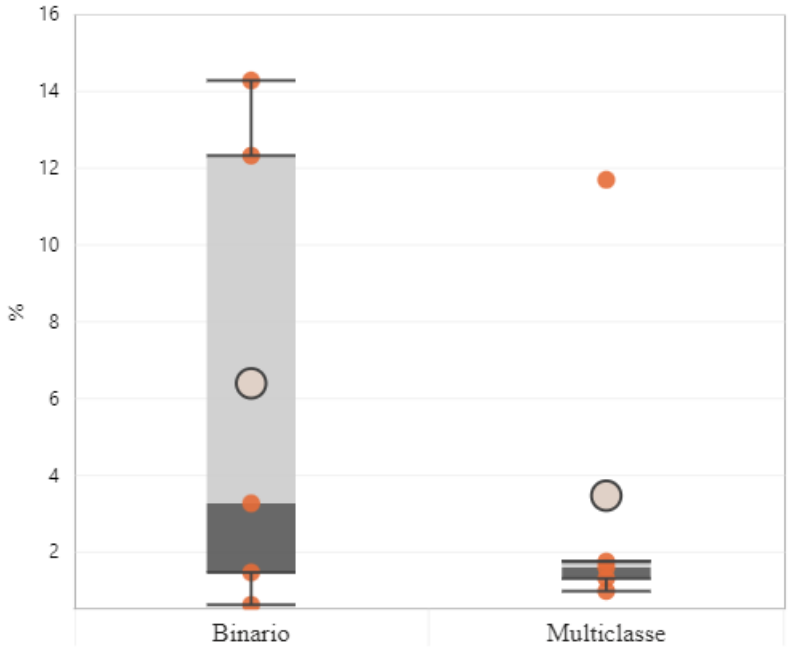


Figura 5.4: Cenário 1: Média de Ganho/Perda Acurácia por Tipo Label

Nos contextos tradicionais da aplicação de modelos de aprendizado de máquina supervisionados, a quantidade de instâncias em um conjunto de dados tem sido reconhecida como um fator influente nos resultados de desempenho dos modelos. O gráfico na Figura 5.5 ilustra a distribuição dos ganhos observados com a inclusão de características derivadas de grafos em conjuntos de dados com até 310 instâncias, excluindo os conjuntos VCB, uma vez que este último é idêntico ao VCM. Constatamos que os conjuntos de

dados com menos de 310 instâncias experimentaram ganhos mais substanciais quando características de grafo foram adicionadas. Esse resultado sugere que a incorporação de características extraídas do grafo tende a ser particularmente benéfico em conjuntos de dados menores, destacando a capacidade dessa abordagem em melhorar o desempenho dos modelos nesses cenários.

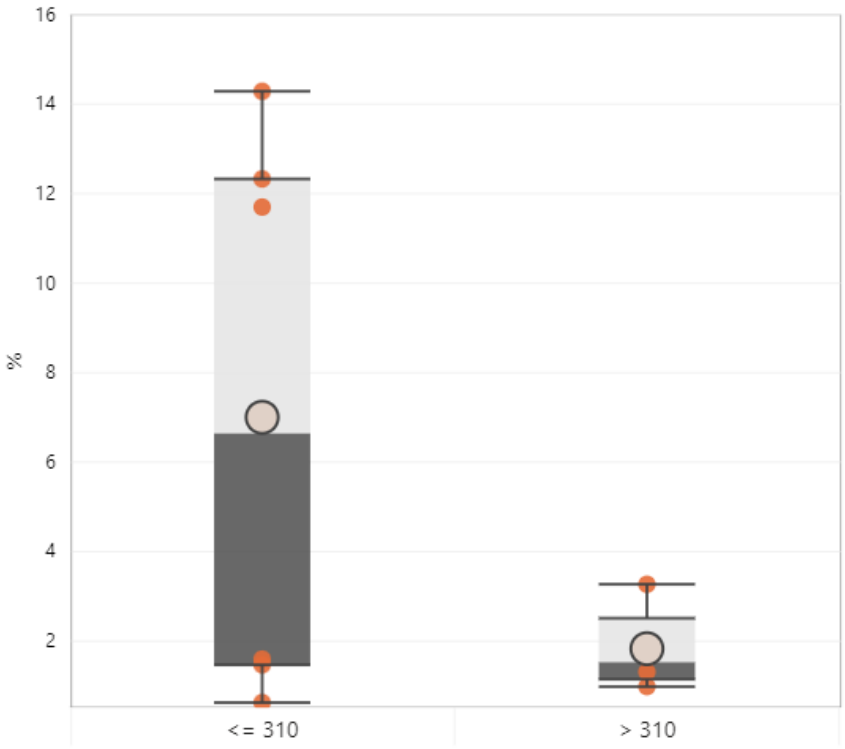


Figura 5.5: Cenário 1: Média de Ganho/Perda Acurácia (%) por Nr. de Instâncias

O balanceamento das classes em um conjunto de dados emerge como um fator que exerce impacto importante nos aprimoramentos observados na métrica da Acurácia. O gráfico na Figura 5.6 apresenta uma tendência em que os conjuntos de dados originalmente desequilibrados manifestam incrementos mais substanciais na métrica da Acurácia (5,2%) quando submetidos à inclusão de características derivadas de estatísticas do grafo, em comparação aos conjuntos equilibrados (4,5%). Este resultado sugere que a adição de novas características, com base nas estatísticas extraídas do grafo, pode conferir uma vantagem mais acentuada aos modelos na superação dos desafios relacionados ao desequilíbrio intrínseco das classes.

Essa análise realça a importância de considerar cuidadosamente as características intrínsecas de um conjunto de dados ao aplicar características de grafo. Compreender como o número, tipo e equilíbrio das características, bem como a natureza do rótulo, afetam o desempenho dos modelos é importante

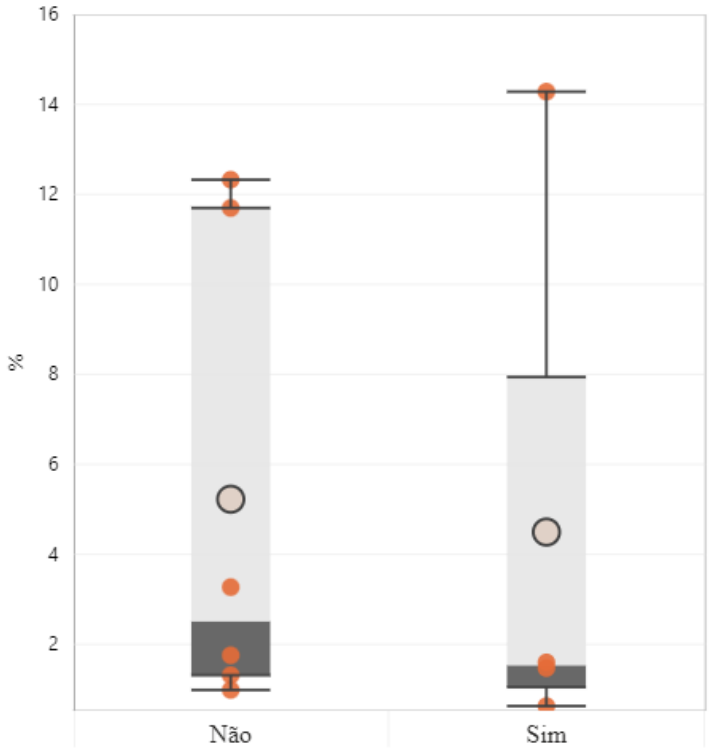


Figura 5.6: Cenário 1: Média de Ganho/Perda Acurácia (%) por Balanceamento das Classes

para tomar decisões ao aplicar essa técnica em contextos de aprendizado de máquina.

Outro destaque desse experimento está relacionado a influência de cada estatística do grafo no resultado dos modelos. A Tabela 5.4 apresenta a frequência com que determinada estatística do grafo esteve entre os melhores valores médios por modelo nos conjuntos de dados. Essa tabela proporciona uma visão esclarecedora sobre a frequência com que diferentes estatísticas de grafo demonstraram ganhos importantes na métrica da Acurácia para vários modelos de aprendizado de máquina supervisionados de classificação.

Um ponto importante foi a frequência com que a adição do conjunto de de todas as características selecionadas do grafo (FG) esteve consistentemente entre as melhores em todos os modelos, em especial nos modelos Árvore de Decisão (DT), Random Forest (RFOR) e Extreme Gradient Boosting (XGB). Além disso, em muitos modelos, a adição de apenas uma característica, como *hits* (HT), *pagerank* (PR) e *load centrality* (CL), resultou em melhorias importantes, contribuindo para melhorias significativas na Acurácia. Essa variação entre as estatísticas ressalta a importância da escolha criteriosa das características de grafo, adaptadas ao modelo em questão.

Observando agora a Tabela 5.4 pelo lado do incremento médio para a métrica Acurácia por cada estatística do grafo, a estatística *triangles* (TR) se

Tabela 5.4: Cenário 1– Ganho/Perda Médio e Frequência em Melhores Resultados por Estatística Grafo e Modelo – Acurácia (%)

Estat. Grafo	**DT**	**LPA**	**NBG**	**RFOR**	**RLOG**	**SVM**	**XGB**	**Geral**
AN		2,1	2,8				1,2	**2,0**
		1	1				1	**3**
BC		4,8			2,8		-0,5	**3,0**
		2			1		1	**4**
CC	0,6				4,1		2,8	**2,5**
	1				1		1	**3**
CL					3,4	2,8		**3,2**
					2	1		**3**
DG		4,5	5,6	2,9				**4,6**
		2	2	1				**5**
DC			10,3	2,3				**6,3**
			1	1				**2**
EC	0,0	0,8		12,5		10,9		**6,1**
	1	1		1		1		**4**
FG	6,5	2,3	5,5	2,3	2,5	6,3	5,5	**5,0**
	6	1	3	3	2	3	3	**21**
HT	16,2	5,4	0,0	5,1	5,5		13,5	**6,6**
	1	3	1	3	2		1	**11**
LC			1,2			1,1	1,9	**1,3**
			1			2	1	**4**
PR				15,4	0,6	-0,8	6,3	**4,7**
				1	2	1	2	**6**
TR	4,3		25,0			5,7		**10,2**
	1		1			2		**4**
Geral	6,0	4,0	6,7	5,5	3,1	4,5	4,8	**4,9**
	10	**10**	**10**	**10**	**10**	**10**	**10**	**70**

destaca com o maior incremento médio (13,1%) mas limitado principalmente ao modelo NBG no conjunto de dados BCC. Além disso, as estatísticas FG e *hits* (HT) foram as que apresentaram as maiores presenças em ganhos médios de Acurácia em conjunto com frequência, destacando sua importância geral na melhoria do desempenho dos modelos. No entanto, também se observam resultados indesejados, uma vez que algumas estatísticas mostraram um impacto nulo ou negativo de incremento em média para os modelos, tais como *eigenvector centrality* (EC), *betweenness centrality* (BC) e *hits* (HT),

o que requer uma análise mais aprofundada das situações em que essas estatísticas são vantajosas ou desvantajosas em conjuntos de dados específicos ou modelos.

No geral, esses resultados enfatizam a relevância de escolher cuidadosamente as estatísticas do grafo para otimizar o desempenho do modelo ML, uma vez que diferentes estatísticas são mais eficazes para diferentes modelos ML e conjunto de dados.

Tabela 5.5: Cenário 1: Frequência/Ganho (%) Melhores Resultados por Similaridade Categórica

Similaridade Categórica	Similaridade Numérica	Freq.	Ganho Acurácia
jaccard	**Total**	**28**	**1,5**
	euclidean	11	1,6
		10	0,9
	cosine	3	2,0
	mahalanobis	3	1,9
	manhattan	1	3,4
eskin	**Total**	**6**	**1,7**
		4	0,7
	mahalanobis	2	3,9
overlap	**Total**	**1**	**2,3**
	cosine	1	2,3

Da parte das medidas de similaridade, pela Tabela 5.5, verificamos o destaque da medida *Jaccard* que teve a maior frequência entre os melhores resultados para métrica Acurácia entre as medidas de similaridade para características categóricas, a maior parte delas associada com medidas de similaridade numéricas, principalmente *Euclidean*. Uma vez que a medida *Jaccard* trabalha melhor características binárias, esse resultado sugere que a transformação de codificação binária dos domínios de características categóricas ajudou essa medida na captação de similaridade entre as instâncias dos conjuntos dados.

Pelo lado das medidas de similaridade para características numéricas, pela Tabela 5.6, observa-se a concentração dos melhores resultados da métrica Acurácia nas medidas *Euclidean* e *Mahalanobis*. Tanto a *Euclidean* quanto a *Mahalanobis* são medidas de distância que avaliam a proximidade entre pontos de dados no espaço de características e que podem ser particularmente influenciadas se as características extraídas do grafo se baseiam principalmente na proximidade ou na distância entre as instâncias, então essas métricas de similaridade podem capturar com eficácia tais relações.

Tabela 5.6: Cenário 1: Frequência/Ganho (%) Melhores Resultados por Similaridade Numérica

Similaridade Numérica	Similaridade Categórica	Freq.	Ganho Acurácia
	Total	**24**	**4,0**
euclidean		13	6,0
	jaccard	11	1,6
	Total	**19**	**10,0**
mahalanobis		14	12,7
	jaccard	3	1,9
	eskin	2	3,9
	Total	**10**	**4,2**
cosine		6	5,7
	jaccard	3	2,0
	overlap	1	2,3
	Total	**3**	**2,0**
manhattan		2	1,4
	jaccard	1	3,4

Segundo análise baseada em princípios de medidas de similaridade, métricas como a *Euclidean* e a *Mahalanobis* são frequentemente consideradas apropriadas para problemas em que as características possuem conotações geométricas ou espaciais. Se as características extraídas do grafo estão relacionadas à geometria ou ao espaço (que é o nosso caso com *hits* ou *pagerank* ou *closeness centrality*, por exemplo), essas medidas se tornam mais relevantes. Também não podemos descartar que alguns modelos podem se beneficiar mais de medidas de distância, enquanto outros podem preferir medidas de similaridade baseadas em ângulos, como a *Cosine*, por exemplo.

Da análise das medidas de similaridade, verifica-se que é preciso observar a estrutura do grafo e testar diferentes medidas, ajustando o modelo de acordo com os resultados obtidos.

Finalmente, os resultados obtidos nesse Cenário 1 de modelos com hiperparâmetros com valores padrão, alcançaram um ganho médio de acurácia 4,9% e com mediana de 2,3%, sugerindo que existem evidências que permitem supor uma melhoria geral nas métricas de desempenho, por meio da aplicação do método proposto, dos modelos de aprendizado de máquina supervisionados de classificação aqui selecionados.

5.2 Cenário 2: Pesquisa em Grade e Ajuste de Parâmetros

Neste cenário, as características originais foram submetidas ao processo de pesquisa em grade (*grid search* em inglês), Vide Capítulo 2 Seção 2.8.4, para otimizar hiperparâmetros dos modelos, ampliando o espaço de hiperparâmetros a que os modelos tiveram acesso. Da mesma forma que no Cenário 1, efetuamos a apresentação da performance dos modelos, e sua avaliação nos diversos conjuntos de dados. Ao final, efetuamos uma avaliação do cenário como um todo. Assim como no Cenário 1, na análise desse cenário apresentaremos informações agregadas, mas todas as informações de resultados detalhadas estão disponíveis no Apêndice J.2 desse documento.

A combinação dos hiperparâmetros para cada modelo foi descrita no Apêndice I Seção I.3 e vale destacar que a mesma combinação será aplicada a todos os conjuntos de dados, sem modificação.

Foi mantido o procedimento efetuado no Cenário 1 quanto a transformação aplicada nos conjuntos de dados, isto é, a única transformação que esses dados receberam foram a sua codificação, no caso de características categóricas, e a padronização, no caso de características numéricas, além, é claro, do enriquecimento com as características extraídas de estatísticas do grafo de similaridade.

Da mesma forma que no Cenário 1, nesse cenário os modelos são submetidos aos mesmos conjuntos de dados.

Os resultados decorrentes da aplicação de distintos modelos de aprendizado de máquina em conjuntos de dados evidenciam uma influência importante sobre os desfechos obtidos mediante o enriquecimento desses conjuntos por meio da introdução de novas características derivadas das estatísticas do grafo, acompanhada da exploração de um espaço ampliado de hiperparâmetros ao qual os modelos ganharam acesso.

A Tabela 5.7 e o gráfico na Figura 5.7 apresentam um resumo dos resultados de ganhos obtidos ao comparar o desempenho dos modelos de aprendizado de máquina quando submetidos a subconjuntos de características originais (FO) em contraposição aos subconjuntos enriquecidos com características derivadas das estatísticas de grafos, obtidas por meio de métricas de similaridade (FG e FG_i). As colunas de medida são a média aritmética, Desvio Padrão (DP) e Mediana dos ganhos na medida Acurácia nos melhores resultados dos modelos em cada conjunto de dados.

Tabela 5.7: Cenário 2: Resumo de Ganhos na Acurácia (%) por Dataset

Dataset	**Média**	**DP**	**Mediana**
BCC	17,3	7,5	16,6
CAR	0,8	3,1	0,0
DM	1,3	0,7	1,2
HDH	1,8	1,7	1,2
HDHNI	1,3	1,4	1,1
PIMA	2,5	0,7	2,6
VCB	10,0	5,2	11,6
VCM	10,8	3,1	11,5
WM	0,8	1,3	0,0
WQ	2,3	0,8	1,9
Geral	**4,9**	**6,4**	**2,2**

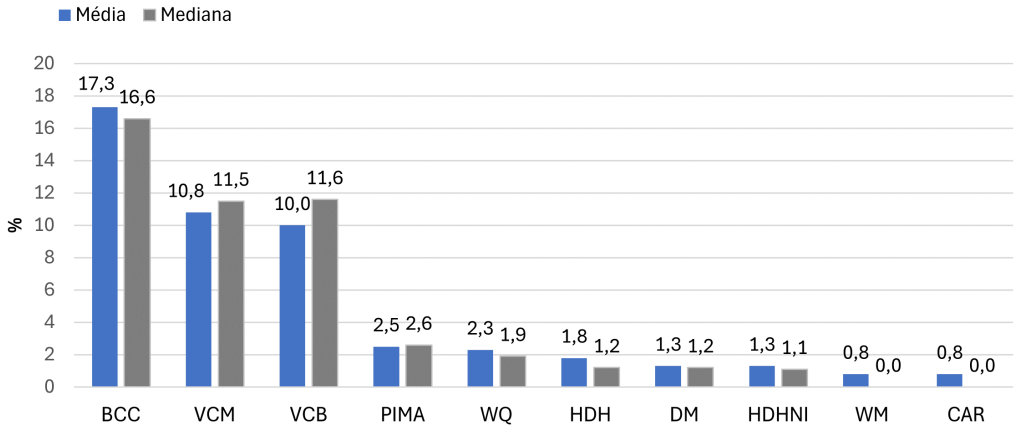


Figura 5.7: Cenário 2: Resumo de Ganhos Média e Mediana na Acurácia (%) por Dataset. Ordenação: Média (desc)

Inicialmente já se percebe, assim como no Cenário 1, uma alta variação nas médias de ganhos por conjunto de dados, o que também já era esperado pela natureza diversificada do perfil de cada um destes em relação a quantidade de características, tipo de características, tipo de label e balanceamento das classes, assim como o fato de utizarmos as mesmas medidas de similaridade com o mesmo perfil de tipo de carcaterísticas e as mesmas estatísticas do grafo para todos eles. Isto posto, sempre que possível, devemos analisar os resultados observando a sua variabilidade e medidas de média e mediana. De uma forma geral, a média de ganhos nesse cenário quando da inclusão das características extraídas do grafo ficou em 4,9% com alta variabilidade (6,4%) e mediana 2,2%.

A Tabela 5.8 sintetiza os resultados da avaliação do desempenho dos modelos de aprendizado de máquina nesse Cenário 2 em todas as métricas. Esses resultados não somente revelam um aumento nas métricas em todos os conjuntos de dados, mas, também, evidenciam variações substanciais entre as grandezas métricas associadas a diferentes conjuntos de dados.

Tabela 5.8: Cenário 2: Resultados (%) por Dataset

Dataset	**Medida**	**Acurácia**	**Precisão**	**Recall**	F_1
BCC	Média	17,3	16,6	17,3	17,4
	Mediana	16,6	16,6	16,6	16,6
CAR	Média	0,8	0,3	0,8	0,5
	Mediana	0	0,4	0	0
DM	Média	1,3	1,2	1,3	1,4
	Mediana	1,2	1,4	1,2	1,2
HDH	Média	1,8	1,7	1,8	1,8
	Mediana	1,2	1,1	1,2	1,2
HDHNI	Média	1,3	1,3	1,3	1,3
	Mediana	1,1	1,2	1,1	1,1
PIMA	Média	2,5	2,7	2,5	2,7
	Mediana	2,6	3,1	2,6	2,8
VCB	Média	10,0	9,7	10,0	9,9
	Mediana	11,6	10,8	11,6	11,3
VCM	Média	10,8	11,4	10,8	11,2
	Mediana	11,5	12,4	11,5	12,1
WM	Média	0,8	0,7	0,8	0,8
	Mediana	0	0	0	0
WQ	Média	2,3	1,7	2,3	2,0
	Mediana	1,9	1,7	1,9	1,9
Geral	**Média**	**4,9**	**4,7**	**4,9**	**4,8**
	Mediana	**2,2**	**2,2**	**2,2**	**2,2**

Avançando na análise da Tabela 5.8, tal como no Cenário 1, é possível notar que a inclusão das características derivadas do grafo exerce um impacto importante e generalizado sobre o desempenho global dos modelos de aprendizado de máquina selecionados. De maneira análoga ao Cenário 1, uma correlação é discernível entre os incrementos métricos obtidos pela incorporação das características do grafo, onde qualquer variação em uma métrica (por exemplo, acurácia) está associada a alterações proporcionais nas demais métricas (precisão, *recall* e F_1).

Comparativamente ao Cenário 1, alguns conjuntos de dados registram melhorias mais substanciais, com especial destaque para BCC e HDHNI, sugerindo que o benefício advindo da inclusão dessas características tornou-se mais proeminente quando se ampliou o espaço de hiperparâmetros disponíveis aos modelos nesses conjuntos de dados. Contudo, em contraste, outros conjuntos de dados exibem melhorias mais modestas do que as observadas no Cenário 1, sugerindo que o ganho pela adição de características extraídas do grafo é minorado para alguns modelos quando se expande o espaço de hiperparâmetros. Não obstante, como abordaremos a seguir, é importante salientar que dependendo da dimensão sob análise os dados revelam variações importantes.

Da mesma forma que no Cenário 1, daqui para frente, nossa avaliação se concentrará na métrica de Acurácia, dado que as demais métricas seguiram uma direção e magnitude consistentes nas variações com essa medida.

A Tabela 5.9 condensa as descobertas relativas aos incrementos e decréscimos na métrica Acurácia provenientes da comparação do desempenho por modelos de aprendizado de máquina. Essa análise incorpora o uso de subconjuntos de características originais (FO) e subconjuntos enriquecidos com características derivadas das estatísticas do grafo (FG e FG_i), obtidas através de métricas de similaridade. Os resultados documentados nessa tabela se caracterizam por variações significativas, tanto entre os distintos conjuntos de dados quanto entre os modelos analisados. Notavelmente, os conjuntos BCC, VCB e VCM se destacam positivamente, registrando aumentos substanciais em todos os modelos, e exibindo uma diferença reduzida entre as médias e medianas desses ganhos.

Em contrapartida, os conjuntos CAR e WM se destacam negativamente, apresentando resultados de ganhos nulos ou mesmo perdas de desempenho com a inclusão de características extraídas do grafo, especialmente nos modelos mais complexos, como SVM e modelos ensemble, a exemplo de RFOR e XGB. Esse desfecho sugere que fatores particulares e específicos exercem influência sobre esses resultados, além dos parâmetros analisados, e vamos investiga-los mas a frente.

No caso do conjunto de dados CAR verificaremos mais a frente as especificidades que conduziram esses resultados. Pelo lado do conjunto de dados WM a performance dos resultados antes da inclusão das novas características já era próxima de 100%, o que reduz o espaço de melhoria.

Tabela 5.9: Cenário 2: Ganho/Perda por Dataset e Modelo – Acurácia (%)

Dataset	**DT**	**LPA**	**NBG**	**RFOR**	**RLOG**	**SVM**	**XGB**
BCC	29,2	8,4	25,0	12,5	8,3	20,8	16,6
CAR	0,9	0,9	7,8	-1,5	0,0	-0,3	-2,2
DM	1,2	2,3	0,0	2,3	1,1	1,2	1,1
HDH	2,3	1,2	1,2	5,6	1,1	1,2	0,0
HDHNI	1,1	1,1	1,1	1,1	0,0	0,0	4,5
PIMA	3,9	1,3	2,6	2,2	2,6	2,2	3,0
VCB	16,2	5,1	0,0	13,5	10,3	11,6	13,5
VCM	10,3	7,7	12,8	14,1	5,1	14,1	11,5
WM	2,8	0,0	2,8	0,0	0,0	0,0	0,0
WQ	4,2	2,1	1,9	2,8	1,7	1,8	1,7
Média	**7,2**	**3,0**	**5,5**	**5,3**	**3,0**	**5,3**	**5,0**
Mediana	**3,4**	**1,7**	**2,3**	**2,5**	**1,4**	**1,5**	**2,4**

De forma abrangente, conforme ilustrado no gráfico da Figura 5.8, o cenário observado no Cenário 1 se repetiu no Cenário 2. Nele, a inclusão das características de grafo proporcionou melhorias nas métricas em sua maioria, com ganhos médios oscilando entre 2,9% e 7,2% e medianas situadas na faixa de 1,3% a 2,8%.

É digno de destaque o desempenho dos modelos DT, NBG, RFOR e SVM, que apresentaram, em média, as maiores melhorias. Esse cenário sugere que a incorporação das características derivadas do grafo detém o potencial de aprimoramento importante na capacidade preditiva dos modelos, inclusive quando combinada com a ampliação do espaço de hiperparâmetros e o emprego de técnicas como pesquisa em grade.

No entanto, é importante ressaltar que os resultados exibem uma variação considerável que, em determinados modelos, superou até mesmo a observada no Cenário 1, notavelmente nos casos de NBG e SVM.

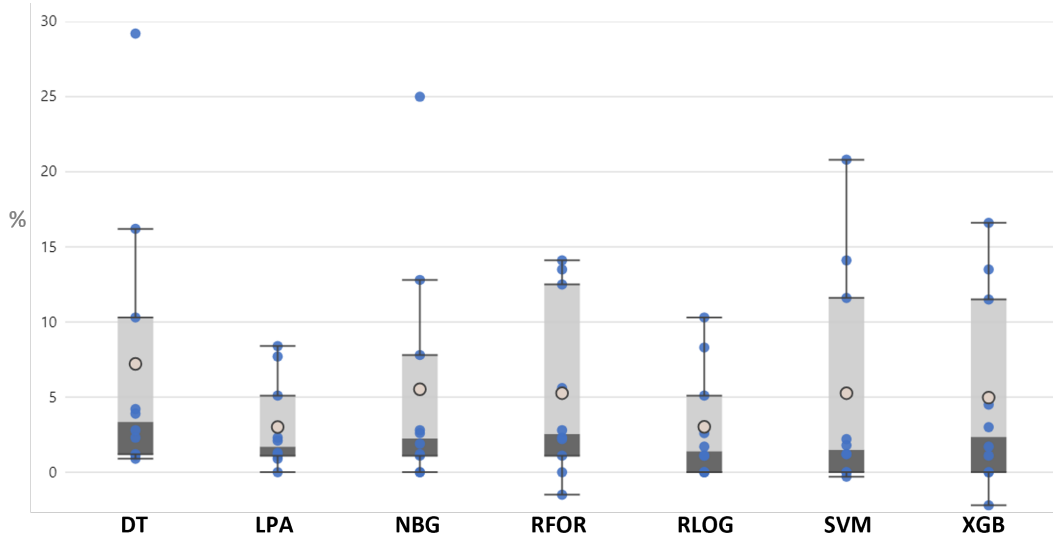


Figura 5.8: Cenário 2: Média (%) de Ganho/Perda Acurácia por Modelo

Em particular, o conjunto de dados BCC teve um destaque importante, revelando ganhos significativos na média (17,3%) e na mediana (16,6%) da métrica Acurácia. Esses ganhos foram observados em diversos modelos, enfatizando a eficácia das características derivadas do grafo na melhoria da capacidade preditiva nesse cenário específico. No entanto, outros conjuntos de dados, a exemplo de CAR e HDHNI, não demonstraram melhorias de destaque, sugerindo que a relevância das características de grafo pode estar condicionada ao contexto e às especificidades do conjunto de dados em questão.

Ademais, as análises das médias indicam um incremento geral na métrica acurácia de 4,9% e 2,2% na média e mediana, respectivamente. Essa constatação sugere uma tendência positiva associada à inclusão de características de grafo nos modelos de aprendizado de máquina, particularmente quando se considera a ampliação do espaço de hiperparâmetros e a aplicação de técnicas como pesquisa em grade.

Tais resultados reforçam a importância de se considerar a incorporação de informações provenientes de grafos para enriquecer a qualidade das previsões em diversos contextos de aplicação de aprendizado de máquina supervisionados de classificação. Isso possui implicações de magnitude, sobretudo em domínios onde pequenos ganhos na precisão podem acarretar impactos substanciais.

É imperativo destacar que, em determinados conjuntos de dados, os modelos demonstraram um aumento médio de aproximadamente 1%. A despeito de representarem ganhos mínimos, essa elevação assume uma relevância substancial em cenários onde os modelos convencionais atingiram o limite de otimização. Um ganho de 1% ou mais na métrica de Acurácia do modelo pode se mostrar decisivo, notadamente em domínios em que a precisão das previsões

tem implicações diretas na vida e no bem-estar de uma extensa coletividade.

Aprofundando a análise dos resultados, em relação a esses e aos perfis dos conjuntos de dados, emergem tendências substanciais que oferecem informações relevantes para a aplicação de características de grafo em modelos de aprendizado de máquina. Como constatado no Cenário 1, há indícios significativos de que o número de características em um conjunto de dados desempenha um papel crucial na resposta dos modelos a essa técnica. Conjuntos de dados com menos características originais apresentam uma variação mais ampla nos ganhos de desempenho ao adicionar características de grafo.

Essa constatação sugere que a inclusão de características de grafo tende a exercer um impacto maior em conjuntos de dados com uma riqueza menor de características originais. A exceção a essa tendência fica por conta dos conjuntos de dados HDHNI e CAR, que se caracterizam por particularidades específicas, como a presença exclusiva de características categóricas e desequilíbrio de classe, aspectos que serão abordados a seguir.

Através da análise do gráfico na Figura 5.9, verifica-se que a natureza das características contidas no conjunto de dados exerce uma influência considerável. Concretamente, a presença de muitas características categóricas pode limitar o incremento na métrica de Acurácia quando se introduzem características extraídas do grafo, conforme exemplificado pelos conjuntos de dados HDHNI e CAR. Isso sugere que as características categóricas originais já fornecem informações suficientemente distintas para satisfazer as necessidades do modelo, ou que as novas características derivadas do grafo podem carecer de informação relevante para influenciar o modelo, o que leva o modelo a desconsiderá-las em suas decisões.

Adicionalmente, cabe ponderar a possibilidade de que as medidas de similaridade empregadas para características categóricas (*Jaccard, Overlap e Eskin*) possam não estar adequadamente ajustadas para a criação das conexões no grafo, sendo esta inadequação refletida nas estatísticas obtidas a partir do grafo. Por outro lado, conjuntos de dados que englobam um número substancial de características numéricas, como o conjunto BCC, demonstram ganhos mais pronunciados, sugerindo que as características do grafo podem se mostrar particularmente eficazes nesse contexto.

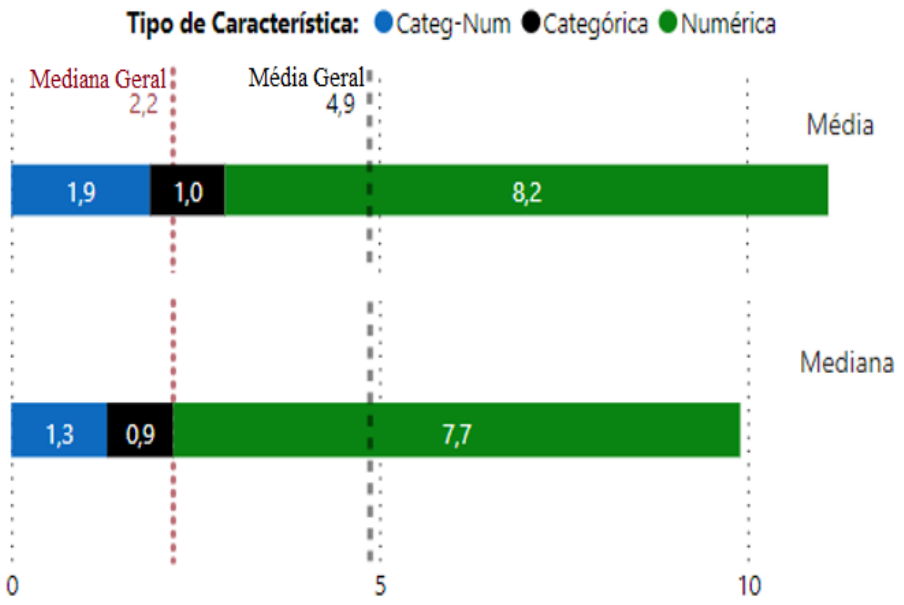


Figura 5.9: Cenário 2: Média (%) de Ganho/Perda Acurácia por Modelo

Em linha com os resultados observados, no Cenário 1, observamos uma alta influência do tipo de rótulo no conjunto de dados sobre os resultados, no Cenário 2. O tipo de rótulo binário alcançou uma melhoria importante sobressaindo em relação aos conjuntos de dados com rótulos multiclasse. O gráfico na Figura 5.10 ilustra que, em média, bem como na mediana, os conjuntos de dados com rótulos binários apresentam um desempenho superior, com destaque para os resultados superiores dos modelos BCC e VCB. Essas descobertas sugerem que os problemas com classes binárias podem colher benefícios substanciais a partir da incorporação de características de grafo, inclusive quando o espaço de hiperparâmetros é ampliado e estratégias de pesquisa em grade são empregadas. Esses resultados sugerem que essas características desempenham um papel crucial ao ajudar os modelos a discernir entre diferentes classes em tais cenários.

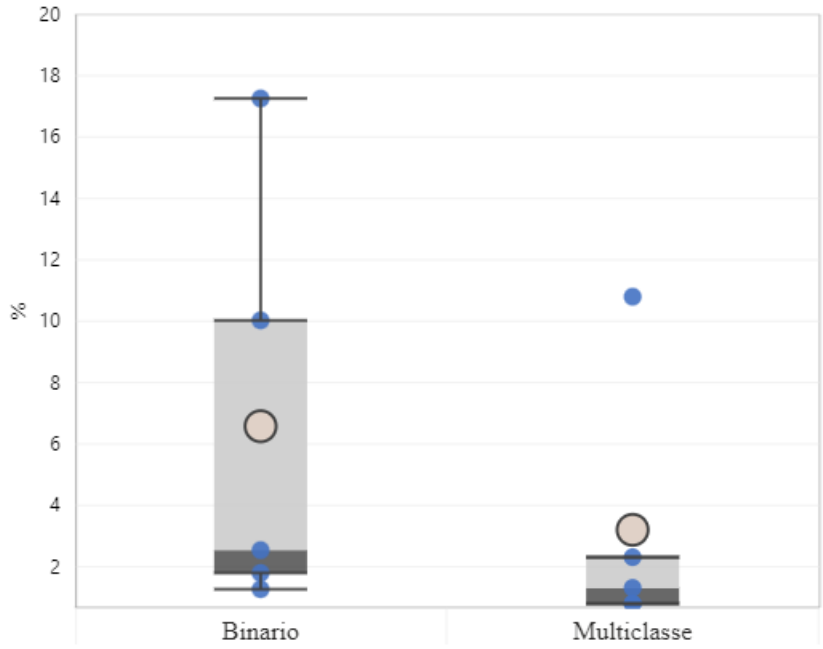


Figura 5.10: Cenário 2: Média de Ganho/Perda Acurácia (%) por Tipo Label

Da mesma forma que no Cenário 1, a quantidade de instâncias em um conjunto de dados foi fator influente nos resultados de desempenho dos modelos. O gráfico Figura 5.11 ilustra a distribuição dos ganhos observados com a inclusão de características derivadas de grafos em conjuntos de dados com até 310 instâncias. Os conjuntos de dados com menos de 310 instâncias experimentaram ganhos mais substanciais quando características de grafo foram adicionadas. Esse resultado sugere que a incorporação de características extraídas do grafo em conjunto com a ampliação do espaço de hiperparâmetros e o emprego de pesquisa em grade tendem a ser particularmente benéficos em conjuntos de dados menores, destacando a capacidade dessa abordagem em melhorar o desempenho dos modelos nesses cenários.

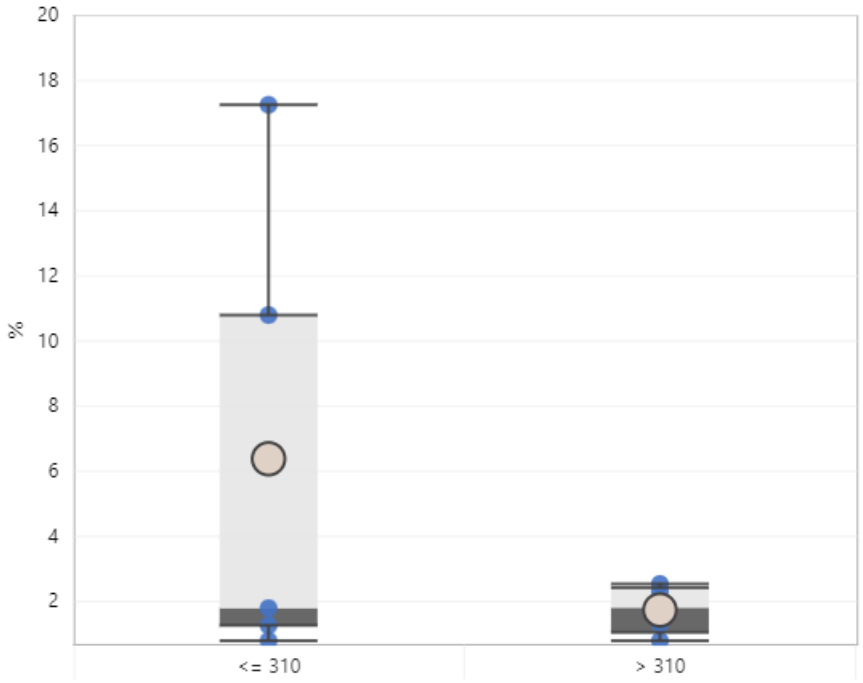


Figura 5.11: Cenário 2: Média de Ganho/Perda Acurácia (%) por Tipo Label

O equilíbrio das classes dentro de um conjunto de dados também emerge como um fator que exerce um impacto maior nos aprimoramentos verificados na métrica de Acurácia. O gráfico na Figura 5.12 destaca que conjuntos de dados equilibrados apresentaram um ganho superior (5,3%) em relação aos conjuntos de dados desequilibrados (4,6%). Contudo, é essencial observar que essa pequena diferença é influenciada significativamente por um valor extremo obtido a partir do conjunto de dados BCC no grupo de dados equilibrados.

Sem a contribuição do conjunto de dados BCC, a diferença de ganho entre os conjuntos de dados desequilibrados e equilibrados seria de 4,6% versus 1,3%, respectivamente. Esse resultado sugere que a inclusão de novas características derivadas de grafos em conjuntos de dados desequilibrados pode conferir uma vantagem mais acentuada aos modelos na superação dos desafios relacionados ao desequilíbrio intrínseco das classes, mesmo quando consideramos a ampliação do espaço de hiperparâmetros e o uso de pesquisa em grade.

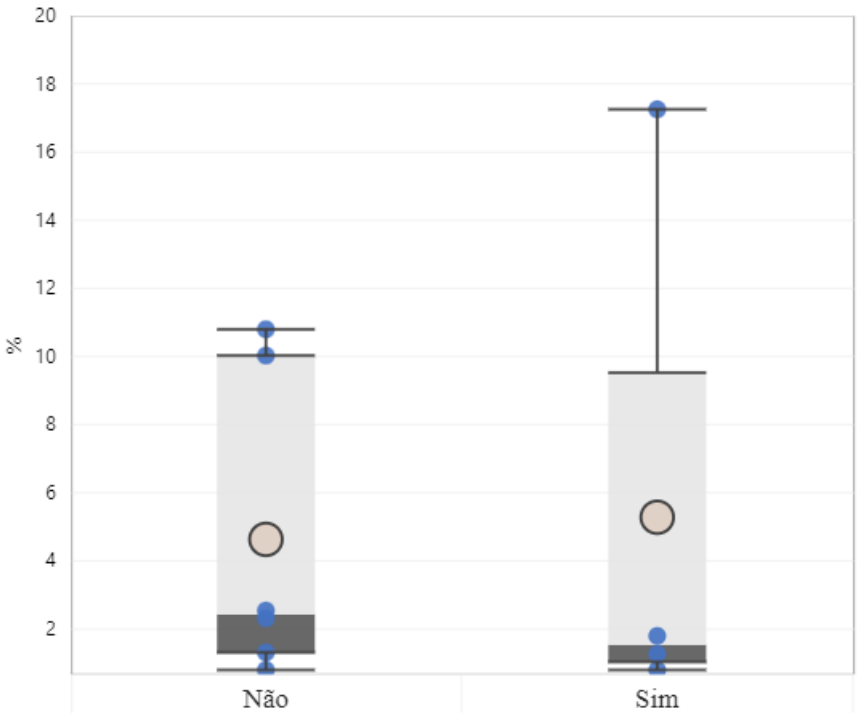


Figura 5.12: Cenário 2: Média de Ganho Acurácia (%) por Balanceamento das Classes

Até o momento, as análises enfatizam a relevância de uma consideração criteriosa das características inerentes a um conjunto de dados ao implementar a inclusão de características de grafo. A compreensão de como elementos como a quantidade, natureza e distribuição de características, assim como a tipificação dos rótulos, impactam o desempenho dos modelos, se revela como um elemento importante na tomada de decisões embasadas ao aplicar essa técnica em cenários de aprendizado de máquina supervisionados de classificação.

Outro aspecto de destaque no contexto deste experimento está diretamente relacionado à influência das estatísticas do grafo nos resultados obtidos pelos modelos. A Tabela 5.10 oferece uma síntese que expõe quais estatísticas de grafo demonstraram maior impacto positivo na métrica de Acurácia em diversos modelos de aprendizado de máquina. Assim como identificado no Cenário 1, esses resultados revelam uma alta variabilidade nas estatísticas de grafo que tendem a aprimorar a Acurácia, enfatizando a ideia de que diferentes estatísticas de grafo possuem níveis variados de eficácia, dependendo do modelo de ML considerado. Essa tendência se mantém mesmo com a expansão do espaço de hiperparâmetros e a aplicação de pesquisa em grade.

Os achados sugerem que a escolha criteriosa da estatística do grafo se configura como um fator de significativa importância na busca por melhorias no desempenho dos modelos. Um ponto de destaque reside na frequência com

que o conjunto de características selecionadas do grafo (FG) se posicionou como uma das melhores escolhas em todos os modelos. Embora não não tenha alcançado um ganho médio expressivo na métrica geral, esta alcançou bons ganhos em alguns modelos, com maior ênfase nos modelos Random Forest (RFOR) e Extreme Gradient Boosting (XGB). Ademais, nos casos de diversos modelos, a inclusão de uma única característica, como *degree* (DG), *hits* (HT) e *triangles* (TR) resultou em melhorias de perfomance, contribuindo para aprimoramento na métrica de Acurácia. Essa diversificação entre as estatísticas sublinha a necessidade de adotar uma abordagem criteriosa na seleção das características de grafo, de forma a adaptá-las ao modelo específico em análise e ao grafo de similaridade associado.

Tabela 5.10: Cenário 2– Ganho Médio e Frequência em Melhores Resultados por Estatística Grafo e Modelo – Acurácia (%)

Estat. Grafo	**DT**	**LPA**	**NBG**	**RFOR**	**RLOG**	**SVM**	**XGB**	**Geral**
AN		2,1						**2,1**
		1						**1**
BC		4,2	1,7		0,0			**2,3**
		2	3		1			**6**
CC				2,8		0,0		**1,4**
				1		1		**2**
CL	2,8						9,8	**7,5**
	1						2	**3**
DC	10,3				3,1	1,8		**4,3**
	1				3	1		**5**
DG	16,6	6,4		2,2		14,1		**10,4**
	2	2		1		1		**6**
EC	4,2	0,9	12,8			11,6		**7,4**
	1	1	1			1		**4**
FG	1,2	2,3	1,3	4,8	1,8	7,7	3,1	**3,6**
	1	1	2	4	3	3	6	**20**
HT	8,7	1,3		6,0	7,7	2,2	13,5	**6,9**
	2	1		2	2	1	1	**9**
LC	2,3		0,0	2,3	0,0	0,0		**0,9**
	1		1	1	1	1		**5**
PR				14,1		-0,3		**6,9**
				1		1		**2**
TR	0,9	1,2	11,6				-2,2	**5,1**
	1	2	3				1	**7**
Geral	**7,2**	**3,0**	**5,5**	**5,3**	**3,0**	**5,3**	**5,0**	**4,9**
	10	**10**	**10**	**10**	**10**	**10**	**10**	**70**

Observa-se que a estatística degree (DG) emerge com um alto incre-

mento médio de 10,4%. No entanto, é importante salientar que esse valor é consideravelmente influenciado pelo modelo DT no conjunto de dados BCC. Logo em seguida, as estatísticas eigenvector centrality (EC), clustering (CL) e hits (HT) se destacam por suas sólidas contribuições para a melhoria geral do desempenho dos modelos, com presenças significativas nos ganhos médios de Acurácia.

É relevante notar que também se evidenciam resultados indesejados, uma vez que algumas estatísticas não demonstraram um impacto positivo médio ou, até mesmo, apresentaram um impacto negativo na métrica Acurácia para os modelos. Isso se aplica particularmente às estatísticas como pagerank (PR), triangles (TR), betweenness centrality (BC), closeness centrality (CC) e load centrality (LC). Tais achados suscitam a necessidade de uma análise mais aprofundada, a fim de discernir as circunstâncias em que essas estatísticas podem oferecer vantagens significativas ou desvantagens em conjuntos de dados ou modelos específicos.

De forma abrangente, esses resultados destacam a importância da seleção criteriosa das estatísticas do grafo como uma etapa no aprimoramento do desempenho do modelo. Isso advém da constatação de que diversas estatísticas do grafo têm o potencial de influenciar os modelos de ML de maneiras distintas e variadas. Deste modo, a presente investigação ressalta a necessidade de uma abordagem criteriosa e orientada a dados na seleção de métricas de grafo, atentando para a adequação destas às particularidades de cada modelo de ML, com vistas a otimizar a capacidade preditiva em cenários de enriquecimento de dados por meio da inclusão de características de grafo. Outro aspecto importante, como veremos mais a frente, é dar atenção ao esforço de cálculo da estatística antes de seleciona-la.

No âmbito das medidas de similaridade, os desfechos revelados neste cenário se assemelham as constatações do Cenário 1. Uma análise da Tabela 5.11 realça a proeminência da medida *Jaccard*, que se destacou como a medida de similaridade mais frequentemente associada aos melhores resultados em termos de métrica de Acurácia entre as medidas de similaridade para características categóricas. É importante notar que boa parte dessas ocorrências foi observada em combinação com medidas de similaridade numéricas, notavelmente a medida *Euclidean*.

Esse resultado corrobora a noção de que a medida *Jaccard*, que é otimizada para características binárias, colheu substanciais vantagens a partir da conversão dessas características categóricas em formato binário, potencializando, assim, sua capacidade de identificar e quantificar similaridades entre as instâncias nos conjuntos de dados examinados. Essa observação lança luz sobre

a influência da engenharia de características na aplicação efetiva de medidas de similaridade, enaltecendo o impacto das estratégias de codificação binária nas análises de dados em que características categóricas estão em foco.

Tabela 5.11: Cenário 2: Frequência/Ganho (%) Melhores Resultados por Similaridade Categórica

Similaridade Categórica	Similaridade Numérica	Freq.	Ganho Acurácia
	Total	**24**	**1,2**
jaccard	euclidean	11	2,0
		7	-0,1
	mahalanobis	3	1,1
	manhattan	2	1,2
	cosine	1	2,3
	Total	**8**	**1,9**
overlap		6	2,4
	cosine	1	1,2
	mahalanobis	1	0,0
	Total	**3**	**3,0**
eskin		1	1,1
	euclidean	1	2,2
	mahalanobis	1	5,6

No que diz respeito às medidas de similaridade aplicadas a características numéricas, uma análise abrangente da Tabela 5.12 evidencia uma concentração dos resultados mais favoráveis em relação à métrica de Acurácia nas medidas *Mahalanobis* e *Euclidean* da mesma forma que observado no Cenário 1.

