PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

Sistema Tutor para Aplicação de Ciência de Dados

Luca Vasconcellos Ribeiro

RELATÓRIO DE PROJETO FINAL DE GRADUAÇÃO

CENTRO TÉCNICO CIENTÍFICO - CTC

DEPARTAMENTO DE INFORMÁTICA

Curso de Graduação em Ciência da Computação

Rio de Janeiro, novembro de 2024

Luca Vasconcellos Ribeiro

Sistema Tutor para Aplicação de Ciência de Dados

Relatório de Projeto Final, apresentado ao programa Ciência da Computação da PUC-Rio como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Marcos Vianna Villas

Departamento de Informática

Rio de Janeiro

Novembro de 2024

Resumo

Vasconcellos Ribeiro, Luca. Viana Villas, Marcos. **Sistema Tutor para Aplicação de Ciência de Dados**. Rio de Janeiro, 2024. 85 p. Relatório de Projeto Final – Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

Este projeto de graduação apresenta a especificação e o desenvolvimento de um sistema voltado a auxiliar pessoas sem especialização em ciência de dados na realização de análises sobre conjuntos de dados. O sistema promove sugestões de técnicas de ciência de dados com base nas características do dataset fornecido, informando motivos de sugestão da técnica, formas de executá-la e resultados que seriam esperados.

Palavras-chave

Ciência de Dados; Análise de Dados; Sistema; Prolog

Abstract

Vasconcellos Ribeiro, Luca. Viana Villas, Marcos. **Data Science Techniques Tutoring Software**. Rio de Janeiro, 2024. 85 p. Final Project Report - Department of Informatics. Pontifical Catholic University of Rio de Janeiro..

This graduation project presents the specification and development of a software aimed at assisting individuals without a background in data science in performing analyses on datasets. The software provides suggestions for data science techniques based on the characteristics of the provided dataset, explaining the reasons for suggesting each technique, ways to implement it, and the expected results.

Keywords

Data Science; Data Analysis; Software; Prolog

Sumário

Sumário

1. Introdução

Com o aumento contínuo do uso de redes sociais, sistemas de IoT (Internet das Coisas) e de outras tecnologias gerando grandes quantidades de informações nos dias atuais, o fluxo de dados se torna cada vez maior. Em vista disso, observa-se uma crescente importância da área de ciência de dados, a qual, de acordo com a IBM [1], é um campo da informática que busca compreender o conhecimento proveniente de conjuntos de dados, aplicando técnicas matemáticas e estatísticas, programação e experiência de domínio.

A ciência de dados apresenta diversos benefícios em múltiplos setores, como o financeiro, permitindo análises de vendas de produtos e serviços, procurando otimizar o desempenho de empresas; ambiental, como análises geográficas visando à identificação de áreas que mais seriam afetadas por desastres naturais; além de atuar em conjunto com aprendizado de máquina e inteligência artificial, permitindo criar sistemas de recomendação e modelos preditivos para encontrar padrões valiosos dentro de complexos conjuntos de dados.

No campo da ciência de dados, existem diversas ferramentas e sistemas para auxiliar a atender às demandas de projetos e ter melhor compreensão a respeito de dados disponíveis. Entretanto, por diversas vezes, usuários não especializados (em técnicas de ciência de dados) não encontram facilidade para escolher as ferramentas adequadas ou explorar com mais detalhes os seus dados, o que faz com que informações relevantes, decorrentes de uma correta análise de dados, não sejam consideradas.

2. Situação Atual

Segundo a IBM [1], devido à crescente abundância de fontes de dados que vem ocorrendo nas últimas décadas, a área de ciência de dados gera cada vez mais uma necessidade de profissionais especializados no tema. A compreensão da grande gama de informações disponíveis, aliada ao acesso a ferramentas e técnicas apropriadas, torna possível providenciar sugestões bastante valiosas na tomada de decisão.

Na seção 2.1 é apresentado um estudo preliminar de ferramentas que tem por objetivo analisar as características de ferramentas que apoiem a execução de tarefas de análise e ciência de dados, buscando compreender suas limitações e funcionalidades. De modo a entender como usuários especializados e não especializados em ciência de dados escolhem as ferramentas e métodos que utilizam para executarem suas análises, foram realizadas entrevistas preliminares, descritas na seção 2.2. Por fim, a seção 2.3 apresenta a conclusão a respeito do estudo de ferramentas e das entrevistas.

2.1 Estudo Preliminar de Ferramentas

Existem diversas ferramentas desenvolvidas para apoiar cientistas de dados em suas tarefas, oferecendo uma variedade de funcionalidades para análises. Um exemplo é o Orange Data Mining [2], que permite aos usuários construir sequências de etapas para analisar conjuntos de dados, ilustrando os processos de forma interativa. Com esse recurso, o Orange se destaca como uma ferramenta educativa, proporcionando suporte visual e interativo que facilita o aprendizado.

Diante da grande quantidade de informações coletadas de múltiplas fontes, surge uma alta variedade de tipos de dados que possibilita análises extremamente diversificadas.