Assim como no Cenário 1, convém salientar que a eficácia de uma medida de similaridade específica pode variar de acordo com as particularidades de um modelo e vemos agora que isso também se aplica com o espaço de hiperpârametros ampliado. Algumas abordagens de aprendizado de máquina podem extrair maiores benefícios de medidas de distância, como *Mahalanobis* e *Euclidean*, enquanto outras podem demonstrar uma predileção por medidas de similaridade que se fundamentam em conceitos de ângulos, como a medida do Cosseno (*Cosine*), por exemplo.

Tabela 5.12: Cenário 2: Frequência/Ganho (%) Melhores Resultados por Similaridade Numérica

Similaridade Numérica	Similaridade Categórica	Freq.	Ganho Acurácia
mahalanobis	**Total**	**22**	**9,7**
	-	17	11,9
	jaccard	3	1,1
	eskin	1	5,6
	overlap	1	1,2
euclidean	**Total**	**18**	**4,2**
	jaccard	11	2,0
	-	6	8,5
	eskin	1	2,2
cosine	**Total**	**11**	**2,0**
	-	9	2,1
	jaccard	1	2,3
	overlap	1	0,0
manhattan	**Total**	**5**	**3,5**
	-	3	5,0
	jaccard	2	1,2

Da análise das medidas de similaridade deste Cenário 2, assim como no Cenário 1, verifica-se que é preciso testar várias medidas e ajustar o modelo de acordo com os resultados obtidos. Além disso, considerar as características do domínio do problema, explorar e analisar a estrutura do grafo gerado, são passos importantes para escolher as medidas de similaridade mais apropriadas para otimizar o desempenho do modelo.

No desfecho deste segundo cenário, uma análise mais aprofundada revela que, de modo geral, as evidências indicam um impacto positivo na métrica de Acurácia quando as características enriquecidas com estatísticas extraídas do grafo são incorporadas. Esse aprimoramento se torna maior em alguns conjuntos de dados ou modelos com a expansão do espaço de hiperparâmetros disponíveis para os modelos, juntamente com a aplicação de pesquisa em grade. No entanto, a alta variabilidade de resultados, que também foi observada no cenário anterior, ressalta a necessidade de uma abordagem minuciosa e adaptada ao contexto.

Resumimos os resultados agregados na Tabela 5.13 onde apresentamos os resultados dos dois Cenários para comparação e verificamos que houve

apenas diferenças pontuais entre os dois cenários, tais como nos conjunto de dados BCC e HDHNI em que os resultados do Cenário 2 (C2) apresentaram um ganho maior com menor dispersão em relação ao Cenário 1 (C1), o que sugere que a ampliação do espaço de hiperparâmetros disponíveis para os modelos, juntamente com a aplicação de pesquisa em grade fez com que as novas características do grafo ajudassem os modelos em suas predições. Por outro lado, os conjuntos VCB e WM que apresentaram um ganho maior para o Cenário 1 (C1) em relação ao Cenário 2 (C2), o que sugere que as características originais já fornecem informações suficientemente distintas para satisfazer as necessidades do modelo, ou que as novas características derivadas do grafo podem carecer de informação relevante para influenciar o modelo, o que leva o modelo a desconsiderá-las em suas decisões.

Tabela 5.13: Cenário 1 e 2: Ganho/Perda Acurácia (%) por Dataset, Cenário e Modelo

Dataset	**Cenário**	**DT**	**LPA**	**NBG**	**RFOR**	**RLOG**	**SVM**	**XGB**	**Média**
BCC	C1	20,8	8,4	25,0	12,5	4,1	16,7	12,5	**14,3**
	C2	29,2	8,4	25,0	12,5	8,3	20,8	16,6	**17,3**
CAR	C1	0,0	0,8	9,1	-1,7	0,0	-0,8	-0,5	**1,0**
	C2	0,9	0,9	7,8	-1,5	0,0	-0,3	-2,2	**0,8**
DM	C1	0,0	2,3	1,1	2,3	2,3	0,0	1,2	**1,3**
	C2	1,2	2,3	0,0	2,3	1,1	1,2	1,1	**1,3**
HDH	C1	0,0	1,2	1,2	3,4	1,1	1,1	2,3	**1,5**
	C2	2,3	1,2	1,2	5,6	1,1	1,2	0,0	**1,8**
HDHNI	C1	0,0	1,1	0,0	2,2	0,0	1,1	0,0	**0,6**
	C2	1,1	1,1	1,1	1,1	0,0	0,0	4,5	**1,3**
PIMA	C1	4,3	6,5	2,6	3,5	2,6	2,1	1,3	**3,3**
	C2	3,9	1,3	2,6	2,2	2,6	2,2	3,0	**2,5**
VCB	C1	16,2	9,6	10,3	14,8	11,0	10,9	13,5	**12,3**
	C2	16,2	5,1	0,0	13,5	10,3	11,6	13,5	**10,0**
VCM	C1	17,9	7,7	12,8	15,4	5,1	10,2	12,8	**11,7**
	C2	10,3	7,7	12,8	14,1	5,1	14,1	11,5	**10,8**
WM	C1	0,0	0,0	2,8	0,0	2,8	2,8	2,8	**1,6**
	C2	2,8	0,0	2,8	0,0	0,0	0,0	0,0	**0,8**
WQ	C1	0,6	2,1	2,1	2,9	1,7	1,0	1,9	**1,8**
	C2	4,2	2,1	1,9	2,8	1,7	1,8	1,7	**2,3**
Média	C1	**6,0**	**4,0**	**6,7**	**5,5**	**3,1**	**4,5**	**4,8**	**4,9**
	C2	**7,2**	**3,0**	**5,5**	**5,3**	**3,0**	**5,3**	**5,0**	**4,9**
Mediana	C1	**0,3**	**2,2**	**2,7**	**3,1**	**2,4**	**1,6**	**2,1**	**-**
	C2	**3,4**	**1,7**	**2,3**	**2,5**	**1,4**	**1,5**	**2,4**	**-**

Concluindo, de uma maneira geral, os dois Cenários experimentados alcançaram um ganho médio de acurácia 4,9% e com mediana entorno de 2,2%, apresentando evidências que sugerem que a inclusão das características com estatísticas extraídas do grafo melhorou os resultados de ganhos nas métricas

avaliadas nos dois cenários experimentados. No entanto, a variabilidade dos resultados, nos dois cenários experimentados, aponta para a necessidade da observação cuidadosa do perfil do conjunto de dados de dados que se está trabalhando, assim como da seleção criteriosa da adequação dos modelos, medidas de similaridade e estatísticas do grafo a serem trabalhados.

5.3 Tempos de Processamento

Em relação a duração do processamento do pipeline do método, selecionamos para análise os tempos médios de um processamento do pipeline de uma medida de Similaridade. Notamos que a quantidade de instâncias no conjunto de dados, assim como a existência de características categóricas e numéricas e a densidade do grafo de similaridade gerado, aumentam significativamente o consumo de tempo para algumas etapas, especificamente, para as etapas de Extração de Estatísticas do Grafo e de Treino e Teste.

Pelo gráfico na Figura 5.13, as etapas 3a e 4a foram agrupadas porque representam os tempos de construção dos grafos de Treino e de Teste, respectivamente, com pequena participação como consumidoras de tempo no pipeline do método. As etapas 3b e 4b também foram agrupadas porque representam os tempos de extração das estatísticas dos grafos de Treino e de Teste, respectivamente, sendo estas as principais consumidoras de tempo no pipeline do método. Como já citado, alguns conjuntos de dados podem gerar grafos mais densos o que pode elevar desproporcionalmente o tempo de criação do grafo nesses conjuntos de dados, assim como o esforço de cálculo na extração de estatísticas do grafo, afetando os tempos médios do pipeline para comparação.

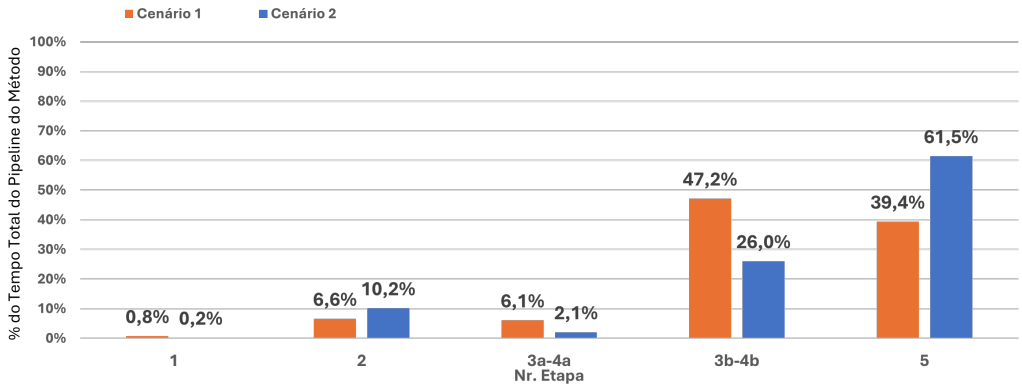


Figura 5.13: Cenário 1x2: Proporção média do tempo por etapa

A quantidade de instâncias no conjunto de dados, assim como a existência de características categóricas e numéricas, aumenta o consumo de tempo para as etapas de 2 a 5. O consumo de tempo pelo treino e teste dos modelos (etapas

2 e 5), para ambos os Cenários, sofre uma influência importante porque são efetuados, na realidade, um processamento para FO na etapa 2 e sete na etapa 5 com FO+FG e FG_i i=(1,..,6).

No Cenário 1, a maior proporção do tempo de processamento é consumida entre a extração das estatísticas dos grafos de treino/teste e o treino/teste dos modelos de ML, sendo que a extração das estatísticas dos grafos de treino e teste (etapas 3b e 4b) consumiram cerca de 47,2% do tempo total, ficando o treino e teste dos modelos (etapas 2 e 5) com cerca de 39,4% do total.

O maior consumo de tempo, no Cenário 1, pela extração das estatísticas do grafo (etapas 3b e 4b) é direcionado não só pela quantidade de instâncias e tipo de dados em cada conjunto de dados, mas, também, pela quantidade de estatísticas selecionadas (11) e, principalmente, pelo esforço de cálculo que algumas dessas estatísticas demandam. Estatísticas do grafo onde o cálculo envolve a análise de todos os pares de vértices, tornam o processo ainda mais lento, como as medidas de centralidade, em especial BC e LC, e as medidas de estrutura local, em especial CL.

Comparando esses resultados com os resultados da Tabela 5.4, verificamos que apenas a CL gerou resultados mais expressivos do que as demais estatísticas.

No Cenário 2, o aumento do espaço de parâmetros no Treinamento e Teste dos modelos aumentou o consumo de tempo nas etapas de Treino e Teste (etapas 2 e 5) de forma significativa, deslocando o protagonismo de tempo para essas etapas nesse Cenário com cerca de 71,6% do consumo de tempo contra 46% no Cenário 1 e a extração de estatísticas do Grafo (etapas 3b e 4b) com cerca de 39,4% nesse cenário contra 47,2% no Cenário 1.

O consumo de tempo nas etapas de extração de estatísticas do Grafo foram mantidos em relação ao Cenário 1, pois os dados, medidas de proximidade e estatísticas do grafo são os mesmos do Cenário 1, o que fez com que, de forma relativa, essas etapas tenham diminuído no Cenário 2 em relação ao Cenário 1.

Analisando o consumo de tempo nas etapas 3b e 4b (Extração de Estatísticas do Grafo de Treino e Teste), o gráfico na Figura 5.14 apresenta a proporção média de tempo que as estatíticas LC (*load centrality*), BC (*between centrality*), CL (*cluster*) e Outras 8 (AN, etc..) consomem no cálculo. Verificamos que LC (*load centrality*), BC (*between centrality*) e CL (*cluster*) consomem cerca de 97% do tempo dessas etapas, isto é, essas três medidas juntas consomem mais de 30x o consumo de tempo das demais 8 medidas juntas. o consumo de tempo de processamento para extração das estatísticas BC, LC e CL, é reflexo da complexidade de tempo dessas estatísticas

$O(n.m + n^2 log\ n)$, $O((m+n).n^2)$ e $O(n^2)$ respectivamente, onde n é o número de vértices (nós) e m o numero de arestas no grafo.

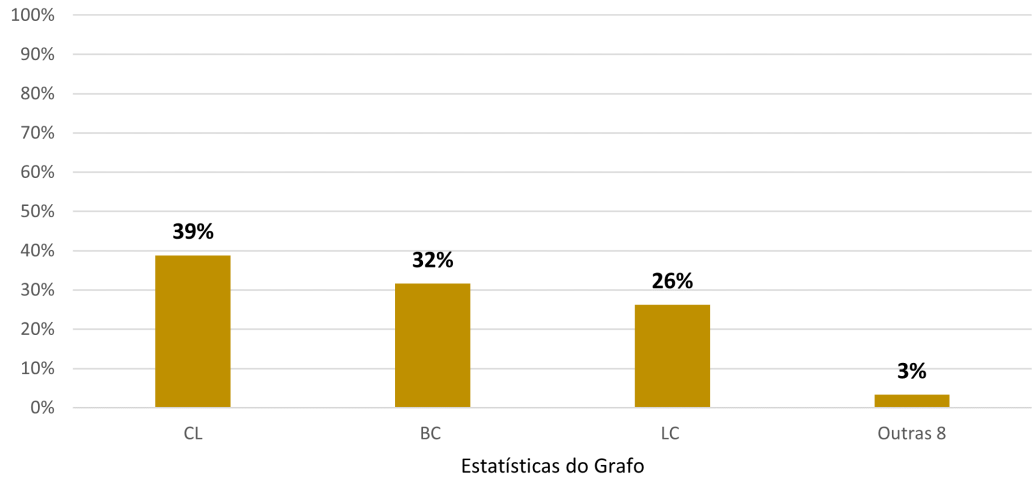


Figura 5.14: Proporção média do tempo nas etapas de Extração de Estatísticas do Grafo

O gráfico na Figura 5.15 apresenta, no Cenário 2, os ganhos médios de Acurácia (%) obtidos pelas estatísticas CL (*cluster*), BC (*between centrality*), LC (*load centrality*) e a sua média agrupada, comparando com a média de ganho das Outras 8 estatísticas do grafo. Essa comparação evidência que, no contexto dessa pesquisa, apenas a estatística CL (*cluster*) gerou resultados mais expressivos e que as estatísticas BC (*between centrality*) e LC (*load centrality*) alcançaram uma contribuição menor do que as demais estatísticas apesar do alto consumo de processamento, sendo que a média de ganho dessas três estatísticas ficou bem abaixo da média das Outras 8 estatísticas.

Como acontece com todos os métodos de enriquecimento de dados, a implementação do método proposto requer um esforço adicional em termos de tempo de processamento no pipeline original. Como discutido anteriormente, diversos fatores podem influenciar esse esforço.

Considerando, no Cenário 2, conforme apresentado apresentado no gráfico da Figura 5.16, o tempo de processamento de todo o pipeline de Treino-Teste com apenas o conjunto de dados contendo as características originais (FO) como linha de base (*Baseline*) com valor 1, a razão entre o pipeline original e o pipeline com as 11 estatísticas seleciondas pode chegar a 14x.

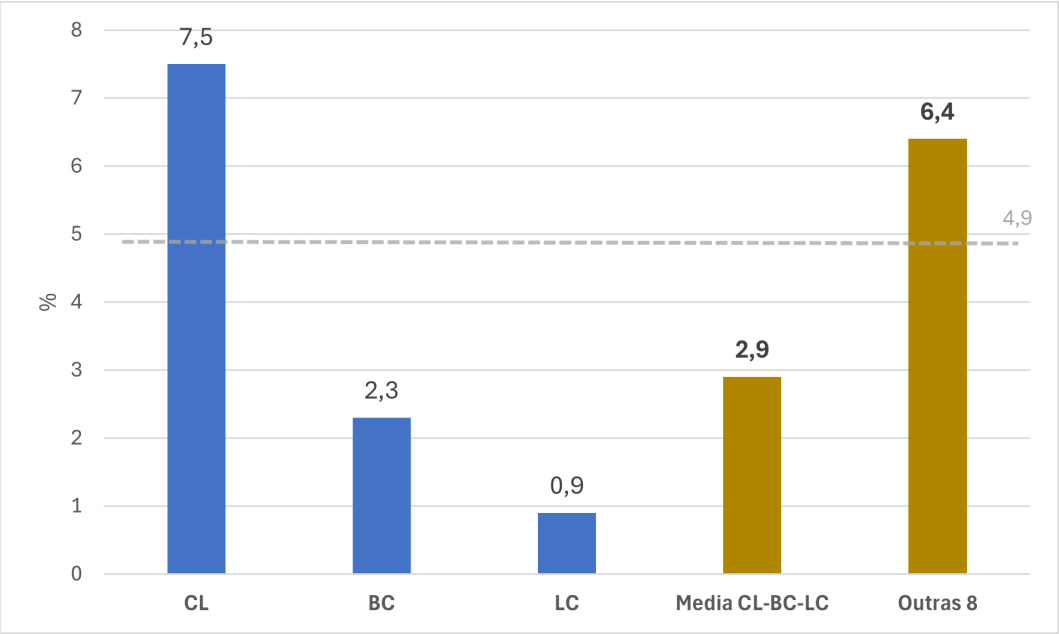


Figura 5.15: Ganho Médio no Cenário 2 por Estat. Grafo CL-BC-LC x Outras 8 – Acurácia (%)

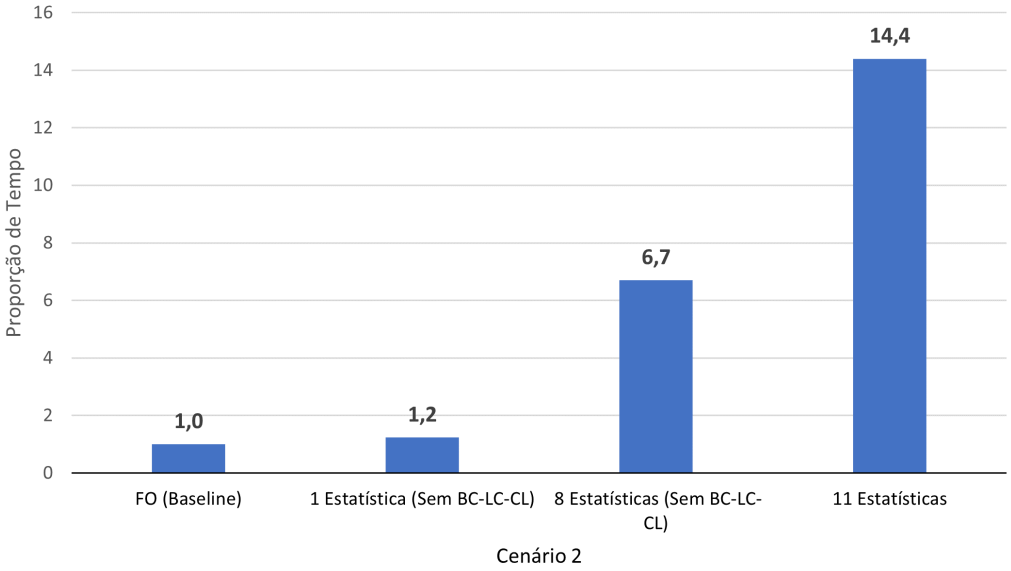


Figura 5.16: Cenário 2: Proporção média do tempo do pipeline

No entanto, como já discutido anteriormente, esse aumento no tempo de processamento deve-se, principalmente, as estatísticas LC (*load centrality*), BC (*between centrality*) e CL (*cluster*). Excluindo essas medidas do pipeline e mantendo as outras 8 estatísticas do grafo selecionadas, o esforço de tempo se reduz para cerca de 7x o *Baseline* (FO), o que está em linha com o esperado, pois na etapa de Treino e Teste dos Modelos com a inclusão das estatísticas do grafo (etapas 3b-4b), todos os modelos são submetidos a sete versões diferentes

do conjunto de dados, isto é, FO+FG e FG_i (*i=1,2,..,6*). Com uma estatística do grafo (sem BC, LC e CL) o esforço de tempo do pipeline se reduz para 1,2x o Baseline (FO), isto é, o incremento de tempo do pipeline em relação ao Baseline (FO) é de apenas 20%.

Esses dados de consumo de tempo no pipeline do método reforçam a relevância de que, dentre os critérios para a seleção de estatísticas a serem extraídas do grafo, deve também ser observado o impacto do esforço de cálculo dessas estatísticas no tempo de processamento. Da mesma forma, as medidas de proximidade adotadas devem também ser selecionadas com base em uma análise exploratória prévia do grafo gerado e observar sua estrutra na busca de inferir a sua adequação para utilização no pipeline.

Mais detalhes do tempo de processamento nas etapas do pipeline do método estão disponíveis no Apêndice J.4.

5.4 Comparação de Resultados com Pesquisas Anteriores

Nesta seção, conduziremos uma análise dos resultados obtidos em nosso estudo, comparando-os com os desempenhos alcançados em pesquisas científicas anteriores que empregaram os mesmos conjuntos de dados e métrica de acurácia. Inicialmente, apresentaremos uma comparação entre os melhores resultados obtidos em nosso estudo para cada conjunto de dados e modelo de aprendizado de máquina. Em seguida, expandiremos nossa análise, explorando como o melhor desempenho de nosso estudo se compara com os resultados mais destacados dos artigos de referência, proporcionando uma perspectiva abrangente do impacto das estatísticas de grafo em nossa pesquisa e o valor em relação à literatura científica existente.

A escolha da métrica de acurácia, em particular, como medida de avaliação, foi em função de que foi a única medida presente em todos os artigos selecionados e, portanto, serve como um denominador comum para a avaliação do desempenho dos modelos de aprendizado de máquina. A métrica de acurácia é frequentemente utilizada em tarefas de classificação, sendo importante em muitos domínios de aplicação. Assim, ao focalizar a acurácia, estamos alinhando nossa análise com os requisitos típicos da literatura acadêmica e prática.

Pela leitura dos dados apresentados na Tabela 5.14, verificamos que no conjunto de dados BCC os resultados alcançados no Cenário 2 foram superiores em alguns modelos tais como DT, KNN e outros. Por outro lado, outros modelos, principalmente aqueles que utilizaram a técnica de bagging, foram superados pelos resultados do artigo de referência.

No conjunto de dados CAR os modelos do Cenário 2 alcançaram resultados superiores aos observados no artigo referência, mas com pequena margem, exceto o modelo NBG. Da mesma forma, nos conjuntos de dados DM, HDH, PIMA e WQ os resultados ficaram muito próximos em todos os modelos. No conjunto de dados HDHNI e VCB todos os modelos do Cenário 2 tiveram desempenho superior em relação ao artigo de referência, sugerindo que o método proposto se destacou para esses conjuntos de dados e esses modelos específicos.

Nos conjuntos de dados VCM2 (VCM aplicando a Estratégia 2) e WM a maior parte dos modelos do Cenário 2 alcançou resultados superiores com algumas exceções, tais como MLP no conjunto de dados WM e o modelo RLOG no conjunto VCM2. Na mesma linha os modelos no conjunto de dados VCM1 (VCM aplicando a Estratégia 1) do Cenário 2 foram superados em todos os modelos, exceto nos modelos RLOG e SVM.

Tabela 5.14: Melhores Resultados Cenário x Artigo por Dataset e Modelo – Acurácia (%)

Dataset	**Modelo**	**C2**	**Artigo**	**Dataset**	**Modelo**	**C2**	**Artigo**
BCC	BAGG_DT	91,7	**100,0**	CAR	KNN	**94,4**	93,5
	BAGG_KNN	91,7	**100,0**		NBG	**81,1**	69,3
	BAGG_RFOR	**91,7**	90,9		RFOR	**95,4**	93,8
	BAGG_RLOG	87,5	**91,7**				
	BAGG_SVM	**91,7**	83,3	DM	MLP	**98,8**	98,0
	DT	**91,7**	66,7		SVM	**98,8**	97,0
	KNN	**95,8**	89,9				
	RFOR	**91,7**	75,0				
	RLOG	**83,3**	75,0				
	SVM	**95,8**	83,3				
HDHNI	DT	**86,5**	82,9	PIMA	DT	**77,1**	75,7
	GB	**85,4**	82,3		NBG	74,9	**77,8**
	MLP	**86,5**	81,5		RFOR	78,4	**79,6**
	NBB	**86,5**	78,7				
	RFOR	**86,5**	81,3	VCB	SVM	**96,1**	86,9
	RLOG	**85,4**	81,9				
VCM1	ADAB	92,3	**95,0**	VCM2	ADAB	**92,3**	86,0
	ETREE	93,6	**99,0**		ETREE	**93,6**	85,0
	RFOR	97,4	**98,0**		RFOR	**97,4**	86,0
	RLOG	**87,2**	85,0		RLOG	87,2	**93,0**
	SVM	**94,9**	80,0		SVM	**94,9**	90,0
WM	DT	**97,2**	74,6	WQ	NBG	**57,3**	55,9
	MLP	**100,0**	99,1		RFOR	**69,0**	65,5
	NBG	**100,0**	97,3		SVM	65,8	**68,6**
	SVM	**100,0**	71,9				
HDH	DT	**98,9**	95,5				
	GB	98,9	98,8				
	MLP	**96,6**	94,0				
	NBB	**91,0**	82,8				
	RFOR	98,9	99,0				
	RLOG	96,6	**98,8**				

A Tabela 5.15 apresenta os resultados da Acurácia, por conjunto de dados, nos modelos que alcançaram maiores valores nessa medida no Cenário 2 e os maiores valores alcançados nos artigos de referência selecionados para esses

conjuntos de dados. Analisando esses dados verificamos que os resultados do Cenário 2 ficaram bem próximos e, em alguns casos, até superaram os resultados dos artigos de referência, com destaque positivo para os resultados nos conjuntos de dados DM, HDHNI, VCB, VCM2 (VCM aplicando a Estratégia 2) e destaque negativo para os resultados nos conjuntos de dados BCC e VCM1 (VCM aplicando a Estratégia 1).

Tabela 5.15: Melhor Modelo Cenário 2 e Artigo por Dataset – Acurácia (%)

	Cenário 2		**Artigo**	
Dataset	**Modelo**	**(FO+FG)**	**Medida**	**Melhor Modelo**
BCC	SVM	95,8	**100,0**	BAGG_KNN
CAR	SVM	**98,5**	96,0	EAC
DM	SVM	98,8	99,0	ANN-SVM
HDH	RFOR	98,9	99,0	RFOR
HDHNI	DT	**86,5**	82,9	DT
PIMA	RFOR	78,4	**79,6**	RFOR
VCB	XGB	**98,7**	92,1	ANN
VCM1	RFOR	97,4	**99,0**	ETREE
VCM2	RFOR	**97,4**	93,0	RLOG
WM	SVM	100,0	100,0	MONT
WQ	RFOR	69,0	68,6	SVM

De uma maneira geral, a análise das tabelas entre os resultados do Cenário 2 e os respectivos artigos de referência selecionados, apresentam evidências que sugerem que o método proposto neste estudo é competitivo em relação aos resultados apresentados nesses artigos e que tem o potencial de aprimorar o desempenho de modelos de aprendizado de máquina supervisionados de classificação, particularmente em conjuntos de dados e modelos específicos. Vale a pena mencionar que pequenas variações podem ser esperadas devido a diferentes configurações experimentais, versões de modelos usadas ou pré-processamento/tratamento de dados aplicados.

6
Conclusão e Trabalhos Futuros

Este trabalho apresentou uma abordagem para enriquecimento de conjuntos de dados tabulares, baseados em estatísticas de grafo de similaridade de suas instâncias, para melhora na performance de modelos de aprendizado de máquina supervisionados de classificação.

Para alcançar esse objetivo, foram delineados objetivos específicos e questões de pesquisa que foram respondidas por meio da metodologia proposta. A questão de pesquisa (Q1) sobre a utilização de métricas de grafo para enriquecer conjuntos de dados que não possuam características explícitas, foi abordada pela aplicação de medidas de proximidade entre as instâncias do conjunto de dados, permitindo a construção do grafo e a extração de estatísticas visando o aprimoramento do conjunto de dados original.

A metodologia proposta envolveu a utilização de técnicas de teoria dos grafos, redes complexas, medidas de similaridade e distância, estatísticas do grafo, seleção de características, modelos de aprendizado de máquina supervisionados para problemas de classificação e a avaliação de modelos.

Esse método foi avaliado em experimentos com dez conjuntos de dados reais de domínios e perfis diversos, em dois cenários diferentes e pela comparação dos resultados do segundo cenário com os resultados de artigos científicos relevantes na área de Aprendizado de Máquina, que utilizaram o mesmo conjunto de dados em suas avaliações.

No cenário 1, em que as características originais foram utilizadas com parâmetros padrão nos modelos de aprendizado de máquina selecionados, os resultados sugerem que a inclusão de estatísticas de grafo pode melhorar significativamente a performance dos modelos de aprendizado de máquina supervisionados de classificação em conjuntos de dados com características distintas, com a média de ganhos de Acurácia de cerca de 4,9% e com mediana de 2,3%.

No cenário 2, a metodologia foi avaliada com a ampliação do espaço de parâmetros para otimização do desempenho em um processo de pesquisa em grade para otimização de hiperparâmetros dos modelos, e os resultados de Acurácia mostraram praticamente o mesmo desempenho de média (4,9%) e mediana (2,2%) do Cenário 1 com diferentes variações entre os modelos e conjuntos de dados.

Dentre os modelos selecionados nesse estudo, aqueles que mais se beneficiaram do método, dentro dos conjuntos de dados selecionados, foram DT,

NBG, RFOR, SVM e XGB. As medidas de similaridade que mais se destacaram foram *Euclidean* e *Mahalanobis* pelas características numéricas e *Jaccard* pelas características categóricas. Pelo lado das estatísticas extraídas do grafo, aquelas que mais contribuíram para os ganhos de performance nos resultados foram *degree* (DG), *eigenvector centrality*, (EC), *clustering* (CL), *hits* (HT) e o conjunto de todas as estatísticas selecionadas (FG).

Os resultados obtidos, sugerem uma influência positiva das características derivadas de estatísticas extraídas do grafo de similaridade na performance dos modelos de predição e na avaliação das medidas de similaridade e estatísticas de grafo selecionadas, endereçando a questão Q2.

Através da comparação dos resultados do Cenário 2 com os *benchmarks* estabelecidos por meio de artigos científicos relevantes na área de Aprendizado de Máquina e que utilizaram o mesmo conjunto de dados em suas avaliações, foi possível avaliar a eficácia do método em diferentes configurações e situá-lo em relação às melhores práticas e resultados previamente reportados na literatura acadêmica.

A maioria dos resultados de Acurácia do Cenário 2 ficaram acima dos resultados de artigos de referência, sugerindo que a metodologia proposta neste estudo é competitiva em relação aos resultados apresentados nesses artigos, indicando que tem o potencial de ser mais uma alternativa competitiva para aprimoramento do desempenho de modelos de aprendizado de máquina supervisionados de classificação, particularmente em conjuntos de dados e modelos específicos, endereçando a questão de pesquisa Q3.

Vale a pena mencionar que pequenas variações podem ser esperadas devido a diferentes configurações experimentais, versões de modelos usadas ou pré-processamento/tratamento de dados aplicados.

Em resumo, este trabalho abordou questões importantes relacionadas ao enriquecimento de dados com base em estatísticas de grafo de similaridade para melhorar o desempenho em Modelos de ML supervisionados de classificação, apresentando uma técnica que habilita a sua utilização em conjuntos de dados que não possuem as características que permitam a criação direta do grafo e sem a necessidade de acesso a bases externas para o enriquecimento.

Os resultados obtidos nos Cenário 1 e 2, bem como a comparação com os *benchmarks* estabelecidos por meio de artigos científicos relevantes na área de Aprendizado de Máquina, indicam que a técnica proposta pode ser uma alternativa eficaz para melhorar o desempenho de modelos de Aprendizado de Máquina Supervisionados de Classificação em diferentes configurações.

Além disso, o trabalho forneceu informações sobre teoria dos grafos, redes complexas, medidas de similaridade e distância, estatísticas do grafo, seleção de

características e avaliação de modelos, que podem ser úteis para pesquisadores e profissionais que trabalham com aprendizado de máquina. Esperamos que este trabalho possa contribuir para o avanço da área e inspirar novas pesquisas sobre o tema.

Limitações

A variabilidade dos resultados, nos dois cenários experimentados, aponta para a necessidade da observação cuidadosa do perfil do conjunto de dados que se está trabalhando, assim como da seleção criteriosa da adequação dos modelos, medidas de similaridade e estatísticas do grafo a serem trabalhados.

Além disso, observamos que os conjuntos de dados contendo características categóricas apresentaram, ao incluir estatísticas de grafo de similaridade, um desempenho menor em relação aos conjuntos de dados com características numéricas. Especificamente, esse padrão foi mais evidente ao utilizar as medidas de similaridade categóricas *Overlap* e *Eskin*. Esses achados indicam que tais medidas podem não ser as mais adequadas para os conjuntos de dados analisados nesta pesquisa, sugerindo a necessidade de considerar abordagens alternativas ao lidar com características categóricas em estudos futuros.

Adicionalmente, identificamos um aumento significativo no tempo de processamento do *pipeline* do método devido ao cálculo das estatísticas do grafo *betweenness centrality* (BC), *load centrality* (LC) e *clustering* (CL), ressaltando a importância de considerar cuidadosamente a complexidade computacional das estatísticas do grafo ao aplicar essa metodologia.

Essas limitações ressaltam a necessidade de investigações futuras para abordar esses desafios e refinamentos na técnica proposta.

Trabalhos Futuros

Os resultados desta pesquisa identificaram oportunidades significativas para avanços futuros no enriquecimento de dados tabulares com base em estatísticas de grafo de similaridade, visando melhorar a performance de modelos de aprendizado de máquina supervisionados de classificação.

Uma área para investigação futura é a avaliação de novas medidas de similaridade e dissimilaridade, especialmente direcionadas para características numéricas e categóricas. A exploração de medidas mais específicas e adequadas para diferentes tipos de atributos, pode enriquecer a compreensão das relações entre instâncias, aprimorando a qualidade do grafo gerado, das estatísticas extraídas do grafo e, por conseguinte, dos modelos de predição.

Além disso, expandir as estatísticas extraídas do grafo, com foco na captura da estrutura dos relacionamentos (arestas), apresenta-se como outra área relevante para investigação. O desenvolvimento de novos atributos derivados

do grafo pode enriquecer ainda mais a representação dos dados, contribuindo significativamente para a eficácia dos modelos de aprendizado de máquina.

Refinar os procedimentos para calcular os pesos dos relacionamentos no grafo e desenvolver novos procedimentos para estabelecer limites (*thresholds*) para a aceitação das arestas é fundamental. A exploração de novos métodos adaptados à especificidade dos conjuntos de dados e modelos pode aprimorar a precisão e relevância das informações extraídas.

Explorar estatísticas de nível global do grafo permitiriam uma compreensão mais abrangente do comportamento do grafo como um todo, o que poderia ajudar a identificar padrões de conectividade, comunidades ou centralidades globais que poderiam antecipar se as estatísticas de nível nodal e/ou de arestas levariam a melhorias significativas no desempenho dos modelos de ML quando incorporadas aos conjuntos de dados originais.

Por fim, uma outra perspectiva para pesquisas futuras reside na adaptação e avaliação do método proposto para aplicação em modelos supervisionados de regressão e modelos não supervisionados. Esta ampliação do escopo de utilização permitiria explorar o potencial do método em contextos adicionais de aprendizado de máquina, possibilitando a sua aplicação em tarefas de regressão, onde a predição de valores contínuos é o foco principal, e em abordagens não supervisionadas, onde a ênfase recai na identificação de padrões e estruturas nos dados sem a necessidade de rótulos de classe.

Essas áreas de investigação representam direções para futuras pesquisas, visando aprimorar a metodologia proposta e sua eficiência em diferentes contextos de aplicação de aprendizado de máquina.

7
Referências bibliográficas

ABDELMAGEED, N. Towards transforming tabular datasets into knowledge graphs. In: **The Semantic Web: ESWC 2020 Satellite Events: ESWC 2020 Satellite Events, Heraklion, Crete, Greece, May 31 – June 4, 2020, Revised Selected Papers**. Berlin, Heidelberg: Springer-Verlag, 2020. p. 217—228. ISBN 978-3-030-62326-5. Disponível em: <https://doi.org/10.1007/978-3-030-62327-2_37>.

ABUSAMRA, H. A comparative study of feature selection and classification methods for gene expression data of glioma. **Procedia Computer Science**, v. 23, p. 5–14, 12 2013.

AEBERHARD, S.; FORINA, M. **Wine**. 1991. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5PC7J.

ALBREIKI, B.; HABUZA, T.; ZAKI, N. Extracting topological features to identify at-risk students using machine learning and graph convolutional network models. **Int. J. Educ. Technol. High. Educ.**, Springer Science and Business Media LLC, v. 20, n. 1, abr. 2023.

ALBUQUERQUE, R. W. d. **Monitoramento da cobertura do solo no entorno de hidrelétricas utilizando o classificador SVM (Support Vector Machines)**. Tese (Doutorado), 2012. DOI: 10.11606/D.3.2011.tde-06062012-164051. Disponível em: <https://www.teses.usp.br/teses/disponiveis/3/3138/tde-06062012-164051/>.

ALHARBI, A.; ALSUBHI, K. Botnet detection approach using graph-based machine learning. **IEEE Access**, v. 9, p. 99166–99180, 07 2021. DOI: 10.1109/ACCESS.2021.3094183. Disponível em: <https://ieeexplore.ieee.org/document/9471889>.

ANSARI, S. et al. Diagnosis of vertebral column disorders using machine learning classifiers. In: **2013 International Conference on Information Science and Applications, ICISA**. [s.n.], 2013. p. 1–6. ISBN 978-1-4799-0602-4. DOI: 10.1109/ICISA.2013.6579446. Disponível em: <https://ieeexplore.ieee.org/document/6579446>.

ANTHONISSE, J. **The Rush in a Directed Graph**. Stichting Mathematisch Centrum. Mathematische Besliskunde, 1971. Disponível em: <https://books.google.com/books?id=tVH-rQEACAAJ>.

BAAZOUZI, W.; KACHROUDI, M.; FAIZ, S. Towards an efficient fairification approach of tabular data with knowledge graph models. **Procedia Computer Science**, v. 207, p. 2727–2736, 2022. ISSN 1877-0509. Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 26th International Conference KES2022. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050922012200>.

BACH, N.; BADASKAR, S. A review of relation extraction. 2007. Disponível em: <https://api.semanticscholar.org/CorpusID:9249117>.

BARABASI, A.-L.; MARTINO, M.; POSFAI, M. **Network Science**. Cambridge University Press, PDF version., 2012. Disponível em: <https://eravilaipnada.files.wordpress.com/2016/01/barabasi-2012.pdf>.

BARRAT, A. et al. The architecture of complex weighted networks. **Proceedings of the National Academy of Sciences**, v. 101, n. 11, p. 3747–3752, 2004. Disponível em: <https://www.pnas.org/doi/abs/10.1073/pnas.0400087101>.

BARRETO, G.; NETO, A. **Vertebral Column**. 2011. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5K89B.

BAUMANN, A. et al. Changing perspectives: Using graph metrics to predict purchase probabilities. **Expert Systems with Applications**, v. 94, 10 2017. DOI: https://doi.org/10.1016/j.eswa.2017.10.046. Disponível em: <https://www.researchgate.net/publication/320580414_Changing_Perspectives_Using_Graph_Metrics_to_Predict_Purchase_Probabilities>.

BHAGAVATULA, C. S.; NORASET, T.; DOWNEY, D. Methods for exploring and mining tables on wikipedia. In: **Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics**. New York, NY, USA: Association for Computing Machinery, 2013. (IDEA '13), p. 18—-26. ISBN 9781450323291. DOI: https://doi.org/10.1145/2501511.2501516. Disponível em: <https://doi.org/10.1145/2501511.2501516>.

BHAVANA, N.; MEGHANA, S. C.; PRADEEP, K. R. Review of ensemble machine learning approach in prediction of diabetes diseases. **International Journal on Future Revolution in Computer Science Communication Engineering**, v. 4, p. 463—466, 2018. Disponível em: <https://www.academia.edu/36692804/A_Review_of_Ensemble_Machine_Learning_Approach_in_Prediction_of_Diabetes_Diseases>.

BHOI, S. K. Prediction of diabetes in females of pima indian heritage: A complete supervised learning approach. 2021. Disponível em: <https://api.semanticscholar.org/CorpusID:236616594>.

BOEHMKE, B.; GREENWELL, B. **Hands-On Machine Learning with R**. Chapman and Hall/CRC, 2020. DOI: https://doi.org/10.1201/9780367816377. ISBN 9780367816377. Disponível em: <https://bradleyboehmke.github.io/HOML/>.

BOHANEC, M. **Car Evaluation**. 1997. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5JP48.

BORIAH, S.; CHANDOLA, V.; KUMAR, V. Similarity measures for categorical data: A comparative evaluation. In: _____. **Proceedings of the 2008 SIAM International Conference on Data Mining (SDM)**. [S.l.: s.n.], 2008. p. 243–254. DOI: https://doi.org/10.1137/1.9781611972788.22.

BRANDES, U. A faster algorithm for betweenness centrality. **The Journal of Mathematical Sociology**, Routledge, v. 25, n. 2, p. 163–177, 2001. DOI: https://doi.org/10.1080/0022250X.2001.9990249.

BRANDES, U. On variants of shortest-path betweenness centrality and their generic computation. **Social Networks**, v. 30, n. 2, p. 136–145, 2008. ISSN 0378-8733. DOI: https://doi.org/10.1016/j.socnet.2007.11.001. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0378873307000731>.

BRIN, S.; PAGE, L. The anatomy of a large-scale hypertextual web search engine. **Computer Networks and ISDN Systems**, v. 30, n. 1, p. 107–117, 1998. ISSN 0169-7552. Proceedings of the Seventh International World Wide Web Conference. Disponível em: <https://www.sciencedirect.com/science/article/pii/S016975529800110X>.

BRYAN, J.; MORIANO, P. Graph-based machine learning improves just-in-time defect prediction. **PLoS ONE**, v. 18, n. 4, 4 2023. ISSN 1932-6203. DOI: https://doi.org/10.1371/journal.pone.0284077. Disponível em: <https://www.osti.gov/biblio/1971024>.

BUCKLAND, M.; GEY, F. The relationship between recall and precision. **Journal of the American Society for Information Science**, v. 45, n. 1, p. 12–19, 1994. DOI: https://doi.org/10.1002/(SICI)1097-4571(199401)45:1<12::AID-ASI2>3.0.CO;2-L. Disponível em: <https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291097-4571%28199401%2945%3A1%3C12%3A%3AAID-ASI2%3E3.0.CO%3B2-L>.

CAFARELLA, M. J.; HALEVY, A.; KHOUSSAINOVA, N. Data integration for the relational web. **Proc. VLDB Endow.**, VLDB Endowment, v. 2, n. 1, p. 1090–1101, aug 2009. ISSN 2150–8097. DOI: https://doi.org/10.14778/1687627.1687750.

CARVALHO, F. de A. T. de. Proximity coefficients between boolean symbolic objects. In: DIDAY, E. et al. (Ed.). **New Approaches in Classification and Data Analysis**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994. p. 387–394. ISBN 978-3-642-51175-2. DOI: https://doi.org/10.1007/978-3-642-51175-2$_4$4.

CHABOT, Y. et al. Dagobah: An end-to-end context-free tabular data semantic annotation system. 2019. Disponível em: <https://api.semanticscholar.org/CorpusID:208222267>.

CHANG, V. et al. Pima indians diabetes mellitus classification based on machine learning (ML) algorithms. **Neural Comput. Appl.**, Springer Science and Business Media LLC, v. 35, n. 22, p. 1–17, mar. 2022. Disponível em: <https://link.springer.com/article/10.1007/s00521-022-07049-z>.

CHAVES, A. d. C. F. **Extração de Regras Fuzzy para Máquinas de Vetor Suporte (SVM) para Classificação em Múltiplas Classes**. Tese (Doutorado), 2006. DOI: https://doi.org/10.17771/PUCRio.acad.9191.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: Association for Computing Machinery, 2016. (KDD '16), p. 785–794. ISBN 9781450342322. DOI: https://doi.org/10.1145/2939672.2939785.

CHICCO, D.; JURMAN, G. The advantages of the matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. **BMC Genomics**, Springer Science and Business Media LLC, v. 21, n. 1, p. 6, jan. 2020. DOI: https://doi.org/10.1186/s12864-019-6413-7.

CORTEZ, P. et al. **Wine Quality**. 2009. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C56S3T.

COSTA, L. d. F. et al. Characterization of complex networks: A survey of measurements. **Adv. Phys.**, Informa UK Limited, v. 56, n. 1, p. 167–242, jan. 2007. DOI: https://doi.org/10.1080/00018730601170527.

CVETKOV-ILIEV, A.; ALLAUZEN, A.; VAROQUAUX, G. Rrelational data embeddings for feature enrichment with background information. **Mach. Learn.**, Springer Science and Business Media LLC, v. 112, n. 2, p. 687–720, fev. 2023. DOI: https://doi.org/10.1007/s10994-022-06277-7.

DAYA, A. A. et al. Botchase: Graph-based bot detection using machine learning. **IEEE Transactions on Network and Service Management**, v. 17, n. 1, p. 15–29, 2020.

DETROJA, K.; BHENSDADIA, C.; BHATT, B. S. A survey on relation extraction. **Intelligent Systems with Applications**, v. 19, p. 200244, 2023. ISSN 2667-3053. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2667305323000698>.

DEZA, M. M.; DEZA, E. **Encyclopedia of distances**. 2009. ed. Berlin, Germany: Springer, 2009.

DIESTEL, R. **Graph Theory**. 5. ed. Berlin, Germany: Springer, 2017. (Graduate texts in mathematics).

DONG, Y.; OYAMADA, M. Table enrichment system for machine learning. In: **Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: Association for Computing Machinery, 2022. (SIGIR '22), p. 3267—3271. ISBN 9781450387323. Disponível em: <https://doi.org/10.1145/3477495.3531678>.

DONG, Y. et al. Efficient joinable table discovery in data lakes: A high-dimensional similarity-based approach. **2021 IEEE 37th International Conference on Data Engineering (ICDE)**, p. 456–467, 2020. Disponível em: <https://api.semanticscholar.org/CorpusID:225070738>.

ELSINGHORST, S. **Machine Learning Basics - Gradient Boosting XGBoost**. 2018. Disponível em: <https://shirinsplayground.netlify.app/2018/11/ml_basics_gbm/>.

ESCOVEDO, T.; KOSHIYAMA, A. **Introducao a Data Science - Algoritmos de Machine Learning e metodos de analise**. Casa do Codigo, 2020. ISBN 9788572540551. Disponível em: <https://books.google.com/books?id=cL7TDwAAQBAJ>.

ESKIN, E. et al. A geometric framework for unsupervised anomaly detection. In: _____. **Applications of Data Mining in Computer Security**. Boston, MA: Springer US, 2002. p. 77–101. Disponível em: <https://doi.org/10.1007/978-1-4615-0953-0_4>.

FETAHU, B.; ANAND, A.; KOUTRAKI, M. Tablenet: An approach for determining fine-grained relations for wikipedia tables. In: **The World Wide Web Conference**. New York, NY, USA: Association for Computing Machinery, 2019. (WWW '19), p. 2736—-2742. ISBN 9781450366748. Disponível em: <https://doi.org/10.1145/3308558.3313629>.

FREEMAN, L. A set of measures of centrality based on betweenness. **Sociometry**, American Sociological Association, Sage Publications, Inc., v. 40, p. 35–41, 03 1977.

GAN, G.; MA, C.; WU, J. **Data Clustering: Theory, Algorithms, and Applications**. Society for Industrial and Applied Mathematics, 2007. (ASA-SIAM Series on Statistics and Applied Probability). ISBN 9780898716238. Disponível em: <https://books.google.com/books?id=rIQZAQAAIAAJ>.

GARATE-ESCAMILA, A. K.; Hajjam El Hassani, A.; ANDRES, E. Classification models for heart disease prediction using feature selection and pca. **Informatics in Medicine Unlocked**, v. 19, p. 100330, 2020. ISSN 2352-9148. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2352914820300125>.

GARCíA, S. et al. An empirical comparison of botnet detection methods. **Computers Security**, v. 45, p. 100–123, 2014. ISSN 0167-4048. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167404814000923>.

GERON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. 2nd. ed. [S.l.]: O'Reilly Media, Inc., 2019. ISBN 1492032646.

GEURTS, P.; ERNST, D.; WEHENKEL, L. Extremely randomized trees. **Machine Learning**, v. 63, p. 3–42, 04 2006.

GOH, K.-L.; KAHNG, B.; KIM, D. Universal behavior of load distribution in scale-free networks. **Physical review letters**, v. 87 27 Pt 1, p. 278701, 2001. Disponível em: <https://api.semanticscholar.org/CorpusID:15746304>.

GOTTSCHALK, S.; DEMIDOVA, E. Tab2kg: Semantic table interpretation with lightweight semantic profiles. **Semantic Web**, IOS Press, v. 13, n. 3, p. 571––597, abr. 2022. ISSN 1570-0844. Disponível em: <http://dx.doi.org/10.3233/SW-222993>.

GOWER, J. C. A general coefficient of similarity and some of its properties. **Biometrics**, Wiley, International Biometric Society, v. 27, n. 4, p. 857–871, 1971. ISSN 0006341X, 15410420. Disponível em: <http://www.jstor.org/stable/2528823>.

GRANDINI, M.; BAGLI, E.; VISANI, G. Metrics for multi-class classification: an overview. **ArXiv**, abs/2008.05756, 2020. Disponível em: <https://api.semanticscholar.org/CorpusID:221112671>.

GRUS, J. **Data Science do Zero**. Alta Books, 2016. ISBN 9788550803876. Disponível em: <https://books.google.com.br/books?id=2LZwDwAAQBAJ>.

GULUM, M. **Horse racing prediction using graph-based features**. Tese (Doutorado), 2018.

GUPTA, R.; SARAWAGI, S. Answering table augmentation queries from unstructured lists on the web. **Proc. VLDB Endow.**, VLDB Endowment, v. 2, n. 1, p. 289—300, aug 2009. ISSN 2150-8097. Disponível em: <https://doi.org/10.14778/1687627.1687661>.

GUYON, I.; ELISSEEFF, A. An introduction of variable and feature selection. **J. Machine Learning Research Special Issue on Variable and Feature Selection**, v. 3, p. 1157–1182, 01 2003.

HAGBERG, A. A.; SCHULT, D. A.; SWART, P. J. Exploring network structure, dynamics, and function using networkx. In: VAROQUAUX, G.; VAUGHT, T.; MILLMAN, J. (Ed.). **Proceedings of the 7th Python in Science Conference**. Pasadena, CA USA: [s.n.], 2008. p. 11–15.

HAMILTON, W. L. Graph representation learning. **Synthesis Lectures on Artificial Intelligence and Machine Learning**, Morgan and Claypool, v. 14, n. 3, p. 1–159, 2020.

HAN, J.; KAMBER, M.; PEI, J. **Data Mining: Concepts and Techniques**. Elsevier Science, 2012. (The Morgan Kaufmann Series in Data Management Systems). ISBN 9780123814807. Disponível em: <https://books.google.com/books?id=pQws07tdpjoC>.

HAND, D.; MANNILA, H.; SMYTH, P. **Principles of Data Mining**. MIT Press, 2001. (Adaptive Computation and Machine Learning series). ISBN 9780262082907. Disponível em: <https://books.google.com/books?id=SdZ-bhVhZGYC>.

HARARI, A.; KATZ, G. Few-shot tabular data enrichment using fine-tuned transformer architectures. In: MURESAN, S.; NAKOV, P.; VILLAVICENCIO, A. (Ed.). **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**. Dublin, Ireland: Association for Computational Linguistics, 2022. p. 1577–1591. Disponível em: <https://aclanthology.org/2022.acl-long.111>.

HERNÁNDEZ, J. M.; MIEGHEM, P. V. Classification of graph metrics. In: . [s.n.], 2015. Disponível em: <https://api.semanticscholar.org/CorpusID:37136216>.

HUTTER, F.; KOTTHOFF, L.; VANSCHOREN, J. (Ed.). **Automatic machine learning**. 1. ed. Basel, Switzerland: Springer International Publishing, 2019. (The Springer Series on Challenges in Machine Learning). Disponível em: <https://doi.org/10.1007/978-3-030-05318-5>.

ILTER, N.; GUVENIR, H. **Dermatology**. 1998. UCI Machine Learning Repository. Disponível em: <https://doi.org/10.24432/C5FK5P>.

IMANDOUST, S.; BOLANDRAFTAR, M. Application of k-nearest neighbor (knn) approach for predicting economic events theoretical background. **Int J Eng Res Appl**, v. 3, p. 605–610, 01 2013.

JALALI, V.; LEAKE, D.; FOROUZANDEHMEHR, N. Learning and applying case adaptation rules for classification: An ensemble approach. In: **Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17**. [s.n.], 2017. p. 4874–4878. Disponível em: <https://doi.org/10.24963/ijcai.2017/685>.

JANOSI, A. et al. **Heart Disease**. 1988. UCI Machine Learning Repository. Disponível em: <https://doi.org/10.24432/C52P4X>.

JIMÉNEZ-RUIZ, E. et al. Semtab 2019: Resources to benchmark tabular data to knowledge graph matching systems. In: HARTH, A. et al. (Ed.). **The Semantic Web**. Cham: Springer International Publishing, 2020. p. 514–530. ISBN 978-3-030-49461-2. Disponível em: <https://doi.org/10.1007/978-3-030-49461-2_30>.

JIOMEKONG, A.; FOKO, B. Towards an approach based on knowledge graph refinement for tabular data to knowledge graph matching. **Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab), CEUR-WS. org**, 2022.

JOSEPH, V. R. Optimal ratio for data splitting. **Statistical Analysis and Data Mining: The ASA Data Science Journal**, v. 15, n. 4, p. 531–538, 2022. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/sam.11583>.

KAYAALP, F.; BAŞARSLAN, M. Performance analysis of filter based feature selection methods on diagnosis of breast cancer and orthopedics. In: . [S.l.: s.n.], 2019.

KIM, S.-B. et al. Some effective techniques for naive bayes text classification. **IEEE Transactions on Knowledge and Data Engineering**, v. 18, n. 11, p. 1457–1466, 2006.

KLEINBERG, J. M. Authoritative sources in a hyperlinked environment. **J. ACM**, Association for Computing Machinery, New York, NY, USA, v. 46, n. 5, p. 604—-632, sep 1999. ISSN 0004-5411. Disponível em: <https://doi.org/10.1145/324133.324140>.

KNAPP, E. D.; LANGILL, J. T. Chapter 11 - exception, anomaly, and threat detection. In: KNAPP, E. D.; LANGILL, J. T. (Ed.). **Industrial Network Security (Second Edition)**. Second edition. Boston: Syngress, 2015. p. 323–350. ISBN 978-0-12-420114-9. Disponível em: <https://www.sciencedirect.com/science/article/pii/B9780124201149000113>.

KOHAVI, R.; JOHN, G. H. Wrappers for feature subset selection. **Artificial Intelligence**, Elsevier BV, v. 97, n. 1-2, p. 273–324, dez. 1997.

KOTSIANTIS, S. Supervised machine learning: A review of classification techniques. **Informatica (Slovenia)**, v. 31, p. 249–268, 01 2007.

KUMAR, S.; AGRAWAL, K.; MANDAN, N. Red wine quality prediction using machine learning techniques. In: **2020 International Conference on Computer Communication and Informatics (ICCCI)**. [S.l.: s.n.], 2020. p. 1–6.

KUNCHEVA, L. **Combining Pattern Classifiers: Methods and Algorithms**. [S.l.: s.n.], 2004. 376 p. ISBN 0471210781.

LANGVILLE, A.; MEYER, C. A survey of eigenvector methods of web information retrieval. **SIAM Review**, v. 47, 01 2004.

LEGENDRE, P.; LEGENDRE, L. **Numerical Ecology**. Elsevier Science, 1998. (Developments in Environmental Modelling). ISBN 9780080537870. Disponível em: <https://books.google.com/books?id=KBoHuoNRO5MC>.

LEHMBERG, O.; BIZER, C. Stitching web tables for improving matching quality. **Proc. VLDB Endow.**, VLDB Endowment, v. 10, n. 11, p. 1502—-1513, aug 2017. ISSN 2150-8097. Disponível em: <https://doi.org/10.14778/3137628.3137657>.

LEHMBERG, O. et al. The mannheim search join engine. **Journal of Web Semantics**, v. 35, p. 159–166, 2015. ISSN 1570-8268. Semantic Web Challenge 2014. Disponível em: <https://www.sciencedirect.com/science/article/pii/S157082681500030X>.

LIU, J.; HAN, J. **Data Clustering: Algorithms and Applications**: Chapter 8 - spectral clustering. 1st. ed. New York: Chapman and Hall/CRC, 2014. 177–200 p.

LIU, T. et al. An evaluation on feature selection for text clustering. In: . [S.l.: s.n.], 2003. p. 488–495.

LUXBURG, U. A tutorial on spectral clustering. **Statistics and Computing**, v. 17, p. 395–416, 01 2004.

MCCALLUM, A.; NIGAM, K. A comparison of event models for naive bayes text classification. In: **AAAI Conference on Artificial Intelligence**. [s.n.], 1998. Disponível em: <https://api.semanticscholar.org/CorpusID:7311285>.

METZ, J. et al. **Redes complexas: conceitos e aplicações**. ICMC-USP, 2007. Disponível em: <https://repositorio.usp.br/item/001579913>.

MITCHELL, T. **Machine Learning**. McGraw-Hill, 1997. (McGraw-Hill International Editions). ISBN 9780071154673. Disponível em: <https://books.google.com/books?id=EoYBngEACAAJ>.

MUTANOV VLADISLAV KARYUKIN, Z. M. G. Multi-class sentiment analysis of social media data with machine learning algorithms. **Computers, Materials & Continua**, v. 69, n. 1, p. 913–930, 2021. ISSN 1546-2226. Disponível em: <http://www.techscience.com/cmc/v69n1/42767>.

NARGESIAN, F. et al. Table union search on open data. **Proc. VLDB Endow.**, VLDB Endowment, v. 11, n. 7, p. 813—825, mar 2018. ISSN 2150-8097. Disponível em: <https://doi.org/10.14778/3192965.3192973>.

NAVEEN; SHARMA, R. K.; NAIR, A. R. Efficient breast cancer prediction using ensemble machine learning models. In: **2019 4th International Conference on Recent Trends on Electronics, Information, Communication Technology (RTEICT)**. [S.l.: s.n.], 2019. p. 100–104.

NEEDHAM, M.; HODLER, A. **Graph Algorithms: Practical Examples in Apache Spark and Neo4j**. O'Reilly Media, 2019. ISBN 9781492047636. Disponível em: <https://books.google.com/books?id=z4WZDwAAQBAJ>.

NEWMAN, M. **Networks**. Oxford University Press, 2018. ISBN 9780198805090. Disponível em: <https://doi.org/10.1093/oso/9780198805090.001.0001>.

NEWMAN, M. J. A measure of betweenness centrality based on random walks. **Social Networks**, v. 27, n. 1, p. 39–54, 2005. ISSN 0378-8733. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0378873304000681>.

OJHA, V.; NICOSIA, G. Multi-objective optimisation of multi-output neural trees. **2020 IEEE Congress on Evolutionary Computation (CEC)**, IEEE, p. 1–8, 2020. Disponível em: <https://api.semanticscholar.org/CorpusID:221568128>.

ONNELA, J.-P. et al. Intensity and coherence of motifs in weighted complex networks. **Phys. Rev. E**, American Physical Society, v. 71, p. 065103, Jun 2005. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevE.71.065103>.

PAGE, L. et al. **The PageRank Citation Ranking: Bringing Order to the Web**. [S.l.], 1998. Disponível em: <http://www.eecs.harvard.edu/~michaelm/CS222/pagerank.pdf>.

PARR, T.; HOWARD, J. **How to explain gradient boosting**. 2018. Disponível em: <https://explained.ai/gradient-boosting/>.

PATRICIO, M. et al. **Breast Cancer Coimbra**. 2018. UCI Machine Learning Repository. Disponível em: <https://doi.org/10.24432/C52P59>.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in python. **Journal of Machine Learning Research**, v. 12, n. 85, p. 2825–2830, 2011. Disponível em: <http://jmlr.org/papers/v12/pedregosa11a.html>.

PESERICO, E.; PRETTO, L. Hits can converge slowly, but not too slowly, in score and rank. **SIAM Journal on Discrete Mathematics**, v. 26, n. 3, p. 1189–1209, 2012. Disponível em: <https://doi.org/10.1137/090762002>.

PIMPLIKAR, R.; SARAWAGI, S. Answering table queries on the web using column keywords. **Proc. VLDB Endow.**, VLDB Endowment, v. 5, n. 10, p. 908—919, jun 2012. ISSN 2150-8097. Disponível em: <https://doi.org/10.14778/2336664.2336665>.

POENARU-OLARU, L. Master's thesis, **Credit Scoring Prediction using Graph Features**. Amsterdam, Netherlands: [s.n.], 2018. Disponível em: <https://repository.tudelft.nl/islandora/object/uuid%3A0704be9f-fdea-4660-8723-4f1776853adc>.

RAMASWAMI, M.; BHASKARAN, R. A study on feature selection techniques in educational data mining. **Journal of Computing**, v. 1, 12 2009. Disponível em: <https://doi.org/10.48550/arXiv.0912.3924>.

RASCHKA, S. Naive bayes and text classification i - introduction and theory. 10 2014.

RASCHKA, S. **Python Machine Learning: Unlock Deeper Insights Into Machine Learning with this Vital Guide to Cutting-edge Predictive Analytics**. Packt Publishing, 2015. (Community experience distilled). ISBN 9781783555130. Disponível em: <https://books.google.com/books?id=HuxuawEACAAJ>.

RESHI, A. A. et al. Diagnosis of vertebral column pathologies using concatenated resampling with machine learning algorithms. **PeerJ Comput. Sci.**, PeerJ, v. 7, n. e547, p. e547, jul. 2021. Disponível em: <https://doi.org/10.7717/peerj-cs.547>.

RUDD, J. Application of support vector machine modeling and graph theory metrics for disease classification. **Model Assisted Statistics and Applications**, v. 13, 07 2017.

SAEYS, Y.; INZA, I.; LARRAñAGA, P. A review of feature selection techniques in bioinformatics. **Bioinformatics**, v. 23, n. 19, p. 2507–2517, 08 2007. ISSN 1367-4803. Disponível em: <https://doi.org/10.1093/bioinformatics/btm344>.

SALOD, Z.; SINGH, Y. A five-year (2015 to 2019) analysis of studies focused on breast cancer prediction using machine learning: A systematic review and bibliometric analysis. **Journal of Public Health Research**, v. 9, n. 1, p. jphr.2020.1772, 2020. Disponível em: <https://doi.org/10.4081/jphr.2020.1772>.

SANZ, I.; DUARTE, O. Graph-based feature enrichment for online intrusion detection in virtual networks. In: **Anais Estendidos do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**. Porto Alegre, RS, Brasil: SBC, 2019. p. 129–136. ISSN 2177-9384. Disponível em: <https://sol.sbc.org.br/index.php/sbrc_estendido/article/view/7779>.

SARKAR, D.; BALI, R.; SHARMA, T. **Practical Machine Learning with Python: A Problem-Solver's Guide to Building Real-World Intelligent Systems**. Apress, 2018. ISBN 9781484240496. Disponível em: <https://books.google.com/books?id=v4xHzQEACAAJ>.

SARMA, D. A. et al. Finding related tables. In: **Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data**. New York, NY, USA: Association for Computing Machinery, 2012. (SIGMOD '12), p. 817—-828. ISBN 9781450312479. Disponível em: <https://doi.org/10.1145/2213836.2213962>.

SEDGEWICK, R.; WAYNE, K. **Algorithms**. 4th edition. ed. [S.l.: s.n.], 2011. ISBN 978-0-321-57351-3.

SEWELL, M. Ensemble learning. Amsterdam, Netherlands, 2008. Disponível em: <http://machine-learning.martinsewell.com/ensembles/ensemble-learning.pdf>.

SHARMA, D. K.; HOTA, H. S. Data mining techniques for prediction of different categories of dermatology diseases. **Journal of Management Information and Decision Sciences**, v. 16, p. 103, 2013. Disponível em: <https://api.semanticscholar.org/CorpusID:57512304>.

SHINDE, R. M. et al. An intelligent heart disease prediction system using k-means clustering and naïve bayes algorithm. In: . [s.n.], 2015. v. 6, p. 637–639. Disponível em: <https://api.semanticscholar.org/CorpusID:18292226>.

SRINIVASAN, S. et al. Chapter three - machine learning techniques for fractured media. In: MOSELEY, B.; KRISCHER, L. (Ed.). **Machine Learning in Geosciences**. Elsevier, 2020, (Advances in Geophysics, v. 61). p. 109–150. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0065268720300042>.

STEENWINCKEL, B. et al. Csv2kg: Transforming tabular data into semantic knowledge. In: . [s.n.], 2019. Disponível em: <ttps://www.cs.ox.ac.uk/isg/challenges/sem-tab/2019/papers/IDLab.pdf>.

SULC, Z.; ŘEZANKOVA, H. Evaluation of recent similarity measures for categorical data. In: . [s.n.], 2014. Disponível em: <https://api.semanticscholar.org/CorpusID:62578671>.

TAN, P.-N. et al. **Introduction to Data Mining (2nd Edition)**. 2nd. ed. [S.l.]: Pearson Education, 2019. ISBN 978-0-13-312890-1.

TYAGI, S.; JIMÉNEZ-RUIZ, E. Lexma: Tabular data to knowledge graph matching using lexical techniques. In: **SemTab@ISWC**. [s.n.], 2020. Disponível em: <https://api.semanticscholar.org/CorpusID:229326521>.

VENETIS, P. et al. Recovering semantics of tables on the web. **Proc. VLDB Endow.**, VLDB Endowment, v. 4, n. 9, p. 528—538, jun 2011. ISSN 2150-8097. Disponível em: <https://doi.org/10.14778/2002938.2002939>.

WANG, D. et al. Tcn: Table convolutional network for web table interpretation. In: **Proceedings of the Web Conference 2021**. New York, NY, USA: Association for Computing Machinery, 2021. (WWW '21), p. 4020—4032. ISBN 9781450383127. Disponível em: <https://doi.org/10.1145/3442381.3450090>.

WASSERMAN, S.; FAUST, K. **Social Network Analysis: Methods and Applications**. [S.l.]: Cambridge University Press, 1994. (Structural Analysis in the Social Sciences).

WU, X. et al. Top 10 algorithms in data mining. **Knowledge and Information Systems**, v. 14, 12 2007.

XU, Y.; GOODACRE, R. On splitting training and validation set: A comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. **J. Anal. Test.**, Springer Science and Business Media LLC, v. 2, n. 3, p. 249–262, out. 2018. Disponível em: <https://doi.org/10.1007/s41664-018-0068-2>.

YAKOUT, M. et al. Infogather: entity augmentation and attribute discovery by holistic matching with web tables. In: **Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data**. New York, NY, USA: Association for Computing Machinery, 2012. (SIGMOD '12), p. 97—-108. ISBN 9781450312479. Disponível em: <https://doi.org/10.1145/2213836.2213848>.

YU, T.; ZHU, H. Hyper-parameter optimization: A review of algorithms and applications. **ArXiv**, abs/2003.05689, 2020. Disponível em: <https://api.semanticscholar.org/CorpusID:212675087>.

ZAKI, N.; MOHAMED, E.; HABUZA, T. From tabulated data to knowledge graph: A novel way of improving the performance of the classification models in the healthcare data. **SSRN Electronic Journal**, 2021.

ZHANG, H. The optimality of naive bayes. In: **Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference**. [S.l.: s.n.], 2004. v. 2.

ZHANG, M.; CHAKRABARTI, K. Infogather+: semantic matching and annotation of numeric and time-varying attributes in web tables. In: **Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data**. New York, NY, USA: Association for Computing Machinery, 2013. (SIGMOD '13), p. 145—-156. ISBN 9781450320375. Disponível em: <https://doi.org/10.1145/2463676.2465276>.

ZHANG, S.; BALOG, K. Entitables: Smart assistance for entity-focused tables. In: **Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: Association for Computing Machinery, 2017. (SIGIR '17), p. 255—-264. ISBN 9781450350228. Disponível em: <https://doi.org/10.1145/3077136.3080796>.

ZHANG, S.; BALOG, K. Web table extraction, retrieval, and augmentation: A survey. **ACM Trans. Intell. Syst. Technol.**, Association for Computing Machinery, New York, NY, USA, v. 11, n. 2, jan 2020. ISSN 2157-6904. Disponível em: <https://doi.org/10.1145/3372117>.

ZHENG, A.; CASARI, A. **Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists**. O'Reilly, 2018. ISBN 9781491953242. Disponível em: <https://books.google.com/books?id=Ho0UvgAACAAJ>.

ZHENG, A.; SHELBY, N.; VOLCKHAUSEN, E. Evaluating machine learning models. **Machine Learning in the AWS Cloud**, 2019. Disponível em: <https://api.semanticscholar.org/CorpusID:51991287>.

ZHOU, Z.-H. **Ensemble Methods: Foundations and Algorithms**. 1st. ed. [s.n.], 2012. v. 14. ISBN 9781439830031. Disponível em: <https://doi.org/10.1201/b12207>.

ZHU, X.; GHAHRAMANI, Z.; LAFFERTY, J. Semi-supervised learning using gaussian fields and harmonic functions. In: . [S.l.: s.n.], 2003. v. 3, p. 912–919.

A
Detalhes Similaridade e Dissimilaridade

A.1
Similaridade em Atributos Categóricos

Atributos categóricos representam categorias discretas que não possuem uma ordem natural. Segundo Boriah, S. et al (BORIAH; CHANDOLA; KUMAR, 2008, p.243), várias medidas de similaridade foram propostas na literatura para calcular a similaridade entre duas instâncias de dados categóricos.

No trabalho conduzido por Boriah, S. et al (BORIAH; CHANDOLA; KUMAR, 2008, p.243) foram analisadas diversas métricas diferentes (14), voltadas para atributos categóricos e sua performance foi avaliada no contexto específico da tarefa de detecção de outliers. Os resultados demonstraram que, embora nenhuma medida domine outras para todos os tipos de problemas, algumas medidas foram capazes de ter um desempenho consistentemente elevado. Apesar desse estudo possuir um contexto específico diferente desta pesquisa e com base nos critérios que citamos anteriormente, selecionamos as métricas *Overlap* e *Eskin* que constam do estudo e, adicionalmente, a métrica *Jaccard* que não consta do estudo. A escolha da métrica de *Jaccard* para a avaliação da similaridade de dados categóricos é justificada pela ampla referência na literatura e aplicabilidade em diversas áreas, especialmente quando lidamos com atributos categóricos binários (HAN; KAMBER; PEI, 2012, p.71)

Overlap

A métrica *Overlap*, também chamada de simple matching coeficiente (SULC; ŘEZANKOVA, 2014, p.250), tornou-se a medida de similaridade mais utilizada no contexto de dados categóricos. A prevalência desta métrica pode ser atribuída, em parte, à sua simplicidade e acessibilidade, que a tornam amplamente adotada e aplicável em uma variedade de cenários de análise de dados categóricos (BORIAH; CHANDOLA; KUMAR, 2008, p.245).

A métrica de *Overlap* é uma medida direta que quantifica a similaridade entre duas instâncias de dados através da contagem dos atributos que apresentam correspondência. Neste domínio de aplicação, cada atributo é considerado dentro de um intervalo de similaridade que varia de 0 a 1, onde o valor 0 indica a ausência de correspondência entre os atributos e o valor 1 denota correspondência perfeita, significando que os valores dos atributos coincidem integralmente. Essa abordagem oferece uma interpretação intuitiva e estruturada para avaliar a similaridade entre dados (BORIAH; CHANDOLA; KUMAR, 2008, p.246).

Uma interpretação simplificada da medida *Overlap* foi dada por Tan, P.-N. et al. (TAN et al., 2019, p.100), nesse caso chamada de *simple matching coeficiente* (SMC):

$$SMC = \frac{\text{NÚMERO DE VALORES DE ATRIBUTOS CORRESPONDENTES}}{\text{NÚMERO DE ATRIBUTOS}}$$

Formalmente, de acordo com Sulc e Řezanková (SULC; ŘEZANKOVA, 2014, p.250), ao determinar a semelhança entre os objetos x_i e x_j, assume-se o valor 1 para a c-ésima variável caso os objetos correspondam e, caso contrário, o valor 0. Seja $S_c\left(x_{ic}, x_{jc}\right)$ a similaridade no atributo *c* no par de objetos *i* e *j*:

$$S_c\left(x_{ic}, x_{jc}\right) = \{1, \text{ se } x_{ic} = x_{jc} \text{ ou } 0, \text{ se } x_{ic} \neq x_{jc}\}$$

A medida de similaridade *Overlap* entre dois objetos x_i e x_j, é então calculada como:

$$S(x_i, x_j) = \frac{\sum_{c=1}^{m} S_c(x_{ic}, x_{jc})}{m},$$

onde *m* é a quantidade de atributos nas instâncias *i* e *j*.

Como já citado anteriormente, a métrica Overlap é amplamente adotada em virtude de sua simplicidade e facilidade de aplicação. Entretanto, Boriah, S. et al (BORIAH; CHANDOLA; KUMAR, 2008, p.243) destaca que uma desvantagem da medida *Overlap* é que ela não distingue entre os diferentes valores obtidos por um atributo, isto é, todas as correspondências, bem como as correspondências erradas, são tratadas como iguais. Outra limitação que essa medida apresenta é que ela desconsidera aspectos cruciais de um conjunto de dados, tais como a quantidade de categorias presentes ou as frequências das categorias em uma variável específica. Essas características, muitas vezes, desempenham um papel significativo na caracterização da similaridade entre objetos e podem contribuir de forma substancial para uma formulação mais precisa e abrangente da similaridade entre elementos do conjunto de dados (SULC; ŘEZANKOVA, 2014, p.250).

Eskin

Segundo Sulc e Řezanková (SULC; ŘEZANKOVA, 2014, p.250), a medida *Eskin* foi proposta por Eskin, E. et al. (ESKIN et al., 2002) e sua ideia básica é atribuir maiores pesos aos descasamentos por variáveis com maior número de categorias. Em outras palavras, ela busca capturar a importância relativa das variáveis e o impacto na similaridade entre objetos, ponderando mais fortemente os desacordos nas variáveis com maior diversidade de categorias. Originalmente, a medida *Eskin* foi fundamentada como uma métrica de distância na qual era atribuído $\frac{2}{n_c^2}$ para discordâncias, onde n_c é o número de categorias

da c-ésima variável. No entanto, quando adaptada para avaliar a similaridade, essa ponderação se transforma em $\frac{n_c^2}{n_c^2+2}$. Essa adaptação atribui maior ênfase a discordâncias que surgem em atributos que apresentam uma ampla gama de valores possíveis. A medida *Eskin* define uma faixa de valores para similaridade que varia de $\left[\frac{2}{3}, \frac{n^2}{n^2+2}\right]$, onde *n* é o número de categorias, sendo o valor mínimo atingido quando o atributo assume apenas dois valores distintos e o valor máximo alcançado quando o atributo possui todos os valores como únicos e distintos (BORIAH; CHANDOLA; KUMAR, 2008, p.248).

Formalmente, a similaridade entre dois objetos para a c-ésima variável é então expressa como (SULC; ŘEZANKOVA, 2014, p.250):

$$S_c\left(x_{ic}, x_{jc}\right) = \{1, \text{ se } x_{ic} = x_{jc} \text{ ou } 0, \text{ se } x_{ic} \neq x_{jc}\},$$

onde n_c é o número de categorias da c-ésima variável.

No contexto de conjuntos de dados nos quais os atributos exibem uma alta cardinalidade, é observado que o desempenho do método de Eskin se mostra significativamente baixo (BORIAH; CHANDOLA; KUMAR, 2008, p.251).

Jaccard

Medidas de similaridade aplicadas a objetos que consistem exclusivamente de atributos categóricos binários são comumente referidas como "coeficientes de similaridade" (TAN et al., 2019, p.99).

O coeficiente de similaridade de *Jaccard* é uma medida de similaridade amplamente utilizada (HAN; KAMBER; PEI, 2012, p.71).

No contexto das medidas de similaridade binária, é possível distinguir entre dois tipos principais de coeficientes: os coeficientes simétricos e os coeficientes assimétricos. A distinção fundamental entre essas duas categorias reside na consideração dos chamados "zeros duplos". Os coeficientes simétricos incorporam esses zeros duplos em seus cálculos, enquanto os coeficientes assimétricos os excluem deliberadamente. Esse conceito foi destacado por Legendre, L. e Legendre, P. (LEGENDRE; LEGENDRE, 1998, p.254-257) como parte da análise e classificação das medidas de similaridade binária. No caso do coeficiente de Jaccard, trata-se de um coeficiente assimétrico.

Simplificadamente, o coeficiente *Jaccard*, frequentemente representado por J (TAN et al., 2019, p.99), de acordo com Han, J. et al (HAN; KAMBER; PEI, 2012, p.71), para um par de objetos *i* e *j* da matriz de dados, é dado por:

$$J(i,j) = \frac{\text{NÚMERO DE CORRESPONDENCIAS EM } 1}{\text{NÚMERO DE ATRIBUTOS NÃO ENVOLVIDOS EM } 0 \text{ EM I E } 0 \text{ EM J}} = \frac{q}{q+r+s}$$

onde *q*, *r*, *s* e *t* são descritos na Tabela $A.1$ de contingência para atributos binários

Tabela A.1: Tabela de Contingência para Atributos Binários

		Objeto j	
Objeto i		1	0
	1	q	r
	0	s	t

O coeficiente de *Jaccard* tem algumas propriedades, como variar de 0 a 1, onde 0 indica nenhuma sobreposição e 1 indica sobreposição completa. Quanto mais próximo o valor estiver de 1, maior a similaridade entre os conjuntos; e não ser sensível à ordem dos elementos nos conjuntos. Ele apenas considera a presença ou ausência de elementos, não sua sequência.

Para mais detalhes sobre métricas de similaridade em dados binários, De Carvalho (CARVALHO, 1994) apresenta uma compilação de métricas de similaridade para esse fim.

A.2
Similaridade em Atributos Numéricos

A adoção de métricas de similaridade direcionadas a atributos numéricos se fundamenta na necessidade de capturar a proximidade entre valores quantitativos em conjuntos de dados. Em contraste com atributos categóricos, onde a presença ou ausência de categorias define a similaridade, atributos numéricos demandam uma abordagem diferenciada. Essas métricas levam em consideração a magnitude das diferenças entre valores, possibilitando a quantificação de similaridade com base em escalas contínuas. Portanto, a utilização de métricas de similaridade específicas para atributos numéricos é uma prática fundamentada na natureza intrínseca desses dados e em suas aplicações analíticas.

Como citado anteriormente, medidas de dissimilaridade também podem ser denominadas distâncias quando satisfazem propriedades específicas (TAN et al., 2019, p.96). Como já citado anteriormente, selecionamos, para essa pesquisa, as medidas de distância *Euclidean, Manhattan, Mahalanobis* e *Cosine*.

Euclidean e Manhattan

Dentre as medidas numéricas selecionadas, a medida de distância *Euclidean* é a mais popular e, junto com a distância de *Manhattan* (também conhecida como city block) e a distância máxima (essa última não faz parte da nossa seleção), são casos particulares da distância de *Minkowski* que é dada por (GAN; MA; WU, 2007, p.71): Seja dois objetos de dados *x* e *y* em um espaço *d* de atributos, a distância $d_{min}\,(x, y)$ de *x* e *y* será

$$d_{min}(x, y) = (\sum_{j=1}^{d} |x_i - y_j|^r)^{\frac{1}{r}}, r \geq 1,$$

onde *r* é chamado de ordem (ou grau) da distância de Minkowski. Observe que se considerarmos $r = 2$, 1 e ∞, obteremos a distância Euclidiana, a distância de *Manhattan* e a distância máxima, respectivamente.

Considerando os pontos *x*=(1, 5, 4) e *y*=(2, 8, 3), a distância de *Minkowski* entre eles seria:

$$d_{min}(x, y) = \left[|1-2|^3 + |5-8|^3 + |4-3|^3\right]^{\frac{1}{3}} = (1+27+1)^{\frac{1}{3}} \cong 3,07$$

Portanto, para distância *Euclidean* entre o par *x* e *y*, $d_{euc}(x, y)$, teremos *r* igual a 2 e será dada por:

$$d_{euc}(x, y) = \left(\sum_{j=1}^{d}(x_j - y_j)^2\right)^{\frac{1}{2}} = \sqrt{\sum_{j=1}^{d}(x_j - y_j)^2}$$

Considerando os pontos *x*=(1, 5, 4) e *y*=(2, 8, 3), a distância de *Euclidean* entre eles seria:

$$d_{euc}(x, y) = \left[(1-2)^2 + (5-8)^2 + (4-3)^2\right]^{\frac{1}{2}} = \sqrt{1+9+1} \cong 3,32$$

Da mesma forma, para distância*Manhattan* entre o par *x* e *y*, $d_{man}(x, y)$, teremos *r* igual a 1 e será dada por:

$$d_{man}(x, y) = \sum_{j=1}^{d}[x_j - y_j]$$

Considerando os pontos $x = (1, 5, 4)$ e $y = (2, 8, 3)$, a distância de *Manhattan* entre eles seria:

$$d_{man}(x, y) = |1-2| + |5-8| + |4-3| = 1+3+1 = 5$$

De acordo com Tan, P.-N. (TAN et al., 2019, p.97), as distâncias *Euclidean* e *Manhattan* satisfazem as seguintes propriedades matemáticas: Se d(x,y) é a distância entre o par de objetos *x* e *y* então

1. Não negatividade: $d(x, y) \geq 0$: A distância é um número não negativo para todo *x* e *y*;
2. d(x,y) = 0: Para x=y, isto é, a distância de um objeto a si mesmo é 0;
3. Simetria: : d(x,y) = d(y,x): A distância é uma função simétrica, para todo x e y;
4. Desigualdade triangular: $d(x, z) \leq d(x, y) + d(y, z)$: para todos os objetos *x*, *y* e *z*.

As medidas que atendem a todas as propriedades acima listadas, são comumente denominadas métricas em contextos de análise de similaridade e distância. Em certos cenários, o termo "distância" é estritamente reservado para se referir a medidas de dissimilaridade que obedecem a essas propriedades, enquanto as medidas que não atendem a todas essas propriedades podem ser mais amplamente classificadas como medidas de similaridade (TAN et al., 2019, p.97).

Mahalanobis

A distância de *Mahalanobis* é uma métrica de distância multivariada que pode mitigar a distorção da distância causada pelas combinações lineares de atributos (GAN; MA; WU, 2007, p.72). Ela é útil nas situações em que os atributos estão correlacionados, têm intervalos de valores diferentes (variâncias diferentes) e a distribuição dos dados é aproximadamente gaussiana (normal) (TAN et al., 2019, p.116). Antes de definir a distância de *Mahalanobis*, precisamos definir a covariância, que é uma medida estatística, e a matriz de covariância, que é utilizada no cálculo da distância de *Mahalanobis*.

Para entender um pouco do conceito da medida estatística covariância, vamos a um exemplo: Suponha que um investidor deseja criar um portfólio diversificado de ativos financeiros, como ações, títulos, ou outros instrumentos, ele precisa considerar não apenas o retorno esperado de cada ativo, mas também como esses ativos se comportam em conjunto. Nesse contexto, a covariância poderia ser utilizada para avaliar os retornos entre os dois ativos x e y, representando, por exemplo, ações e títulos de renda fixa, respectivamente. Se os retornos de dois ativos *x* e *y* têm uma covariância positiva significativa, significa que eles tendem a se mover na mesma direção. Por outro lado, se têm uma covariância negativa entre *x* e *y*, significa que tendem a se mover em direções opostas. Uma covariância próxima de zero entre *x* e *y* indica que os ativos têm movimentos relativamente independentes.

Definindo formalmente a covariância, seja *D* um conjunto de dados com *n* objetos com *d* atributos ou variáveis $v_1, v_2, \ldots, v_d$. A covariância c_{rs} entre duas variáveis v_r e v_s é definido por (GAN; MA; WU, 2007, p.70)

$$c_{rs} = \frac{1}{n} \sum_{i=1}^{n} \left(x_{ir} - \overline{x_r}\right) \left(x_{is} - \overline{x_s}\right),$$

onde x_{ij} é o j-ésimo elemento do objeto x_i e x_j é a média de todos os objetos na j-ésima variável.

$$\overline{x_j} = \frac{1}{n} \sum_{i=1}^{n} x_{ij}, j = 1, \cdots, d$$

A matriz de covariância é uma matriz quadrada, no nosso caso *d* x *d*, que fornece informações detalhadas sobre as relações entre as variáveis em um conjunto

de dados multivariado e é usada para entender como as variáveis se relacionam entre si em termos de suas variações conjuntas (TAN et al., 2019, p.116).

Formalmente, a matriz de covariância é *d* x *d* e utiliza a notação $\sum$ onde a posição (r,s) contém a covariância entre a variável (atributo) v_r e v_s (GAN; MA; WU, 2007, p.70):

$$\sum = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1d} \\ c_{21} & c_{22} & \cdots & c_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ c_{d1} & c_{d2} & \cdots & c_{dd} \end{pmatrix}$$

Agora que definimos covariância e matriz de covariância, podemos definir a distância de *Mahalanobis*.

A distância de *Mahalanobis*, para dois objetos *x* e *y* no conjunto de dados, é definida por (GAN; MA; WU, 2007, p.72):

$$d_{man}(x, y) = \sqrt{(x - y)\sum^{-1}}(x - y)^T,$$

onde $\sum^{-1}$ é o inverso da matriz de covariância do conjunto de dados.

Cosine

A medida de similaridade *Cosine* (cosseno) é também conhecida como Semelhança de Orchini ou semelhança angular ou produto escalar normalizado (DEZA; DEZA, 2009, p.308), sendo uma das medidas mais comuns de semelhança de documentos (TAN et al., 2019, p.101).

A similaridade de cosseno é uma medida que quantifica a afinidade entre dois vetores que pertencem a um espaço de produto interno. O cálculo envolve a obtenção do cosseno do ângulo formado entre esses vetores, permitindo avaliar se eles compartilham uma orientação aproximadamente semelhante (HAN; KAMBER; PEI, 2012, p.77-79).

Para elucidar esse conceito, vamos lançar mão de um exemplo básico inspirado no exemplo contido em Tan, P.-N. et al (TAN et al., 2019, p.102):

Suponha que temos uma coleção de documentos e um usuário faz uma consulta de pesquisa. Queremos encontrar os documentos mais relevantes para a consulta do usuário. Cada documento e a consulta (query) são representados como vetores onde cada elemento representa a frequência das palavras-chave (termos) em relação ao documento ou à consulta. Simplificadamente, podemos representar documentos como vetores e a consulta também como um vetor, da seguinte forma: vetor de Documento 1: "machine learning": 2, "data mining": 3, "algorithms": 1, "classification": 2,; vetor do Documento 2: "machine learning": 1, "data mining": 0, "algorithms": 2, "classification": 1; vetor de Consulta do Usuário: "machine

learning": 1, "data mining": 1, "algorithms": 1, "classification": 0. Para classificar os documentos pela sua relevância em relação a consulta do usuário, vamos calcular a similaridade de *Cosine* (cosseno) entre a consulta do usuário e os documentos da coleção, conseguindo, assim, determinar quais documentos são mais relevantes para a consulta. Mas, antes, precisamos definir como calcular a similaridade de *Cosine* (cosseno). Se *x* e *y* são dois vetores de documento, então (TAN et al., 2019, p.101):

$$cos(x, y) = \frac{\langle x,y \rangle}{\|x\|\|y\|} = \frac{x'y}{\|x\|\|y\|},$$

onde ' indica um vetor ou matrix transposta e $\langle x, y \rangle$ indica o produto interno dos dois vetores,

$$\langle x, y \rangle = \sum_{k=1}^{n} x_k y_k = x'y,$$

e $\|x\|$ é, conceitualmente, o tamanho do vetor *x* ou é a norma Euclidiana do vetor $x = (x_1, x_2, ..., x_n)$, definida como

$$\langle x, y \rangle = \sum_{k=1}^{n} x_k^2,$$

similarmente $\|y\|$ é o tamanho do vetor *y* ou é a norma Euclidiana do vetor y. A medida calcula o cosseno do ângulo entre os vetores *x* e *y*. Um valor de cosseno de 0 significa que os dois vetores estão a 90 graus um do outro (ortogonal) e não tem correspondência. Quanto mais próximo o valor do cosseno de 1, menor será o ângulo e maior a correspondência entre os vetores (HAN; KAMBER; PEI, 2012, p.78).

O produto interno entre dois vetores é adequado para lidar com atributos assimétricos, uma vez que se baseia unicamente nos componentes não nulos presentes em ambos os vetores. Consequentemente, a medida de similaridade entre dois documentos depende exclusivamente das palavras que são compartilhadas por ambos. Isso estabelece uma abordagem robusta para avaliar a concordância entre documentos, focando nas características comuns e desconsiderando aquelas que não estão presentes em ambos os contextos (TAN et al., 2019, p.101).

Voltando ao nosso exemplo, podemos agora calcular a similaridade de *Cosine* (cosseno) entre a consulta do usuário e os documentos da coleção para determinar quais documentos são mais relevantes para a consulta.

Primeiro, calculamos a similaridade de *Cosine* (cosseno) entre a consulta do usuário e o Documento 1, onde o vetor *x* será o Documento 1 e *y* será a Consulta do usuário:

x=(2,3,1,2) e y=(1,1,1,0)

Similaridade de Cosseno (Documento 1, Consulta)

$$cos(x,y) = \frac{(2.1+(3.1)+(1.1)+(2.0)}{\sqrt{(2^2+3^2+1^2+2^2)}.\sqrt{(1^2+1^2+1^2+0^2)}} = \frac{6}{\sqrt{18}.\sqrt{3}} \cong 0,82$$

Segundo, calculamos a similaridade de *Cosine* (cosseno) entre a consulta do usuário e o Documento 2, onde o vetor *x* será o Documento 2 e *y* será a Consulta do usuário:

x=(1,0,2,1) e y=(1,1,1,0)

Similaridade de Cosseno (Documento 2, Consulta)

$$cos(x,y) = \frac{(1.1+(0.1)+(2.1)+(1.0)}{\sqrt{(1^2+0^2+2^2+1^2)}.\sqrt{(1^2+1^2+1^2+0^2)}} = \frac{3}{\sqrt{6}.\sqrt{3}} \cong 0,71$$

Portanto, o Documento 1, que teve similaridade de *Cosine* (cosseno) de aproximadamente 0,82, é mais relevante para consulta do usuário. Este é um exemplo de como a similaridade de *Cosine* (cosseno) pode ser utilizada em sistemas de recuperação de informações para encontrar documentos semelhantes com base na sobreposição de termos (palavras-chave) entre a consulta e os documentos da coleção.

Vários autores na literatura relacionam essa medida de similaridade com a busca ou comparação ou indexação de documentos como exemplo de aplicação. De fato, os documentos são frequentemente representados como vetores, onde cada componente (atributo) representa a frequência com que um determinado termo (palavra) ocorre no documento. Embora os documentos tenham milhares ou dezenas de milhares de atributos (termos), cada documento é esparso, pois possui relativamente poucos atributos diferentes de zero.

Para encerrar, algumas observações relacionadas a essa medida *Cosine* (cosseno):

- Aplicações que trabalham com vetores termo-frequência, como no nosso exemplo, tem esses vetores, normalmente, muito longos e esparsos (ou seja, eles têm muitos valores 0) e que a medida de *Cosine* (cosseno) trabalha bem nesses casos (HAN; KAMBER; PEI, 2012, p.77);
- A medida de similaridade *Cosine* (cosseno) possui a característica de não considerar o comprimento dos objetos de dados durante o cálculo de similaridade. Esta peculiaridade destaca-se, uma vez que a distância Euclidiana pode ser uma escolha mais apropriada quando a magnitude ou o comprimento dos objetos de dados são fatores críticos a serem considerados. A desconsideração do comprimento é uma característica distintiva da similaridade de cosseno

e, em muitos contextos, torna-se uma vantagem, especialmente quando o foco está na direção relativa dos vetores, independentemente de sua escala. No entanto, em situações em que o comprimento dos objetos de dados é de importância primordial, outras métricas, como a distância Euclidiana, podem ser mais apropriadas para a análise (TAN et al., 2019, p.102);

- A medida de similaridade de *Cosine* (cosseno) não atende a todas as propriedades que definem uma medida ser chamada de métrica. Portanto, ela é classificada como uma medida não métrica. A não conformidade com todas as propriedades métricas, como a desigualdade triangular, implica que a similaridade de *Cosine* (cosseno) não pode ser estritamente considerada como uma métrica no sentido matemático, embora ainda seja amplamente utilizada e eficaz na prática para avaliar a semelhança entre objetos em espaços vetoriais (HAN; KAMBER; PEI, 2012, p.78).

A.3 Similaridade em Atributos Heterogêneos

Todas as medidas de similaridade e dissimilaridades que abordamos anteriormente, tinham como premissa que os atributos (variáveis) dos objetos (instâncias, exemplos, amostras etc.) na matriz de dados tinham o mesmo tipo. Essa premissa foi necessária para viabilizar a definição teórica de cada medida sem misturar as especificidades que a mistura de tipos de dados provoca nessas medidas.

No entanto, como vamos ver mais adiante nesta pesquisa, alguns dos conjuntos de dados que vamos utilizar em nossas análises tem atributos categóricos (nominais, ordinais ou binários) e numéricos (discretos e flutuante). O desafio agora é como aplicar a avaliação da similaridade entre o par de objetos nesse espaço de características (atributos) heterogêneo.

Uma abordagem sugerida por Tan, P.-N. et al. (TAN et al., 2019, p.117) para lidar com atributos de diferentes tipos envolve o cálculo da similaridade para cada atributo individualmente. Em seguida, essas similaridades são combinadas para obter um valor entre 0 e 1. A partir daí existem algumas possibilidades de definir a similaridade global do par de objetos, como a média das similaridades de todos os atributos individuais. Mas o próprio autor conclui que essa abordagem pode revelar-se inadequada quando se lida com atributos assimétricos (que não levam duplo 0 em conta).

Uma outra abordagem, proposta por Han, J et al. (HAN; KAMBER; PEI, 2012, p.75), envolve a categorização de cada tipo de atributo, com análises de mineração de dados específicas para cada categoria. No entanto, Han, J et al. (HAN; KAMBER; PEI, 2012, p.75) observa que essa abordagem seria eficaz apenas se as análises conduzidas para cada tipo de atributo produzissem resultados

consistentes e que, em aplicações práticas, é improvável que uma abordagem que separe as análises por tipo de atributo resulte em resultados compatíveis.

Portanto, é necessário adotar uma abordagem mais robusta e sensível à natureza desses atributos na análise de similaridade.

Tan, P.-N. et al. (TAN et al., 2019, p.118) propõe uma medida similaridade geral entre o par de objetos *x* e *y*, que é a soma das medidas de similaridade calculadas para cada atributo, multiplicadas pelos valores indicadores correspondentes a variável δ_k. Essa abordagem permite que o algoritmo leve em consideração a natureza de cada atributo e quaisquer valores faltantes (missing values) ao calcular a similaridade geral entre os objetos. O resultado estará na faixa de 0 a m, onde 0 indica nenhuma similaridade e m indica similaridade máxima.

Formalmente a proposta de Tan, P.-N. et al. (TAN et al., 2019, p.118) para uma medida de similaridade geral entre o par de objetos *x* e *y*, $sim_{het}(x, y)$, é dada por:

$$sim_{het}(x, y) = \frac{\sum_{k=1}^{n} \delta_k s_k(x,y)}{\sum_{k=1}^{n} \delta_k}$$

onde *k* é o *k*-ésimo atributo dos objetos *x* e *y*, *n* o número de atributos no par de objetos, $s_k(x, y)$ é a similaridade do *k*-ésimo atributo do par de objetos *x* e *y* utilizando a medida ajustada a natureza desse atributo e δ_k uma variável que assume 0 quando o *k*-ésimo atributo é um atributo assimétrico e ambos os objetos têm um valor de 0, ou se um dos objetos tem um valor ausente para o *k*-ésimo atributo.

Em resumo, o algoritmo considera cada atributo separadamente, calcula uma medida de similaridade ajustada para a natureza do atributo e, em seguida, combina essas medidas ajustadas para obter a similaridade geral entre os objetos com atributos heterogêneos.

Uma outra abordagem para o problema foi proposta por Han, J. et al (HAN; KAMBER; PEI, 2012, p.75) em que o objetivo é calcular uma matriz de dissimilaridade que coloca todos os atributos em uma escala comum no intervalo [0, 1], permitindo a comparação de objetos com diferentes tipos de atributos. Basicamente, o algoritmo proposto por an, J. et al (HAN; KAMBER; PEI, 2012, p.75) considera todos os tipos de atributos juntos, normalizando-os para uma escala comum, o que permite calcular uma matriz de dissimilaridade que reflete as diferenças entre objetos com atributos heterogêneos. Segundo o autor, essa abordagem facilitaria a comparação e análise de objetos com diferentes tipos de atributos em um conjunto de dados.

Formalmente a proposta de Han, J. et al (HAN; KAMBER; PEI, 2012, p.75-76) para uma medida de dissimilaridade geral entre o par de objetos *i* e *j*, *d(i,j)*, é dada por:

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{f} d_{ij}^{f}}{\sum_{f=1}^{p} \delta_{ij}^{f}}$$

onde δ_{ij}^{f} assume o valor 0 se o valor do atributo *f* nos pares de objeto *i* e *j* não estiverem preenchidos (ambos ou somente um) ou se ambos forem zero e o atributo f for binário assimétrico (não aceita a combinação 0 e 0), caso contrário 1. d_{ij}^{f} representa a dissimilaridade entre o par de objetos *i* e *j* do atributo *f*, que assume valores conforme o tipo de dados: Se numérico $d_{ij}^{(f)} = \frac{|x_{if} x_{jf}|}{max_h x_{hf} - min_h x_{hf}}$, onde *h* vale para todos os objetos preenchidos do atributo *f*; Se nominal ou binário, então 0 se os valores dos atributos no par de objetos *i* e *j* forem iguais e 1 caso contrário; Se ordinal calcula o ranking normalizado (z_{if}) de r_{if} que é estado do atributo *f* no objeto *i* e M_f é a quantidade de estados no domínio do atributo *f*. É importante destacar que os atributos numéricos foram previamente normalizados para o intervalo [0,1].

Avançando nas diversas propostas disponíveis na literatura para lidar com esse problema, de calcular similaridade entre par de objetos com atributos (características) heterogêneos, Gan, G. et al (GAN; MA; WU, 2007, p.79) propôs o chamado coeficiente geral de similaridade que, ainda segundo Gan, G. et al (GAN; MA; WU, 2007, p.79) é amplamente utilizado e foi originalmente proposto por Gower (GOWER, 1971, p.858-861), sendo que este pode ser utilizado em objetos com valores faltantes (missing values).

O algoritmo é utilizado para calcular o coeficiente de similaridade geral entre um par de objetos multidimensionais,*x* e *y*, em um espaço d-dimensional. O coeficiente de similaridade geral, chamado $s_{gower}(x,y)$, é definido da seguinte forma (GAN; MA; WU, 2007, p.79):

$$s_{gower}(x,y) = \frac{\sum_{k=1}^{d} w_{(x_k,y_k)} s_{(x_k,y_k)}}{\sum_{k=1}^{d} w_{(x_k,y_k)}}$$

onde $s(x_k, y_k)$ é a similaridade entre os pares de objetos *x* e *y* para o *k*-ésimo atributo (característica) e $w(x_k, y_k)$ um ou zero dependendo se uma comparação é válida ou não para o k-ésimo atributo do par de objetos *x* e *y*. Eles são definidos, respectivamente, para diferentes tipos de atributos. Sejam x_k e y_k denotando o *k*-ésimo atributo do par de objetos *x* e *y*, respectivamente, então $s(x_k, y_k)$ e $w(x_k, y_k)$ serão definidos por: Para atributos quantitativos, x_k e y_k, $s(x_k, y_k)$ são $s(x_k, y_k) = 1 - \frac{|x_k - y_k|}{R_k}$, onde R_k é a amplitude do domínio do *k*-ésimo atributo; $w(x_k, y_k)$=0 se um dos objetos x e y tem valores faltantes no *k*-ésimo atributo ou 1 caso contrário. Para atributos binários, x_k e y_k, $s(x_k, y_k) = 1$ se ambos os objetos onde $s(x_k, y_k)$ é a similaridade entre os pares de objetos do *k*-ésimo atributo estão preenchidos, caso contrário 0; $w(x_k, y_k)$=0 se ambos os objetos *x* e *y* do *k*-ésimo atributo não estão preenchidos, caso contrário 1; Para atributos

nominais ou categóricos, x_k e y_k, $s(x_k, y_k) = 1$ se o par de objetos*x* e *y* do*k*-ésimo atributo são iguais e 0 caso contrário; $w(x_k, y_k) = 0$ se ambos os objetos *y* do *k*-ésimo atributo não estão preenchidos (missing values) e 1 caso contrário.

O resultado é uma medida de similaridade entre os objetos *x* e *y*, com valor mínimo de 0 quando os dois pontos de dados são idênticos e um valor máximo de 1 quando os dois pontos de dados são extremamente diferentes (GAN; MA; WU, 2007, p.80). O cálculo leva em consideração tanto a diferença nos valores dos atributos quanto a disponibilidade dos dados para cada atributo.

Com base nos levantamentos efetuados até agora, depreende-se que a seleção apropriada de uma medida de proximidade deve ser diretamente relacionada ao tipo de dados sob análise, assim como do nível de medição das variáveis.

Nesse estudo, vamos adotar uma versão simplificada para estabelecer a proximidade entre cada par de objetos, conforme abaixo:

1. Calculamos a similaridade entre o par de objetos do tipo categóricos utilizando uma medida alinhada a esse tipo categórico;

2. Calculamos a similaridade entre o par de objetos do tipo numéricos utilizando uma medida alinhada a esse tipo numérico;

3. Se a similaridade entre os pares de objeto for superior a um limite arbitrário, em qualquer um dos dois tipos, então consideramos que o par de objetos são próximos.

B
Detalhes Estatísticas Grafo

B.1
Medidas de Centralidade

As medidas de centralidade em um grafo avaliam o grau de importância de cada vértice dentro da estrutura do grafo e seu impacto nessa estrutura (NEWMAN, 2018). Muitas dessas medidas foram inventadas para análise de redes sociais, desde então elas têm sido utilizadas em uma variedade de indústrias e campos (NEEDHAM; HODLER, 2019). Diversas áreas já se utilizam dessas medidas tais como ciência da computação, física, estatística e biologia, constituindo-se uma ferramenta importante para análise de redes (NEWMAN, 2018).

A seguir descrevemos as medidas de centralidade selecionadas nesse trabalho e amplamente utilizadas na literatura.

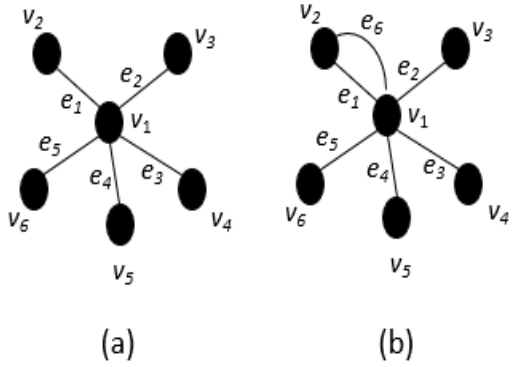
Degree (DG)

O grau (*degree*) de um vértice (*v*) em uma rede não direcionada é o número total de arestas (*edges*) conectadas a esse vértice. No caso de um grafo não direcionado, isso significa o número de conexões que um vértice possui (NEWMAN, 2018). Em outras palavras, o grau de um vértice *v* é o número de relacionamentos (arestas) diretos que esse vértice possui na rede (NEEDHAM; HODLER, 2019). Em uma rede de transporte aéreo entre aeroportos, por exemplo, o grau de um aeroporto poderia ser o número de pousos e decolagens deste aeroporto, isto é, que tem origem ou de destino esse aeroporto. Newman (NEWMAN, 2018, p.126) observa que a definição de grau é em termos do número de arestas, não de números de vértices vizinhos e que isso é especialmente importante em multigrafos. Portanto, se um vértice *v* tiver duas arestas paralelas ao mesmo vizinho, ambas contribuem para o grau deste vértice. A Figura B.1 procura demonstrar esse conceito: em (a) o vértice v1 tem grau cinco porque possui cinco arestas (e_1, e_2, e_3, e_4 e e_5) relacionadas a ele, ao passo que em (b) o vértice v_1 tem o mesmo número de vértices ligados a ele, mas tem grau seis, pois possui seis arestas (e_1, e_2, e_3, e_4, e_5 e e_6) relacionadas a ele:

Formalmente, Newman (NEWMAN, 2018) define o grau de um vértice *i* por k_i. Para uma rede de *n* vértices o grau pode ser representado em termos de sua matriz de adjacência como

$$k_i = \sum_{j=1}^{n} A_{ij}$$

Figura B.1: Grau de um vértice em uma rede para diferentes situações



Uma vez que cada aresta em uma rede não direcionada tem duas extremidades e se houver m arestas no total, há $2m$ de extremidades de arestas. Mas o número de extremidades das arestas é também igual à soma dos graus de todos os vértices (NEWMAN, 2018), resultando em:

$$2m = \sum_{i=1}^{n} k_i = \sum_{ij} A_{ij}$$

Para redes direcionadas onde os relacionamentos (arestas) têm uma origem e um destino, em vez de conexões mútuas, existem duas medidas de grau: grau de entrada (*in-degree*) e grau de saída (*out-degree*). In-degree é o número de conexões que apontam para dentro de um vértice. *Out-degree* é o número de conexões que se originam em um vértice e apontam para outros vértices. Um exemplo de rede direcionada seria uma rede de aeroportos (vértices) com as partidas e chegadas dos voos como relacionamentos (arestas) desses aeroportos, sendo as partidas como *out-degree* e chegadas como *in-degree*.

Para Newman (NEWMAN, 2018), o grau (*degree*) às vezes é chamado de centralidade de grau (*degree centrality*) na literatura de redes sociais, para enfatizar o uso como medida de centralidade.

Needham (NEEDHAM; HODLER, 2019, p. 81) fornece como exemplo da importância da medida do grau de um vértice, uma pessoa com um elevado grau de iteração social teria muitos contatos imediatos em sua rede e teria maior probabilidade de pegar um resfriado circulando em sua rede de contatos.

Para essa pesquisa, faz todo o sentido a utilização dessa métrica, pois se um vértice *v*, no nosso caso uma instância do conjunto de dados, estiver relacionado (aresta) a muitos outros vértices, então terá um grau maior e pode ser considerado mais central do que um vértice que possui menos relacionamentos (arestas).

Degree Centrality (DC)

Segundo Needham (NEEDHAM; HODLER, 2019, p.81), a centralidade de grau (*Degree Centrality*) foi proposta por Linton C. Freeman em artigo de 1979

com título "Centrality in Social Networks: Conceptual Clarification". Srinivasan (SRINIVASAN et al., 2020, p.17) define que a centralidade de grau de um vértice é um valor normalizado que representa o número de arestas que tocam um vértice. Em outras palavras, para uma rede de n vértices os valores de centralidade de grau são normalizados dividindo-se pelo grau máximo possível em um grafo simples *n*-1. Portanto, para um vértice *i*, a centralidade de grau (*degree centrality*) *D(i)* é dada por:

$$D(i) = \frac{1}{n-1} \sum_{j=1}^{n} A_{ij} \tag{B-1}$$

onde A_{ij} é o *ij*-ésimo elemento da matriz de adjacência *A* do grafo e *n* é o número de vértices do grafo.

Pela Equação B-1, verificamos que a centralidade de grau (*degree centrality*) de um vértice é uma medida normalizada que indica a importância relativa desse vértice em um grafo. Portanto, vértices com baixa centralidade de grau (*degree centrality*) geralmente estarão na periferia da rede ou nos ramos de baixo fluxo (SRINIVASAN et al., 2020, p.17). Valores de centralidade de grau variam entre 0 e 1, onde 0 indica que o vértice não é central em absoluto, e 1 indica que o vértice é central em relação a todos os outros.

Newman (NEWMAN, 2018) fornece um exemplo da importância da centralidade de grau (*degree centrality*): pensando em uma rede social, parece razoável supor que indivíduos que têm muitos amigos (relacionamentos-arestas) ou conhecidos (relacionamentos-arestas) podem ter mais influência, mais acesso à informação ou mais prestígio do que aqueles que têm menos.

Outro exemplo, fornecido por Needham, M. (NEEDHAM; HODLER, 2019), é o de identificar indivíduos poderosos através de seus relacionamentos, como conexões de pessoas em uma rede social. No relatório "Most Influential Men and Women on Twitter 2017" da BrandWatch, as 5 melhores pessoas em cada categoria têm mais de 40 milhões de seguidores cada. Na área científica, por exemplo, o número de citações (relacionamentos-arestas) que um artigo (vértice) recebe de outros artigos (vértices), que é o seu grau na rede (direcionada) de citações, fornece uma medida quantitativa de quão influente o artigo tem sido e é amplamente utilizado para avaliar o impacto da pesquisa científica (NEWMAN, 2018).

Para nossa pesquisa essa medida é importante, pois adiciona informações ao conjunto de dados levando em consideração toda a rede para cada instância do conjunto de dados.

Average Neighbor Degree (AN)

Segundo documentação do Networkx (HAGBERG; SCHULT; SWART, 2008) a medida *average neighbor degree* é o grau (*degree*) médio da vizinhança de cada vértice. Em outras palavras, é uma métrica de análise de redes utilizada para

calcular a média dos graus (*degree*) dos vizinhos de cada vértice em um grafo. Para um vértice específico, o *average neighbor degree* é calculado da seguinte maneira: (1) Para cada vértice no grafo, identifica-se seus vizinhos diretos, ou seja, os vértices conectados a ele por uma ou mais arestas. (2) Calcula-se o grau (*degree*) de cada vizinho, que é o número de arestas conectadas a esse vizinho. (3) Calcula-se a média dos graus (*degree*) dos vizinhos do vértice de interesse.

Formalmente, segundo a documentação do Networkx (HAGBERG; SCHULT; SWART, 2008), em um grafo não direcionado, a vizinhança *N(i)* do vértice *i* contém os vértices que estão conectados a *i* por uma aresta. O grau médio de vizinhança (*average neighbor degree*) de um vértice *i* é dada por:

$$k^w_{nn,i} = \frac{1}{|N(i)|} \sum_{j \in N(i)} k_j$$

onde *N(i)* são os vizinhos do vértice *i* e k_j é o grau do vértice (nó) *j* que pertence a *N(i)*.

Para grafo ponderados, uma medida análoga foi definida por Barrat A. et al. (BARRAT et al., 2004):

$$k^w_{nn,i} = \frac{1}{s_i} \sum_{j \in N(i)} w_{ij} k_j$$

onde s_i é o grau ponderado do vértice *i*, w_{ij} é o peso da aresta que liga *i* e *j* e *N(i)* são os vizinhos do vértice *i*.

É interessante observar que se caminha através dos vértices de uma rede e calcula a média dos graus dos vizinhos de cada um, muitos desses vizinhos aparecerão em mais de uma média. Isso significa que vértices de alto grau tem uma representação superestimada nos cálculos em comparação com os de baixo grau e esse viés eleva o valor médio geral (NEWMAN, 2018, p.380).

Para nossa pesquisa, essa estatística do grafo é importante pois ajuda a entender a conectividade local de um vértice, no nosso caso uma instância do conjunto de dados, analisando o grau médio de seus vizinhos imediatos. É útil para identificar vértices que estão em regiões mais densas ou menos densas do grafo.

Eigenvector Centrality (EC)

A EC é uma medida de importância de um vértice em um Grafo. Nesse caso, a importância do vértice é determina em função do grau de seus vizinhos e, também, pela importância desses vizinhos (ALHARBI; ALSUBHI, 2021). Essa definição se baseia na premissa de que nem todos os vizinhos de um vértice são iguais e que podem ser diferenciados por sua importância, influenciando na própria importância do vértice ligado a eles. Portanto, nesse caso, a importância de um vértice (nó) em uma rede aumenta quando há conexões com outros nós que são importantes. Um exemplo fornecido por Newman (NEWMAN, 2018, p.159) esclarece esse ponto:

você pode ter apenas um amigo no mundo, mas se esse amigo for o presidente dos Estados Unidos, então você mesmo pode ser uma pessoa importante. Logo, conclui Newman (NEWMAN, 2018, p.159), a EC não se trata apenas de quantas pessoas você conhece, mas, também, quem você conhece.

A EC pode ser vista como uma extensão da DC que leva esse fator da importância dos seus vizinhos em consideração (ALHARBI; ALSUBHI, 2021). Em vez de atribuir um valor unitário a cada vizinho de um vértice na rede, a medida EC atribui pontuações proporcionais às medidas de centralidade de seus vizinhos (NEWMAN, 2018, p.159).

Formalmente, uma definição para EC foi fornecida por Newman (NEWMAN, 2018, p.160): para uma rede não direcionada com *n* vértices, a EC denominada x_i de um vértice *i* como sendo proporcional à soma das centralidades dos vértices *i*'s vizinhos

$$x_i = k^{-1} \sum_{\textit{vértices de j que são vizinhos de i}} x_j \tag{B-2}$$

onde k^{-1} é uma constante de proporcionalidade arbitrária

Alternativamente, a Equação B-2 pode ser definida em função da matriz de adjacência *A* ($A_{ij} = 1$ se existe uma aresta entre *i* e *j* ou 0 caso contrário):

$$x_i = k^{-1} \sum_{j=1}^{n} A_{ij}.x_j$$

onde *k* é o maior autovalor associado ao autovetor da matriz de adjacência *A*.

Para nossa pesquisa, essa estatística do grafo é importante pois ela captura as relações entre os vértices em sua rede e adicionar essa medida como uma característica nova pode ajudar o modelo a considerar a estrutura da rede ao fazer previsões. Além do mais os vértices, no nosso caso as instancias do conjunto de dados, com valores de EC mais elevados também estão ligados a vértices com valores de EC mais elevados, o que pode nos levar ao agrupamento destes vértices mais próximos ajudando o modelo de predição.

Closeness Centrality (CC)

Closeness Centrality é uma métrica de centralidade que avalia a importância de um vértice com base em sua distância média para todos os outros vértices no grafo, entendendo como distância o caminho mais curto (WASSERMAN; FAUST, 1994, p.183). Em outras palavras, é uma medida da distância média mais curta de cada vértice a outro vértice. Especificamente, é o inverso da distância média mais curta entre o vértice e todos os outros vértices da rede.

A definição formal de CC vária levemente entre os autores, e nessa pesquisa vamos adotar a definição de Freeman (FREEMAN, 1977, p.226) que é aquela

utilizada pela ferramenta (Networkx) que suporta essa pesquisa no trabalho com grafos: dado os vértices *u* e *v*, a CC de um vértice *u*, *C(u)*, é o inverso da distância média do caminho mais curto para *u* em todos os *n*-1 vértices alcançáveis:

$$C(u) = \frac{n-1}{\sum_{v=1}^{n-1} d(v,u)}$$

onde *d(v, u)* é a distância do caminho mais curto entre *v* e *u*, e *n*-1 é o número de vértices acessíveis a partir de *u*.

Needham, M. (NEEDHAM; HODLER, 2019) fornece como exemplo de utilização da CC: a avaliação da importância das palavras em um documento, com base em um processo de extração de frases-chave baseado em grafo. Este processo é descrito por F. Boudin em "A Comparison of Centrality Measures for Graph-Based Keyphrase Extraction". Outro exemplo interessante de utilização da CC é como heurística para estimar o tempo de chegada em telecomunicações e entrega de pacotes, onde o conteúdo flui pelos caminhos mais curtos até um destino predefinido (NEEDHAM; HODLER, 2019).

Para nossa pesquisa, essa estatística do grafo é útil pois ela captura a importância dos vértices, no nosso caso as instancias do conjunto de dados, em relação a toda a rede e adicionar essa medida como uma característica nova pode ajudar o modelo a considerar a estrutura da rede ao fazer predições.

Betweenness Centrality (BC)

Segundo Newman, M. (NEWMAN, 2018, p.173), a ideia por trás *Betweenness Centrality* é geralmente atribuída a Freeman (FREEMAN, 1977) que, ainda segundo Newman, M. (NEWMAN, 2018), por sua vez atribui à Anthonisse (ANTHONISSE, 1971).

Alharbi A. (ALHARBI; ALSUBHI, 2021) definiu a BC de um vértice sendo o número de caminhos mais curtos através do vértice. De uma maneira menos formal, pode-se dizer que BC é o número de vezes que um vértice atua como uma ponte ao longo do caminho mais curto da rede entre dois outros vértices (ALHARBI; ALSUBHI, 2021). Newman (NEWMAN, 2018) definiu a Betweenness Centrality como a métrica que mede até que ponto um vértice se encontra em caminhos entre outros vértices. Newman (NEWMAN, 2018, p.173) fornece um exemplo baseado na Internet utilizando as redes sociais em que temos mensagens, notícias, informações ou rumores sendo passados e repassados de uma pessoa (vértice) para outra na rede. Conclui que, após um período longo para que essas mensagens fluam pelas diferentes pessoas (vértices) dessa rede e que essas mensagens sempre seguem o caminho mais curto entre as pessoas (vértices), o número de caminhos mais curtos que essas mensagens passam por uma pessoa (vértice) é o que chamamos de BC do vértice.

Com base no exemplo descrito acima e para entender melhor esse conceito, buscamos um exemplo no nosso dia a dia em que temos uma rede complexa de aeroportos com aeronaves partindo e chegando, de aeroporto em aeroporto, em pousos e decolagens. Nesse exemplo, temos uma rede com algo fluindo (aeronaves) por essa rede (de aeroportos), de vértice (aeroporto) em vértice (aeroporto), através de suas arestas (origens e destinos). Supondo que cada par de aeroportos (vértices) nessa rede tem pousos e decolagens de aeronaves à mesma taxa média (para pares que estão conectados) e que as operadoras dessas aeronaves sempre planejam as rotas dessas aeronaves para seguir o caminho mais curto disponível nessa rede, ou se fizer isso de forma aleatória no caso de haver vários desses caminhos.

Nesse caso, se esperarmos por um período longo até que muitas aeronaves tenham passado entre cada par de aeroportos (vértices) e medirmos quantas aeronaves terão passado por cada par de aeroporto a caminho do destino, vamos verificar que, como as aeronaves viajam ao longo das rotas mais curtas planejadas pelas suas operadoras, o número de vezes que esta passa por cada aeroporto (vértice) é proporcional ao número de rotas mais curtas em que o aeroporto está, e esse número de rotas ou caminhos mais curtos é o que se denomina de BC do vértice.

Formalmente adotamos a definição de BC fornecida por Brandes, U. (BRANDES, 2008, p.3) e Brandes, U. (BRANDES, 2001) que é aquela utilizada pela ferramenta (Networkx) que suporta essa pesquisa no trabalho com grafos: A BC de um vértice *v* é a soma da fração dos caminhos mais curtos de todos os pares que passam por *v*

$$C_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)}$$

onde *V* é o conjunto de vértices, $\sigma(s,t)$ é o número de caminhos mais curtos *(s, t)* e $\sigma(s,t|v)$ é o número desses caminhos que passam por algum vértice *v* diferente de *s*, *t*. Se $s = t$, $\sigma(s,t)$=1, e se $v \in s,t$,$\sigma(s,t|v)$=0. Esse algoritmo requer espaço $O(n+m)$, executam em $O(n.m)$ e $O(n.m+n^2 log\ n)$ para o tempo em redes não ponderadas e ponderadas, respectivamente, onde *n* é o número de vértices (nós) e *m* o numero de arestas no grafo. (BRANDES, 2001, p.9).

Apesar do esforço de processamento que o algoritmo consome para o cálculo do BC, ele poderá ser útil para nossa pesquisa, pois é uma medida que capta variações associadas a um vértice, no nosso caso uma instância do conjunto de dados, conseguindo captar variações também dos relacionamentos entre esse vértice e diversos outros vértices da rede e seus relacionamentos, onde aqueles com alto BC tem sua importância aumentada em virtude do seu controle sobre a passagem de informações.

Load Centrality (LC)

Segundo a documentação da ferramenta NetworkX (HAGBERG; SCHULT; SWART, 2008), que suporta essa pesquisa na utilização de grafos, a medida *Load Centrality* foi originalmente introduzida por Goh, K. (GOH; KAHNG; KIM, 2001) e implementada no algoritmo proposto por Newman, M.E.J. (NEWMAN, 2005), sendo ligeiramente diferente da BC.

Conforme descrevemos na seção anterior, a *Betweenness Centrality* é uma medida de centralidade que avalia a importância de um vértice em uma rede, geralmente calculada como a fração de caminhos mais curtos entre pares de vértices que passam pelo vértice de interesse. Embora essa métrica tradicional considere apenas os caminhos mais curtos, Newman, M.E.J. (NEWMAN, 2005) propôs uma medida de *betweenness* que leva em conta contribuições de praticamente todos os caminhos entre vértices, não apenas os mais curtos. Essa medida se baseia em caminhadas aleatórias e quantifica com que frequência um vértice é atravessado por uma caminhada aleatória entre dois outros vértices.

Para descrever, resumidamente, o que seria uma caminhada aleatória na rede, vamos considerar uma "mensagem", que pode ser informação de qualquer tipo, originando-se em um vértice de origem *s* em uma rede. Desejamos enviar essa mensagem a algum destino *t* nessa rede, mas a mensagem, ou aqueles que a transmitem, não têm ideia de onde *t* está nessa rede. Portanto, a mensagem simplesmente é passada aleatoriamente até que encontre o destino *t*. Assim, em cada etapa de suas viagens pela rede, a mensagem se move de sua posição atual na rede para um dos vértices adjacentes, escolhido de forma uniforme e aleatória dentre as possibilidades (NEWMAN, 2005).

Ainda segundo Newman, M.E.J. (NEWMAN, 2005, p.4), esse procedimento pode ser calculado, para todos os vértices de uma rede no pior caso de tempo $O((m+n).n^2)$ usando métodos matriciais, tornando comparável em suas demandas computacionais com a BC, onde *n* é o número de vértices (nós) e *m* o numero de arestas no grafo..

Concluindo, enquanto a *Betweenness Centrality* convencional procura analisar os caminhos mais curtos, a *Load Centrality* de Newman, M.E.J. (NEWMAN, 2005) abrange praticamente todos os caminhos da rede, empregando um método de passeio aleatório que incorpora pesos baseados no comprimento dos caminhos.

Apesar de poder ser considerado uma variação da BC, o Load Centrality será útil para nossa pesquisa, pois é uma estatística que pode captar variações associadas a um vértice, no nosso caso uma instância do conjunto de dados, que a BC pode não captar, pois procura abranger praticamente todos os caminhos da rede sem uma diferenciação expressiva de tempo computacional.

B.2
Medidas de Estrutura Local

Medidas de estrutura local em uma rede fornecem informações sobre padrões de conexão em torno de nós individuais e são úteis para avaliar a conectividade e a coesão local na rede, o que pode indicar a presença de grupos de vértices (nós) fortemente interconectados. Portanto, essas medidas são voltadas para entender a estrutura da rede em nível local.

Local Clustering (CL)

O *Local Clustering Coeficient* mede a conectividade da vizinhança de um vértice, isto é, trata-se de uma métrica para determinar o quão próximo os vizinhos de um vértice estão entre si (ALHARBI; ALSUBHI, 2021).

Newman, M. E. J. (NEWMAN, 2005, p.186) resume que o *Local Clustering Coeficient* representa a probabilidade média de que um par de amigos de *i* sejam amigos um do outro.

Para chegar na definição do coeficiente *Local Clustering Coeficient*, Newman, M. E. J. (NEWMAN, 2005, p.185) lança mão do conceito de quantificação da transitividade em uma rede. Segundo Newman, M. E. J. (NEWMAN, 2005, p.185), a transitividade em uma rede é parcial e que a transitividade perfeita só ocorre em redes onde todos os vértices de um componente estão conectados a todos os outros, mas, mesmo parcial, essa transitividade é útil. Em muitas redes, especialmente nas redes sociais, o fato de *u* conhecer *v* e *v* conhecer *w* não garante que você conheça *w*, mas torna isso muito mais provável.

Ainda segundo Newman, M. E. J. (NEWMAN, 2005, p.185), a partir dessa premissa, se *u* conhece *v* e *v* conhece *w*, então temos um caminho *u*-*v*-*w*, com duas arestas de comprimento na rede. Se *u* também conhece *w* então o caminho *u*-*v*-*w* é considerado fechado, formando um laço de comprimento três, ou um triângulo, na rede. No jargão das redes sociais, diz-se que *u*, *v* e *w* formam uma tríade fechada.

Finalmente, Newman, M. E. J. (NEWMAN, 2005, p.185) define o *Local Clustering Coeficient* como a fração de caminhos de comprimento dois na rede que estão fechados e fornece uma ideia de cálculo do *Local Clustering Coeficient* - C_i para o vértice *i*:

$$C_i = \frac{(\textit{número de pares de vizinhos de i que estão conectados})}{(\textit{número de pares de vizinhos de i})}$$

Portanto, para calcular C_i percorremos todos os pares distintos de vértices vizinhos de *i*, contamos o número desses pares que estão conectados entre si e dividimos pelo número total de pares, que é $frac12k_i(k_i - 1)$, onde k_i é o grau de *i* (NEWMAN, 2005, p.186).

Formalmente, adotamos as definições do *Local Clustering Coeficient* fornecida por Onnela J.P. (ONNELA et al., 2005, p.186), que é aquela utilizada pela

ferramenta (Networkx) que suporta essa pesquisa no trabalho com grafos, onde *Local Clustering Coeficient* - C_i para o vértice *i*:

$$C_i = \frac{2t_i}{k_i(k_i-1)}$$

onde k_i é o grau do vértice *i* e t_i denota o número de triângulos em torno de *i*. Por isso C_i= 0 se nenhum dos vizinhos de um vértice estiver conectado, e C_i= 1 se todos os vizinhos estiverem conectados. Na análise de rede, essa quantidade pode então ser calculada em média em toda a rede ou por grau de vértice.

Em termos gerais, o *Local Clustering Coeficient* C_i representa a probabilidade de conexão entre dois vizinhos de um nó específico. Quando C_i é igual a 0,5, isso indica que existe uma probabilidade de 50% de que dois vizinhos desse nó estejam conectados entre si. Em essência, esse valor descreve a tendência à formação de conexões entre os contatos imediatos de um nó na rede. Quanto maior o valor de C_i, maior a probabilidade de que os vizinhos de um nó estejam conectados, o que pode indicar um alto grau de coesão local na rede (BARABASI; MARTINO; POSFAI, 2012, p.41 sec.10).

Onnela et al. (ONNELA et al., 2005, p.186) também fornece uma definição específica para grafo ponderado:

$$\overline{C_i} = \frac{2}{k_i(k_i-1)} \sum_{j,k} (\overline{w}_{i,j}.\overline{w}_{j,k}.\overline{w}_{k,i})^{1/3}$$

onde $w_{u,v}$ é o peso da aresta que liga *u-v*. Os pesos das arestas $w_{u,v}$ são normalizados pelo peso máximo na rede $\overline{w}_{u,v} = \frac{w_{u,v}}{max(w)}$

Needham, M. (NEEDHAM; HODLER, 2019) fornece alguns exemplos de utilização do *Local Clustering Coeficient*: Identificar características para classificar um determinado site como conteúdo de spam; Explorar a estrutura temática da web e detectar comunidades de páginas com temas comuns com base nas ligações recíprocas entre elas.

Para nossa pesquisa, a detecção da conectividade entre os vértices, no nosso caso uma instância do conjunto de dados, pode ajudar na detecção de agrupamentos de instâncias que se comunicam e que possuem um CL mais alto.

Triangles (TR)

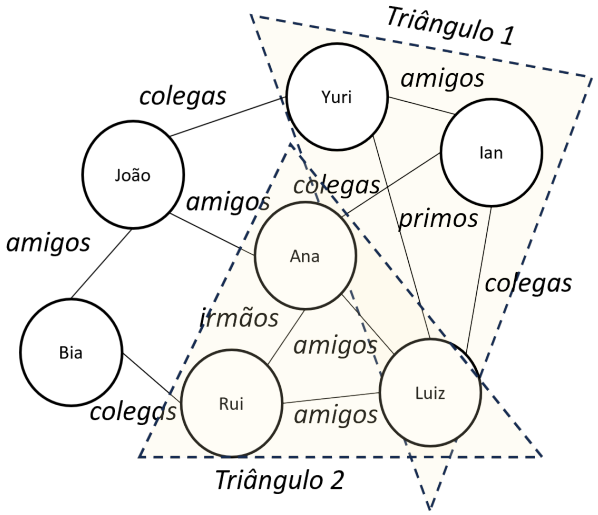
Quando definimos a métrica *Local Clustering Coeficient*, foi preciso, também, definir o que é um triângulo no grafo, que, resumidamente, é um agrupamento (ou subgrafos) de três vértices, onde cada vértice está conectado a todos os outros vértices do agrupamento (ou subgrafos), formando um triângulo. Nesse sentido, a métrica denominada *Triangles* quantifica o número de subgrafos completos de três vértices (triângulos) que contêm um vértice específico como um de seus participantes. Em outras palavras, mede a quantidade de triângulos nos quais

um dado vértice está presente como parte integrante, representando, assim, a participação desse vértice na formação de tais estruturas triangulares no grafo (NEEDHAM; HODLER, 2019).

Intuitivamente, quantificar o numero de subgrafos completos (triângulos) que um vértice participa é interessante, pois triângulos em uma rede podem representar a indicação de agrupamentos o que, em redes sociais, como Facebook por exemplo, podem representar comunidades de vértices com interações mais fortes no seu grupo, abrindo a possibilidade de identificar comunidades nessa rede.

A Figura B.2 procura demonstrar o conceito de triângulos em uma pequena rede de pessoas (vértices) e seus relacionamentos (arestas). Nesta temos dois triângulos, sendo o Triângulo 1 temos as pessoas (vértices) Yuri, Ian e Luiz com os relacionamentos (arestas) de amigos, primos e colegas de trabalho. No Triângulo 2 temos as pessoas Ana, Rui e Luiz com os relacionamentos (arestas) de irmãos e amigos. Nesse exemplo, Luiz teria o Triangles com o valor dois, Yuri, Ian, Ana e Rui com o valor um e as demais pessoas (vértices) com valor zero.

Figura B.2: Triangulos em uma rede de pessoas e relacionamentos



A métrica Triangles é apropriada para ser utilizada quando desejamos avaliar a coesão ou solidez de um conjunto de vértices em uma rede, ou quando necessitamos incorporá-la como componente na computação de outras medidas de rede (NEEDHAM; HODLER, 2019).

Para nossa pesquisa, a utilização de Triangles pode indicar a eficácia da disseminação de informação em grupos de vértices, no nosso caso instâncias de um conjunto de dados, representando comunidades ou grupos de vértices com interações mais fortes entre si, e ajudar os modelos com suas predições da variável dependente.

B.3
Análise de Links

Medidas de análise de links em uma rede visam compreender a organização das conexões e padrões de relacionamento (ou link ou aresta) entre pares de nós na rede. Essas medidas são voltadas para análise de redes complexas, pois ajudam a identificar nós importantes e influentes com base nas relações de link (aresta) dentro da rede.

PageRank (PR)

O algoritmo *PageRank* foi desenvolvido por Larry Page e Sergey Brin (PAGE et al., 1998), criadores do Google, e é um método, segundo os autores, para calcular uma classificação para cada página da web com base no grafo da web. *PageRank* pode ser definido como um algoritmo que atribui uma pontuação numérica a cada página da web com base no grau de vinculação (número de links) e na qualidade dos links que apontam para ela. Quanto mais links de alta qualidade apontam para uma página, maior será seu *PageRank* (PAGE et al., 1998).

O Google usa o *PageRank* como uma parte central de sua tecnologia de classificação na web para pesquisas na web, que estima a importância das páginas da web e, portanto, permite que o mecanismo de pesquisa liste as páginas mais importantes primeiro (BRIN; PAGE, 1998).

Essencialmente, o algoritmo *PageRank* avalia a relevância de um vértice em uma rede considerando tanto a quantidade quanto a qualidade dos links (arestas) que apontam para ele e os links (arestas) que ele possui para outros vértices na rede. Ele é calculado por meio de um processo iterativo de redistribuição de pontuações de classificação entre os vértices da rede até que a convergência seja alcançada (ALHARBI; ALSUBHI, 2021).

O algoritmo *PageRank* inovou ao considerar a importância relativa dos links e ao ajustar os valores de acordo com o número de links em uma página, permitindo uma avaliação mais precisa da relevância e qualidade das páginas da web em relação umas às outras (BRIN; PAGE, 1998).

Needham, M. (NEEDHAM; HODLER, 2019) fornece como exemplo da ideia por trás do *PageRank*: ter alguns amigos poderosos pode torná-lo mais influente do que ter muitos amigos menos poderosos.

Formalmente, o *PageRank* foi definido pelos autores (BRIN; PAGE, 1998, p.109) como: Assuma que temos a página *A* e que temos as páginas $T_1, T_2, ...T_n$ que apontam para ela. O parâmetro *d* é um fator de amortecimento que pode ser definido como entre 0 e 1. *C(A)* é definido como o número de links de saída da página A. O *PageRank* da página *A*, *PR(A)*, é dado por:

$$PR(A) = (1-d) + d\left(\frac{PR(T1)}{C(T1)} + \cdots + \frac{PR(Tn)}{C(Tn)}\right)$$

Needham, M. (NEEDHAM; HODLER, 2019) sugere que podemos pensar o fator de amortecimento d como a probabilidade de um usuário continuar clicando e $(1 - d)$ a probabilidade de um vértice ser alcançado diretamente sem seguir nenhum relacionamento.

Como o PR é uma distribuição de probabilidade sobre todos os vértices da rede, então a soma de todos os PR´s de cada vértice será 1 (BRIN; PAGE, 1998, p.110).

Uma outra maneira de entender o *PageRank* é como se fosse uma representação do comportamento de um usuário fictício que navega na web. Esse usuário explora links aleatórios em páginas web, sem retornar atrás, mas as vezes decide acessar outra página aleatória devido ao tédio. A probabilidade de visitar uma página específica corresponde ao seu *PageRank*. O fator de amortecimento d representaria a probabilidade de que, em cada página, o usuário fictício fique entediado e saia desta para uma nova página aleatória (BRIN; PAGE, 1998, p.110).

O *PageRank* agora é usado em muitos domínios fora da indexação da web, tais como, entre outros: Apresentar aos usuários recomendações de outras contas de usuário que eles possam desejar seguir; Prever o fluxo de tráfego e o movimento humano em espaços ou ruas públicas; Como parte de sistemas de detecção de anomalias e fraudes nos setores de saúde e seguros (NEEDHAM; HODLER, 2019).

Para nossa pesquisa, PR pode ser um indicador importante que se um vértice, no nosso caso uma instância do conjunto de dados, tem pouco relacionamento com as demais instâncias deve ter baixo PR. Considerando que instâncias do conjunto de dados que são similares devem ter respostas similares na variável dependente.

Nota: Fica claro, com as definições acima, que o algoritmo *PageRank* foi projetado para grafos direcionados e, nessa pesquisa, trabalhamos com grafo não direcionado. A documentação da ferramenta Networkx (HAGBERG; SCHULT; SWART, 2008) esclarece que, nesse caso, ela automaticamente converte as arestas direcionadas em duas arestas não direcionadas, de modo que o algoritmo *PageRank* possa ser aplicado de maneira apropriada nesse contexto. E com o fator de amortecimento d padrão com valor 0,85.

HITS (HT)

Segundo Langville, A e Meyer, C. (LANGVILLE; MEYER, 2004, p.136) HITS é a sigla para *Hypertext Induced Topic Search* e foi desenvolvida em 1997 por Kleinberg J.M. (KLEINBERG, 1999), pouco depois foi criado o método *PageRank*.

O algoritmo HITS estabelece duas métricas de centralidade, conhecidas como autoridade e *hub*. De maneira comparável ao *PageRank*, o HITS representa uma técnica de análise de links criada com o propósito de categorizar as páginas com base em sua autoridade e em sua posição como *hub* na rede (ALHARBI; ALSUBHI, 2021).

O conceito subjacente ao HITS envolve atribuir uma centralidade significativa a um vértice (autoridade) quando este é referenciado por outros vértices de elevada centralidade. Além disso, é também aplicável conceder uma alta centralidade a um vértice (*hub*) quando este se liga a outros vértices que possuem alta centralidade (NEWMAN, 2005, p.168).

Hubs e autoridades se complementam em uma relação de reforço mútuo: um bom *hub* aponta para várias boas autoridades, e uma boa autoridade é apontada por muitos bons *hubs* (KLEINBERG, 1999, p.8).

Formalmente, adotamos a definição de HITS fornecida por Langville e Meyer (LANGVILLE; MEYER, 2004, p.137-138), que é aquela utilizada pela ferramenta (Networkx) que suporta essa pesquisa no trabalho com grafos: Seja *E* o conjunto de todas as arestas no grafo e seja e_ij a representação de uma aresta direta do vértice *i* para o vértice *j*. Dado que, de alguma forma, os scores iniciais de autoridade $x_i^{(0)}$ e *hub* $y_i^{(0)}$ foram inicializados, HITS refina sucessivamente esses scores computando:

$$x_i^{(k)} = \sum_{j:e_{ji}\in E} y_j^{(k-1)} \quad \text{e} \quad y_i^{(k)} = \sum_{j:e_{ij}\in E} x_j^{(k)} \quad \text{para k=1,2,3,...}$$

Uma das vantagens do HITS é que ele é eficaz na identificação de vértices de alta qualidade que, nesse caso, são conhecidas como autoridade. Isso é útil em aplicações onde descobrir vértices relevantes e confiáveis em uma rede precisam ser priorizados. Outra vantagem é que ele ajuda para uma melhor compreensão e mapeamento de relações em uma grande rede, identificando os vértices que são *hubs* (que apontam para outros) quanto os vértices que são autoridade (que são apontados por outros) e isso pode ajudar, por exemplo, em sistemas de (LANGVILLE; MEYER, 2004, p.142-144)

Uma das desvantagens é que o HITS pode ser influenciado negativamente por elementos no vértice que apontam para outros vértices, mas sem que haja, de fato, a intenção de relacionar esse vértice ao outro. É o caso, por exemplo, de uma página web que carrega imagens de banner de anúncios, scripts, chamadas de API e CSS em outros locais da rede, aumentando a contagem de hits artificialmente. Outra desvantagem é que o cálculo do HITS pode ser computacionalmente intensivo em redes grandes, exigindo recursos significativos de processamento (LANGVILLE; MEYER, 2004, p.142-144). Para mais detalhes sobre a complexidade de tempo para HITS, consulte o trabalho de Peserico, E. e Pretto, L. (PESERICO; PRETTO, 2012).

Embora HITS tenha sido desenvolvida para grafos direcionados, a ferramenta Networkx (HAGBERG; SCHULT; SWART, 2008), que suporta essa pesquisa no trabalho com grafos, descreve em sua documentação que não verifica se o grafo de entrada é direcionado e será executado em grafo não direcionado. Como essa pesquisa se trata de um grafo não direcionado, vamos utilizar apenas a

métrica *hub*. Portanto, para nossa pesquisa, a utilização de HITS pode ajudar a identificar vértices, no nosso caso instâncias de um conjunto de dados, que se conectam a vértices muito influentes ajudando os modelos de predição a identificar comportamentos padrões com outros vértices e, assim, ajudar na melhora da predição da variável dependente.

C
Detalhes Seleção de Características

As técnicas de filtro avaliam a relevância das características observando apenas as propriedades intrínsecas dos dados (SAEYS; INZA; LARRAñAGA, 2007). É uma abordagem que avalia a relação entre cada característica e o rótulo do conjunto de dados usando métricas estatísticas, como correlação, qui-quadrado ou informações mútuas (KOTSIANTIS, 2007). Ainda segundo Kotsiantis (KOTSIANTIS, 2007), uma pontuação de relevância estatística da característica é calculada e as características de baixa pontuação são removidas com base em um limite definido, formando um novo subconjunto de dados. Uma das vantagens dos métodos de filtro, é que são independentes dos modelos que se utilizam de métodos que buscam o subconjunto ideal de recursos com base em métricas predefinidas (DAYA et al., 2020). Ainda segundo Daya (DAYA et al., 2020), as características selecionadas não têm dependência de um modelo de ML específico utilizado, o que auxilia na construção de uma visão mais precisa e generalizante.

Por outro lado, essa independência de um modelo de ML é vista como uma desvantagem por Saeys (SAEYS; INZA; LARRAñAGA, 2007) fazendo com que a busca no espaço de características do subconjunto fique separado da pesquisa no espaço de hipóteses do modelo, e acrescenta outra desvantagem de que a maioria das técnicas propostas são univariadas, isto é, cada característica é considerada separadamente, ignorando dependências entre as características. Saeys (SAEYS; INZA; LARRAñAGA, 2007) complementa que essa limitação pode levar a piores desempenhos de classificação quando comparado a outros tipos de técnicas de seleção de características. Para endereçar esse problema, uma série de técnicas de filtros multivariados foram introduzidas, visando a incorporação de dependências de características até certo ponto (SAEYS; INZA; LARRAñAGA, 2007).

As técnicas de *wrapper* selecionam o subconjunto ideal de características através da pesquisa no espaço de caraterísticas, avaliando todas as combinações possíveis de características em relação ao critério de avaliação (DAYA et al., 2020). Segundo Saeys (SAEYS; INZA; LARRAñAGA, 2007) a avaliação de um subconjunto específico de características é obtida treinando e testando um modelo de classificação específico, tornando esta abordagem voltada a um algoritmo de classificação específico. O modelo é treinado várias vezes com diferentes combinações de subconjuntos de características, e a eficácia é medida por meio da validação cruzada ou outra métrica de desempenho (KOHAVI; JOHN, 1997).

Saeys (SAEYS; INZA; LARRAñAGA, 2007) e Daya et al. (DAYA et al., 2020) concordam que uma desvantagem deste método é que ele se utiliza

de uma estratégia de busca exaustiva para alcançar o subconjunto ideal entre todas as combinações possíveis de subconjuntos de características no espaço de características, resultando em alta complexidade computacional. Saeys Saeys (SAEYS; INZA; LARRAñAGA, 2007) complementa que outra desvantagem é que ela apresenta um risco maior de overfitting do que técnicas de filtro. Daya et al. (DAYA et al., 2020) aponta como desvantagem que esse método encontra as características de um subconjunto ideais para um algoritmo de ML específico, o que torna a técnica dependente do modelo.

Nas técnicas de *embedded* o processo de seleção de características é incorporado ao algoritmo de aprendizagem, isto é, esse mecanismo é introduzido no estágio de aprendizagem do modelo para orientar o processo de seleção de características e encontrar o subconjunto de características ideal (DAYA et al., 2020). Portanto, os modelos de aprendizado de máquina incorporam a seleção de características durante o treinamento Saeys (GUYON; ELISSEEFF, 2003). Segundo Saeys (SAEYS; INZA; LARRAñAGA, 2007), as técnicas de *embedded*, da mesma forma que as técnicas *wrapper*, são específicas para um determinado algoritmo de aprendizagem. Saeys (SAEYS; INZA; LARRAñAGA, 2007) ainda observa que os métodos *embedded* têm a vantagem de incluir a interação com o modelo de classificação no mesmo momento, sendo menos intensivo em termos computacionais do que os métodos *wrapper*.

Ramaswami e R. Bhaskaran (RAMASWAMI; BHASKARAN, 2009) abordaram o uso de técnicas de seleção de características em mineração de dados educacionais. O estudo investigou várias técnicas de seleção de características e avaliaram o impacto na eficácia dos modelos de mineração de dados educacionais. Foram aplicadas técnicas de seleção de características em conjuntos de dados educacionais e realizaram experimentos para comparar o desempenho de modelos de aprendizado de máquina antes e depois da seleção de características. Eles avaliaram métricas como precisão, *recall* e *F1-score* para medir o impacto das técnicas de seleção de características. Os resultados do estudo demonstraram que a seleção de características desempenha um papel crucial na melhoria do desempenho dos modelos de mineração de dados educacionais. As técnicas de seleção de características permitiram reduzir a dimensionalidade dos dados, eliminar características irrelevantes e ruidosas, e melhorar a precisão e a eficácia dos modelos.

Estudo conduzido por Abusamra (ABUSAMRA, 2013) efetuou uma comparação entre métodos de seleção de características e classificação para dados de expressão gênica relacionados ao glioma, um tipo de tumor cerebral. O estudo teve como objetivo identificar quais métodos são mais eficazes na seleção das características mais relevantes e na classificação precisa desses dados. O autor analisou uma série de técnicas de seleção de características, incluindo métodos baseados

em filtro, *wrapper* e *embedded*, juntamente com diferentes algoritmos de classificação. O estudo também considerou métricas de desempenho, como acurácia, sensibilidade e especificidade, para avaliar a eficácia dos métodos. O autor concluiu que a seleção de características desempenha um papel fundamental na classificação precisa de dados de expressão gênica do glioma. Além disso, os resultados demonstraram que não existe um método único de seleção de características ou algoritmo de classificação que seja o mais adequado para todos os conjuntos de dados.

Liu, T. (LIU et al., 2003) abordou a avaliação de técnicas de seleção de características aplicadas a problemas de agrupamento de textos. O estudo tem como objetivo analisar o impacto da seleção de características na qualidade do agrupamento de documentos de texto. Os autores examinaram várias abordagens de seleção de características, incluindo medidas estatísticas, técnicas baseadas em informações e estratégias de redução de dimensionalidade. Eles aplicaram essas técnicas a conjuntos de dados de texto e utilizaram métricas de avaliação, como índice de Rand e índice Fowlkes-Mallows, para medir a qualidade dos agrupamentos resultantes. A conclusão foi que a seleção de características desempenha um papel fundamental na qualidade dos agrupamentos de texto. A escolha das técnicas de seleção de características adequadas pode melhorar significativamente o desempenho do agrupamento, tornando-o mais preciso e coerente.

Os estudos listados acima indicam que a escolha adequada das técnicas de seleção de características pode levar a modelos mais precisos e úteis na análise de dados educacionais, auxiliando na identificação de tendências e padrões relevantes para a melhoria da educação e do ensino.

D
Detalhes Algoritmos de Classificação

D.1
K-vizinhos mais próximos (KNN)

O KNN (K-Nearest *Neighbours*) é um algoritmo simples que se baseia no quanto cada observação está próxima uma da outra (K vizinhos) para medir a sua similaridade e concluir sua predição (BOEHMKE; GREENWELL, 2020). Por basear sua classificação somente nessa semelhança entre K vizinhos, é conhecido como algoritmo de aprendizado preguiçoso (BHAVANA; MEGHANA; PRADEEP, 2018). É também de fácil entendimento e que funciona bem para problemas de Classificação e Regressão (ESCOVEDO; KOSHIYAMA, 2020).

Os passos dados pelo algoritmo para classificar uma instância podem ser resumidos da seguinte forma: 1) Obtém um novo dado não classificado; 2) Verifica, em todas as instâncias do conjunto, os K mais próximos (menores distâncias) segundo alguma métrica; 3) Desses K mais próximos (vizinhos), quantifica as classes associadas a eles; 4) Classifica essa instância com a classe com maior frequência nesses K mais próximos (vizinhos) (BOEHMKE; GREENWELL, 2020).

As medidas de distância mais comumente utilizadas são: Euclidiana, *Manhattan* e *Minkowski* (ESCOVEDO; KOSHIYAMA, 2020). A precisão e performance do KNN estará ligada, principalmente, ao melhor balanceamento entre o tamanho dos K vizinhos mais próximos e da métrica para medir a distância (IMANDOUST; BOLANDRAFTAR, 2013). Isso nos leva a algumas desvantagens do método, como a excessiva dependência do valor de K e a necessidade de um conjunto de dados grande para alcançar uma boa acurácia, além de ser lento para o treinamento (BHAVANA; MEGHANA; PRADEEP, 2018). Algumas vantagens são que é muito simples de entender, intuitivo e que, para conjuntos de dados grandes, produz uma boa classificação (IMANDOUST; BOLANDRAFTAR, 2013).

O algorítimo KNN é altamente vulnerável ao sobreajuste (*overfitting*, em inglês), resultante do fenômeno conhecido como "maldição da dimensionalidade". Essa expressão descreve o fenômeno no qual o espaço de características torna-se cada vez mais escasso à medida que o número de dimensões de um conjunto de dados de treinamento de tamanho fixo aumenta. De maneira intuitiva, pode-se considerar que, em espaços de alta dimensão, mesmo os vizinhos mais próximos estão tão distantes que não oferecem uma estimativa precisa. Uma das formas de contornar esse problema seria usar a seleção de características ou técnicas de redução de dimensionalidade (RASCHKA, 2015, p.93).

D.2 *Naïve* Bayes (NBG)

Naive Bayes (Bayes Ingênuo) é baseado no teorema de Bayes e fornece um método rápido de criação de modelos preditivos (BHAVANA; MEGHANA; PRADEEP, 2018). Ele é dito ingênuo porque parte da premissa que as características contidas no conjunto de dados são independentes entre si (ESCOVEDO; KOSHIYAMA, 2020).

O algoritmo *Naive* Bayes é uma técnica de aprendizado de máquina que integra o Teorema de Bayes com a suposição de independência condicional das variáveis preditoras em relação à classe (KIM et al., 2006).

Nesse método, para cada instância do conjunto de dados trabalhado, analisa-se o relacionamento entre cada característica e a classe, derivando uma probabilidade condicional para os relacionamentos entre os valores das características e a classe [Bhavana N, M. S. C. e Pradeep K. R., A. 2018]. Essa probabilidade condicional para os relacionamentos pode ser decomposta utilizando o teorema de Bayes que fornece uma maneira de calcular a chamada probabilidade a posteriore P (H | X) de P (H), P (X) e P (X | H)), através da equação

$$P(H|X) = \frac{(P(X|H)P(H))}{P(X)},$$

onde P(H|X) é a probabilidade a posteriori de H condicionada em X; P(X|H) é a probabilidade a posteriori de X condicionado em H; P (H) é a probabilidade a priori de H e P (X) é a probabilidade a priori de X (SHINDE et al., 2015).

O algoritmo *Naive* Bayes é uma técnica de aprendizado de máquina que integra o Teorema de Bayes com a suposição de independência condicional das variáveis preditoras em relação à classe (KIM et al., 2006).

Por necessitar de um conjunto de dados pequeno para treinamento e ser rápido computacionalmente, ele é um dos métodos mais utilizados para Classificação [Escovedo, T. e Koshiyama, A., 2020]. No entanto, a falta do pressuposto de independência e problemas de classificação não linear podem levar a desempenhos muito fracos (RASCHKA, 2014, p.3).

Zhang, H. (ZHANG, 2004) pondera que o que eventualmente afeta a performance da classificação no Naive Bayes é a combinação de dependências entre todos os atributos. Quando as dependências entre todos os atributos trabalham juntas, elas podem se anular e não afetar mais a classificação. O estudo de Zhang, H. (ZHANG, 2004) conclui que é a distribuição de dependências entre todos os atributos sobre as classes que afeta a classificação do Naive Bayes, e não apenas as dependências.

Alguns dos conjuntos de dados utilizados como *benchmark* nesse trabalho utilizaram o Naive Bayes Bernoulli (NBB) (MCCALLUM; NIGAM, 1998) que é

uma variação do Naive Bayes aplicado principalmente a dados binários, onde cada característica é representada como uma variável booleana, indicando a presença ou ausência da característica. Por exemplo, em um conjunto de dados de texto, cada termo pode ser considerado como um recurso booleano, representando se está presente (1) ou ausente (0) no documento. Nesse estudo estamos utilizando as classes GaussianNB e BernoulliNB da biblioteca Sckit-learn na implementação desses algoritmos.

D.3 Máquina de Vetores de Suporte (SVM)

Uma Máquina de Vetores de Suporte (*Support Vector Machine*) é um modelo de aprendizado de máquina supervisionado para problemas de Classificação e Regressão e, também, para detecção de outliers (GERON, 2019, p.155) . O objetivo do SVM, em uma tarefa de aprendizagem de duas classes, é buscar encontrar, dentro do espaço de características, o melhor hiperplano que separe as duas classes (BOEHMKE; GREENWELL, 2020). Em outras palavras, dado um conjunto de instâncias de treinamento que pertencem a duas classes, um SVM busca um hiperplano que separe essas instâncias de modo a que o maior número de instâncias da mesma classe fique do mesmo lado, ao mesmo tempo que procura maximizar a distância de cada classe a esse hiperplano (CHAVES, 2006). O SVM busca o ajuste à via mais larga possível entre essas classes (GERON, 2019, p.155). As instâncias que estão mais próximas do separador de ambas as classes são chamadas de vetores de suporte e a sua determinação é importante para o estabelecimento da função separadora das classes, uma vez que o algoritmo utiliza os mesmos na classificação (ALBUQUERQUE, 2012). A distância do vetor de suporte até o hiperplano é chamada de margem de separação (CHAVES, 2006). Em um conjunto de dados linearmente separável, utiliza-se uma função de classificação linear para a separação das duas classes, que corresponde a um hiperplano de separação que passa pelo meio das duas classes, no entanto, existem muitos desses hiperplanos lineares, e o que o SVM garante, adicionalmente, é que maximizando a margem entre as duas classes encontramos a melhor dessas funções (WU et al., 2007).

Mas encontrar um hiperplano que separe perfeitamente as classes usando somente as características originais é, na prática, difícil (se não impossível) e a SVM busca contornar essa dificuldade ampliando a ideia de encontrar esse hiperplano de separação relativizando um pouco o que queremos dizer com uma "separação" perfeita, admitindo que algumas instâncias possam estar em lados incorretos, além de efetuar o procedimento de ampliar o espaço de características até o ponto que é mais provável a separação perfeita das classes, sendo esse procedimento chamado

de truque do Kernel (BOEHMKE; GREENWELL, 2020). Em outras palavras, o SVM utiliza funções kernel, em um conjunto de dados que não é linearmente separável, para mapear o espaço de características para uma dimensão maior que a original, ajustando o classificador a esse novo espaço (ESCOVEDO; KOSHIYAMA, 2020). As funções kernel mais populares são a linear, polinomial, radial e tangente hiperbólica (BOEHMKE; GREENWELL, 2020).

D.4 Regressão Logística (RLOG)

A Regressão Logística, de forma semelhante a Regressão Linear, é utilizada para aproximar a relação linear entre uma variável resposta e um conjunto de variáveis preditoras, sendo que a diferença é que, no caso da Regressão Linear, a variável resposta é contínua e, no caso da Regressão Logística (também chamada de Regressão Regressão Logit (GERON, 2019, p.144), a variável resposta é binária (ou seja, Sim / Não ou 1 / 0) (BOEHMKE; GREENWELL, 2020). Dessa forma, a Regressão Logística é muito utilizada para estimar a probabilidade de uma instância pertencer a uma determinada classe e, se essa for maior que 50%, então o modelo prevê que a instância pertence a chamada classe positiva (rotulada como "1"), ou, senão, pertence à classe negativa (rotulada como "0") (GERON, 2019, p.145). Para estimar a probabilidade de uma instância pertencer a determinada classe, a Regressão Logística calcula uma soma ponderada das características de entrada mais um termo de polarização, gerando como resultado a chamada logística (também chamada de *logit*) (EquaçãoD-1) (GERON, 2019, p.144).

$$\hat{p} = h_\theta(x) = \sigma(x^T\theta) \tag{D-1}$$

A função logística também é chamada função sigmóide que é uma curva em forma de S e que pode mapear qualquer número entre 0 e 1 (ESCOVEDO; KOSHIYAMA, 2020).

A equação abaixo demonstra a função logística:

$$\sigma(t) = \frac{1}{1 + exp^{(-t)}} \tag{D-2}$$

Estimada a probabilidade $\hat{p} = h_\theta(x)$ que uma determinada instância *x* pertença a classe positiva, então o modelo de Regressão Logística efetua a sua previsão *y* sendo 0 se $\hat{p}$ for menor que 0,5 e 1 caso contrário (GERON, 2019, p.145). Observando a Equação D-2, verificamos que quando $t < 0$ então $\sigma(t) < 0{,}5$ e que quando $t \geq 0$ então $\sigma(t) \geq 0{,}5$, com isso, concluímos então que o modelo prevê 1 se $\sigma(x^T\theta)$ for positivo, e 0 se for negativo (GERON, 2019, p.145).

D.5
Árvore de Decisão (DT)

Árvores de Decisão são algoritmos de Aprendizado de Máquina utilizados para executar tarefas de classificação, regressão e multioutput (cada rótulo pode ser multiclasse) (GERON, 2019, p.71). Elas são representadas por uma estrutura de árvore com diversos possíveis caminhos de decisão e, para cada caminho, um resultado (GRUS, 2016, p.286). De uma forma geral, os modelos baseados em árvore funcionam particionando o espaço de características, utilizando um conjunto de regras de divisão, em várias regiões menores, agrupando dados com valores de resposta semelhantes que não se sobrepõe (BOEHMKE; GREENWELL, 2020). A árvore é composta de um nó raiz no topo e suas divisões em diversos nós filhos, sendo o nó folha aquele que não tem nenhuma divisão abaixo dele e que conterá a classe prevista (GERON, 2019, p.173). Em outras palavras, os nós internos representam as decisões que determinam como os dados serão particionados para os seus nós filhos, sendo que, para classificação de uma nova instância, devemos percorrer os caminhos dessas decisões até atingir o nó folha que contém a classe predita (ESCOVEDO; KOSHIYAMA, 2020). Existem diferentes algoritmos para a elaboração de uma Árvore de Classificação, são eles: CHAID, CTree, C4.5, CART e Hoeffding Tree (ESCOVEDO; KOSHIYAMA, 2020).

Neste estudo, nosso problema é de classificação e vamos nos ater ao algoritmo CART, pois utilizamos o Scikit-learn (PEDREGOSA et al., 2011) para o treinamento de nossos modelos e, segundo a documentação do Scikit-learn (PEDREGOSA et al., 2011), ele utiliza uma versão otimizada do algoritmo CART.

No algoritmo CART os nós sem folhas sempre têm dois filhos, isto é, as decisões só contêm perguntas que geram respostas binárias do tipo sim/não (GERON, 2019, p.182). O algoritmo funciona, basicamente, com um particionamento recursivo binário em que cada divisão ou condição depende das divisões acima dela, sendo que cada nó interno busca, para particionar os dados restantes, a característica do conjunto de dados que melhor separa os dados nas duas opções (sim/não) de modo a minimizar o erro geral entre a resposta real e a prevista (BOEHMKE; GREENWELL, 2020). De modo geral, em problemas de classificação, esse particionamento dos dados é feito procurando maximizar a redução da entropia ou do coeficiente de Gini (BOEHMKE; GREENWELL, 2020). A entropia, que varia entre 0 e 1, pode ser entendida como a representação da impureza associada com os dados do nó, isto é, se a entropia for baixa, então não há impureza e ele é puro (valor 0). A entropia será baixa se todas as instâncias de um conjunto de dados pertencerem a uma única classe, e será alta se o conjunto de dados estão equilibrados nas classes (GERON, 2019, p.183). Da mesma forma que a entropia, o Coeficiente de Gini, também pode representar a impureza associada com

os dados, sendo que coeficiente será baixo (puro) se todas as instâncias de dados do nó pertencerem a uma única classe (valor 0), e será alto se os dados estão equilibrados nas classes (GERON, 2019, p.180).

Algumas das vantagens das Arvores de Decisão são: a simplicidade e transparência do funcionamento e como chegam a uma previsão, tornam essas árvores muito fáceis de entender e interpretar [Grus, J., 2016]; elas podem trabalhar com características de diferentes tipos, tais como numéricos, categóricas e até valores faltantes (missing values) (GRUS, 2016); ela pode utilizar características de diferentes tipos de dados numéricos sem a necessidade de transformações (BOEHMKE; GREENWELL, 2020). Algumas das desvantagens são: as árvores profundas tendem a ter alta variação (e baixo viés) e árvores rasas tendem a ter alto viés (mas baixa variação) (BOEHMKE; GREENWELL, 2020); É alta a chance de um sobreajuste aos dados de treinamento se a árvore for deixada sem restrição, isto é, a estrutura da árvore se adaptará muito de perto aos dados de treinamento (GERON, 2019, p.184).

D.6
Multi-Layer Perceptron (MLP)

O MLPClassifier é um classificador de redes neurais artificiais baseado em perceptrons multicamadas (MLP, do inglês "*Multilayer Perceptron*"). Ele é amplamente utilizado para problemas de classificação em que os dados têm estruturas complexas e não lineares (GERON, 2019, p.284).

Utiliza um modelo de rede neural artificial com múltiplas camadas para aprender a relação entre as características de entrada e a classe de destino. A rede neural é composta por neurônios artificiais organizados em camadas interconectadas. Cada neurônio aplica uma função não linear aos inputs e gera um output que é propagado para a próxima camada. O processo de aprendizado da rede neural envolve a otimização dos pesos das conexões entre os neurônios para minimizar o erro de classificação. O algoritmo de otimização utilizado é o *backpropagation*, que é um método eficiente para ajustar os pesos da rede neural e melhorar o desempenho. Isso permite que o modelo aprenda a representação dos dados e faça previsões precisas em novos exemplos (RASCHKA, 2015, p.345).

D.7
Ensemble

A ideia por trás do método *Ensemble* é que agregar as previsões de um conjunto de previsores, sejam eles classificadores ou regressores, muitas vezes, deverá gerar uma previsão melhor do que com o melhor desses previsores (GERON, 2019, p.191). A esse conjunto de previsores chamamos de *Ensemble*, sendo que

a aplicação da técnica de agregar as previsões desses *Ensemble* é chamada de *Ensemble* Learning e, finalmente, um algoritmo que implementa o *Ensemble Learning* é chamado de método *Ensemble* (ou *Ensemble method*) (GERON, 2019, p.191).

O *Ensemble Learning* combina vários chamados aprendizes fracos (*weak learners*) produzindo um modelo forte (GRUS, 2016, p.302). Isto posto, os métodos *Ensemble* buscam construir um conjunto de previsores e combiná-los para solucionar o mesmo problema, diferentemente dos previsores comuns que buscam construir um previsor a partir do treinamento dos dados (ZHOU, 2012). Para ilustrar essa ideia, podemos pensar em um treinamento de um conjunto de classificadores de Árvores de Decisão individuais (base), cada uma utilizando um subconjunto aleatório diferente do mesmo conjunto de treinamento e fazendo as suas previsões individuais, sendo que, para fazer a previsão do *Ensemble*, devemos, então, obter essas previsões individuais dos previsores base e prever a classe que obteve a maioria dos votos nesses previsores, sendo essa a classe do *Ensemble* (GERON, 2019, p.193).

A arquitetura dos *Ensemble* tem uma quantidade de previsores base que, de forma geral, são construídos individualmente e baseados em dados de treinamento por algoritmos de aprendizado que podem ser Árvores de Decisão, Redes Neurais ou outros tipos de algoritmos de aprendizado (ZHOU, 2012). O tipo e a forma de como esses previsores base são utilizados fornecem diferentes dimensões dos métodos *Ensemble*. Quando o método *Ensemble* utiliza previsores do mesmo tipo, isto é, todos os previsores do conjunto são, por exemplo, arvores de decisão, esses são chamados de *Ensemble* homogêneos, mas existem também alguns métodos que utilizam múltiplos algoritmos de aprendizados diferentes, isto é, previsores de diferentes tipos, nesse caso chamados de *Ensemble* heterogêneo (ZHOU, 2012). A forma como os previsores base são gerados também fornece um paradigma diferente nos métodos *Ensemble*, podendo ser sequenciais e paralelos, isto é, se os previsores base são gerados sequencialmente, então são chamados de métodos *Ensemble* sequenciais, por outro lado, se os previsores base são gerados em paralelo, são chamados de métodos *Ensemble* paralelos, sendo o *Bagging* como um representante deste último (ZHOU, 2012).

Os métodos *Ensemble* mais populares são: *Bagging*, *Boosting* e *Stacking* (GERON, 2019, p.191). Concluindo, os métodos de *Ensemble* podem ser resumidos como meta-algoritmos que combinam várias técnicas de aprendizados de máquinas, geradas individualmente, em um modelo preditivo único para diminuir a variância (*Bagging*) e o viés (*Boosting*) (BHAVANA; MEGHANA; PRADEEP, 2018).

A seguir, vamos nos ater aos métodos *Ensemble Bagging* e *Boosting* que foram utilizadas em alguns dos nossos modelos neste estudo.

D.7.1 Bagging

O método *Ensemble Bagging*, que também é conhecido como *bootstrap aggregating*, utiliza no treinamento dos modelos base, que utilizam todos o mesmo algoritmo (por exemplo árvore de decisão), várias versões diferentes do conjunto de dados de treinamento usando a técnica de bootstrap para seleção desses dados em cada modelo base, ou seja, amostragem com substituição, e as saídas dos modelos base, isto é, suas previsões, são combinadas por média (no caso de regressão) ou votação (no caso de classificação) para criar uma única saída, que será a previsão do *Bagging* (SEWELL, 2008). Em outras palavras, o *Bagging* permite, ao mesmo tempo, que as instâncias de treinamento sejam amostradas várias vezes em diferentes previsores (todos do mesmo algoritmo) e que essas mesmas instâncias sejam amostradas diversas vezes pelo mesmo previsor, sendo que, concluídas todas previsões individuais, o *Ensemble* efetua a previsão para uma nova instância agregando as previsões de todos esses previsores, utilizando a previsão mais frequente (no caso de um problema de classificação) ou a média (no caso de um problema de regressão) (GERON, 2019, p.196).

Uma vantagem de utilizar a técnica de *Bagging* é que ela tem um forte efeito na redução da variância e é particularmente eficaz com previsores base instáveis (ZHOU, 2012). Previsores base instáveis são aqueles em que pequenas mudanças no conjunto de dados de treinamento geram mudanças consideráveis no modelo (SEWELL, 2008). Finalmente, outra vantagem importante do *Bagging* é que os previsores base podem ser processados em paralelo, o que abre a possibilidade de que alcance uma boa performance nos ambientes tecnológicos que suportam esse tipo de processamento (GERON, 2019, p.196).

D.7.1.1 Florestas Aleatórias (RFOR)

A Floresta Aleatória é uma técnica que utiliza uma combinação (*Ensemble*) de Árvores de Decisão [(GERON, 2019, p.193). Vimos, anteriormente, que o treinamento das Árvores de Decisão é efetuado com todos os dados de treinamento, mas, no caso das Árvores de Decisão utilizadas na técnica de Floresta Aleatória, a seleção de dados para treinamento de cada Árvore de Decisão é efetuado com a técnica de *Bagging*(*bootstrap aggregating*) (GRUS, 2016, p.301). Esse componente adicional de aleatoriedade se dá no momento da busca da melhor característica ao dividir um nó de cada Árvore de Decisão da combinação, isto é, ao invés de buscar o par com a melhor característica e um limiar mais puro, como na Árvore de Decisão, a Floresta Aleatória busca em um subconjunto aleatório dessas características (GERON, 2019, p.199). Nesse caso, o algoritmo está utilizando

o que se chama de *Ensemble Learning*, combinando vários chamados aprendizes fracos (*weak learners*) produzindo um modelo forte (GRUS, 2016, p.302). Em função da técnica de busca aleatória de características, o espaço de características pesquisadas diminui e, com isso, a Floresta Aleatória tende a ser mais rápida que o *Bagging* (BOEHMKE; GREENWELL, 2020). Note que, dependendo da utilização dos hiperpârametros, o *Bagging* de Arvores de Decisão e a Floresta Aleatória podem ser equivalentes (GERON, 2019, p.199).

D.7.1.2 Árvores Extras (ETREE)

Assim como a Floresta Aleatória, a Árvores Extras é uma técnica que utiliza uma combinação (*Ensemble*) de Árvores de Decisão, e é também chamada de um *Ensemble* de Árvores Extremamente Aleatórias (GERON, 2019, p.200).

A Árvores Extras é uma técnica que constrói Arvores de Decisão totalmente aleatórias e que suas estruturas independem do valor da variável dependente da amostra de treinamento, através da escolha aleatória de uma única característica e de limiar em cada nó (GEURTS; ERNST; WEHENKEL, 2006, p.3). A Árvores Extras é uma técnica que deixa a técnica de Floresta Aleatória ainda mais aleatória ao adicionar também a seleção aleatória dos limiares para cada característica (GERON, 2019, p.200).

Portanto, essa técnica, além de aumentar a busca por uma variância menor em troca de maior viés, que a Floresta Aleatória também faz, tende a ser mais rápida que a Floresta Aleatória, pois a busca pelo melhor limiar possível para cada característica, que é a tarefa que toma mais tempo na construção das Árvores de Decisão, é efetuada de forma aleatória (GERON, 2019, p.200).

D.7.2 Boosting

Boosting é um método de *Ensemble* que pode ser usado para qualquer tipo de modelo de classificação ou regressão e que, geralmente, é aplicado a modelos de árvore de decisão (SEWELL, 2008).

Originalmente chamado de *hypothesis boosting*, ele tem como ideia geral treinar, sequencialmente, previsores fracos em que cada um tenta corrigir as previsões incorretas do predecessor (GERON, 2019, p.201). Esse é um dos métodos *Ensemble* mais utilizados e uma das ideias de aprendizagem de máquina mais poderosas (SEWELL, 2008).

O método se inicia com a criação de um classificador fraco (com precisão ligeiramente melhor do que uma adivinhação aleatória no treinamento), em seguida é construído novo classificador fraco, treinando este agora em um conjunto de

dados em que as instâncias classificadas incorretamente pelo classificador anterior recebem mais peso, sendo esses procedimentos repetidos iterativamente, ao final, os modelos recebem pesos conforme a precisão na previsão gerada (quanto mais preciso for o modelo mais alto será seu peso) e, então, gera a previsão do *Ensemble* por votação (para problema de classificação) ou pela média (para problemas de regressão) *Ensemble*.

Uma das desvantagens desta técnica de *Ensemble* com aprendizado sequencial é que ela não pode ser paralelizada (ou pode parcialmente), pois os previsores de uma etapa necessitam dos resultados dos previsores anteriores (GERON, 2019, p.203). A Tabela D.1 apresenta uma comparação básica dos métodos *Ensemble Bagging* e *Boosting*.

Tabela D.1: Comparação entre *Bagging* e *Boosting* (BHAVANA; MEGHANA; PRADEEP, 2018)

	Bagging	Boosting
Example	Random Forest	Gradient Boosting Method
Bias/Variance Trade Off	Decreases Variance	Decreases Bias
Training Data	Partition before training	Adaptive data weighting
Base Learner	Complex	Simple
Classifier Strenght	Tries to overfit with a flexible model and then average to decrease variance	Tries to underfit using weak learners and then improve model based on classifier performance

Os métodos de *Boosting* mais populares são AdaBoost e *Gradient Boosting* [Geron, A., 2019, pp 202]. Neste trabalho, além dos métodos *Ensemble* de *Boosting* AdaBoost e *Gradient Boosting*, vamos tratar também do método XGBoost

D.7.2.1 AdaBoost (ADAB)

O AdaBoost, é o mais popular entre os algoritmos que utilizam o método *Ensemble* de *Boosting*, sendo este a abreviação de '*adaptive boosting*', ele pode combinar um número arbitrário de previsores base fracos que utilizam o mesmo conjunto de treinamento de forma aleatória e repetidamente, permitindo, assim, que esse conjunto não precise ser grande (SEWELL, 2008).

Resumidamente, é uma técnica que encadeia sequencialmente previsores base em que cada novo previsor base corrigi o antecessor prestando mais atenção

(fornecendo maior peso relativo) às instâncias de treinamento que o antecessor subajustou [Geron, A., 2019, pp 202]. Para fazer a previsão do conjunto, o AdaBoost antes pondera as previsões encontradas por cada previsor base do conjunto (quanto mais preciso for um previsor, mais alto será seu peso) e calcula a previsão final com todas essas previsões ponderadas (GERON, 2019, p.202).

D.7.2.2
Gradient Boosting (GB)

O *Gradient Boosting* é popular entre os algoritmos que utilizam o método *Ensemble* de *Boosting*, e que também adiciona previsores sequencialmente a um conjunto, cada um corrigindo o antecessor (GERON, 2019, p.205). Na realidade, assim como Adaboost, o *Gradient Boosting* é um meta-modelo que encadeia, sequencialmente, vários modelos fracos cuja saída é adicionada para obter uma previsão geral, no entanto, o que o difere é que ele não está preocupado com a otimização dos parâmetros do modelos fracos que lhe servem de base, mas, sim, na previsão do modelo composto [Parr, T. e Howard, J., 2018]. Desta forma, ao invés de ajustar os pesos das instâncias incorretas a cada iteração, como dissemos que o Adaboost faz, o *Gradient Boosting* procura ajustar o novo previsor aos erros residuais feitos pelo previsor anterior (GERON, 2019, p.205).

Portanto, o treinamento do *Gradient Boosting* pode ser visto como ocorrendo em dois níveis, sendo um para treinar os modelos fracos e outro no modelo composto geral (PARR; HOWARD, 2018).

D.7.2.3
Extreme Gradient Boosting

(XGB)

O *Extreme Gradient Boosting* (ou XGBoost) pode ser encarado como uma implementação específica do método *Gradient Boosting*, mas utilizando aproximações mais precisas para encontrar o melhor modelo de árvore e, para tanto, emprega uma série de truques diferenciados que o tornam bem-sucedido, particularmente com dados estruturados (ELSINGHORST, 2018).

Portanto, o XGBoost é uma implementação otimizada do *Gradient Boosting* (GERON, 2019, p.210). Ele é muito utilizado em desafios de aprendizado de máquina para resultados de última geração, sendo o fator de maior sucesso por trás deste modelo a sua escalabilidade em todos os cenários, alcançando uma performance mais do que dez vezes mais rápida, em uma máquina, se comparada a outras soluções populares (CHEN; GUESTRIN, 2016, p.785).

Resumidamente, os truques que diferenciam o XGBoost são: 1) Ele utiliza a 2ª. derivada parcial da função de perda (semelhante ao método de Newton) como

uma aproximação para minimizar o erro desta função, que fornece mais informações sobre a direção dos gradientes e como chegar ao mínimo da função de perda; 2) Ele utiliza uma regularização avançada (L1 e L2), que melhora a generalização do modelo. Dessa forma, o treinamento se torna mais rápido e pode ser paralelizado / distribuído entre clusters (ELSINGHORST, 2018).

O XGBoost fornece alguns hiperparâmetros adicionais se comparados com os algoritmos baseados em árvore e *boosting*, que o colocam em vantagem sobre estes, tais como a Regularização, Parada Antecipada, Processamento Paralelo, Funções de Perda, continuar com o modelo existente, Diferentes Previsores Base e Múltiplos idiomas (BOEHMKE; GREENWELL, 2020).

D.8 Label Propagation (LPA)

Zhu et al. (ZHU; GHAHRAMANI; LAFFERTY, 2003) introduziram o Label Propagation (Propagação de Rótulos) como uma técnica para propagação de informações em redes.

O método *Label Propagation* (Propagação de Rótulos) é uma técnica de aprendizado semi-supervisionado utilizada para classificação em conjuntos de dados onde apenas uma pequena parte dos dados é rotulada. O principal objetivo é propagar rótulos de exemplos conhecidos para exemplos desconhecidos. Ele se baseia na propagação de rótulos a partir de exemplos rotulados para exemplos não rotulados, considerando a estrutura de similaridade dos dados. A propagação ocorre iterativamente, atualizando os rótulos dos dados não rotulados com base na similaridade entre os dados rotulados e não rotulados (ZHU; GHAHRAMANI; LAFFERTY, 2003).

O *Label Propagation*, após a fase de propagação dos rótulos, utiliza a informação de similaridade entre os dados rotulados e não rotulados para prever os rótulos dos dados não rotulados.

O método começa com um conjunto de dados onde apenas alguns pontos têm rótulos. Ele calcula a similaridade entre todos os pontos de dados, usando a função de similaridade gaussiana, que é definida como uma função exponencial negativa da distância Euclidiana entre duas instâncias. Em seguida, os rótulos conhecidos são propagados para os dados não rotulados com base nessa similaridade. Os rótulos são atribuídos aos dados não rotulados de acordo com a similaridade ponderada dos dados rotulados mais próximos. Ao fazer a previsão para novos dados, o algoritmo determina os rótulos dos exemplos não rotulados com base na similaridade com os exemplos rotulados já propagados. Este processo de propagação de rótulos e atribuição para os dados não rotulados é iterativo e continua até a convergência do algoritmo ou após um número fixo de iterações.

Para efetuar o procedimento acima, o algoritmo constrói um grafo totalmente conectado (ou completo), ponderado e não direcionado, onde cada nó representa uma instância e as arestas conectam pares de nós em que os pesos são a similaridade entre os pares. O grafo é construído de tal forma que os dados rotulados são conectados diretamente aos dados não rotulados, permitindo que os rótulos sejam propagados dos dados rotulados para os dados não rotulados.

(ZHU; GHAHRAMANI; LAFFERTY, 2003) define formalmente o *Label Propagation*: Sejam $(x_1, y_1) \ldots (x_l, y_l)$ dados rotulados, onde $Y_L = \{y_1, ..., y_l\}$ são os rótulos de classe. Assumimos que o número de classes *C* é conhecido e todas as classes estão presentes nos dados rotulados. Sejam $(x_{l+1}, y_{l+1}) \ldots (x_{l+u}, y_{l+u})$ dados não rotulados onde $Y_U = \{y_{l+1}, \cdots, y_{l+u}\}$ não são observados, geralmente $l \ll u$. Seja $X = \{x_1, \cdots, x_{l+1}\}$ onde $x_i \in \mathbb{R}$. O problema é estimar Y_U a partir de X e Y_L.

Queremos que pontos de dados próximos tenham rótulos semelhantes. Criamos um grafo totalmente conectado onde os nós são todos pontos de dados, rotulados e não rotulados. A aresta entre quaisquer nós *i,j* é ponderada de modo que quanto mais próximos os nós estiverem na distância euclidiana, d_{ij} , maior será o peso w_{ij} . Os pesos são controlados por um parâmetro:

$$w_{ij} = exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) = exp\left(\sum_{d=1}^{D} \frac{(x_i^d - x_j^d)^2}{\sigma^2}\right) \tag{D-3}$$

Breve, Fabricio A. [Breve, Fabricio A., 2000] resumiu o algoritmo proposto por Zhu et al [Zhu et al., 2003] conforme abaixo:

1. Calcule a matriz de afinidade W utilizando a Equação D-3;
2. Calcula a matriz diagonal D utilizando $D_{ii} \leftarrow \sum_j W_{ij}$;
3. Inicialize $\overline{Y}^{(0)} \leftarrow (y_1, y_1, \cdots, y_l, 0, \cdots, 0)$;
4. Repita
5. $\quad \overline{Y}^{(t+1)} \leftarrow D^{-1}W\overline{Y}^{(t)}$;
6. $\quad \overline{Y}^{(t+1)} = Y_l$;
7. até a convergência para $\overline{Y}^{\infty}$;
8. Rotule a amostra x_i, usando o sinal de $\overline{y_i}^{\infty}$;

Nesse estudo utilizaremos a classe LabelPropagation da biblioteca Scikit-learn (PEDREGOSA et al., 2011) que implementa o algoritmo com base no trabalho de Zhu et al (ZHU; GHAHRAMANI; LAFFERTY, 2003)

E
Detalhes Separação de Dados

Formalmente, Kuncheva, L. I. (KUNCHEVA, 2004, p.9) define e destaca os tipos de separação de um conjunto de dados. Seja *Z* o conjunto de dados *N* x *n*, um classificador *D* qualquer e $\overline{P}_d$ a predição gerada por esse classificador D, as principais alternativas para fazer o melhor uso de *Z* podem ser resumidas da seguinte forma:

- (1) Método de Resubstituição (método-R): Projete o classificador *D* em *Z* e teste-o em *Z*. Nesse caso, Kuncheva, L. I. (KUNCHEVA, 2004, p.9) informa que a predição gerada $\overline{P}_d$ seria tendenciosamente (bias) otimista;
- (2) Método de divisão ou hold-out em inglês (Método-H). Tradicionalmente, divide-se *Z* em duas metades, utilizando uma metade para treinamento e a outra metade para calcular $\overline{P}_d$. Nesse caso, Kuncheva, L. I. (KUNCHEVA, 2004, p.9) informa que $\overline{P}_d$ será uma estimativa pessimista (bias). Também são utilizadas divisões em outras proporções para *Z* (o que será abordado na subseção seguinte). Alternativamente, é possível trocar os dois subconjuntos de *Z* para obter outra estimativa $\overline{P}_d$ e, em seguida, calcular a média entre as duas. Uma variante desse método é a permutação dos dados, na qual se realiza *L* divisões aleatórias de *Z* em partes de treinamento e teste, calculando a média de todas as *L* estimativas de $\overline{P}_d$ com base nos respectivos conjuntos de teste;
- (3) Método da Validação cruzada ou *cross-validation* em inglês (também chamada de método de rotação ou método-p). Seleciona-se um inteiro *K* (preferencialmente um fator de *N*) e divide-se *Z* aleatoriamente em *K* subconjuntos de tamanho $\frac{N}{K}$. Em seguida, utiliza-se um subconjunto para testar o desempenho de *D* treinado na união dos *K*-1 subconjuntos restantes. Esse procedimento é repetido *K* vezes, escolhendo uma parte diferente para teste a cada iteração. Para obter o valor final de $\overline{P}_d$, calcula-se a média das *K* estimativas. Quando *K* é igual a *N*, o método é conhecido como "leave-one-out"(ou método-U).
- (4) Método Bootstrap: Este método tem como finalidade corrigir o viés otimista do método-R (1). Isso é realizado gerando aleatoriamente *L* conjuntos de tamanho *N* a partir do conjunto original *Z*, permitindo a reposição de elementos. Em seguida, avalia-se e calcula-se a média da taxa de erro dos classificadores construídos com base nesses conjuntos.

Grus, J. (GRUS, 2016, cap.11) sugere que devemos dividir os dados em três partes: um conjunto de treinamento para construir modelos, um conjunto de validação para escolher entre os modelos treinados e um conjunto de teste para avaliar o modelo final.

Na mesma linha, Geron, A. (GERON, 2019, p.32) sugere uma solução chamada holdout validation, em que os dados devem ser, primeiramente, divididos em conjunto de treino e teste e, em seguida, o conjunto de treino separado deve ser dividido em um conjunto de treino menor e um conjunto de validação. O conjunto de treino menor seria utilizado para treinamento dos modelos e avaliados com o conjunto de validação. O conjunto de teste seria utilizado para avaliação do modelo final. Nesse caso, de acordo com Kuncheva, L. I. (KUNCHEVA, 2004, p.10), o conjunto de dados de validação atuaria como um "pseudo-teste", isto é, continua-se o processo de treinamento até que a melhoria de performance no conjunto de treino não se reflita mais na melhoria de performance no conjunto de validação.

Raschka, S (RASCHKA, 2015, p.174) observa que uma desvantagem do método holdout (ou método de divisão) reside na sensibilidade da estimativa de desempenho em relação à estratégia de particionamento do conjunto de treinamento em subconjuntos de treinamento e validação, gerando variações nas estimativas quando diferentes amostras de dados são empregadas. Raschka, S (RASCHKA, 2015, p.174) então sugere uma abordagem mais robusta para a avaliação de desempenho, conhecida como validação cruzada k-fold, já descrita anteriormente.

As sugestões de Grus, J., Geron, A. e Raschka, S., se baseiam no fato de que serão avaliados diversos algoritmos com diversos parâmetros (ou hiperpârametros) em um espaço de parâmetros a serem ajustados em um processo de tunning a cada avaliação no conjunto de treino para a escolha do melhor deles. No entanto, nesse trabalho, os parâmetros (ou hiperpârametros) de *tunning* serão fixos e não ajustados, além do mais, não vamos escolher um melhor modelo entre modelos diferentes, mas, sim, avaliar, entre diversos modelos diferentes, o comportamento da performance de cada um deles quando o conjunto de dados original é enriquecido com estatísticas extraídas de um grafo derivado desse mesmo conjunto de dados.

Portanto, vamos adotar nessa pesquisa a separação de treino e teste *hold-out*, isto é, o conjunto de dados original será separado entre um conjunto de treino e teste, com uma proporção arbitrária (será visto com detalhes adiante).

Em seguida, os algoritmos passarão por uma fase de treinamento, utilizando o conjunto de treino, com um conjunto de parâmetros (ou hiperpârametros) sem ajustes e o modelo com o melhor resultado da combinação desses parâmetros será submetido ao conjunto de teste com dados ainda não vistos por esse modelo para avaliação dos resultados. Outro aspecto que nos leva a adotar esse método é que

as estatísticas extraídas do grafo têm que ser apuradas em conjuntos de treino e teste separados. Esse procedimento se justifica para que as estatísticas extraídas do grafo de treino não tenham acesso aos dados de teste, evitando viés na avaliação dos modelos.

E.0.1
Dimensionamento dos Conjuntos de Treinamento e Teste

Vimos, na subseção anterior, que a divisão de dados é parte importante para avaliação de modelos e construção de um modelo com bom desempenho de generalização. Uma vez estabelecido o método de separação dos dados de treinamento e teste, nos deparamos com a difícil decisão de como dimensionar os tamanhos desses conjuntos de dados.

No estudo de Xu Y. e Goodacre R. (XU; GOODACRE, 2018, p.249) descobriu-se que ter muitas ou poucas amostras no conjunto de treinamento teve um efeito negativo no desempenho estimado do modelo, sugerindo que é necessário ter um bom equilíbrio entre os tamanhos do conjunto de treinamento (treino e validação) e conjunto de teste para ter uma estimativa confiável do desempenho do modelo.

Uma proporção frequentemente adotada é a relação 80:20, que implica a alocação de 80% dos dados para treinamento e 20% para teste. Outras relações, como 70:30, 60:40 e até mesmo 50:50, são também amplamente empregadas na prática. Entretanto, não se evidencia uma orientação definitiva quanto à proporção considerada superior ou ideal em contextos específicos de conjuntos de dados. A escolha da relação 80:20, por exemplo, encontra justificação no princípio de Pareto amplamente reconhecido, mas é, em última análise, uma diretriz rudimentar adotada por profissionais do campo (JOSEPH, 2022, p.531).

Joseph, V. R. (JOSEPH, 2022, p.533), investigou a questão da proporção ideal para divisão de dados e chegou a propor um novo critério para avaliar a escolha da razão (ou proporção ou *ratio* em inglês) de divisão com uma solução simples e ótima, que, segundo o autor, parece concordar com a intuição e a prática comum. Essa proporção ideal de divisão de treinamento/teste é dada por $\sqrt{p} : 1$, onde *p* é o número de parâmetros em um modelo de regressão linear utilizado como base de estudo. Essa proporção é derivada da equação $\gamma^* = \frac{1}{\sqrt{p+1}}$, que fornece a proporção de teste. O autor destaca que, embora tenha derivado a razão de divisão ideal para problemas de regressão, ele acredita que ela também pode servir como uma diretriz para fins de classificação porque o ajuste de um modelo de regressão logística pode ser visto como o ajuste de um modelo de regressão linear a alguns dados latentes contínuos. No entanto, mais pesquisas são necessárias para compreender sua utilidade em problemas de classificação.

Geron, A. (GERON, 2019, p.31) observa que é comum usar 80% dos dados para treinamento e reter 20% para teste. No entanto, isso depende do tamanho do conjunto de dados: se o conjunto de dados contém 10 milhões de instâncias, então manter 1% significa que o conjunto de teste conterá 100.000 instâncias: isso é, conclui o autor, provavelmente mais do que o suficiente para obter uma boa estimativa do erro de generalização.

F
Detalhes Métricas de Desempenho

Geron, A. (GERON, 2019, p.92) observa que várias métricas são propostas para avaliar modelos de ML em diferentes aplicações e sugere que uma boa maneira de avaliar o desempenho de um classificador é olhar para a matriz de confusão. Segundo Escovedo e Koshiyama (ESCOVEDO; KOSHIYAMA, 2020, p.90), a matriz de confusão oferece um detalhamento do desempenho do modelo de Classificação, mostrando, para cada classe, o número de classificações corretas em relação ao número de classificações indicadas pelo modelo.

Ainda segundo Escovedo e Koshiyama (ESCOVEDO; KOSHIYAMA, 2020, p.91), a partir da matriz de confusão é possível verificar o número de Falsos Positivos, (FP - também conhecido como Alarme Falso; ou seja, quando o resultado esperado é negativo, mas o modelo resulta em positivo) e Falsos Negativos (FN - também conhecido como Alarme Defeituoso; ou seja, quando o resultado esperado é positivo, mas o modelo resulta em negativo), bem como os números de Verdadeiros Positivos, (VP, quando o resultado esperado é positivo e o modelo resulta em positivo - TP: *True Positive* em inglês) e Verdadeiros Negativos (VN, quando o resultado esperado é negativo e o modelo resulta em negativo – TN: *True Negative* em inglês).

Adicionalmente, a matriz de confusão revela a localização das classificações incorretas, o que se torna particularmente relevante em cenários com um número de classes alto. Um valor alto fora da diagonal principal na matriz pode sinalizar desafios distintos entre duas classes, requerendo, portanto, abordagens separadas (KUNCHEVA, 2004, p.11).

Tabela F.1: Matriz de Confusão para problema de classificação binária

	Predição		
Real	**Classe**	**Negativo**	**Positivo**
	Negativo	TN	FP
	Positivo	FN	TP

Partindo da observação da matriz de confusão na Tabela F.1, diversas métricas podem ser reconhecidas. Escovedo e Koshiyama (ESCOVEDO; KOSHIYAMA, 2020, p.92) relacionam as métricas de Sensibilidade, que representa a capacidade de se identificar corretamente os indivíduos que apresentam a característica de interesse, e Especificidade, que mede a quantidade de negativos verdadeiros corretamente classificados, e representa a capacidade em identificar corretamente os indivíduos que não apresentam a condição de interesse. Geron, A. (GERON, 2019,

p.93-95) acrescenta as métricas Precisão, que mede a acurácia das predições positivas, e a pontuação F1, que combina a Precisão e Sensibilidade através da sua média harmônica.

Detalhamos a seguir as métricas selecionadas para avaliação dos modelos nesse trabalho.

Acurácia: é a razão entre o número de instâncias (ou amostras ou exemplo) classificadas corretamente e o número geral de instâncias [Geron, A., 2019, pag.4]. Ela pode ser calculada a partir da matriz de confusão como:

$$Acurácia = \frac{TN + TP}{TP + FP + TN + FN} \tag{F-1}$$

Alternativamente, de forma mais simples, temos a definição de Zheng, A. et al (Zheng, A. et al, 2019, pag.8) para Acurácia:

$$Acurácia = \frac{\text{nr.de predições corretas}}{\text{nr.total de instâncias}} \tag{F-2}$$

Em outras palavras, acurácia seria a resposta à pergunta: "Com que frequência o classificador está correto ?"

Outra forma alternativa de calcular a Acurácia, sem a utilização da matriz de confusão, foi sugerida por Kuncheva, L. I. (KUNCHEVA, 2004, p.8), de uma maneira mais formal, como sendo 1 menos o erro total de classificação, isto é, 1 – Erro(D), em que D é um classificador qualquer e Erro(D) é dado por: $Erro(D) = \frac{N_{erro}}{N}$, onde N_{erro} é o numero de instâncias (ou exemplos ou amostras) que o classificador D errou na predição e N o número total de instâncias no conjunto de dados D.

Raschka, S. (RASCHKA, 2015, p.191) descreve o erro total de classificação utilizando a matriz de confusão:

$$ErroTotal = \frac{FP + FN}{FP + FN + TP + TN} \tag{F-3}$$

Precisão (ou *Precision* em inglês): mede a fração de exemplos classificados como positivos e que são verdadeiramente positivos.

$$\text{Precisão} = \frac{TP}{TP + FP} \tag{F-4}$$

Geron, A. (GERON, 2019, p.93) define a Precisão como a Acurácia das predições positivas. Zheng, A. et al (ZHENG; SHELBY; VOLCKHAUSEN, 2019, p.8) define a Precisão com uma pergunta: "Dos itens que o classificador previu serem positivos, quantos são realmente positivos ?", sendo representada por:

$$\text{Precisão} = \frac{\text{nr.predições positivas corretas}}{\text{nr.de predições positivas}} \tag{F-5}$$

A inadequação da métrica de Precisão, quando analisada de forma isolada, é apontada por Geron, A. (GERON, 2019, p.93), destacando que a sua obtenção perfeita pode ser alcançada de modo trivial mediante a realização de uma única

previsão positiva, cuja correção é assegurada $(precisão = 1/1 = 100\%)$. Isso não seria muito útil, pois o classificador ignoraria todas as instâncias positivas, exceto uma. Por isso, ainda de acordo com Geron, A. (GERON, 2019, p.93), a precisão é normalmente utilizada junto com outra métrica chamada Recall, também chamada de Sensibilidade ou Taxa de Verdadeiro Positivo (TPR em inglês) que veremos a seguir.

Sensibilidade (ou *Recall* em inglês ou TPR): é a proporção de instâncias positivas que são detectadas corretamente pelo classificador.

$$Sensibilidade = \frac{TP}{TP + FN} \tag{F-6}$$

Em outras palavras, é a proporção de verdadeiros positivos entre todos os elementos que são positivos.

Simplificadamente, Sensibilidade pode ser representada por

$$Sensibilidade = \frac{\text{nr.predições positivas corretas}}{\text{nr.total de instâncias positivas}} \tag{F-7}$$

ou a Acurácia das predições positivas.

Citamos, anteriormente, que a métrica Precisão deveria ser observada em conjunto com a métrica Sensibilidade (ou *Recall*). Buckland, M.K., e Gey, F.C. (BUCKLAND; GEY, 1994, p.12) examinaram como essas métricas estão relacionadas e destacam que, em algumas circunstâncias, é difícil obter um alto valor tanto de Precisão quanto de *Recall* simultaneamente. Os autores sugerem que é inevitável, sob certas condições, uma compensação (*rade-off* em inglês) entre Precisão e *Recall*. Isso significa que, ao ajustar o modelo ou as configurações de classificação para melhorar uma das métricas, a outra pode ser afetada negativamente.

Uma forma de harmonizar essa relação entre Precisão e Sensibilidade (*Recall*) é dada pela métrica F_1 que veremos a seguir.

F$_1$: é definida como a média harmônica entre a Precisão e Sensibilidade (*Recall*) (ZHENG; SHELBY; VOLCKHAUSEN, 2019, p.14)

$$F_1 = 2\frac{\text{Precisão x Sensibilidade}}{\text{Precisão+Sensibilidade}} \tag{F-8}$$

Chicco, D. e Jurman, G. (CHICCO; JURMAN, 2020, p.5) apresentam também o cálculo da F_1 a partir da matriz de confusão e sugerem que, em determinadas condições, esse é preferível:

$$F_1 = \frac{2TP}{\text{2TP+FP+FN}} \tag{F-9}$$

Ao contrário da média aritmética, a média harmônica tende para o menor dos dois elementos. Portanto, a métrica F_1 será pequena se a Precisão ou a Sensibilidade (*Recall*) são pequenos (ZHENG; SHELBY; VOLCKHAUSEN, 2019, p.14). Essa característica ressalta a sensibilidade do F_1 com a métrica de desempenho mais fraca, incentivando um equilíbrio entre Precisão e Sensibilidade

(*Recall*), pois o F_1 será elevado apenas quando ambos forem elevados. Em todas as medidas, o valor varia de 0 a 1.

Resumindo:

- TN (*True Negative*): A quantidade de instâncias da classe negativa que o modelo previu corretamente como negativo.
- FP (*False Positive*): A quantidade de instâncias que originalmente pertencem a classe negativa, mas o modelo previu como positivo.
- FN (*False Negative*): A quantidade de instâncias que originalmente pertencem a classe positiva, mas o modelo previu como negativo.
- TP (*True Positive*): A quantidade de instâncias da classe positiva que o modelo previu corretamente como positivo.

A Tabela F.2 resume as fórmulas de cálculo das métricas para problema de classificação binária.

Tabela F.2: Fórmulas de Medidas de Desempenho - Classe Binária

Métrica	Pela Matriz de Confusão	Descrição
Acurácia	$\frac{TN+TP}{TP+FP+TN+FN}$	$\frac{\text{nr.de predições corretas}}{\text{nr.total de instâncias}}$
Precisão	$\frac{TP}{TP+FP}$	$\frac{\text{nr.predições positivas corretas}}{\text{nr.de predições positivas}}$
Sensibilidade (Recall ou TPR)	$\frac{TP}{TP+FN}$	$\frac{\text{nr.predições positivas corretas}}{\text{nr.total de instâncias positivas}}$
F_1	$\frac{2TP}{2TP+FP+FN}$	$2\frac{\text{Precisão x Sensibilidade}}{\text{Precisão+Sensibilidade}}$

Múltiplas Classes

Até o momento trabalhamos com a matriz de confusão, e as métricas derivadas desta, para o caso das classes binárias, resultando em uma matriz de confusão 2 x 2. Na verdade, as classes podem tomar valores de tamanho n e, nesse caso, a matriz de confusão que vai representar essas classes deverá assumir a dimensão n x n. A Tabela F.3 apresenta uma matriz de confusão para uma classe com 3 valores distintos (A, B e C), ficando com a dimensão 3 x 3, onde M_ij é a quantidade de instâncias entre o par de classes i e j em que $i, j = A, B, C$

Tabela F.3: Matriz de confusão em problema de classificação múltipla com 3 classes

	Predição				
	Classe	A	B	C	$\sum$
Real	A	M_{AA}	M_{AB}	M_{AC}	$\sum M_{Aj}$
	B	M_{BA}	M_{BB}	M_{BC}	$\sum M_{Bj}$
	C	M_{CA}	M_{CB}	M_{CC}	$\sum M_{Cj}$
	$\sum$	$\sum M_{iA}$	$\sum M_{iB}$	$\sum M_{iC}$	Total

Para o caso da matriz de confusão com múltiplas classes, seguimos a mesma lógica utilizada na matriz binária, mas efetuando esse procedimento para cada classe. Fixando uma classe, identificamos os elementos TP, FP, TN, FN conforme apresentado na Tabela F.4, onde a matriz de confusão da Tabela F.3 é representada considerando apenas a classe B como foco principal.

Tabela F.4: Matriz de confusão com 3 classes onde a **classe B** é a referência.

	Predição			
	Classe	A	B	C
Real	A	TN	FP	TN
	B	FN	TP	FN
	C	TN	FP	TN

Da Tabela F.4, identificados os elementos TP, FP, FN, FP para classe B, podemos calcular os valores para essa classe, isto é, $TP = M_{BB}$, $FP = (M_{AB} + M_{CB})$, $TN = (M_{AA} + M_{AC} + M_{CA} + M_{CC})$ e $FN = (M_{BA} + M_{BC})$.

Agora que temos os elementos básicos para cálculo das métricas nessa classe B, podemos utilizar para esse cálculo as fórmulas (Vide resumo na Tabela F.2) que descrevemos para matriz de confusão binária.

Repetindo o procedimento acima para as classes A e C, teríamos então as métricas calculadas para cada classe, isto é, Acurácia, Precisão, Sensibilidade (*Recall*) e F_1 para cada classe A, B e C. O próximo passo é calcular uma medida que expresse as métricas de cada classe em um só número.

De acordo com Raschka, S (RASCHKA, 2015, p.197) o procedimento descrito acima é conhecido como "Um contra todos (*One vs. All* em inglês - OvA)" e essas métricas podem ter duas especificações diferentes, dando origem a duas métricas diferentes: média Micro e média Macro.

Grandini, M. et al (GRANDINI; BAGLI; VISANI, 2020, p.6-8) fornece as fórmulas de cálculo de média Micro e média Macro para K classes.

$$Precisão_k = \frac{TP_k}{TP_k + FP_k} \tag{F-10}$$

$$Recall_k = \frac{TP_k}{TP_k + FN_k} \tag{F-11}$$

Média Micro: Calculamos como a soma de todos os verdadeiros positivos para todas as classes, dividido por todas as predições (Precisão) ou todas as instâncias (*Recall*).

$$Precisão_{micro} = \frac{\sum_{k=1}^{K} TP_k}{\sum_{k=1}^{K}(TP_k + FP_k)} \tag{F-12}$$

$$Recall_{micro} = \frac{\sum_{k=1}^{K} TP_k}{\sum_{k=1}^{K}(TP_k + FN_k)} \tag{F-13}$$

$$F_{1_{micro}} = 2\left(\frac{Precisão_{micro}.Recall_{micro}}{Precisão_{micro} + Recall_{micro}}\right) \tag{F-14}$$

Média Macro: Calculamos como a média aritmética de todos os valores da métrica para todas as classes.

$$Precisão_{macro} = \frac{\sum_{k=1}^{K} Precisão_k}{K} \tag{F-15}$$

$$Recall_{micro} = \frac{\sum_{k=1}^{K} TP_k}{\sum_{k=1}^{K}(TP_k + FN_k)} \tag{F-16}$$

$$F_{1_{micro}} = 2\left(\frac{Precisão_{micro}.Recall_{micro}}{Precisão_{micro} + Recall_{micro}}\right) \tag{F-17}$$

Mutanov, G. et al (MUTANOV VLADISLAV KARYUKIN, 2021, p.922) fornece as fórmulas de cálculo para essas métricas para o exemplo com 3 classes (A, B e C).

$$Precisão_{micro} = \frac{TP_A + TP_B + TP_C}{TP_A + TP_B + TP_C + FP_A + FP_B + FP_C} \tag{F-18}$$

$$Recall_micro = \frac{TP_A + TP_B + TP_C}{TP_A + TP_B + TP_C + FN_A + FN_B + FN_C} \tag{F-19}$$

$$F_{1_{micro}} = \frac{Precisão_{micro}.Recall_{micro}}{Precisão_{micro} + Recall_{micro}} \tag{F-20}$$

$$Precisão_{macro} = \frac{Precisão_A + Precisão_B + Precisão_C}{3} \tag{F-21}$$

$$Recall_{macro} = \frac{Recall_A + Recall_B + Recall_C}{3} \tag{F-22}$$

$$F_{1_{micro}} = \frac{F_{1_A} + F_{1_B} + F_{1_C}}{3} \tag{F-23}$$

Mutanov, G. et al (MUTANOV VLADISLAV KARYUKIN, 2021, p.922) ainda introduz a fórmula de cálculo para média ponderada (Weighted em inglês)

dos valores das classes. O cálculo da média ponderada (Weighted) é semelhante ao da média Macro. No entanto, cada classe tem um peso que será o número de instâncias (ou exemplos ou amostras) que a classe possui.

Média Ponderada (Weighted):

$$Precision_{weighted} = \frac{W_A.Precision_A + W_B.Precision_B + W_C.Precision_C}{W_A + W_B + W_C} \tag{F-24}$$

$$Recall_{weighted} = \frac{W_A.Recall_A + W_B.Recall_B + W_C.Recall_C}{W_A + W_B + W_C} \tag{F-25}$$

$$F_{1_{weighted}} = \frac{W_A.F_{1_A} + W_B.F_{1_B} + W_C.F_{1_C}}{W_A + W_B + W_C} \tag{F-26}$$

onde w_A, w_B e w_C são os pesos (número de instâncias) correspondentes as classes.

De acordo com Raschka, S. (RASCHKA, 2015, p.197-198), a média Micro é útil se queremos ponderar cada instância ou previsão igualmente, a média Macro pondera todas as classes igualmente para avaliar o desempenho geral de um classificador em relação aos rótulos de classe mais frequentes. A média ponderada (*Weighted*) é útil se estivermos lidando com classes desbalanceadas.

Os conjuntos de dados que utilizaremos nesse estudo são bem diferentes, tendo várias situações como classes binárias e múltiplas, classes balanceadas e desbalanceadas etc. Isto posto, vamos adotar a média ponderada (*Weighted*) como padrão.

G
Detalhes Otimização de Hiperparâmetros

Hiperparâmetros em modelos de aprendizado de máquina são parâmetros que não são aprendidos diretamente a partir dos dados de treinamento, mas são configurados antes do início do processo de treinamento do modelo. Eles desempenham um papel crítico na determinação do comportamento e desempenho do modelo. Ao contrário dos parâmetros do modelo, que são ajustados durante o treinamento, os hiperparâmetros são definidos pelo desenvolvedor ou pelo cientista de dados antes de iniciar o processo de treinamento (ZHENG; SHELBY; VOLCKHAUSEN, 2019, p.27-28).

Hutter, F. et al (HUTTER; KOTTHOFF; VANSCHOREN, 2019, p.3) observa que todo sistema de aprendizado de máquina possui hiperparâmetros, e uma das tarefas mais básicas no aprendizado de máquina é definir automaticamente esses hiperparâmetros para otimizar o desempenho.

Zheng, A. et al (ZHENG; SHELBY; VOLCKHAUSEN, 2019, p.27-28) chama atenção que as escolhas relacionadas aos hiperparâmetros podem ter uma grande influência na precisão das previsões geradas pelo modelo treinado. As configurações ótimas de hiperparâmetros muitas vezes variam de um conjunto de dados para outro, o que implica que um ajuste cuidadoso é necessário para adaptar os hiperparâmetros a cada cenário específico de dados.

Zheng, A. et al (ZHENG; SHELBY; VOLCKHAUSEN, 2019, p.27-28) ainda introduz uma definição para otimização de hiperparâmetros, em que visto que os hiperparâmetros não são diretamente determinados durante o processo de treinamento, torna-se imperativo instituir um metaprocesso dedicado à sintonia (ou ajuste) desses hiperparâmetros e que essa etapa é frequentemente denominada otimização de hiperparâmetros (ou *hyperparameter tuning* em língua inglesa). Portanto, conclui, o ajuste de hiperparâmetros é uma tarefa de meta-otimização.

Hutter et al (HUTTER; KOTTHOFF; VANSCHOREN, 2019, p.3) destacam uma série de vantagens inerentes à otimização de hiperparâmetros, que incluem aprimorar significativamente o desempenho dos algoritmos de aprendizado de máquina, adaptando-os de maneira precisa às nuances do problema em questão. Além disso, essa prática contribui para melhorar a reprodutibilidade e a imparcialidade em estudos científicos, viabilizando a comparação justa entre diferentes métodos, onde cada um recebe o mesmo nível de ajuste.

No entanto, Hutter et al (HUTTER; KOTTHOFF; VANSCHOREN, 2019, p.4) também apontam desvantagens relevantes, como o custo considerável associado ao processo de otimização de hiperparâmetros, particularmente quando se lida

com modelos de grande porte ou conjuntos de dados extensos. Além disso, o espaço de configuração frequentemente se mostra complexo, abrangendo uma variedade de hiperparâmetros contínuos, categóricos e condicionais, sendo caracterizado por uma alta dimensionalidade.

Conforme observado por Yu e Zhu (YU; ZHU, 2020, p.16), a pesquisa em grade enfrenta o desafio da maldição da dimensionalidade, onde o consumo de recursos computacionais se intensifica à medida que um maior número de hiperparâmetros e suas respectivas listas de valores são inter-relacionados para fins de otimização. Esse fenômeno é inerente à complexidade do espaço de hiperparâmetros e representa uma barreira significativa no contexto da otimização sistemática.

Como meio de atenuar as desvantagens mencionadas anteriormente, Yu e Zhu (YU; ZHU, 2020, p.2) identificam uma abordagem viável e econômica que se baseia no conhecimento prévio de especialistas, isto é, aproveitando a experiência de especialistas para direcionar o processo, permitindo uma seleção criteriosa de parâmetros que exercem influência significativa. Consequentemente, restringe a amplitude do espaço de busca e mitiga os custos computacionais associados à otimização de hiperparâmetros.

O resultado do ajuste de hiperparâmetros culmina na identificação da configuração ótima dos hiperparâmetros, enquanto o resultado do processo de treinamento do modelo está associado à busca da configuração ideal dos parâmetros do modelo. Cada configuração de hiperparâmetros proposta inicia um processo interno de treinamento do modelo, que resulta na geração de um modelo ajustado aos dados e na obtenção de métricas de avaliação em conjuntos de validação cruzada ou validação.

Superada a análise de diversas configurações de hiperparâmetros, o processo de ajuste de hiperparâmetros determina a configuração que resulta no modelo de melhor desempenho. Concluindo, a etapa final envolve o treinamento no modelo de melhor desempenho (ou somente melhor modelo) usando o conjunto de dados completo de treinamento, empregando a configuração de hiperparâmetros otimizada previamente identificada (ZHENG; SHELBY; VOLCKHAUSEN, 2019, p.29).

Zheng, A. et al (ZHENG; SHELBY; VOLCKHAUSEN, 2019, p.27-28) observa que, conceitualmente, a otimização de hiperparâmetros é uma tarefa de otimização, assim como o treino de modelos. No entanto, na prática, estas duas tarefas são bastante diferentes. Ao treinar um modelo, a qualidade de um conjunto proposto de parâmetros do modelo pode ser escrita como uma fórmula matemática.

No entanto, ao ajustar os hiperparâmetros, a qualidade desses hiperparâmetros não pode ser escrita numa fórmula fechada, porque depende do resultado do

processo de treino do modelo.

Zheng, A. et al (ZHENG; SHELBY; VOLCKHAUSEN, 2019, p.30) aponta que até há alguns poucos anos, os únicos métodos disponíveis eram a pesquisa em grade e a pesquisa aleatória (Random Search, em inglês). Nos últimos anos, tem havido um interesse crescente na otimização de hiperparâmetros com novos métodos sendo apresentados.

Nesse momento, vamos nos concentrar apenas na pesquisa em grade, que foi aquela utilizada neste estudo, mas o leitor pode se aprofundar em outros métodos consultando as referências citadas.

Raschka, S. (RASCHKA, 2015, p.185) considera a abordagem da pesquisa em grade bastante simples, em que é adotado um paradigma de pesquisa sistemática e exaustiva (força-bruta), em que se define uma lista de valores para diversos hiperparâmetros. O computador, então, realiza uma avaliação metódica do desempenho do modelo para cada combinação desses valores, visando a identificação da configuração ótima de hiperparâmetros.

Esse processo implica em uma análise minuciosa de todas as combinações possíveis, representando um método de busca intensiva que visa aperfeiçoar o desempenho do modelo.

Zheng, A. et al (ZHENG; SHELBY; VOLCKHAUSEN, 2019, p.31) observa que algumas suposições são necessárias para especificar o valor mínimo e valores máximos dos hiperparâmetros e que, às vezes, as pessoas executam uma pequena lista de opções ou grade, para ver se o ideal está em qualquer um dos extremos e, em seguida, expande a grade naquela direção.

Isso é chamado de pesquisa manual em grade. Ainda segundo Zheng, A. et al (ZHENG; SHELBY; VOLCKHAUSEN, 2019, p.31), a pesquisa em grade é extremamente simples de configurar e trivial de paralelizar, mas é o método mais caro em termos de tempo total de cálculo.

Geron, A. (GERON, 2019, p.79) sugere que para implementar a pesquisa em grade e se liberar do tempo que o ajuste manual para explorar as muitas combinações de hiperparametos levaria, devemos utilizar o GridSearchCV do Scikit-Learn para efetuar essa procura autoA. (GERON, 2019, p.79)[Geron, A., 2019, pag.79], é dizer quais hiperparâmetros você deseja experimentar e quais valores experimentar e ele avaliará todas as combinações possíveis de valores de hiperparâmetros, usando validação cruzada.

Concluindo, Yu e Zhu (YU; ZHU, 2020, p.16) observam que, na prática, a pesquisa em grade quase que só é preferível quando o analista tem experiência sobre esses hiperparâmetros para permitir a definição de um espaço de pesquisa estreito e não mais do que três hiperparâmetros precisam ser ajustados simultaneamente.

H
Detalhes Trabalhos Relacionados

(1) Enriquecimento para o aumento de linhas e/ou pelo aumento de colunas

No campo do enriquecimento pelo aumento de linhas e/ou colunas em conjunto de dados tabulares, Dong, Y. e Oyamada, M. (DONG; OYAMADA, 2022) desenvolveram um método para enriquecimento de dados tabulares automático, com colunas externas com origens em data lakes para melhorar a predição de modelos de ML. O trabalho propõe uma solução de enriquecimento de tabelas para recuperar, selecionar, unir e agregar tabelas candidatas passo a passo. O método alcançou bons resultados quando submetido a diversos modelos de ML.

Na mesma linha, Cvetkov-Iliev, A. et al (CVETKOV-ILIEV; ALLAUZEN; VAROQUAUX, 2023) abordam o procedimento de incorporação de dados relacionais para enriquecer tabelas de dados com fontes externas, com o objetivo de melhorar a precisão dos modelos de aprendizado de máquina. Algumas fontes externas que podem ser usadas para enriquecer tabelas de dados incluem coordenadas GPS, população, renda média e outros atributos numéricos relacionados às entidades de interesse. A técnica proposta envolve a criação de representações vetoriais de entidades (por exemplo, cidades) que capturam informações relevantes para o problema em questão.

Os dados relacionais sobre as entidades são representados como um grafo e métodos de incorporação de grafos são adaptados para criar vetores de características para cada entidade. A técnica envolve modelar bem as diferentes relações entre as entidades e capturar atributos numéricos relevantes. Algumas fontes externas que podem ser usadas para enriquecer tabelas de dados incluem coordenadas GPS, população, renda média e outros atributos numéricos relacionados às entidades de interesse. Os autores concluem que essa abordagem é mais escalável e requer menos esforço humano do que a criação manual de características.

Por outro lado, com uma linha mais simples e prática, Zhang, S. e Balog, K. (ZHANG; BALOG, 2017) apresentam uma nova abordagem para equipar programas de planilhas com recursos de assistência inteligente, especificamente para tabelas com foco em entidade. Os autores desenvolveram modelos probabilísticos generativos para preencher linhas com instâncias adicionais e preencher colunas com novos cabeçalhos.

Com uma abordagem mais robusta e com base em tabelas web, Zhang, M. e Chakrabarti, K. (ZHANG; CHAKRABARTI, 2013) apresentam o InfoGather+, um sistema para automatizar a tarefa de coleta de informações sobre entidades por

meio de tabelas web. O sistema usa um grafo semântico e uma API de aumento de entidade para combinar e anotar, com precisão, atributos numéricos e que variam no tempo, mesmo quando expressos em unidades ou escalas diferentes.

Os autores propõem uma nova técnica baseada em modelos probabilísticos para descobrir os rótulos semânticos das colunas e as correspondências semânticas entre as colunas em todas as tabelas da web coletivamente, em vez de individualmente. Também foram desenvolvidos algoritmos eficientes para resolver a tarefa de descoberta conjunta. Os autores concluem que, ao aproveitar o grafo semântico, o InfoGather+ pode responder a consultas de aumento de entidade com muito mais precisão em comparação com o estado da arte.

Ainda na linha de trabalho com tabelas web, Lehmberg, O. e Bizer, C. (LEHMBERG; BIZER, 2017) exploram os desafios da correspondência entre tabelas web e bases de dados de conhecimento e propõe uma solução para melhorar a precisão. A solução proposta envolve a combinação de tabelas de cada site em tabelas maiores e depois combinar essas tabelas ampliadas com a base de conhecimento ou tabela base. Para este processo, foram avaliados diferentes métodos de correspondência de esquemas em combinação com o refinamento holístico da correspondência.

A Limitação do procedimento de junção a tabelas da web do mesmo site diminuiu a heterogeneidade e permitiu unir tabelas com altíssima precisão. Os autores concluem que os experimentos mostram que aplicar a junção da tabela antes de executar o método de correspondência real melhora a correspondência.

(2) Enriquecimento com pesquisa de tabela

Para a tarefa de enriquecimento de dados com a pesquisa de tabelas no ambiente web, Cafarella, M. J., Halevy, A e Khoussainova, N. (CAFARELLA; HALEVY; KHOUSSAINOVA, 2009) apresentam o sistema WebTables, que extrai e analisa dados estruturados de tabelas HTML na web. A análise de um conjunto grande de dados estruturados permite a integração e análise de dados na web em uma escala que não era possível antes. O sistema pesquisa Tabelas na Web para montar um índice de tabelas relacionadas da Web e suas semânticas. Também em ambiente web, Pimplikar, R. e Sarawagi, S. (PIMPLIKAR; SARAWAGI, 2012) apresentam um mecanismo de pesquisa estruturado que utiliza tabelas disponíveis na web para fornecer resultados de tabelas com várias colunas em resposta a consultas de palavras-chave.

Ainda com pesquisa de tabelas no ambiente web Bhagavatula, C. S., Noraset, T. e Downey, D. (BHAGAVATULA; NORASET; DOWNEY, 2013) desenvolveram o WikiTables, um aplicativo da web que permite aos usuários explorar interativamente o conhecimento tabular extraído da Wikipedia.

Mas a pesquisa pode ser efetuada não apenas com palavra-chave, mas

também pode ser uma tabela, sendo chamada de pesquisa baseada em tabela (*table matching*, em inglês). No trabalho de Sarma, D.A., et al (SARMA et al., 2012) discutem uma estrutura e algoritmos para detectar tabelas HTML na web relacionadas em um grande corpus de tabelas heterogêneas. O principal benefício da criação desses repositórios é fomentar a integração de dados, facilitando a descoberta e reutilização de conjuntos de dados existentes. Nesse caso o objetivo é apresentar ao usuário os resultados da pesquisa para atender sua necessidade de informação.

Também na linha de fornecer ao usuário os resultados da pesquisa para atender sua necessidade de informação para enriquecimento, Yakout, M. (YAKOUT et al., 2012) apresentam o sistema InfoGather, que automatiza tarefas de coleta de informações utilizando tabelas web. Os autores propõem uma estrutura de correspondência holística que alcança alta precisão e cobertura, e uma nova arquitetura que aproveita o pré-processamento no MapReduce para obter tempos de resposta extremamente rápidos no momento da consulta.

Ainda na linha de apresentação de resultado da pesquisa ao usuário, Gupta, R. e Sarawagi, S. (GUPTA; SARAWAGI, 2009) desenvolveram uma abordagem ponta a ponta totalmente não supervisionada que, dadas as linhas de consulta de exemplo - (a) recupera listas HTML relevantes para a consulta a partir de um rastreamento pré-indexado de listas da web, (b) segmenta os registros da lista e mapeia os segmentos para o esquema de consulta usando um modelo estatístico, (c) consolida o resultados de várias listas em uma tabela mesclada unificada, (d) e apresenta ao usuário os registros consolidados classificados por sua participação estimada na relação alvo.

Por outro lado, no trabalho de Lehmberg, O., et al (LEHMBERG et al., 2015) que apresentam o mecanismo "Mannheim Search Join (MSJ)", o objetivo é ser uma das etapas para dar entrada em outra tarefa de enriquecimento. Nesse trabalho, o mecanismo MSJ usa uma combinação de técnicas de normalização, pesquisa e consolidação de dados para combinar os dados descobertos com a tabela local. O trabalho também discute possíveis melhorias no mecanismo MSJ, como detecção de atributos e cabeçalhos de assunto e métodos de correspondência aprimorados.

Também na linha de ser uma das etapas para dar entrada em outra tarefa de enriquecimento, Nargesian et al. (NARGESIAN et al., 2018) apresentam uma solução probabilística para encontrar tabelas que podem ser unidas a uma tabela de consulta em grandes repositórios. Duas tabelas podem ser unidas se compartilharem atributos do mesmo domínio.

A solução formaliza três modelos estatísticos que descrevem como atributos eleitos para união são aqueles gerados a partir de domínios de conjunto, domínios

semânticos com valores de uma ontologia e domínios de linguagem natural. Segundo os autores, a abordagem proposta supera os algoritmos existentes em termos de velocidade e precisão, e pode pesquisar com eficiência repositórios de dados abertos contendo mais de um milhão de atributos.

Ainda na linha de tabelas como entrada, mas agora utilizando um grafo de conhecimento para auxiliar nas relações entre as tabelas encontradas, o trabalho de Fetahu, B., Anand, A. e Koutraki, M. (FETAHU; ANAND; KOUTRAKI, 2019) apresentam TableNet, uma abordagem para determinar relações refinadas para tabelas da Wikipedia.

Os autores propõem um grafo de conhecimento de tabelas interligadas com subPartOf e relações equivalentes, usando um algoritmo eficiente para encontrar todas as tabelas relacionadas candidatas e uma abordagem baseada em redes neurais (RNN), levando em consideração os esquemas da tabela e os dados da tabela correspondentes, que determinam com alta precisão a relação da tabela para um par de tabelas.

Os autores concluem que o TableNet pode capturar com eficácia relações refinadas entre tabelas, com aplicações potenciais na recuperação de informações, resposta a perguntas e construção de gráficos de conhecimento.

No campo da pesquisa por tabelas em data lakes, Dong, Y. (DONG et al., 2020) propõe um framework chamado PEXESO para descoberta eficiente de tabelas que podem ser unidas em data lakes. O PEXESO utiliza abordagens baseadas em similaridade de alta dimensão para encontrar tabelas que podem ser unidas de forma mais significativa e útil para o enriquecimento de dados em tarefas de aprendizado de máquina. A saída do PEXESO é um conjunto de tabelas uníveis que correspondem às colunas de consulta da tabela de consulta. A arquitetura geral do PEXESO consiste em dois componentes principais: indexação e busca.

(3) Enriquecimento com interpretação de tabela (metadados e semânticas)

No campo do enriquecimento dos metadados de tabelas com a sua semântica na web, Venetis, P., et al (VENETIS et al., 2011) desenvolveram um sistema para recuperação da semântica de tabelas na web. O sistema enriquece as tabelas com anotações para facilitar operações como procurar tabelas e localizar tabelas relacionadas. Também com base em tabelas web, Wang, D., et al (WANG et al., 2021) apresentam uma nova abordagem para extrair conhecimento de tabelas relacionais em páginas da web.

O modelo Table Convolutional Network (TCN) considera informações contextuais intra e inter-tabelas, permitindo uma extração de informações mais precisa e abrangente. O TCN aprende primeiro a incorporação latente de cada célula da tabela relacional, agregando informações contextuais intra e inter-tabelas. Essas

incorporações de células aprendidas são então resumidas em incorporações de colunas que são usadas para prever o tipo de coluna e a relação de colunas entre pares.

Steenwinckel, B. (STEENWINCKEL et al., 2019) apresentam o sistema CSV2KG que transforma dados tabulares em conhecimento semântico, tornando-os disponíveis na DBPedia. O objetivo é permitir que os dados sejam integrados e reutilizados em diferentes contextos, melhorando a interoperabilidade e a descoberta de informações.

O sistema proposto consiste em seis fases: Primeiro, anotações iniciais são atribuídas a cada célula da tabela considerada como entrada. Os candidatos passam por desambiguação por medidas de similaridade aplicadas ao rótulo de cada candidato. Em seguida, os tipos de colunas e as propriedades entre as colunas são inferidas usando as anotações iniciais. Na próxima etapa, os tipos e propriedades de coluna inferidos são usados para criar anotações de células de cabeçalho mais refinadas. Em seguida, usa as células de cabeçalho recém-geradas para corrigir outras células da tabela, usando anotações de propriedade. Concluindo, novos tipos de colunas são inferidos usando todas as células corrigidas disponíveis.

Também existem iniciativas que buscam informações em bases de dados da web, como no trabalho que Chabot, Yoan et al (CHABOT et al., 2019) apresentam o sistema DAGOBAH, que tem como objetivo realizar a anotação semântica de tabelas com entidades do Wikidata e do DBpedia. O sistema é capaz de realizar a anotação de células e colunas, bem como a identificação de relacionamentos entre elas.

O DAGOBAH é implementado como um conjunto de ferramentas que fornecem três funcionalidades principais: 1. Identificação de relacionamentos de mapeamento entre dados tabulares e grafos de conhecimento. 2. Enriquecimento de grafos de conhecimento por meio da transformação em triplas do conhecimento contido na tabela. 3. Produção de metadados que podem ser usados para processos de referência, busca e recomendação de conjuntos de dados.

Para fornecer esses recursos, o DAGOBAH é estruturado da seguinte maneira: 1. Os módulos de pré-processamento realizam a limpeza das tabelas e uma primeira caracterização de alto nível de sua forma e conteúdo. 2. Os módulos de vinculação de entidades realizam as tarefas do desafio em si, ou seja, a anotação de células-entidade (CEA), a anotação de tipo de coluna (CTA) e a anotação de propriedade de coluna (CPA). 3. Os módulos de pós-processamento realizam a agregação de resultados e a produção de metadados.

Alguns trabalhos também se utilizam do suporte de grafos para aprimorar o processo de anotação de tabelas, como o trabalho de Baazouzi, W., Kachroudi, M. e Faiz, S. (BAAZOUZI; KACHROUDI; FAIZ, 2022) que propõe uma abordagem

eficiente chamada Kepler-aSI utilizando FAIRification de dados tabulares com grafo de conhecimento. Nesse trabalho, FAIRification é o processo de tornar os dados encontráveis, acessíveis, interoperáveis e reutilizáveis.

Portanto, FAIRification refere-se ao processo de melhoria da qualidade dos dados tabulares, tornando-os mais facilmente detectáveis, acessíveis e utilizáveis. Os autores argumentam que pode ser difícil trabalhar com dados tabulares devido a lacunas e inconsistências semânticas, e que a FAIRificação pode ajudar a superar esses desafios. Ao usar modelos de grafo de conhecimento, foram capazes de anotar dados tabulares com recursos do grafo de conhecimento, o que pode ajudar a superar ambiguidades semânticas e melhorar a localização e interoperabilidade dos dados.

A abordagem dos autores combina serviços de pesquisa e filtro com técnicas de pré-processamento de texto para anotar dados tabulares de forma eficiente com recursos do grafo de conhecimento.

Também na linha da utilização de grafos como suporte ao processo de anotação, Jiomekong, A. e Foko, B. (JIOMEKONG; FOKO, 2022) apresentam uma abordagem baseada em refinamento de grafos de conhecimento para a correspondência entre dados tabulares e grafos de conhecimento, com uma solução que usa métodos internos e externos para prever links entre células em uma tabela e entidades e relações ausentes. Métodos internos são usados para prever links entre células na tabela, enquanto métodos externos são usados para prever entidades e relações ausentes.

Cada arquivo tabular é representado como um grafo em que cada célula é um nó, rotulado pelo conteúdo da célula e pode ser vinculado a outra célula ou título de coluna. O objetivo é corrigir o erro ortográfico do rótulo da célula, vincular a célula à sua anotação correspondente no KG e determinar o tipo de conjunto de células e a relação entre as células. Essa abordagem foi aplicada à anotação de várias tabelas usando DBpedia, Wikidata e Schema.org.

Ainda na linha do suporte de grafos para aprimorar o processo de anotação de tabelas, Gottschalk, S. e Demidova, E. (GOTTSCHALK; DEMIDOVA, 2022) apresentam uma abordagem chamada Tab2KG, que é um método para interpretação semântica de tabelas. o Tab2KG transforma tabelas em grafos de dados semânticos. O método utiliza perfis semânticos leves para interpretar tabelas e pode generalizar para novas ontologias de domínio com uma abordagem de aprendizado de uma única vez.

Tyagi, S. e Jiménez-Ruiz, E. [Tyagi, S. e Jiménez-Ruiz, E., 2020] propõe o sistema LexMa, que permite a anotação semântica automática de dados tabulares usando um grafo de conhecimento. O objetivo é permitir o acesso semântico às informações em dados tabulares, o que é um desafio fundamental na criação de

sistemas de busca eficientes. O LexMa usa técnicas de correspondência lexical para realizar essa anotação e pode ser aplicado em várias áreas, como a recuperação de informações e a integração de dados.

(4) Enriquecimento para construção de grafo de conhecimento (*knowledge graph* - KG)

A construção de grafos de conhecimento a partir de conjunto de dados tabulares pressupõe a identificação de entidades e relacionamentos nesses dados. Vimos anteriormente alguns trabalhos que contribuem para o enriquecimento de conjuntos de dados que possibilitam a identificação de entidades. A identificação da relação entre as instâncias recebe um tratamento em separado por uma técnica conhecida como extração de relação (Relation Extraction, em inglês).

A extração de relações é a tarefa de estabelecer a associação entre um par de colunas em uma tabela, identificando e representando a relação existente entre seus conteúdos. Esse processo também envolve a extração de informações relacionais a partir de conjuntos de dados tabulares, reconfigurando-as em um formato novo ou mais representativo (ZHANG; BALOG, 2020, p.15).

Porém, segundo Abdelmageed, N. (ABDELMAGEED, 2020), apresentam muitas limitações, como a sobrecarga das anotações manuais e são difíceis de estender. No trabalho de Bach, N. e Badaskar, S. (BACH; BADASKAR, 2007) e mais recentemente por Detroja, K. e Bhensdadia , C.K. e Bhatt, B. S. (DETROJA; BHENSDADIA; BHATT, 2023), foram pesquisadas diversas técnicas onde os autores utilizam uma abordagem de extração de relações como um método de aprendizagem supervisionado, mas sem a necessidade de rotular os dados (ABDELMAGEED, 2020). Ainda segundo Abdelmageed, N. (ABDELMAGEED, 2020), essa técnica ajuda a estender uma base de conhecimento existente, mas não é útil para construir uma do zero.

Mais recentemente Detroja, K. e Bhensdadia , C.K. e Bhatt, B. S. (DETROJA; BHENSDADIA; BHATT, 2023), efetuaram pesquisa onde foram levantadas diversas abordagens com métodos de supervisão supervisionada, semi-supervisionada, não-supervisionada e *Depp Learning* (DL), mas com foco predominante nas abordagens baseadas em DL.

Abdelmageed, N. (ABDELMAGEED, 2020) apresenta um framework baseado em aprendizado de máquina para integrar dados tabulares em um grafo de conhecimento com o objetivo de reduzir o esforço manual necessário para criar descrições ricas de dados e metadados, que são essenciais para alcançar a reutilização de dados.

As descrições desejadas são representadas como um grafo de conhecimento (KG). O framework proposto neste artigo consiste em três módulos principais: Predição de Conceitos, Detecção de Relações e Construção de Grafo de Conhecimento.

O módulo de Predição de Conceitos prevê os conceitos do grafo de conhecimento a partir de várias fontes de dados, incluindo cabeçalhos de colunas, células de tabelas e bases de conhecimento externas.

O módulo de Detecção de Relações extrai possíveis relações entre esses conceitos usando uma combinação de abordagens baseadas em regras e aprendizado de máquina.

Finalmente, o módulo de Construção de Grafo de Conhecimento integra os conceitos previstos e as relações extraídas para construir o grafo de conhecimento final. O trabalho é direcionado pelo relato de cientistas de dados envolvidos com o processo manual demorado que realizam para encontrar e integrar dados relevantes para suas pesquisas. O uso de um grafo de conhecimento bem descrito reduziria drasticamente o esforço necessário para realizar essa tarefa.

(5) Extração de Estatísticas de Grafo de Conhecimento (*knowledge graph*)

A extração de Estatísticas de Grafo de Conhecimento (*knowledge graph*), se utilizam de dados tabulares para construir grafo de conhecimento com o objetivo de extrair estatísticas desse grafo para enriquecimento dos dados tabulares originais. Nesses casos, o(s) conjunto(s) de dados tabular já possui(em) as características necessárias para a construção de seu grafo. Sanz, I. e Duarte, O. C. M. B. (SANZ; DUARTE, 2019) apresentam uma abordagem de detecção de intrusão em redes virtuais utilizando enriquecimento do conjunto de dados original com características baseado em grafos.

Os autores exploram o uso da teoria de grafos e técnicas de aprendizado de máquina para descobrir automaticamente padrões de comportamento de grupos de ameaças de rede e conclui com resultados experimentais que demonstram a eficácia da abordagem proposta. O grafo é construído a partir de dados de tráfego de rede com endereços IP como vértices e os dados transmitidos entre dois endereços IP como arestas direcionadas. A direção da aresta é definida como a origem apontando para o destino do fluxo.

Baumann, A. et al (BAUMANN et al., 2017), propõe uma abordagem alternativa para derivar grafo de sessões de usuário na web com base em dados de clickstream. Foram utilizados dois conjuntos de dados do mundo real de varejistas on-line. A abordagem proposta visa extrair informações auxiliares dos dados de clickstream, construindo grafo dentro da sessão e calculando métricas do grafo para enriquecer o conjunto de dados original, que podem então ser utilizadas para prever probabilidades de compra.

O grafo é construído com cada nó representando um site específico que um usuário visitou durante a sessão. Para cada visualização de página, cria-se um nó se ele ainda não existir no grafo da sessão. Uma aresta é criada entre dois nós

(entre páginas) se o usuário os visitar sucessivamente durante a sessão.

Gulum, M. A. (GULUM, 2018) explora o uso de características extraídas do grafo para prever resultados de corridas de cavalos. O autor coletou dados de corridas de cavalos e construiu um grafo direcionado para cada corrida, onde os nós representam cavalos e as arestas representam a ordem de chegada dos cavalos na corrida.

As estatísticas são extraídas do grafo e enriquecem o arquivo com os dados originais para entrada nos modelos de ML. O autor conclui que a utilização de características extraídas do grafo e algoritmos de aprendizado de máquina podem melhorar a precisão da previsão de resultados de corridas de cavalos.

Bryan J e Moriano, P (BRYAN; MORIANO, 2023) exploram o potencial do uso de grafo para capturar a complexidade do desenvolvimento de software e melhorar a previsão de defeitos em tempo real. Os autores propõem um modelo de aprendizado de máquina baseado em grafo que usa informações extraídas do grafo para prever defeitos em tempo real. Os autores construíram grafo bipartido que consiste em desenvolvedores e arquivos fonte para capturar a complexidade das mudanças necessárias para construir software.

A hipótese era que características extraídas do grafo podem ser melhores preditores de mudanças propensas a defeitos do que recursos intrínsecos derivados de características de software. No grafo os vértices representam desenvolvedores e arquivos fonte, sendo que dois vértices são conectados se dois desenvolvedores fizeram alterações nos mesmos arquivos.

Foram realizados experimentos em vários conjuntos de dados de software e descobriram que seu modelo superou os métodos tradicionais de previsão de defeitos em termos de precisão e eficiência.

Poenaru-Olaru, L. (POENARU-OLARU, 2018) apresenta uma abordagem para a previsão de pontuações de crédito. A técnica desenvolvida é baseada em uma abordagem de aprendizado de máquina que utiliza características extraídas de grafo para prever pontuações de crédito.

A técnica envolve a criação de um grafo que representa as relações entre os mutuários e os credores, e a extração de características do grafo. Os vértices do grafo representam as empresas, enquanto as arestas representam as transações financeiras entre elas. Essas características extraídas do grafo são então utilizadas como entrada para um modelo de aprendizado de máquina que é treinado para prever as pontuações de crédito.

Alharbi, A. e Alsubhi, K. (ALHARBI; ALSUBHI, 2021) propõe um modelo de detecção de botnets baseado em aprendizado de máquina (ML) que considera a importância das características de estatísticas extraídas do grafo para detectar botnets com base nas características selecionadas.

O artigo explora essas novas características baseadas em grafo para treinar várias técnicas de ML para detecção de botnets e emprega uma etapa abrangente de engenharia de características para selecionar o conjunto mais discriminativo de características para melhorar o desempenho e a escalabilidade do algoritmo de ML.

O grafo é construído a partir de conjuntos tabulares com fluxos de rede obtidos durante a etapa de inicialização de dados. Cada aresta do grafo é associada a dois vértices representados pelos endereços IP de hosts de origem e destino que constam como características no conjunto de dados original como "SrcAddr" e "DstAddr", respectivamente.

Para cada par de vértices conectados existem arestas direcionadas e os pesos das arestas são obtidos pelas características srcpktsx e dstpktsx que representam a quantidade de pacotes de origem e destino, respectivamente. O modelo é treinado e validado em dois conjuntos de dados reais de botnets e comparado com estado da arte de sistemas de detecção de botnets baseados em fluxo. Os resultados experimentais mostram que o modelo proposto supera os sistemas existentes.

(6) Extração de estatísticas de grafo de similaridade

A extração de estatísticas de grafo de similaridade, se utiliza de dados tabulares para construir grafo de similaridade de suas instâncias para construção do grafo com o objetivo de extrair estatísticas desse grafo para enriquecimento dos dados tabulares originais.

Nesses casos, não é verificado se o(s) conjunto(s) de dados tabular já possui(em) as características necessárias para a construção de seu grafo, utilizando técnicas que efetuam o relacionamento entre as instâncias com abordagens pela similaridade entre elas.

Em um trabalho focado em conjunto de dados para saúde, Zaki, N., Mohamed, E. e Habuza, T. (ZAKI; MOHAMED; HABUZA, 2021) apresentam uma técnica para melhorar o desempenho de modelos de classificação em dados de saúde, convertendo dados tabulados em grafo de conhecimento, que capturam correlações estruturais entre os dados.

Os autores utilizaram uma abordagem baseada em correlação, que envolveu a seleção de características com base na correlação entre os pares de instâncias. Em seguida, eles usaram essas características para construir um grafo de conhecimento, onde cada nó representa uma instância e cada aresta é determinada com base no produto escalar entre as características de cada par de instâncias e a sua conexão estabelecida mediante um limite calculado com base na média da similaridade entre as instâncias, evitando a utilização de pesos nessas arestas e, portanto, um grafo não ponderado. Essa estrutura de grafo foi utilizada para treinar modelos de classificação, que mostraram resultados superiores em comparação com métodos tradicionais.

Agora focado no campo da educação, Albreiki, B., Habuza, T. e Zaki, N. (ALBREIKI; HABUZA; ZAKI, 2023) apresentam um estudo sobre a extração de características topológicas de grafo para identificar estudantes em risco de desempenho acadêmico usando modelos de aprendizado de máquina (ML) e redes convolucionais de grafos (GCN).

A técnica parte de um conjunto de dados educacionais que consiste em informações sobre os alunos, como notas, frequência, histórico acadêmico e checkpoints. O conjunto de dados é representado em uma tabela, onde cada linha representa um aluno e cada coluna representa um atributo. Os atributos podem ser contínuos ou categóricos e são definidos com base nas informações disponíveis sobre os alunos. O objetivo do processo é identificar estudantes em risco de desempenho acadêmico o mais cedo possível, permitindo que as instituições educacionais tomem medidas preventivas para melhorar o sucesso dos alunos.

O conjunto de dados é pré-processado para extrair as informações relevantes e transformado em uma representação em grafo de conhecimento. Cada vértice do grafo representa um aluno e cada aresta representa uma relação entre dois alunos. As relações (arestas) entre os alunos foram definidas com base em dois conjuntos de atributos: checkpoints e características históricas.

Os checkpoints representam eventos importantes no curso, como testes e trabalhos, enquanto as características históricas representam o histórico acadêmico do aluno, como notas e frequência. As relações (arestas) entre os alunos foram estabelecidas com base na similaridade (*Chebyshev, Euclidean, Manhattan, Correlation* e *Cosine*) entre todos seus atributos juntos e a sua conexão estabelecida mediante um limite calculado com base na média da similaridade entre as instâncias, evitando a utilização de pesos nessas arestas e, portanto, um grafo não ponderado.

O grafo resultante é uma representação em rede dos dados educacionais, que é então utilizada para extrair características topológicas e prever o desempenho dos alunos em um modelo de aprendizado de máquina (ML) e uma rede convolucional de grafo (GCN), comparando os resultados em cada uma delas.

Em ambos os estudos, o grafo e a extração de suas estatísticas é efetuado com base em todo o conjunto de dados tabular e, em sequência, a técnica de validação cruzada foi empregada para o treinamento e teste dos modelos de aprendizado de máquina.

I
Detalhes Experimento

I.1
Conjuntos de Dados e Referências

I.1.1
Breast Cancer Coimbra (BCC)

Conjunto de dados relacionado ao câncer de mama disponível no repositório de aprendizado de máquina da UCI Machine Learning Repository (Universidade da Califórnia, Irvine) (PATRICIO et al., 2018), foi coletado na Universidade de Coimbra, Portugal, entre 2009 e 2013. Esse conjunto de dados contém informações médicas e clínicas sobre pacientes com câncer de mama e é frequentemente usado para tarefas de classificação, em particular para prever se um paciente tem câncer de mama maligno ou benigno com base em suas características médicas.

O objetivo principal deste conjunto de dados é prever se um paciente tem câncer de mama maligno ou benigno com base em suas características médicas. A importância desse conjunto de dados pode ser observada pelo estudo conduzido por Salod Z. e Singh Y. (SALOD; SINGH, 2020) em que efetuaram uma revisão sistemática e uma análise bibliométrica dos estudos publicados no período de 2015 a 2019 abordando a utilização de aprendizado de máquina para prever o câncer de mama.

Número de Instâncias: 116 pacientes

Número de Atributos: 9 atributos médicos e clínicos, mais a classe. Todos os atributos são numéricos não categóricos: A idade do paciente, índice de massa corporal do paciente, nível de glicose no sangue do paciente, nível de insulina no sangue do paciente, modelo de avaliação homeostática da Insulina (uma medida que avalia a resistência à insulina e a função das células beta), nível de leptina no sangue do paciente, nível de adiponectina no sangue do paciente, nível de resistina no sangue do paciente e o nível da proteína MCP-1 no sangue do paciente. A classe de saída, que indica se o paciente tem câncer de mama maligno (2) ou benigno (1). A distribuição da classe é de 64 (55%) pacientes com (2) e 52 (45%) pacientes com (1).

Como *benchmark* do nosso trabalho, para esse conjunto de dados, vamos nos basear no estudo conduzido por Naveen e Sharma, R. K. e Ramachandran Nair, Anil (NAVEEN; SHARMA; NAIR, 2019) em que foram empregados diversos modelos de predição, incluindo abordagens de aprendizado de máquina em conjunto (*ensemble*), para realizar previsões de câncer de mama utilizando o conjunto de

dados BCC. Este estudo obteve resultados notáveis, demonstrando o desempenho eficiente e eficaz desses modelos na tarefa de predição desse tipo de câncer.

A performance da Acurácia alcançada, em alguns dos modelos deste artigo, foi utilizada como base para comparação da performance dos modelos da técnica proposta nesse estudo, a saber: DT 66,7%, SVM 83,3%, RLOG 75,0%, RFOR 75,0%, KNN 89,9%, BAGG-KNN 100,0%, BAGG-DT 100,0%, BAGG-SVM 83,3%, BAGG-RLOG 91,7% e BAGG-RFOR 90,9%. Melhor Modelo: BAGG-KNN 100%.

Tamanho Conjuntos de Treino/Teste: 80%/20%

I.1.2 CAR Evaluation (CAR)

O conjunto de dados "CAR Evaluation"contém informações sobre a avaliação de carros. Esses dados foram coletados para auxiliar na classificação de carros com base em suas características. Disponibilizado no repositório da UCI (BOHANEC, 1997). O conjunto de dados de avaliação de carros contém um total de seis atributos, cada instância no conjunto de dados representa um carro, e o objetivo é classificar esses carros em uma das quatro categorias de avaliação de acordo com suas características.

Número de Instâncias: 1728 registros de carros.

Número de Atributos: 5 atributos, mais a variável classe. São informações como preço, custo de manutenção, número de portas e capacidade de passageiros e espaço de carga. Todos os atributos são nominais categóricos.

Classes: O conjunto de dados "CAR Evaluation"possui quatro classes de classificação "inaceitável", "aceitável", "bom"ou "muito bom", traduzidas da seguinte forma internamente: "unacc"(1210 instâncias – 70%), "acc"(384 instâncias – 22%), "good"(69 instâncias – 4%) ou "vgood"(65 – 4%).

Como *benchmark* do nosso trabalho, para esse conjunto de dados, vamos nos basear no estudo conduzido por Jalali, Vahid (JALALI; LEAKE; FOROUZANDEHMEHR, 2017), em que foi apresentada uma abordagem de aprendizado de máquina voltada para a classificação de dados por meio de um conjunto de regras de adaptação de casos. Os autores investigam a criação e aplicação de regras de adaptação de casos em cenários de classificação. A pesquisa avalia a eficácia desse método de ensemble em comparação com abordagens tradicionais de aprendizado de máquina. O conjunto de dados "CAR Evaluation"foi um dos conjuntos de dados utilizados na avaliação do desempenho dos modelos.

A performance da Acurácia alcançada, em alguns dos modelos deste artigo, foi utilizada como base para comparação da performance dos modelos da técnica proposta nesse estudo, a saber: RFOR 93,7%, KNN 93,5% e NBG 69,3%. Melhor Modelo EAC 96%

Proporção dos Conjuntos de Treino/Teste: 70%/30%

I.1.3 Dermatology (DM)

O conjunto de dados "Dermatology"é uma coleção de informações clínicas e diagnósticas relacionadas a doenças de pele, disponibilizado no repositório da UCI (ILTER; GUVENIR, 1998). Os dados foram coletados com o objetivo de auxiliar no diagnóstico de doenças de pele por meio de métodos computacionais. Esse conjunto de dados é frequentemente utilizado como um *benchmark* em tarefas de classificação médica, especialmente no contexto de dermatologia, como pode ser apurado pelo número de citações deste no repositório da UCI.

Número de Instâncias: 366 pacientes com suspeita de doenças de pele.

Número de Atributos: 34 atributos, mais a variável classe, incluindo informações demográficas do paciente e características observadas na área afetada pela doença. A variável classe é a doença de pele diagnosticada para cada paciente. Somente o atributo "Age" é numérico linear, sendo todos os demais numéricos categóricos (a idade também pode ser considerada categorica e a escolha depende do contexto e da análise). O atributo "Age" possui 8 instâncias com valores faltantes que foram substituídos pela mediana dos valores preenchidos.

Classes: O conjunto de dados "Dermatology" possui seis classes de doenças de pele diferentes, numeradas de 1 a 6, cada uma representando um tipo específico de condição dermatológica. Cada classe corresponde a doença de pele diagnosticada para cada paciente: 1 (112 instâncias - 31%), 2 (61 instâncias - 17%), 3 (72 instâncias - 20%), 4 (49 instâncias - 13%), 5 (52 instâncias - 14%) e 6 (20 instâncias - 5%).

Como *benchmark* do nosso trabalho, para esse conjunto de dados, vamos nos basear no estudo conduzido por Sharma, D.K. e Hota, H.S. (SHARMA; HOTA, 2013), que discute técnicas de mineração de dados para prever diferentes categorias de doenças dermatológicas. O artigo explora o uso de Máquina de Vetores de Suporte e Rede Neural Artificial, bem como um conjunto (ensemble) dessas duas técnicas, para classificação precisa de doenças Eritemato-escamosas. Os resultados mostram que o modelo ensemble obteve a maior precisão nas fases de treinamento e teste

A performance da Acurácia alcançada, em alguns dos modelos deste artigo, foi utilizada como base para comparação da performance dos modelos da técnica proposta nesse estudo, a saber: ANN 98%, SVM: 97%. Melhor Modelo Ensemble ANN/SVM: 99%

Proporção dos Conjuntos de Treino/Teste: 73%/27%

I.1.4
Heart Disease Hungarian (HDH)

O conjunto de dados "Heart Disease Hungarian"é uma coleção de informações para avaliação cardíaca, como idade, sexo, pressão arterial, níveis de colesterol, resultados de eletrocardiograma (ECG), frequência cardíaca máxima atingida durante o teste de esforço, entre outros, disponibilizado no repositório da UCI (JANOSI et al., 1988). Os dados foram coletados com o objetivo de auxiliar na previsão da presença ou ausência de doença cardíaca em pacientes com base em atributos médicos e clínicos.

Como pode ser apurado pelo número de citações deste no repositório da UCI, esse conjunto de dados é frequentemente utilizado para construir modelos de aprendizado de máquina que podem ajudar na detecção precoce de doenças cardíacas com base em informações médicas. No repositório da UCI constam três conjuntos de dados, Cleveland, Hungarian e Cleveland-Hungarian, sendo que para nosso estudo utilizaremos o conjunto de dados Hungarian.

Número de Instâncias: 294 pacientes com suspeita de doença cardíaca.

Número de Atributos: 13 atributos, mais a variável classe, que são tipo de dor no peito, pressão arterial em repouso, colesterol sérico, glicemia em jejum, resultados eletrocardiográficos em repouso, frequência cardíaca máxima alcançada, angina induzida por exercício, pico antigo - depressão ST induzida pelo exercício em relação ao repouso, a inclinação do pico do segmento ST do exercício, número de vasos principais e talassemia. São nove atributos categóricos e quatro numéricos.

Segundo informações constantes do repositório da UCI, esta base de dados inclui 76 atributos, mas todos os estudos publicados referem-se à utilização de um subconjunto de 14 deles. A variável classe é binária e indica a presença (1) ou não de doença cardíaca no paciente, distribuída em 188 (64.9%) pacientes com a presença de doença cardíaca (1) e 106 (35.1%) pacientes não (0).

Como *benchmark* do nosso trabalho, para esse conjunto de dados, vamos nos basear no estudo conduzido por Gárate-Escamila, A. et al. (GARATE-ESCAMILA; Hajjam El Hassani; ANDRES, 2020), em que foram desenvolvidos modelos de classificação para a previsão de doenças cardíacas, utilizando técnicas de seleção de características e Análise de Componentes Principais (PCA). Os autores empregaram métodos de seleção de características para identificar os atributos mais relevantes e informativos para o diagnóstico de doenças cardíacas. Além disso, eles utilizaram a técnica de PCA para redução da dimensionalidade dos dados, com o intuito de melhorar a eficiência e a interpretabilidade dos modelos de classificação.

A pesquisa utiliza técnicas de aprendizado de máquina e análise estatística para construir modelos capazes de prever a presença ou ausência de doenças cardía-

cas com base em dados médicos e clínicos. Os modelos propostos foram avaliados quanto à sua eficácia na detecção precoce de doenças cardíacas, contribuindo para um diagnóstico mais preciso e ágil. Embora o artigo trabalhe com referência aos três conjuntos de dados (Cleveland, Hungarian e Cleveland-Hungarian), utilizaremos apenas os dados referentes ao conjunto de dados Hungarian.

A performance da Acurácia alcançada, em alguns dos modelos deste artigo, foi utilizada como base para comparação da performance dos modelos da técnica proposta nesse estudo, a saber: DT 95,5%, RLOG 98,8%, NBB 82,8%, RFOR 99%, GB 98,8% e MLP 94%. Melhor Modelo: RFOR 99%

Proporção dos Conjuntos de Treino/Teste: 70%/30%

I.1.5
Heart Disease Hungarian Non-Invasive (HDHNI)

O conjunto de dados "Heart Disease Hungarian Non-Invasive" é um subconjunto do conjunto de dados original "Heart Disease Hungarian", contendo somente os atributos que foram obtidos de forma não invasiva. Nesse caso, somente os atributos 'EXANG' e "THAL" foram utilizados para os modelos e a variável classe original foi mantida.

Como *benchmark* do nosso trabalho, para esse conjunto de dados, vamos nos basear no mesmo estudo utilizado para o conjunto de dados "Heart Disease Hungarian", conduzido por Gárate-Escamila, A. et al. (GARATE-ESCAMILA; Hajjam El Hassani; ANDRES, 2020), que propôs a verificação da performance dos mesmos modelos para esse subconjunto de dados.

A performance da Acurácia alcançada, em alguns dos modelos deste artigo, foi utilizada como base para comparação da performance dos modelos da técnica proposta nesse estudo, a saber: DT 82,9%, RLOG 81,9%, NBB 78,7%, RFOR 81,3%, GB 82,3% e MLP 81,5%. Melhor Modelo: DT 82,9%.

Proporção dos Conjuntos de Treino/Teste: 70%/30%

I.1.6
Pima Indians Diabetes (PIMA)

O conjunto de dados "PIMA Indians Diabetes"contém informações demográficas e médicas sobre pacientes femininas com pelo menos 21 anos da tribo Pima Indian (BHOI, 2021), disponibilizado no repositório da Kaggle [Pima Indians Diabetes Database] e foi originalmente coletado pelo Instituto Nacional de Diabetes e Doenças Digestivas e Renais dos Estados Unidos [National Institute of Diabetes and Digestive and Kidney Diseases - NIDDK) com o objetivo de investigar a prevalência da diabetes entre essa população (BHOI, 2021).

Número de Instâncias: 768 pacientes.

Número de Atributos: 8 atributos, mais a variável classe, que são o número de gestações, concentração plasmática de glicose, pressão arterial diastólica, espessura da dobra da pele do tríceps, insulina, índice de massa corporal, função de pedigree de diabetes e idade (anos). Todos os atributos são numéricos (a idade também pode ser considerada categorica e a escolha depende do contexto e da análise). A variável classe é binária e indica a presença (1) ou não (0) de diabetes na paciente, distribuída em 268 (35%) pacientes com a presença de diabetes (classe 1) e 500 (65%) pacientes não (classe 0).

Como *benchmark* do nosso trabalho, para esse conjunto de dados, vamos nos basear no estudo conduzido por Chang, V. et al. (CHANG et al., 2022), que propõe um sistema de diagnóstico eletrônico baseado em IA para diabetes tipo 2 usando algoritmos de aprendizado de máquina. Os autores analisam o desempenho de três modelos de ML supervisionados interpretáveis (classificador Naive Bayes, classificador de floresta aleatório e árvore de decisão J48) e avaliam o processo de decisão para melhorar o modelo.

A performance da Acurácia alcançada, em alguns dos modelos deste artigo, foi utilizada como base para comparação da performance dos modelos da técnica proposta nesse estudo, a saber: DT 75,7%, RFOR 79,6%, NBG 77,8%. Melhor Modelo: RFOR 79,6%.

Proporção dos Conjuntos de Treino/Teste: 70

I.1.7 Vertebral Column (VCM)

O conjunto de dados "Vertebral Column"contém características obtidas de radiografias da coluna vertebral. As características incluem medidas específicas das vértebras e regiões da coluna vertebral, como ângulos e comprimentos. Cada entrada no conjunto de dados é uma instância correspondente a um paciente e suas respectivas medições, disponibilizado no repositório da UCI (University of California, Irvine) (BARRETO; NETO, 2011).

Segundo consta do repositório este conjunto de dados biomédicos foi construído pelo Dr. Henrique da Mota durante um período de residência médica no Grupo de Investigação Aplicada em Ortopedia (GARO) do Centro Médico-Chirúrgico de Réadaptation des Massues, Lyon, França.

Segundo Kayaalp, F. e Başarslan, M. (KAYAALP; BAŞARSLAN, 2019), cada paciente é representado no conjunto de dados por seis atributos biomecânicos derivados da forma e orientação da pelve e da coluna lombar. O objetivo principal ao usar esse conjunto de dados é criar modelos de classificação que possam prever com precisão a condição da coluna vertebral com base nas características radiográficas.

Número de Instâncias: 310 pacientes.

Número de Atributos: 6 atributos, mais a variável classe, que são incidência pélvica, inclinação pélvica, ângulo de lordose lombar, inclinação sacral, rádio pélvico e grau de espondilolistese. Todos os atributos são numéricos não categóricos. A variável classe indica pacientes ortopédicos em 3 classes, normal ("NO" com 100 instâncias - 32%), hérnia de disco (DH com 60 instâncias – 19%) ou espondilolistese ("SL" com 150 instâncias – 48%).

Como *benchmark* do nosso trabalho, para esse conjunto de dados, vamos nos basear no estudo conduzido por Reshi AA et al. (RESHI et al., 2021), que apresenta uma abordagem inovadora para o diagnóstico de patologias da coluna vertebral, utilizando técnicas de aprendizado de máquina e resampling concatenado. Os autores abordam o problema de desequilíbrio de classe, comum em conjuntos de dados médicos, e propõem o uso de resampling concatenado para mitigar esse desequilíbrio.

O estudo inclui a avaliação de diferentes algoritmos de aprendizado de máquina em relação ao diagnóstico de patologias da coluna vertebral. Foram desenvolvidas duas estratégias de resampling, com o resampling efetuado antes da separação dos conjuntos de treino e teste (1) e outra após (2).

Cada estratégia alcançou performance diferente e a performance da Acurácia alcançada, em alguns dos modelos deste artigo, foi utilizada como base para comparação da performance dos modelos da técnica proposta nesse estudo, a saber: Estratégia 1 (VCM1): RFOR 98%, ETREE 99%, RLOG 85%, ADAB 95% e SVM 80%. Melhor Modelo: ETREE 99%. Estratégia 2 (VCM2): RFOR 86%, ETREE 85%, RLOG 93%, ADAB 86% e SVM 90%. Melhor Modelo: RLOG 93%.

Proporção dos Conjuntos de Treino/Teste: 75%/25%

I.1.8 Vertebral Column Binary (VCB)

O conjunto de dados "Vertebral Column Binary"é exatamente o mesmo do conjunto "Vertebral Column" mas sua variável classe será binária, sendo aglutinados na classe normal todas as instâncias com "NO" (100 instâncias – 32%) e as demais em abnormal (210 instâncias – 68%).

Como *benchmark* do nosso trabalho, para esse conjunto de dados, vamos nos basear no estudo conduzido por Ansari, S. et al (ANSARI et al., 2013), que apresenta uma pesquisa que se concentra na aplicação de técnicas de aprendizado de máquina para o diagnóstico de distúrbios da coluna vertebral. Os autores investigam o uso de algoritmos de aprendizado de máquina, como classificadores, para analisar dados relacionados a distúrbios da coluna vertebral.

Eles abordam a questão de como esses métodos podem ser eficazes na identificação e classificação desses distúrbios com base em características e informações

específicas, incluindo sintomas, exames médicos e históricos clínicos. Os pesquisadores utilizam técnicas de aprendizado de máquina, incluindo Redes Neurais Artificiais, para treinar e testar o sistema.

A performance da Acurácia alcançada, em alguns dos modelos deste artigo, foi utilizada como base para comparação da performance dos modelos da técnica proposta nesse estudo, a saber: SVM 86,9%. Melhor Modelo: ANN 92,1%.

Proporção dos Conjuntos de Treino/Teste: 50%/50%

I.1.9 Wine (WM)

O conjunto de dados "Wine"contém informações sobre características químicas de vinhos. As características incluem várias propriedades químicas dos vinhos, como concentração de álcool, acidez, fenóis, flavonoides, intensidade de cor, entre outros.

Cada entrada no conjunto de dados é uma instância correspondente a um vinho e suas respectivas características. O conjunto de dados está disponibilizado no repositório da UCI (University of California, Irvine) (AEBERHARD; FORINA, 1991). Esse conjunto de dados é comumente utilizado para tarefas de classificação, onde o objetivo é prever a classe de um vinho com base em suas características químicas

Número de Instâncias: 178.

Número de Atributos: 13 atributos, mais a variável classe, que descrevem várias propriedades químicas dos vinhos. Todos os atributos são numéricos.

Classes: O conjunto de dados “Wine” possui três classes distintas de vinho, representadas pelos valores 1, 2 e 3. Cada classe corresponde a um tipo diferente de vinho: 1 (59 instâncias - 33%), 2 (71 instâncias - 40%) ou 3 (48 instâncias - 27%).

Como *benchmark* do nosso trabalho, para esse conjunto de dados, vamos nos basear no estudo conduzido por Ojha, V. et al. (OJHA; NICOSIA, 2020), que se concentra na otimização de árvores neurais que produzem várias saídas. Essas árvores são utilizadas em tarefas que envolvem previsão de múltiplas variáveis de saída, como em problemas de regressão multivariada ou classificação multirrótulo. Os autores propõem uma estratégia de otimização multiobjetivo para ajustar os parâmetros dessas árvores neurais.

Em vez de otimizar um único objetivo, o estudo busca otimizar múltiplos objetivos simultaneamente, considerando as diferentes saídas da árvore neural. Segundo os autores, a abordagem de otimização multiobjetivo permite encontrar soluções que equilibram os diferentes objetivos, o que é útil em problemas com múltiplas saídas.

A performance da Acurácia alcançada, em alguns dos modelos deste artigo, foi utilizada como base para comparação da performance dos modelos da técnica proposta nesse estudo, a saber: DT 74,6%, SMV 71,9%, NBG 97,3% e MLP 99,1&. Melhor Modelo: MONT 100%.

Proporção dos Conjuntos de Treino/Teste: 80%/20%

I.1.10 Wine (Wine Quality (WQ))

O conjunto de dados "Wine Quality" é composto por dois subconjuntos: "winequality-red"e "winequality-white". Ambos os subconjuntos contêm informações sobre características químicas de vinhos, com o objetivo de prever a qualidade do vinho com base nessas características. Vamos utilizar o conjunto "winequality-red". Cada entrada no conjunto de dados é uma instância correspondente a um vinho e suas respectivas características.

Para cada instância, há 11 atributos numéricos que descrevem várias propriedades químicas do vinho, como teor alcoólico, acidez volátil, ácido cítrico, açúcar residual, cloretos, dióxido de enxofre livre, dióxido de enxofre total, densidade, pH, sulfatos e qualidade (atributo de classe). A qualidade é um valor inteiro que varia de 3 a 8, indicando a qualidade percebida do vinho. O conjunto de dados está disponibilizado no repositório da UCI (University of California, Irvine) (CORTEZ et al., 2009).

Número de Instâncias: 1.599.

Número de Atributos: 11 atributos, mais a variável classe, que descrevem várias propriedades químicas dos vinhos. Todos os atributos são numéricos.

Classes: O conjunto de dados "Wine Quality" possui seis classes distintas de vinho, representadas pelos valores 3 a 8. Cada classe corresponde a qualidade alcançada por este vinho: 3 (10 instâncias - 1%), 4 (53 - 3%), 5 (681 instâncias - 43%), 6 (638 instâncias - 40%), 7 (199 instâncias -12%) e 8 (18 instâncias - 1%).

Como *benchmark* do nosso trabalho, para esse conjunto de dados, vamos nos basear no estudo conduzido por Kumar, S. e Agrawal, K. e Mandan, N., 2020 (KUMAR; AGRAWAL; MANDAN, 2020), apresenta aplicação de técnicas de aprendizado de máquina para prever a qualidade de vinhos tintos. Os autores aplicaram várias técnicas de *machine learning*, como regressão linear e árvores de decisão, para criar modelos preditivos. Eles também realizaram experimentos para avaliar o desempenho desses modelos em prever a qualidade dos vinhos.

A performance da Acurácia alcançada, em alguns dos modelos deste artigo, foi utilizada como base para comparação da performance dos modelos da técnica proposta nesse estudo, a saber: SVM 68,6%, NBG 55,9% e RFOR: 65,5%. Melhor Modelo: SVM 68,6%.

Proporção dos Conjuntos de Treino/Teste: 70%/30%

I.2 Bibliotecas Python

Pandas: A biblioteca Pandas fornece estruturas de dados e funções que permitiram a manipulação, análise e organização eficiente de conjuntos de dados. Essa biblioteca foi essencial para a preparação e limpeza dos dados, bem como para a extração de informações relevantes.

Numpy: O Numpy é uma biblioteca fundamental para computação científica em Python. Ele oferece suporte a arrays multidimensionais e funções matemáticas para realizar operações eficientes, o que se mostrou crucial para o processamento de dados numéricos.

Networkx: A biblioteca Networkx possibilitou a criação, manipulação e análise de grafos e redes. Foi utilizada para construir e explorar estruturas de grafo relacionadas a aspectos do experimento e, também, para extrair as estatísticas do grafo.

Scikit-learn (PEDREGOSA et al., 2011): O scikit-learn é uma das principais bibliotecas de aprendizado de máquina em Python. Diversas funções e classes do scikit-learn foram empregadas, incluindo:

- LabelEncoder: Utilizado para codificar variáveis categóricas em um formato numérico apropriado para os modelos.

- Train_test_split: Essa função foi aplicada para dividir o conjunto de dados em subconjuntos de treinamento e teste. O parâmetro test_size recebeu o tamanho desejado do conjunto de teste referente a cada *benchmark* associado ao conjunto de dados.

- Metrics: A biblioteca metrics contém diversas métricas de avaliação, como precisão, recall, F1-score, entre outras. Foram utilizadas as funções *accuracy_score , precision_score, recall_score* e *f1_score*, com o parâmetro *average="weighted"*.

- Make_score: Foi necessária para criar uma função de pontuação personalizada para ser utilizada na validação cruzada do GridSearchCV. Parâmetros: (precision_score/recall_score/f1_score, average='weighted').

- StandardScaler: Transforma os valores de um conjunto de dados para que tenham média zero e desvio padrão unitário. Essa função foi utilizada na padronização dos dados.

- GridSearchCV: Realiza a busca em grade (*grid search*) e validação cruzada ao mesmo tempo, a fim de encontrar a combinação ideal de hiperparâmetros para um modelo de aprendizado de máquina. Parâmetros utilizados: param_grid: Vide cada Modelo a seguir, scoring=lista de medidas pela função make_score, cv=10.

- Pairwise_distance: é utilizada para calcular matrizes de distâncias entre as amostras em um conjunto de dados. Ela aceita várias métricas de distância, o que permite a escolha da métrica mais apropriada para sua aplicação. As métricas de distância que utilizamos através desta foram *Jaccard, Euclidean, Cosine* e *Manhattan*.

- Scipy.spatial.distance: A biblioteca SciPy é utilizada para cálculos científicos e estatísticos. No nosso experimento foram utilizadas as scipy.spatial.distance e scipy.stats. A scipy.spatial.distance é utilizada para calcular a distância entre duas amostras de dados. No nosso experimento foi utilizada a função de distância *Mahalanobis*. A scipy.stats é utilizada para cálculos estatísticos. No nosso experimento foi utilizada a função *mannwhitneyu* para teste de significância de médias.

MinMaxScaler: Transforma os valores de um conjunto de dados para que estejam entre um intervalo especificado, no nosso caso, entre [0,1]. Utilizada na normalização dos valores de distância.

I.3 Parâmetros e Hiperparâmetros

- MLPClassifier (MLP): Classificador baseado em redes neurais artificiais, com camadas ocultas para aprendizado profundo. Hiperparâmetros utilizados no grid: activation=['identity', 'logistic', 'tanh', 'relu'], learning_rate=['adaptive', 'constant', 'invscaling'], solver=['lbfgs', 'sgd', 'adam'].

- LogisticRegression (RLOG): Modelo de regressão logística usado para tarefas de classificação binária e multiclasse. Hiperparâmetros utilizados no grid: solver=['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'], c_values=[0.01, 0.1, 1].

- ExtraTreesClassifier (ETREE): Classificador de árvore de decisão que forma várias árvores e combina suas previsões. Hiperparâmetros utilizados no grid: criterios=['gini','entropy'], max_depth=[15], max_features=['sqrt', 'log2', None]

- DecisionTreeClassifier (DT): Classificador de árvore de decisão que divide o espaço de características em regiões para classificar instâncias. Hiperparâmetros utilizados no grid: criterios=['gini','entropy'], max_depth=[None, 5, 10], min_samples_leaf=[1, 2, 3], splitter=['best', 'random'], max_features=['sqrt', 'log2', None]

- RandomForestClassifier (RFOR): Classificador de conjunto baseado em várias árvores de decisão. Hiperparâmetros utilizados no grid: n_estimators=[50, 100], criterion=['gini', 'entropy'], max_depth=[5, None], min_samples_leaf = [1, 5], max_features=['sqrt', 'log2'].

- KNeighborsClassifier (KNN): Classificador baseado na proximidade das amostras no espaço de características, classificando novas instâncias com base em seus vizinhos mais próximos. Hiperparâmetros utilizados no grid: metric=

["euclidean", "manhattan", "minkowski"], n_neighbors=[2, 3, 5, 7, 9, 11, 15], weights =['uniform', 'distance'], algorithm= ['auto', 'ball_tree', 'kd_tree', 'brute']

- GaussianNB (NBG): Classificador Naive Bayes que assume uma distribuição Gaussiana para as características. Sempre utilizado com os hiperparâmetros padrão.

- BernoulliNB (NBB): Classificador Naive Bayes que assume uma distribuição de Bernoulli para as características, adequado para dados binários. Hiperparâmetros utilizados no grid: fit_prior=[True, False].

- SVC (SVM): Máquinas de Vetores de Suporte. Classificador que encontra o hiperplano de decisão ótimo para separar classes. Hiperparâmetros utilizados no grid: c_values=[0.001, 0.1, 1.0, 10], kernel_values=['linear','poly', 'rbf'], gamma=['scale', 'auto'].

- GradientBoostingClassifier (GB): Classificador baseado no método de impulsionamento, constrói árvores de decisão sequencialmente, corrigindo os erros dos modelos anteriores. Hiperparâmetros utilizados no grid: loss=['log_loss', 'exponential'], criterion=['friedman_mse', 'mse'], min_samples_leaf=[1, 3, 5], max_depth=[3, 5, 10], max_features=['sqrt', 'log2'], learning_rate=[0.05, 0.01].

- AdaBoostClassifier (ADAB): Classificador de impulsionamento que combina vários classificadores fracos para criar um forte, focando nas instâncias mal classificadas. Hiperparâmetros utilizados no grid: n_estimators=[500], algorithm=['SAMME', 'SAMME.R'], learning_rate=[0.8].

- XGBClassifier (XGB): Classificador baseado em Gradient Boosting, otimizado para eficiência e desempenho. Hiperparâmetros utilizados no grid: max_depth=[3, 6, 10], n_estimators=[50, 100], learning_rate=[0.01, 0.1, 0.3, 0.4].

- BaggingClassifier (BAGG): Classificador que combina múltiplos classificadores de base para melhorar a generalização. Utilizado com os classificadores DT, KNN, RLOG, RFOR e SVM. Hiperparâmetros utilizados no grid: n_estimators=[100], bootstrap=[True, False],bootstrap_features=[True, False].

- LabelPropagation (LPA): Um modelo de aprendizado semi-supervisionado que propaga rótulos de instâncias rotuladas para instâncias não rotuladas com base em similaridades. Hiperparâmetros utilizados no grid: kernel=[knn, rbf], n_neighbors= [3, 5, 7, 9, 11].

J
Detalhes Resultados

J.1
Cenário 1

As Tabelas J.1, J.2, J.3, J.4 e J.5 apresentam os Ganhos/Perdas de Acurácia (%) em cada conjunto de dados aberto por modelo e Similaridade Categórica e Numérica. Os resultados estão abertos pela situação anterior (FO) e posterior (+FG) a inclusão das estatísticas do grafo no conjunto de dados original.

Tabela J.1: Cenário 1 – Ganho/Perda Conjunto de Dados com Somente Características Categóricas - Acurácia (%)

	Similaridade		**Eskin**		**Jaccard**		**Overlap**	
Dataset	**Modelo**	**FO**	**+FG**	**Dif.**	**+FG**	**Dif.**	**+FG**	**Dif.**
CAR	DT	86,9	86,9	0,0	86,9	0,0	86,9	0,0
	LPA	94,0	94,4	0,4	94,8	0,8	94,4	0,4
	NBG	75,1	84,0	8,9	84,2	9,1	84,0	8,9
	RFOR	97,5	95,8	-1,7	95,4	-2,1	95,6	-1,9
	RLOG	90,2	90,2	0,0	90,2	0,0	90,2	0,0
	SVM	95,4	94,6	-0,8	94,6	-0,8	94,6	-0,8
	XGB	98,8	96,1	-2,7	98,3	-0,5	96,1	-2,7
HDHNI	DT	85,4	85,4	0,0	85,4	0,0	85,4	0,0
	LPA	85,4	86,5	1,1	85,4	0,0	85,4	0,0
	NBG	85,4	85,4	0,0	85,4	0,0	85,4	0,0
	RFOR	85,4	87,6	2,2	85,4	0,0	85,4	0,0
	RLOG	85,4	85,4	0,0	85,4	0,0	85,4	0,0
	SVM	85,4	86,5	1,1	85,4	0,0	85,4	0,0
	XGB	85,4	85,4	0,0	85,4	0,0	85,4	0,0

Tabela J.2: Cenário 1 – Ganho/Perda Conjunto de Dados com Somente Características Numéricas - Acurácia (%)

	Similaridade	**Cosine**			**Euclidean**		**Mahalanobis**		**Manhattan**	
Dataset	**Modelo**	**FO**	**+FG**	**Dif.**	**+FG**	**Dif.**	**+FG**	**Dif.**	**+FG**	**Dif.**
	DT	66,7	83,3	16,6	79,2	12,5	87,5	20,8	79,2	12,5
	LPA	83,3	87,5	4,2	91,7	8,4	87,5	4,2	87,5	4,2
	NBG	62,5	79,2	16,7	87,5	25,0	87,5	25,0	87,5	25,0
BCC	RFOR	75,0	87,5	12,5	83,3	8,3	87,5	12,5	87,5	12,5
	RLOG	79,2	83,3	4,1	83,3	4,1	83,3	4,1	83,3	4,1
	SVM	75,0	83,3	8,3	83,3	8,3	91,7	16,7	83,3	8,3
	XGB	79,2	91,7	12,5	87,5	8,3	91,7	12,5	87,5	8,3
	DT	81,9	83,9	2,0	84,5	2,6	98,1	16,2	85,8	3,9
	LPA	85,2	87,1	1,9	88,4	3,2	94,8	9,6	86,5	1,3
	NBG	78,7	79,4	0,7	89,0	10,3	80,0	1,3	85,8	7,1
VCB	RFOR	83,9	85,2	1,3	89,0	5,1	98,7	14,8	87,7	3,8
	RLOG	84,5	86,5	2,0	88,4	3,9	95,5	11,0	90,3	5,8
	SVM	85,2	87,7	2,5	89,0	3,8	96,1	10,9	89,7	4,5
	XGB	85,2	85,2	0,0	89,0	3,8	98,7	13,5	86,5	1,3
	DT	80,8	83,3	2,5	92,3	11,5	98,7	17,9	92,3	11,5
	LPA	87,2	88,5	1,3	88,5	1,3	94,9	7,7	88,5	1,3
	NBG	80,8	89,7	8,9	82,1	1,3	93,6	12,8	83,3	2,5
VCM	RFOR	83,3	87,2	3,9	89,7	6,4	98,7	15,4	89,7	6,4
	RLOG	80,8	85,9	5,1	85,9	5,1	85,9	5,1	85,9	5,1
	SVM	80,8	84,6	3,8	85,9	5,1	91,0	10,2	84,6	3,8
	XGB	82,1	88,5	6,4	94,9	12,8	91,0	8,9	93,6	11,5
	DT	94,4	91,7	-2,7	91,7	-2,7	94,4	0,0	91,7	-2,7
	LPA	100,0	100,0	0,0	100,0	0,0	100,0	0,0	100,0	0,0
	NBG	97,2	100,0	2,8	100,0	2,8	100,0	2,8	100,0	2,8
WM	RFOR	100,0	100,0	0,0	100,0	0,0	100,0	0,0	100,0	0,0
	RLOG	97,2	100,0	2,8	97,2	0,0	97,2	0,0	97,2	0,0
	SVM	97,2	100,0	2,8	100,0	2,8	100,0	2,8	100,0	2,8
	XGB	97,2	100,0	2,8	100,0	2,8	100,0	2,8	100,0	2,8
	DT	56,5	56,9	0,4	57,1	0,6	57,1	0,6	56,9	0,4
	LPA	57,9	60,0	2,1	59,2	1,3	59,4	1,5	60,0	2,1
	NBG	55,4	57,5	2,1	56,9	1,5	56,7	1,3	56,2	0,8
WQ	RFOR	65,4	67,5	2,1	68,3	2,9	68,1	2,7	68,1	2,7
	RLOG	60,4	61,2	0,8	61,2	0,8	61,0	0,6	62,1	1,7
	SVM	61,5	62,3	0,8	62,1	0,6	62,3	0,8	62,5	1,0
	XGB	66,0	67,9	1,9	67,5	1,5	67,1	1,1	67,5	1,5

Tabela J.3: Cenário 1 – Ganho/Perda Conjunto de Dados com Categóricas e Numéricas - Eskin - Acurácia (%)

			Eskin							
	Similaridade		**Cosine**		**Euclidean**		**Mahalanobis**		**Manhattan**	
Dataset	**Modelo**	**FO**	**+FG**	**Dif.**	**+FG**	**Dif.**	**+FG**	**Dif.**	**+FG**	**Dif.**
DM	DT	90,6	90,6	0,0	90,6	0,0	90,6	0,0	90,6	0,0
	LPA	96,5	98,8	2,3	97,6	1,1	97,6	1,1	97,6	1,1
	NBG	91,8	91,8	0,0	91,8	0,0	91,8	0,0	91,8	0,0
	RFOR	96,5	97,6	1,1	97,6	1,1	98,8	2,3	97,6	1,1
	RLOG	96,5	97,6	1,1	97,6	1,1	97,6	1,1	97,6	1,1
	SVM	98,8	98,8	0,0	98,8	0,0	98,8	0,0	98,8	0,0
	XGB	94,1	95,3	1,2	95,3	1,2	95,3	1,2	95,3	1,2
HDH	DT	96,6	96,6	0,0	96,6	0,0	96,6	0,0	96,6	0,0
	LPA	92,1	92,1	0,0	92,1	0,0	93,3	1,2	92,1	0,0
	NBG	96,6	96,6	0,0	96,6	0,0	96,6	0,0	96,6	0,0
	RFOR	94,4	95,5	1,1	96,6	2,2	96,6	2,2	96,6	2,2
	RLOG	95,5	95,5	0,0	95,5	0,0	96,6	1,1	95,5	0,0
	SVM	95,5	96,6	1,1	96,6	1,1	96,6	1,1	96,6	1,1
	XGB	95,5	97,8	2,3	97,8	2,3	97,8	2,3	97,8	2,3
PIMA	DT	73,2	76,2	3,0	76,2	3,0	76,2	3,0	76,2	3,0
	LPA	69,7	75,8	6,1	75,3	5,6	76,2	6,5	74,9	5,2
	NBG	72,3	73,2	0,9	72,7	0,4	72,7	0,4	72,7	0,4
	RFOR	74,9	76,2	1,3	76,2	1,3	77,5	2,6	75,8	0,9
	RLOG	74,0	74,0	0,0	74,5	0,5	74,0	0,0	74,0	0,0
	SVM	74,5	76,2	1,7	76,2	1,7	76,2	1,7	76,2	1,7
	XGB	75,8	76,2	0,4	76,2	0,4	76,6	0,8	77,1	1,3

Tabela J.4: Cenário 1 – Ganho/Perda Conjunto de Dados com Categóricas e Numéricas - Jaccard - Acurácia (%)

	Similaridade		**Jaccard**							
			Cosine		**Euclidean**		**Mahalanobis**		**Manhattan**	
Dataset	**Modelo**	**FO**	**+FG**	**Dif.**	**+FG**	**Dif.**	**+FG**	**Dif.**	**+FG**	**Dif.**
DM	DT	90,6	90,6	0,0	90,6	0,0	90,6	0,0	90,6	0,0
	LPA	96,5	98,8	2,3	98,8	2,3	98,8	2,3	98,8	2,3
	NBG	91,8	91,8	0,0	91,8	0,0	92,9	1,1	91,8	0,0
	RFOR	96,5	97,6	1,1	97,6	1,1	97,6	1,1	97,6	1,1
	RLOG	96,5	98,8	2,3	97,6	1,1	97,6	1,1	97,6	1,1
	SVM	98,8	98,8	0,0	98,8	0,0	98,8	0,0	98,8	0,0
	XGB	94,1	95,3	1,2	95,3	1,2	95,3	1,2	95,3	1,2
HDH	DT	96,6	96,6	0,0	96,6	0,0	96,6	0,0	96,6	0,0
	LPA	92,1	93,3	1,2	93,3	1,2	92,1	0,0	93,3	1,2
	NBG	96,6	96,6	0,0	97,8	1,2	96,6	0,0	96,6	0,0
	RFOR	94,4	96,6	2,2	95,5	1,1	95,5	1,1	97,8	3,4
	RLOG	95,5	95,5	0,0	95,5	0,0	96,6	1,1	95,5	0,0
	SVM	95,5	95,5	0,0	96,6	1,1	96,6	1,1	96,6	1,1
	XGB	95,5	97,8	2,3	96,6	1,1	97,8	2,3	97,8	2,3
PIMA	DT	73,2	76,6	3,4	77,5	4,3	77,5	4,3	77,1	3,9
	LPA	69,7	74,5	4,8	75,3	5,6	74,5	4,8	74,5	4,8
	NBG	72,3	74,0	1,7	74,9	2,6	73,6	1,3	74,5	2,2
	RFOR	74,9	76,2	1,3	76,6	1,7	78,4	3,5	76,2	1,3
	RLOG	74,0	75,8	1,8	76,6	2,6	74,5	0,5	74,5	0,5
	SVM	74,5	75,8	1,3	76,6	2,1	76,2	1,7	76,2	1,7
	XGB	75,8	77,1	1,3	76,6	0,8	76,6	0,8	76,6	0,8

Tabela J.5: Cenário 1 – Ganho/Perda Conjunto de Dados com Categóricas e Numéricas - Overlap - Acurácia (%)

		Similaridade	Overlap							
			Cosine		Euclidean		Mahalanobis		Manhattan	
Dataset	**Modelo**	**FO**	**+FG**	**Dif.**	**+FG**	**Dif.**	**+FG**	**Dif.**	**+FG**	**Dif.**
DM	DT	90,6	90,6	0,0	90,6	0,0	90,6	0,0	90,6	0,0
	LPA	96,5	97,6	1,1	97,6	1,1	97,6	1,1	97,6	1,1
	NBG	91,8	91,8	0,0	91,8	0,0	91,8	0,0	91,8	0,0
	RFOR	96,5	98,8	2,3	98,8	2,3	98,8	2,3	98,8	2,3
	RLOG	96,5	97,6	1,1	97,6	1,1	97,6	1,1	97,6	1,1
	SVM	98,8	98,8	0,0	98,8	0,0	98,8	0,0	98,8	0,0
	XGB	94,1	94,1	0,0	94,1	0,0	94,1	0,0	94,1	0,0
HDH	DT	96,6	96,6	0,0	96,6	0,0	96,6	0,0	96,6	0,0
	LPA	92,1	92,1	0,0	92,1	0,0	93,3	1,2	92,1	0,0
	NBG	96,6	96,6	0,0	96,6	0,0	96,6	0,0	96,6	0,0
	RFOR	94,4	96,6	2,2	96,6	2,2	95,5	1,1	97,8	3,4
	RLOG	95,5	96,6	1,1	95,5	0,0	95,5	0,0	96,6	1,1
	SVM	95,5	96,6	1,1	95,5	0,0	96,6	1,1	96,6	1,1
	XGB	95,5	97,8	2,3	97,8	2,3	97,8	2,3	97,8	2,3
PIMA	DT	73,2	76,2	3,0	76,6	3,4	77,1	3,9	76,2	3,0
	LPA	69,7	74,9	5,2	74,5	4,8	75,8	6,1	74,9	5,2
	NBG	72,3	73,2	0,9	72,3	0,0	73,6	1,3	72,7	0,4
	RFOR	74,9	76,6	1,7	76,2	1,3	77,1	2,2	77,1	2,2
	RLOG	74,0	74,5	0,5	74,0	0,0	74,0	0,0	74,9	0,9
	SVM	74,5	76,2	1,7	76,2	1,7	76,6	2,1	76,6	2,1
	XGB	75,8	75,8	0,0	76,2	0,4	77,1	1,3	76,6	0,8

As Tabelas J.6, J.7 e J.8 apresentam os melhores resultados de Acurácia (%) em cada conjunto de dados aberto por modelo. Os valores percentuais das métricas de Precisão, Recall e F1 associadas também são apresentadas. Os resultados estão abertos pela situação anterior (FO) e posterior (+FG) a inclusão das estatísticas do grafo no conjunto de dados original. Adicionalmente, são apresentados os valores (*p-value*) de verificação se houve diferença significativa ($p<0,05$) entre os valores de Acurácia, utilizando o teste de *Mann Whitney U*.

Tabela J.6: Cenário 1 – Melhores Resultados por Dataset e Modelos (%)

Dataset	**Medida**		**DT**	**LPA**	**NBG**	**RFOR**	**RLOG**	**SVM**	**XGB**
BCC	Acurácia	FO	66,7	83,3	62,5	75,0	79,2	75,0	79,2
		+FG	87,5	91,7	87,5	87,5	83,3	91,7	91,7
	Precisão	FO	69,6	84,5	66,4	76,0	79,5	76,0	79,5
		+FG	89,8	92,9	87,8	87,7	84,5	91,7	91,7
	Recall	FO	66,7	83,3	62,5	75,0	79,2	75,0	79,2
		+FG	87,5	91,7	87,5	87,5	83,3	91,7	91,7
	F_1	FO	66,2	83,3	61,5	75,0	79,2	75,0	79,2
		+FG	87,2	91,7	87,5	87,4	83,3	91,7	91,7
		p-value	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001
CAR	Acurácia	FO	86,9	94,0	75,1	97,5	90,2	95,4	98,8
		+FG	86,9	94,8	84,2	95,8	90,2	94,6	98,3
	Precisão	FO	85,7	94,3	77,6	97,6	90,1	95,6	98,8
		+FG	85,7	95,1	80,6	96,2	90,1	95,0	98,3
	Recall	FO	86,9	94,0	75,1	97,5	90,2	95,4	98,8
		+FG	86,9	94,8	84,2	95,8	90,2	94,6	98,3
	F_1	FO	85,8	93,7	74,7	97,5	89,9	95,3	98,8
		+FG	85,8	94,6	82,3	95,2	89,9	94,5	98,2
		p-value	0,6046	0,3520	0,0001	0,9990	0,7531	0,9905	0,9202
DM	Acurácia	FO	90,6	96,5	91,8	96,5	96,5	98,8	94,1
		+FG	90,6	98,8	92,9	98,8	98,8	98,8	95,3
	Precisão	FO	92,4	97,2	93,6	96,6	97,2	98,9	94,6
		+FG	92,4	98,9	94,2	98,9	98,9	98,9	96,1
	Recall	FO	90,6	96,5	91,8	96,5	96,5	98,8	94,1
		+FG	90,6	98,8	92,9	98,8	98,8	98,8	95,3
	F_1	FO	90,8	96,5	91,4	96,5	96,5	98,8	94,1
		+FG	90,8	98,8	92,7	98,8	98,9	98,8	95,3
		p-value	0,7756	0,0011	0,0477	0,0012	0,0012	0,2834	0,0045
HDH	Acurácia	FO	96,6	92,1	96,6	94,4	95,5	95,5	95,5
		+FG	96,6	93,3	97,8	97,8	96,6	96,6	97,8
	Precisão	FO	96,8	92,5	96,7	94,4	95,5	95,7	95,5
		+FG	96,8	93,5	97,8	97,8	96,6	96,7	97,8
	Recall	FO	96,6	92,1	96,6	94,4	95,5	95,5	95,5
		+FG	96,6	93,3	97,8	97,8	96,6	96,6	97,8
	F_1	FO	96,6	92,0	96,6	94,4	95,5	95,5	95,5
		+FG	96,6	93,1	97,8	97,8	96,6	96,6	97,8
		p-value	0,6476	0,0008	0,0015	0,0006	0,0032	0,0153	0,0011

Tabela J.7: Cenário 1 – Melhores Resultados por Dataset e Modelos (%) - Continuação

Dataset	**Medida**		**DT**	**LPA**	**NBG**	**RFOR**	**RLOG**	**SVM**	**XGB**
HDHNI	Acurácia	FO	85,4	85,4	85,4	85,4	85,4	85,4	85,4
		+FG	85,4	86,5	85,4	87,6	85,4	86,5	85,4
	Precisão	FO	85,3	85,3	85,3	85,3	85,3	85,3	85,3
		+FG	85,3	86,4	85,3	87,7	85,3	86,4	85,3
	Recall	FO	85,4	85,4	85,4	85,4	85,4	85,4	85,4
		+FG	85,4	86,5	85,4	87,6	85,4	86,5	85,4
	F_1	FO	85,2	85,2	85,2	85,2	85,2	85,2	85,2
		+FG	85,2	86,4	85,2	87,4	85,2	86,4	85,2
		p-value	0,4100	0,0204	0,6758	0,0011	0,9010	0,0036	0,0068
PIMA	Acurácia	FO	73,2	69,7	72,3	74,9	74,0	74,5	75,8
		+FG	77,5	76,2	74,9	78,4	76,6	76,6	77,1
	Precisão	FO	72,5	68,7	72,1	74,2	73,3	74,4	75,2
		+FG	78,1	75,6	75,2	77,9	76,6	76,3	76,9
	Recall	FO	73,2	69,7	72,3	74,9	74,0	74,5	75,8
		+FG	77,5	76,2	74,9	78,4	76,6	76,6	77,1
	F_1	FO	72,7	68,9	72,2	74,0	73,3	72,3	75,4
		+FG	75,6	75,5	75,0	77,9	76,6	76,4	77,0
		p-value	0,0001	0,0001	0,0001	0,0012	0,0001	0,0011	0,0014
VCB	Acurácia	FO	81,9	85,2	78,7	83,9	84,5	85,2	85,2
		+FG	98,1	94,8	89,0	98,7	95,5	96,1	98,7
	Precisão	FO	83,0	86,3	85,6	84,2	84,7	86,0	85,7
		+FG	98,1	95,1	90,1	98,7	95,6	96,3	98,7
	Recall	FO	81,9	85,2	78,7	83,9	84,5	85,2	85,2
		+FG	98,1	94,8	89,0	98,7	95,5	96,1	98,7
	F_1	FO	82,2	85,4	79,4	84,0	84,6	85,4	85,3
		+FG	98,1	94,9	89,2	98,7	95,4	96,1	98,7
		p-value	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001
VCM	Acurácia	FO	80,8	87,2	80,8	83,3	80,8	80,8	82,1
		+FG	98,7	94,9	93,6	98,7	85,9	91,0	94,9
	Precisão	FO	80,9	87,7	81,5	82,6	80,4	80,7	81,0
		+FG	98,8	95,0	93,7	98,8	85,9	91,4	95,0
	Recall	FO	80,8	87,2	80,8	83,3	80,8	80,8	82,1
		+FG	98,7	94,9	93,6	98,7	85,9	91,0	94,9
	F_1	FO	80,6	86,9	80,5	82,8	80,6	80,7	81,4
		+FG	98,7	94,9	93,5	98,7	85,8	91,0	94,8
		p-value	0,0001	0,0001	0,0001	0,0001	0,0011	0,0001	0,0001

Tabela J.8: Cenário 1 – Melhores Resultados por Dataset e Modelos (%) - Continuação

Dataset	**Medida**		**DT**	**LPA**	**NBG**	**RFOR**	**RLOG**	**SVM**	**XGB**
WM	Acurácia	FO	94,4	100,0	97,2	100,0	97,2	97,2	97,2
		+FG	94,4	100,0	100,0	100,0	100,0	100,0	100,0
	Precisão	FO	95,1	100,0	97,4	100,0	97,4	97,4	97,4
		+FG	95,1	100,0	100,0	100,0	100,0	100,0	100,0
	Recall	FO	94,4	100,0	97,2	100,0	97,2	97,2	97,2
		+FG	94,4	100,0	100,0	100,0	100,0	100,0	100,0
	F_1	FO	94,5	100,0	97,2	100,0	97,2	97,2	97,2
		+FG	94,5	100,0	100,0	100,0	100,0	100,0	100,0
		p-value	0,9075	0,3547	0,0009	0,5530	0,0005	0,0001	0,0001
WQ	Acurácia	FO	56,5	57,9	55,4	65,4	60,4	61,5	66,0
		+FG	57,1	60,0	57,5	68,3	62,1	62,5	67,9
	Precisão	FO	52,9	58,5	56,9	63,0	58,3	58,5	63,4
		+FG	53,2	57,9	58,0	65,7	59,4	59,5	66,7
	Recall	FO	56,5	57,9	55,4	65,4	60,4	61,5	66,0
		+FG	57,1	60,0	57,5	68,3	62,1	62,5	67,9
	F_1	FO	53,6	58,0	56,1	63,5	57,8	58,5	64,4
		+FG	54,0	58,8	57,6	66,3	59,8	59,6	66,4
		p-value	0,0040	0,0014	0,0005	0,0011	0,0003	0,0405	0,0169

p-value: Teste *Mann Whitney U* para verificar diferença significativa (p<0,05) entre os valores de Acurácia.

As Tabelas J.9 e J.10 apresentam as medidas de Similaridade e Estatísticas que alcançaram os melhores resultados de Acurácia (%) por Dataset e Modelo. .

Tabela J.9: Cenário 1 – Melhores Similaridade e Estatísticas por Dataset e Modelos

Dataset	**Modelo**	**Similaridade**	**Estatística Grafo**
	DT	mahalanobis	FG
	LPA	euclidean	betweenness_centrality
	NBG	euclidean	triangles
BCC	RFOR	cosine	eigenvector_centrality
	RLOG	euclidean	closeness_centrality
	SVM	mahalanobis	FG
	XGB	cosine	pagerank
	DT	jaccard	FG
	LPA	jaccard	eigenvector_centrality
	NBG	jaccard	degree
CAR	RFOR	eskin	hits
	RLOG	jaccard	hits
	SVM	jaccard	pagerank
	XGB	jaccard	betweenness_centrality
	DT	jaccard / euclidean	FG
	LPA	jaccard / euclidean	FG
	NBG	jaccard / mahalanobis	FG
DM	RFOR	overlap / cosine	degree_centrality
	RLOG	jaccard / cosine	FG
	SVM	jaccard / euclidean	FG
	XGB	eskin / mahalanobis	average_neighbor_degree
	DT	jaccard / euclidean	FG
	LPA	jaccard / euclidean	degree
	NBG	jaccard / euclidean	load_centrality
HDH	RFOR	jaccard / manhattan	FG
	RLOG	jaccard / mahalanobis	pagerank
	SVM	jaccard / euclidean	triangles
	XGB	jaccard / cosine	FG
	DT	jaccard	eigenvector_centrality
	LPA	eskin	betweenness_centrality
	NBG	jaccard	hits
HDHNI	RFOR	eskin	hits
	RLOG	jaccard	pagerank
	SVM	eskin	load_centrality
	XGB	jaccard	pagerank

Tabela J.10: Cenário 1 – Melhores Similaridade e Estatísticas por Dataset e Modelos - Continuação

Dataset	**Modelo**	**Similaridade**	**Estatística Grafo**
	DT	jaccard / euclidean	triangles
	LPA	eskin / mahalanobis	hits
	NBG	jaccard / euclidean	FG
PIMA	RFOR	jaccard / mahalanobis	FG
	RLOG	jaccard / euclidean	FG
	SVM	jaccard / euclidean	FG
	XGB	jaccard / cosine	FG
	DT	mahalanobis	hits
	LPA	mahalanobis	hits
	NBG	euclidean	degree_centrality
VCB	RFOR	mahalanobis	hits
	RLOG	mahalanobis	hits
	SVM	mahalanobis	eigenvector_centrality
	XGB	mahalanobis	hits
	DT	mahalanobis	FG
	LPA	mahalanobis	degree
	NBG	mahalanobis	FG
VCM	RFOR	mahalanobis	pagerank
	RLOG	euclidean	clustering
	SVM	mahalanobis	triangles
	XGB	euclidean	FG
	DT	mahalanobis	FG
	LPA	euclidean	hits
	NBG	euclidean	average_neighbor_degree
WM	RFOR	euclidean	FG
	RLOG	cosine	betweenness_centrality
	SVM	euclidean	clustering
	XGB	euclidean	closeness_centrality
	DT	euclidean	closeness_centrality
	LPA	cosine	average_neighbor_degree
	NBG	cosine	degree
WQ	RFOR	euclidean	degree
	RLOG	manhattan	clustering
	SVM	manhattan	load_centrality
	XGB	cosine	load_centrality

J.2
Cenário 2

As Tabelas J.16, J.12, J.13, J.14 e J.15 apresentam os Ganhos/Perdas de Acurácia (%) em cada conjunto de dados aberto por modelo e Similaridade Categórica e Numérica. Os resultados estão abertos pela situação anterior (FO) e posterior (+FG) a inclusão das estatísticas do grafo no conjunto de dados original.

Tabela J.11: Cenário 2 – Ganho/Perda Conjunto de Dados com Somente Características Categóricas - Acurácia (%)

	Similaridade	**Eskin**			**Jaccard**		**Overlap**	
Dataset	**Modelo**	**FO**	**+FG**	**Dif.**	**+FG**	**Dif.**	**+FG**	**Dif.**
CAR	DT	95,6	96,0	0,4	96,5	0,9	96,0	0,4
	LPA	94,2	94,5	0,3	95,1	0,9	94,5	0,3
	NBG	75,1	82,9	7,8	82,1	7,0	82,7	7,6
	RFOR	96,9	95,2	-1,7	94,8	-2,1	95,4	-1,5
	RLOG	90,2	87,7	-2,5	90,2	0,0	87,7	-2,5
	SVM	98,8	96,7	-2,1	98,5	-0,3	96,7	-2,1
	XGB	98,5	95,4	-3,1	96,3	-2,2	95,4	-3,1
HDHNI	DT	85,4	85,4	0,0	85,4	0,0	86,5	1,1
	LPA	85,4	85,4	0,0	85,4	0,0	86,5	1,1
	NBG	85,4	86,5	1,1	85,4	0,0	85,4	0,0
	RFOR	85,4	85,4	0,0	85,4	0,0	86,5	1,1
	RLOG	85,4	84,3	-1,1	85,4	0,0	85,4	0,0
	SVM	85,4	85,4	0,0	85,4	0,0	85,4	0,0
	XGB	85,4	85,4	0,0	85,4	0,0	89,9	4,5

Tabela J.12: Cenário 2 – Ganho/Perda Conjunto de Dados com Somente Características Numéricas - Acurácia (%)

	Similaridade	**Cosine**			**Euclidean**		**Mahalanobis**		**Manhattan**	
Dataset	**Modelo**	**FO**	**+FG**	**Dif.**	**+FG**	**Dif.**	**+FG**	**Dif.**	**+FG**	**Dif.**
	DT	62,5	83,3	20,8	83,3	20,8	91,7	29,2	87,5	25,0
	LPA	83,3	87,5	4,2	91,7	8,4	87,5	4,2	87,5	4,2
	NBG	62,5	79,2	16,7	87,5	25,0	87,5	25,0	87,5	25,0
BCC	RFOR	79,2	91,7	12,5	83,3	4,1	87,5	8,3	87,5	8,3
	RLOG	75,0	79,2	4,2	83,3	8,3	83,3	8,3	83,3	8,3
	SVM	75,0	83,3	8,3	83,3	8,3	95,8	20,8	83,3	8,3
	XGB	79,2	91,7	12,5	87,5	8,3	95,8	16,6	91,7	12,5
	DT	81,9	83,9	2,0	83,2	1,3	98,1	16,2	83,9	2,0
	LPA	85,2	87,1	1,9	88,4	3,2	90,3	5,1	86,5	1,3
	NBG	78,7	78,7	0,0	78,7	0,0	78,7	0,0	78,7	0,0
VCB	RFOR	85,2	85,2	0,0	89,0	3,8	98,7	13,5	87,1	1,9
	RLOG	83,9	86,5	2,6	87,7	3,8	94,2	10,3	89,0	5,1
	SVM	84,5	86,5	2,0	88,4	3,9	96,1	11,6	89,0	4,5
	XGB	85,2	86,5	1,3	86,5	1,3	98,7	13,5	86,5	1,3
	DT	85,9	85,9	0,0	94,9	9,0	96,2	10,3	94,9	9,0
	LPA	87,2	88,5	1,3	88,5	1,3	94,9	7,7	88,5	1,3
	NBG	80,8	89,7	8,9	82,1	1,3	93,6	12,8	83,3	2,5
VCM	RFOR	83,3	84,6	1,3	89,7	6,4	97,4	14,1	89,7	6,4
	RLOG	82,1	85,9	3,8	87,2	5,1	85,9	3,8	85,9	3,8
	SVM	80,8	85,9	5,1	87,2	6,4	94,9	14,1	83,3	2,5
	XGB	82,1	87,2	5,1	93,6	11,5	92,3	10,2	93,6	11,5
	DT	94,4	94,4	0,0	94,4	0,0	97,2	2,8	94,4	0,0
	LPA	100,0	100,0	0,0	100,0	0,0	100,0	0,0	100,0	0,0
	NBG	97,2	100,0	2,8	100,0	2,8	100,0	2,8	100,0	2,8
WM	RFOR	100,0	100,0	0,0	100,0	0,0	100,0	0,0	100,0	0,0
	RLOG	100,0	100,0	0,0	100,0	0,0	100,0	0,0	100,0	0,0
	SVM	100,0	100,0	0,0	100,0	0,0	100,0	0,0	100,0	0,0
	XGB	100,0	100,0	0,0	100,0	0,0	100,0	0,0	100,0	0,0
	DT	57,5	59,0	1,5	59,6	2,1	59,6	2,1	59,6	2,1
	LPA	57,9	60,0	2,1	58,8	0,9	57,9	0,0	58,8	0,9
	NBG	55,4	57,3	1,9	56,5	1,1	56,7	1,3	57,1	1,7
WQ	RFOR	66,2	67,5	1,3	67,5	1,3	68,3	2,1	68,1	1,9
	RLOG	60,4	60,8	0,4	60,8	0,4	60,6	0,2	61,2	0,8
	SVM	64,0	65,8	1,8	65,0	1,0	65,4	1,4	65,0	1,0
	XGB	66,0	66,5	0,5	66,9	0,9	67,7	1,7	67,3	1,3

Tabela J.13: Cenário 2 – Ganho/Perda Conjunto de Dados com Categóricas e Numéricas - Eskin - Acurácia (%)

	Similaridade		**Eskin**							
			Cosine		**Euclidean**		**Mahalanobis**		**Manhattan**	
Dataset	**Modelo**	**FO**	**+FG**	**Dif.**	**+FG**	**Dif.**	**+FG**	**Dif.**	**+FG**	**Dif.**
DM	DT	95,3	94,1	-1,2	95,3	0,0	96,5	1,2	95,3	0,0
	LPA	96,5	98,8	2,3	97,6	1,1	97,6	1,1	97,6	1,1
	NBG	87,1	88,2	1,1	88,2	1,1	87,1	0,0	88,2	1,1
	RFOR	96,5	98,8	2,3	98,8	2,3	98,8	2,3	98,8	2,3
	RLOG	95,3	97,6	2,3	97,6	2,3	97,6	2,3	97,6	2,3
	SVM	97,6	98,8	1,2	98,8	1,2	98,8	1,2	98,8	1,2
	XGB	96,5	97,6	1,1	96,5	0,0	96,5	0,0	96,5	0,0
HDH	DT	96,6	96,6	0,0	96,6	0,0	96,6	0,0	97,8	1,2
	LPA	92,1	92,1	0,0	92,1	0,0	92,1	0,0	92,1	0,0
	NBG	96,6	96,6	0,0	96,6	0,0	96,6	0,0	96,6	0,0
	RFOR	93,3	96,6	3,3	93,3	0,0	98,9	5,6	96,6	3,3
	RLOG	95,5	95,5	0,0	95,5	0,0	96,6	1,1	95,5	0,0
	SVM	96,6	96,6	0,0	96,6	0,0	97,8	1,2	97,8	1,2
	XGB	97,8	97,8	0,0	97,8	0,0	97,8	0,0	97,8	0,0
PIMA	DT	73,2	76,6	3,4	76,2	3,0	75,8	2,6	76,2	3,0
	LPA	76,6	77,9	1,3	77,1	0,5	77,1	0,5	77,9	1,3
	NBG	72,3	73,2	0,9	72,7	0,4	72,3	0,0	72,7	0,4
	RFOR	76,2	77,1	0,9	78,4	2,2	77,1	0,9	77,5	1,3
	RLOG	74,0	74,9	0,9	74,9	0,9	74,9	0,9	74,9	0,9
	SVM	76,2	78,4	2,2	77,9	1,7	78,4	2,2	78,4	2,2
	XGB	73,2	74,9	1,7	75,8	2,6	76,2	3,0	75,3	2,1

Tabela J.14: Cenário 2 – Ganho/Perda Conjunto de Dados com Categóricas e Numéricas - Jaccard - Acurácia (%)

		Similaridade	**Jaccard**							
			Cosine		**Euclidean**		**Mahalanobis**		**Manhattan**	
Dataset	**Modelo**	**FO**	**+FG**	**Dif.**	**+FG**	**Dif.**	**+FG**	**Dif.**	**+FG**	**Dif.**
DM	DT	95,3	95,3	0,0	95,3	0,0	95,3	0,0	95,3	0,0
	LPA	96,5	98,8	2,3	98,8	2,3	98,8	2,3	98,8	2,3
	NBG	87,1	87,1	0,0	87,1	0,0	88,2	1,1	87,1	0,0
	RFOR	96,5	98,8	2,3	97,6	1,1	98,8	2,3	97,6	1,1
	RLOG	95,3	97,6	2,3	97,6	2,3	97,6	2,3	97,6	2,3
	SVM	97,6	98,8	1,2	98,8	1,2	98,8	1,2	98,8	1,2
	XGB	96,5	96,5	0,0	97,6	1,1	96,5	0,0	97,6	1,1
HDH	DT	96,6	97,8	1,2	96,6	0,0	98,9	2,3	96,6	0,0
	LPA	92,1	92,1	0,0	92,1	0,0	92,1	0,0	93,3	1,2
	NBG	96,6	96,6	0,0	97,8	1,2	96,6	0,0	96,6	0,0
	RFOR	93,3	96,6	3,3	93,3	0,0	96,6	3,3	96,6	3,3
	RLOG	95,5	95,5	0,0	95,5	0,0	96,6	1,1	95,5	0,0
	SVM	96,6	96,6	0,0	96,6	0,0	97,8	1,2	97,8	1,2
	XGB	97,8	97,8	0,0	97,8	0,0	97,8	0,0	97,8	0,0
PIMA	DT	73,2	77,1	3,9	77,1	3,9	76,6	3,4	77,1	3,9
	LPA	76,6	76,2	-0,4	77,9	1,3	76,2	-0,4	77,1	0,5
	NBG	72,3	74,0	1,7	74,9	2,6	73,6	1,3	74,0	1,7
	RFOR	76,2	77,9	1,7	76,6	0,4	77,1	0,9	77,1	0,9
	RLOG	74,0	75,8	1,8	76,6	2,6	74,0	0,0	74,5	0,5
	SVM	76,2	77,5	1,3	78,4	2,2	77,5	1,3	78,4	2,2
	XGB	73,2	74,9	1,7	76,2	3,0	74,9	1,7	75,8	2,6

Tabela J.15: Cenário 2 – Ganho/Perda Conjunto de Dados com Categóricas e Numéricas - Overlap - Acurácia (%)

		Similaridade	**Overlap**							
			Cosine		**Euclidean**		**Mahalanobis**		**Manhattan**	
Dataset	**Modelo**	**FO**	**+FG**	**Dif.**	**+FG**	**Dif.**	**+FG**	**Dif.**	**+FG**	**Dif.**
DM	DT	95,3	94,1	-1,2	95,3	0,0	96,5	1,2	95,3	0,0
	LPA	96,5	97,6	1,1	97,6	1,1	97,6	1,1	97,6	1,1
	NBG	87,1	88,2	1,1	87,1	0,0	88,2	1,1	87,1	0,0
	RFOR	96,5	98,8	2,3	98,8	2,3	98,8	2,3	98,8	2,3
	RLOG	95,3	97,6	2,3	97,6	2,3	97,6	2,3	97,6	2,3
	SVM	97,6	98,8	1,2	98,8	1,2	98,8	1,2	98,8	1,2
	XGB	96,5	96,5	0,0	96,5	0,0	96,5	0,0	96,5	0,0
HDH	DT	96,6	96,6	0,0	96,6	0,0	96,6	0,0	96,6	0,0
	LPA	92,1	92,1	0,0	92,1	0,0	92,1	0,0	92,1	0,0
	NBG	96,6	96,6	0,0	96,6	0,0	96,6	0,0	96,6	0,0
	RFOR	93,3	97,8	4,5	93,3	0,0	96,6	3,3	96,6	3,3
	RLOG	95,5	95,5	0,0	95,5	0,0	95,5	0,0	95,5	0,0
	SVM	96,6	96,6	0,0	96,6	0,0	96,6	0,0	96,6	0,0
	XGB	97,8	97,8	0,0	97,8	0,0	97,8	0,0	97,8	0,0
PIMA	DT	73,2	76,2	3,0	76,2	3,0	76,2	3,0	76,6	3,4
	LPA	76,6	77,5	0,9	77,1	0,5	77,5	0,9	77,1	0,5
	NBG	72,3	73,2	0,9	72,3	0,0	73,6	1,3	72,7	0,4
	RFOR	76,2	78,4	2,2	77,1	0,9	77,1	0,9	77,9	1,7
	RLOG	74,0	74,9	0,9	74,9	0,9	74,0	0,0	74,9	0,9
	SVM	76,2	78,4	2,2	78,4	2,2	78,4	2,2	78,4	2,2
	XGB	73,2	74,9	1,7	76,2	3,0	75,3	2,1	74,9	1,7

As Tabelas J.16, J.17 e J.18 apresentam os melhores resultados de Acurácia (%) em cada conjunto de dados aberto por modelo. Os valores percentuais das métricas de Precisão, Recall e F1 associadas também são apresentadas. Os resultados estão abertos pela situação anterior (FO) e posterior (+FG) a inclusão das estatísticas do grafo no conjunto de dados original. Adicionalmente, são apresentados os valores (*p-value*) de verificação se houve diferença significativa ($p<0,05$) entre os valores de Acurácia, utilizando o teste de *Mann Whitney U*.

Tabela J.16: Cenário 2 – Melhores Resultados por Dataset e Modelos (%)

Dataset	**Medida**		**DT**	**LPA**	**NBG**	**RFOR**	**RLOG**	**SVM**	**XGB**
BCC	Acurácia	FO	62,5	83,3	62,5	79,2	75,0	75,0	79,2
		+FG	91,7	91,7	87,5	91,7	83,3	95,8	95,8
	Precisão	FO	64,1	84,5	66,4	79,5	76,0	76,0	79,5
		+FG	92,9	92,9	87,8	91,7	84,5	96,1	96,1
	Recall	FO	62,5	83,3	62,5	79,2	75,0	75,0	79,2
		+FG	91,7	91,7	87,5	91,7	83,3	95,8	95,8
	F_1	FO	62,3	83,3	61,5	79,2	75,0	75,0	79,2
		+FG	91,7	91,7	87,5	91,7	83,3	95,8	95,8
		p-value	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001
CAR	Acurácia	FO	95,6	94,2	75,1	96,9	90,2	98,8	98,5
		+FG	96,5	95,1	82,9	95,4	90,2	98,5	96,3
	Precisão	FO	95,5	94,4	77,6	97,0	90,1	98,8	98,5
		+FG	96,6	95,2	80,7	96,0	90,5	98,5	96,6
	Recall	FO	95,6	94,2	75,1	96,9	90,2	98,8	98,5
		+FG	96,5	95,1	82,9	95,4	90,2	98,5	96,3
	F_1	FO	95,4	94,0	74,7	96,9	89,9	98,8	98,4
		+FG	96,4	94,9	81,5	94,6	89,9	98,4	95,7
		p-value	0,0036	0,0049	0,0001	0,9814	0,7531	0,6615	0,9991
DM	Acurácia	FO	95,3	96,5	91,8	96,5	96,5	97,6	96,5
		+FG	96,5	98,8	91,8	98,8	97,6	98,8	97,6
	Precisão	FO	95,5	97,2	93,6	96,5	97,2	97,6	96,6
		+FG	96,9	98,9	93,6	98,9	97,6	98,9	98,0
	Recall	FO	95,3	96,5	91,8	96,5	96,5	97,6	96,5
		+FG	96,5	98,8	91,8	98,8	97,6	98,8	97,6
	F_1	FO	95,2	96,5	91,4	96,4	96,5	97,6	96,5
		+FG	96,5	98,8	91,4	98,8	97,6	98,8	97,7
		p-value	0,0225	0,0011	0,8295	0,0012	0,0023	0,0031	0,0095
HDH	Acurácia	FO	96,6	92,1	96,6	93,3	95,5	96,6	97,8
		+FG	98,9	93,3	97,8	98,9	96,6	97,8	97,8
	Precisão	FO	96,8	92,5	96,7	93,3	95,5	96,6	97,8
		+FG	98,9	93,5	97,8	98,9	96,6	97,8	97,8
	Recall	FO	96,6	92,1	96,6	93,3	95,5	96,6	97,8
		+FG	98,9	93,3	97,8	98,9	96,6	97,8	97,8
	F_1	FO	96,6	92,0	96,6	93,2	95,5	96,6	97,8
		+FG	98,9	93,1	97,8	98,9	96,6	97,8	97,8
		p-value	0,0012	0,0008	0,0015	0,0001	0,0032	0,0168	0,4397

Tabela J.17: Cenário 2 – Melhores Resultados por Dataset e Modelos (%) - Continuação

Dataset	**Medida**		**DT**	**LPA**	**NBG**	**RFOR**	**RLOG**	**SVM**	**XGB**
HDHNI	Acurácia	FO	85,4	85,4	85,4	85,4	85,4	85,4	85,4
		+FG	86,5	86,5	86,5	86,5	85,4	85,4	89,9
	Precisão	FO	85,3	85,3	85,3	85,3	85,3	85,3	85,3
		+FG	86,5	86,5	86,4	86,5	85,3	85,3	89,8
	Recall	FO	85,4	85,4	85,4	85,4	85,4	85,4	85,4
		+FG	86,5	86,5	86,5	86,5	85,4	85,4	89,9
	F_1	FO	85,2	85,2	85,2	85,2	85,2	85,2	85,2
		+FG	86,3	86,3	86,4	86,3	85,2	85,2	89,9
		p-value	0,0025	0,0204	0,0012	0,0036	0,9010	0,2846	0,0011
PIMA	Acurácia	FO	73,2	76,6	72,3	76,2	74,0	76,2	73,2
		+FG	77,1	77,9	74,9	78,4	76,6	78,4	76,2
	Precisão	FO	72,5	76,8	72,1	75,6	73,3	75,6	72,4
		+FG	76,5	77,6	75,2	77,9	76,6	78,0	75,6
	Recall	FO	73,2	76,6	72,3	76,2	74,0	76,2	73,2
		+FG	77,1	77,9	74,9	78,4	76,6	78,4	76,2
	F_1	FO	72,7	74,9	72,2	75,3	73,3	75,6	72,5
		+FG	76,3	77,0	75,0	78,0	76,6	77,5	75,3
		p-value	0,0001	0,0202	0,0001	0,0001	0,0001	0,0008	0,0001
VCB	Acurácia	FO	81,9	85,2	78,7	85,2	83,9	84,5	85,2
		+FG	98,1	90,3	78,7	98,7	94,2	96,1	98,7
	Precisão	FO	82,1	86,3	85,6	85,4	84,0	85,5	85,2
		+FG	98,1	90,3	85,6	98,7	94,2	96,3	98,7
	Recall	FO	81,9	85,2	78,7	85,2	83,9	84,5	85,2
		+FG	98,1	90,3	78,7	98,7	94,2	96,1	98,7
	F_1	FO	82,0	85,4	79,4	85,3	83,9	84,8	85,2
		+FG	98,1	90,3	79,4	98,7	94,2	96,1	98,7
		p-value	0,0001	0,0001	0,4099	0,0001	0,0001	0,0001	0,0001
VCM	Acurácia	FO	85,9	87,2	80,8	83,3	82,1	80,8	82,1
		+FG	96,2	94,9	93,6	97,4	87,2	94,9	93,6
	Precisão	FO	85,4	87,7	81,5	82,5	82,1	80,7	81,0
		+FG	96,4	95,0	93,9	97,6	87,9	96,0	93,9
	Recall	FO	85,9	87,2	80,8	83,3	82,1	80,8	82,1
		+FG	96,2	94,9	93,6	97,4	87,2	94,9	93,6
	F_1	FO	85,5	86,9	80,5	82,7	81,9	80,7	81,4
		+FG	96,1	94,9	93,6	97,4	87,5	95,0	93,5
		p-value	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001

Tabela J.18: Cenário 2 – Melhores Resultados por Dataset e Modelos (%) - Continuação

Dataset	**Medida**		**DT**	**LPA**	**NBG**	**RFOR**	**RLOG**	**SVM**	**XGB**
WM	Acurácia	FO	94,4	100,0	97,2	100,0	100,0	100,0	100,0
		+FG	97,2	100,0	100,0	100,0	100,0	100,0	100,0
	Precisão	FO	95,1	100,0	97,4	100,0	100,0	100,0	100,0
		+FG	97,4	100,0	100,0	100,0	100,0	100,0	100,0
	Recall	FO	94,4	100,0	97,2	100,0	100,0	100,0	100,0
		+FG	97,2	100,0	100,0	100,0	100,0	100,0	100,0
	F_1	FO	94,5	100,0	97,2	100,0	100,0	100,0	100,0
		+FG	97,2	100,0	100,0	100,0	100,0	100,0	100,0
		p-value	0,0001	0,3547	0,0009	0,5530	0,4040	0,8684	0,0994
WQ	Acurácia	FO	57,5	57,9	55,4	66,2	60,4	64,0	66,0
		+FG	61,7	60,0	57,3	69,0	62,1	65,8	67,7
	Precisão	FO	57,6	58,5	56,9	63,8	58,3	61,4	64,1
		+FG	60,3	57,9	58,6	67,2	59,7	63,6	64,9
	Recall	FO	57,5	57,9	55,4	66,2	60,4	64,0	66,0
		+FG	61,7	60,0	57,3	69,0	62,1	65,8	67,7
	F_1	FO	57,5	58,0	56,1	64,4	57,8	62,5	64,2
		+FG	60,7	58,8	57,9	66,8	59,9	64,4	65,7
		p-value	0,0011	0,0014	0,0005	0,0001	0,0003	0,0076	0,0169

p-value: Teste *Mann Whitney U* para verificar diferença significativa (p<0,05) entre os valores de Acurácia.

As Tabelas J.19 e J.20 apresentam as medidas de Similaridade e Estatísticas que alcançaram os melhores resultados de Acurácia (%) por Dataset e Modelo. .

Tabela J.19: Cenário 2 – Melhores Similaridade e Estatísticas por Dataset e Modelos

Dataset	**Modelo**	**Similaridade**	**Estatística Grafo**
	DT	mahalanobis	degree
	LPA	euclidean	betweenness_centrality
	NBG	euclidean	triangles
BCC	RFOR	cosine	FG
	RLOG	euclidean	degree_centrality
	SVM	mahalanobis	FG
	XGB	mahalanobis	clustering
	DT	jaccard	triangles
	LPA	jaccard	eigenvector_centrality
	NBG	overlap	triangles
CAR	RFOR	overlap	hits
	RLOG	jaccard	degree_centrality
	SVM	jaccard	pagerank
	XGB	jaccard	triangles
	DT	overlap / mahalanobis	FG
	LPA	jaccard / euclidean	FG
	NBG	jaccard / mahalanobis	FG
DM	RFOR	jaccard / cosine	load_centrality
	RLOG	jaccard / euclidean	FG
	SVM	jaccard / euclidean	FG
	XGB	jaccard / euclidean	FG
	DT	jaccard / mahalanobis	load_centrality
	LPA	jaccard / manhattan	triangles
	NBG	jaccard / euclidean	betweenness_centrality
HDH	RFOR	eskin / mahalanobis	FG
	RLOG	jaccard / mahalanobis	degree_centrality
	SVM	jaccard / manhattan	FG
	XGB	overlap / cosine	FG
	DT	overlap	hits
	LPA	overlap	triangles
	NBG	eskin	betweenness_centrality
HDHNI	RFOR	overlap	FG
	RLOG	jaccard	load_centrality
	SVM	jaccard	load_centrality
	XGB	overlap	FG

Tabela J.20: Cenário 1 – Melhores Similaridade e Estatísticas por Dataset e Modelos - Continuação

Dataset	**Modelo**	**Similaridade**	**Estatística Grafo**
	DT	jaccard / euclidean	degree
	LPA	jaccard / euclidean	hits
	NBG	jaccard / euclidean	FG
PIMA	RFOR	eskin / euclidean	degree
	RLOG	jaccard / euclidean	FG
	SVM	jaccard / euclidean	hits
	XGB	jaccard / euclidean	clustering
	DT	mahalanobis	hits
	LPA	mahalanobis	degree
	NBG	euclidean	load_centrality
VCB	RFOR	mahalanobis	hits
	RLOG	mahalanobis	hits
	SVM	mahalanobis	eigenvector_centrality
	XGB	mahalanobis	hits
	DT	mahalanobis	degree_centrality
	LPA	mahalanobis	degree
	NBG	mahalanobis	eigenvector_centrality
VCM	RFOR	mahalanobis	pagerank
	RLOG	euclidean	hits
	SVM	mahalanobis	degree
	XGB	manhattan	FG
	DT	mahalanobis	clustering
	LPA	cosine	betweenness_centrality
	NBG	cosine	betweenness_centrality
WM	RFOR	cosine	FG
	RLOG	cosine	betweenness_centrality
	SVM	cosine	closeness_centrality
	XGB	cosine	FG
	DT	euclidean	eigenvector_centrality
	LPA	cosine	average_neighbor_degree
	NBG	manhattan	triangles
WQ	RFOR	mahalanobis	closeness_centrality
	RLOG	manhattan	FG
	SVM	cosine	degree_centrality
	XGB	mahalanobis	FG

J.3
Cenário 1 e Cenário 2

O gráfico na Figura J.1 apresenta as médias de Ganhos/Perdas de Acurácia (%) por conjunto de dados em cada Cenário.

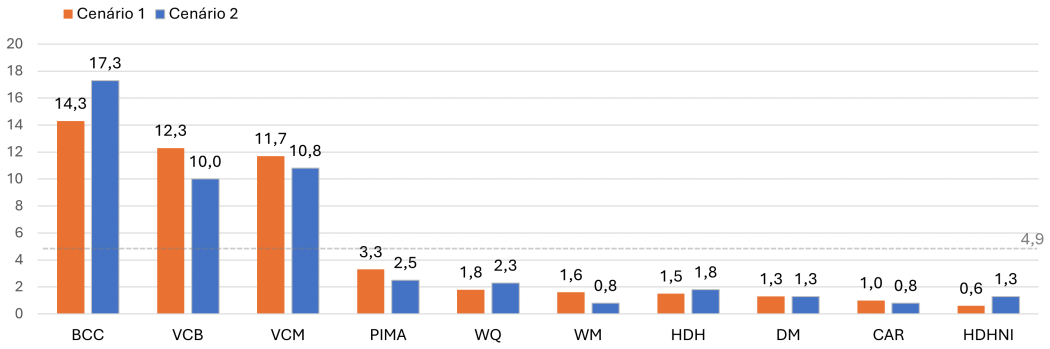


Figura J.1: Cenário 1 e 2: Média de Ganhos na Acurácia (%) por Dataset

O gráfico na Figura J.2 apresenta as médias de Ganhos/Perdas de Acurácia (%) por modelo em cada Cenário.

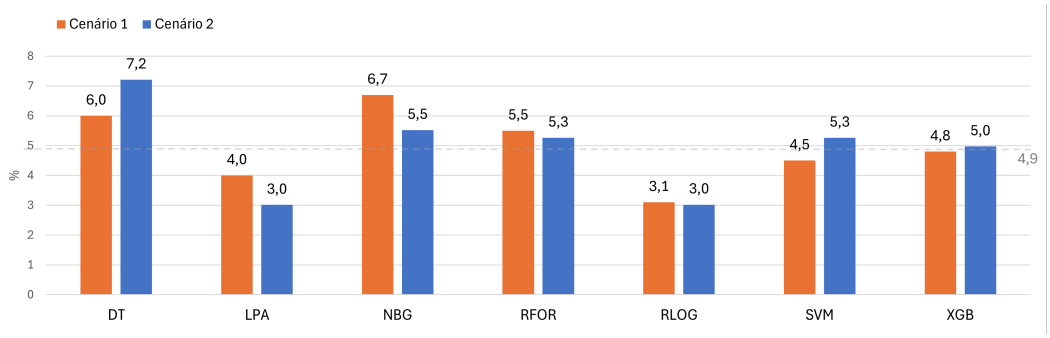


Figura J.2: Cenário 1 e 2: Média de Ganhos na Acurácia (%) por Modelo

O gráfico na Figura J.3 apresenta as médias de Ganhos/Perdas de Acurácia (%) por estatísticas extraídas do grafo em cada Cenário.

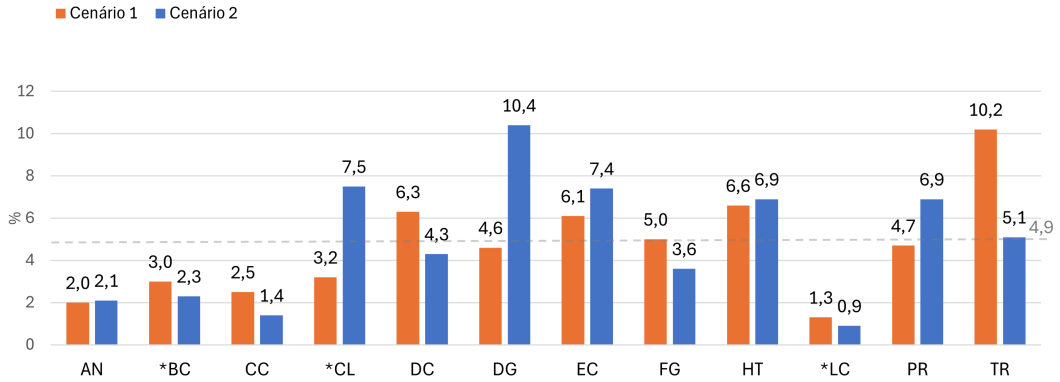


Figura J.3: Cenário 1 e 2: Média de Ganhos na Acurácia (%) por Estatística Grafo

O gráfico na Figura J.4 apresenta a frequência com que as estatísticas extraídas do grafo estiveram nos melhores resultados em cada Cenário.

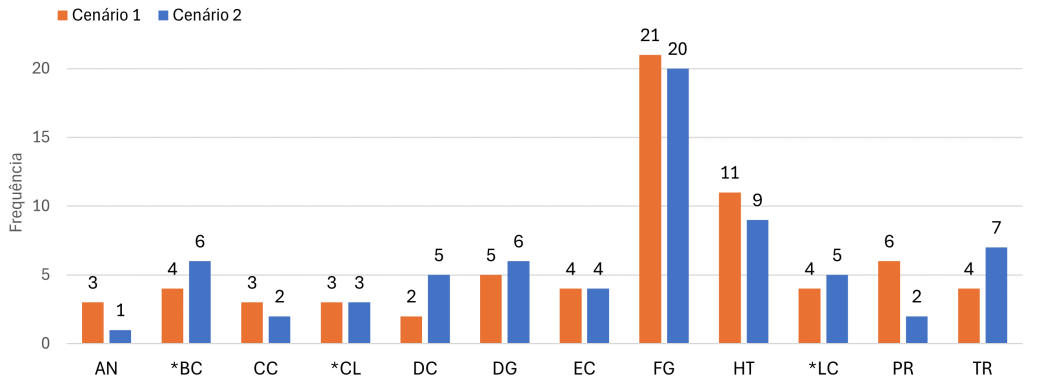


Figura J.4: Cenário 1 e 2: Frequência de Melhores Resultados na Acurácia (%) por Estatística Grafo

O gráfico na Figura J.5 apresenta as médias de Ganhos/Perdas de Acurácia (%) e frequência com que as estatísticas extraídas do grafo estiveram nos melhores resultados em cada Cenário. O tamanho desses pontos de dados está relacionada a média de Ganhos/Perdas de Acurácia (%). Esses dados evidênciam que aquelas que mais contribuíram para os ganhos de performance nos resultados foram *degree* (DG), *eigenvector centrality*, (EC), *clustering* (CL), *hits* (HT) e o conjunto de todas as estatísticas selecionadas (FG).

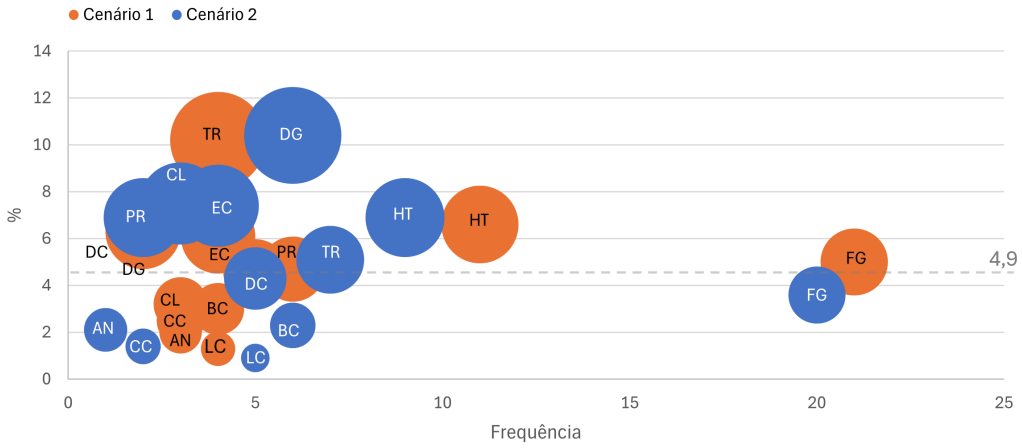


Figura J.5: Cenário 1 e 2: Média e Frequência de Melhores Resultados na Acurácia (%) por Estatística Grafo - Tamanho: Média

O gráfico na Figura J.6 apresenta as médias de Ganhos/Perdas de Acurácia (%) por similaridades categóricas e numéricas em cada Cenário.

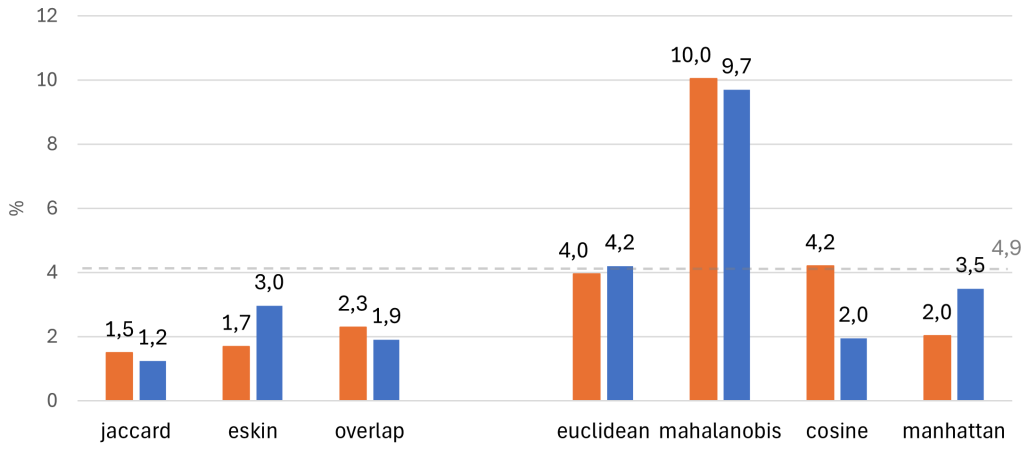


Figura J.6: Cenário 1 e 2: Média de Ganhos na Acurácia (%) por Similaridade

O gráfico na Figura J.7 apresenta a frequência com que as similaridades categóricas e numéricas estiveram nos melhores resultados em cada Cenário.

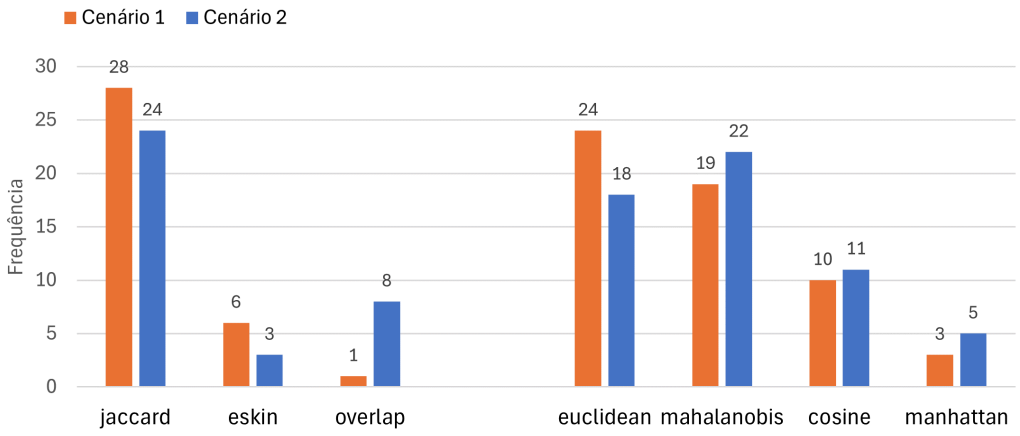


Figura J.7: Cenário 1 e 2: Frequência de Melhores Resultados na Acurácia (%) por Similaridade

O gráfico na Figura J.8 apresenta as médias de Ganhos/Perdas de Acurácia (%) e frequência com que as similaridades categóricas e numéricas estiveram nos melhores resultados em cada Cenário. O tamanho desses pontos de dados está relacionada a média de Ganhos/Perdas de Acurácia (%). Evidênciando melhores resultados nas medidas *Euclidean* e *Mahalanobis* pelas características numéricas e *Jaccard* pelas características categóricas.

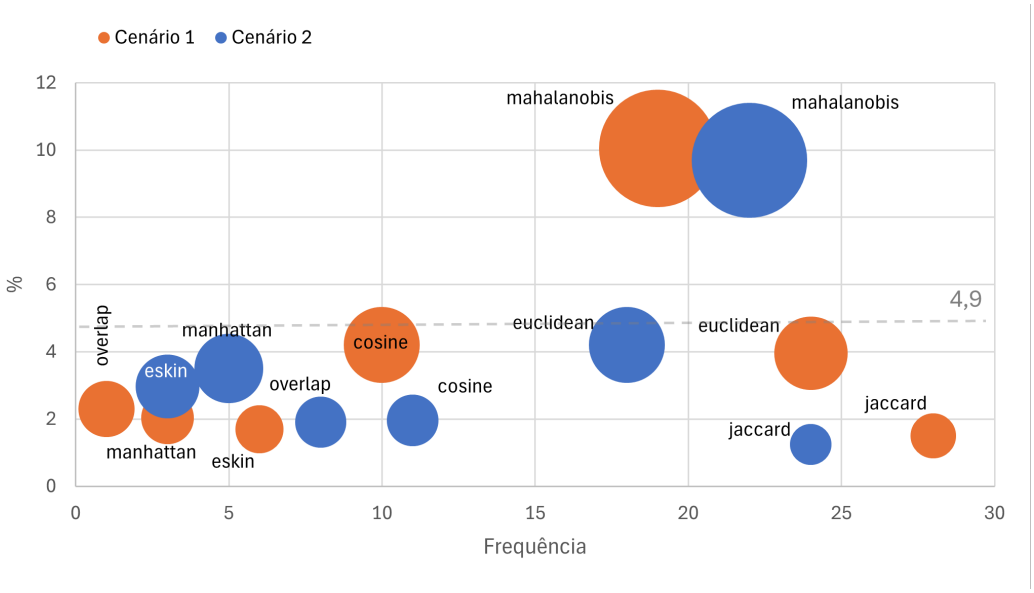


Figura J.8: Cenário 1 e 2: Média e Frequência de Melhores Resultados na Acurácia (%) por Similaridade - Tamanho: Média

J.4 Tempos de Processamento

As tabelas J.21 e J.22 apresentam os tempos médios de processamento nas etapas do pipeline de treino-Teste do método para o Cenário 1. Na Tabela J.22 as etapas 3a-Criação Grafo Treino e 4a-Criação Grafo Teste, foram agrupadas, assim como as etapas 3b-Extração Estatísticas Grafo Treino e 4b-Extração Estatísticas Grafo Teste, para facilitar a comparação.

Tabela J.21: Cenário 1: Tempo (hh:mm:ss) médio por dataset e etapas do método

	Nr. Etapa						
Dataset	**1**	**2**	**3a**	**3b**	**4a**	**4b**	**5**
BCC	00:00:01	00:00:11	00:00:01	00:00:01	00:00:01	00:00:02	00:01:15
CAR	00:00:01	00:00:20	00:00:03	00:24:48	00:00:03	01:09:11	00:02:35
DM	00:00:01	00:00:19	00:00:45	00:00:46	00:01:29	00:01:26	00:02:08
HDH	00:00:01	00:00:19	00:00:13	00:00:15	00:00:18	00:00:45	00:01:09
HDHNI	00:00:01	00:00:07	00:00:01	00:00:19	00:00:01	00:00:45	00:00:51
PIMA	00:00:19	00:00:12	00:00:08	00:05:15	00:00:16	00:15:15	00:01:42
VCB	00:00:02	00:00:10	00:00:01	00:00:05	00:00:01	00:00:31	00:00:58
VCM	00:00:01	00:00:13	00:00:01	00:00:12	00:00:01	00:00:29	00:01:22
WM	00:00:01	00:00:11	00:00:01	00:00:03	00:00:01	00:00:05	00:01:04
WQ	00:00:02	00:00:40	00:00:03	00:20:33	00:00:02	00:56:33	00:04:17

Cabe lembrar que o consumo de tempo pelo treino e teste dos modelos (etapas 2 e 5), para ambos os Cenário, sofre uma influência importante porque são efetuados, na realidade, um processamento para FO na etapa 2 e sete na etapa 5 com FO+FG e FG_i i=1,..,6).

Tabela J.22: Cenário 1: Tempo (hh:mm:ss) médio por dataset e etapas agrupadas método

	Nr. Etapa				
Dataset	**1**	**2**	**3a-4a**	**3b-4b**	**5**
BCC	00:00:01	00:00:11	00:00:02	00:00:03	00:01:15
CAR	00:00:01	00:00:20	00:00:06	01:33:59	00:02:35
DM	00:00:01	00:00:19	00:02:14	00:02:12	00:02:08
HDH	00:00:01	00:00:19	00:00:31	00:01:00	00:01:09
HDHNI	00:00:01	00:00:07	00:00:02	00:01:04	00:00:51
PIMA	00:00:19	00:00:12	00:00:24	00:20:30	00:01:42
VCB	00:00:02	00:00:10	00:00:02	00:00:36	00:00:58
WM	00:00:01	00:00:13	00:00:02	00:00:41	00:01:22
WQ	00:00:01	00:00:11	00:00:02	00:00:08	00:01:04
WQ	00:00:02	00:00:40	00:00:05	01:17:06	00:04:17

As tabelas J.23 e J.24 apresentam os tempos médios de processamento nas etapas do pipeline de treino-Teste do método para o Cenário 2. Na tabela J.24 as

etapas 3a-Criação Grafo Treino e 4a-Criação Grafo Teste, foram agrupadas, assim como as etapas 3b-Extração Estatísticas Grafo Treino e 4b-Extração Estatísticas Grafo Teste, para facilitar a comparação.

Tabela J.23: Cenário 2: Tempo (hh:mm:ss) médio de processamento por etapa método

	Nr. Etapa						
Dataset	**1**	**2**	**3a**	**3b**	**4a**	**4b**	**5**
BCC	00:00:01	00:00:51	00:00:01	00:00:01	00:00:01	00:00:02	00:04:25
CAR	00:00:01	00:01:56	00:00:03	00:24:48	00:00:03	01:15:21	00:12:29
DM	00:00:01	00:02:10	00:00:45	00:00:46	00:01:29	00:01:26	00:13:15
HDH	00:00:01	00:01:00	00:00:13	00:00:15	00:00:18	00:00:45	00:06:09
HDHNI	00:00:01	00:00:50	00:00:01	00:00:19	00:00:01	00:00:45	00:05:11
PIMA	00:00:19	00:03:16	00:00:08	00:05:15	00:00:16	00:15:15	00:23:01
VCB	00:00:02	00:01:21	00:00:01	00:00:05	00:00:01	00:00:31	00:08:26
VCM	00:00:01	00:00:30	00:00:01	00:00:12	00:00:01	00:00:29	00:02:59
WM	00:00:01	00:01:35	00:00:01	00:00:03	00:00:01	00:00:05	00:09:11
WQ	00:00:04	00:04:32	00:00:03	00:20:33	00:00:02	00:56:33	00:31:02

Tabela J.24: Cenário 2: Tempo (hh:mm:ss) médio por dataset e etapas agrupadas método

	Nr. Etapa				
Dataset	**1**	**2**	**3a-4a**	**3b-4b**	**5**
BCC	00:00:01	00:00:51	00:00:02	00:00:03	00:04:25
CAR	00:00:01	00:01:56	00:00:06	01:40:09	00:12:29
DM	00:00:01	00:02:10	00:02:14	00:02:12	00:13:15
HDH	00:00:01	00:01:00	00:00:31	00:01:00	00:06:09
HDHNI	00:00:01	00:00:50	00:00:02	00:01:04	00:05:11
PIMA	00:00:19	00:03:16	00:00:24	00:20:30	00:23:01
VCB	00:00:02	00:01:21	00:00:02	00:00:36	00:08:26
VCM	00:00:01	00:00:30	00:00:02	00:00:41	00:02:59
WM	00:00:01	00:01:35	00:00:02	00:00:08	00:09:11
WQ	00:00:04	00:04:32	00:00:05	01:17:06	00:31:02