Nesse contexto, softwares que oferecem suporte para análises com dados geoespaciais ou textuais tornam-se fundamentais, visto que esses tipos de dados são cruciais para a interpretação de contextos e padrões,

enriquecendo significativamente as análises. Por exemplo, o software ArcGIS [3] permite a manipulação de dados geográficos com ferramentas avançadas de estatística espacial[1] - como mapas de clusters e análise de padrões - e construção de mapas, enquanto a plataforma Lettria [4] dispõe de uma API para processamento de linguagem natural e manipulação de dados textuais.

Entretanto, apesar do grande auxílio que essas ferramentas apresentam, as três citadas requerem que o usuário escolha as técnicas a serem utilizadas sobre seus conjuntos de dados, não as sugerindo automaticamente. Uma vez que pessoas não especializadas ao trabalharem com ciência de dados possuem entendimento e interpretabilidade precários sobre técnicas e modelos, a falta de conhecimento para aplicar as técnicas apropriadas em seus dados pode acabar prejudicando suas análises, seja por torná-las enviesadas e imprecisas, ou por não aproveitar eventuais informações importantes à disposição.

Em algumas outras ferramentas, observa-se uma menor necessidade de seleção de técnicas por parte do usuário, com o próprio sistema já fornecendo sugestões de como lidar com as análises.

Ao trabalhar com a linguagem de programação Python [5], uma das mais utilizadas na área de ciência de dados, pode ser utilizado o serviço Google Colab [6] para armazenar e compartilhar códigos, o qual com o uso de Inteligência Artificial, pode promover a construção de gráficos para que análises simples e práticas sejam feitas. O DataPrep by Trifacta [7] é um serviço de nuvem da Google que facilita o preparo de dados para análises e aprendizado de máquina, de forma inteligente. Por outro lado, em ambas não há explicações que justifiquem os motivos para a aplicação de determinadas escolhas, como o porquê da geração de um gráfico em detrimento de outro ou a aplicação de uma técnica sobre um dataset específico.

[1] ESRI. An overview of the Spatial Statistics toolbox. Disponível em: https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-statistics/an-overview-of-the-spatial-statistics-toolbox.htm. Acesso em: 13 de junho de 2024.

2.2 Entrevistas Preliminares

Visando a compreender melhor sobre a questão de como são executadas tarefas de análise e ciência de dados, foram realizadas entrevistas com um especialista[2] na área e com um profissional[3] que trabalha regularmente com datasets, porém com menos experiência no uso de técnicas de análise de dados mais avançadas, as quais poderiam enriquecer sua compreensão das informações disponíveis.

2.2.1 Entrevistado Não Especializado em Ciência de Dados

A entrevista teve como base o seguinte roteiro:

- Para fazer análise de dados, você tem um passo-a-passo a seguir?
  - Se sim, que etapas você seguiria para analisar o dataset do contexto?
- Existe algum processo que você executa manualmente e gostaria de automatizar e/ou facilitar com alguma ferramenta de análise de dados?
- Que ferramentas você usa para análise de dados?
- Que ferramentas você gostaria de usar para análise de dados?
- Você teria interesse em aprender mais sobre análise de dados?
  - Que recursos você consideraria interessantes para o aprendizado?
- Você teria interesse em uma ferramenta que te sugerisse técnicas/modelos de dados para análise?
- Que recursos uma ferramenta ideal de análise de dados deveria ter?

O entrevistado não especialista comentou que não segue um procedimento específico ao fazer suas análises, optando por uma

---

[2] Cientista da computação, 32 anos de idade. Formado em Ciência da Computação. Tempo de experiência de 5 anos em Ciência de Dados. Doutor em Informática.

[3] Arquiteto, 60 anos de idade. Formado em Arquitetura e Educação. Tempo de experiência de 37 anos em Arquitetura e 30 anos em Educação. Graduado em Arquitetura e Doutor em Educação.

abordagem simples e usando apenas o Excel [8] para criar alguns gráficos, reconhecendo que a ferramenta oferece mais recursos do que por ele são utilizados.

Quando questionado sobre seu interesse em aprender mais sobre análise de dados, ele expressou vontade, mas não demonstrou interesse em usar uma ferramenta didática que o ensinasse sobre técnicas e modelos (de aprendizado de máquina) para melhorar suas análises. No entanto, ele destacou que um software capaz de mostrar correlações e realizar análises preditivas seria útil para as suas demandas. Alguns de seus conjuntos de dados, por ele utilizados com frequência, estão relacionados à gestão de obras, para os quais ele considera que o software MS Project [9] seria mais eficaz, embora não o utilize.

2.2.2 Entrevistado Especializado em Ciência de Dados

Já para a entrevista com o usuário especializado, o seguinte roteiro foi utilizado como base:

- Quais os principais passos para se trabalhar com um conjunto de dados?
- Como abordar variedades de tipos de dados (como geoespaciais e textuais)?
- Quais as principais dificuldades ao se trabalhar com ciência de dados?
- Que ferramentas (linguagens, sistemas) costuma utilizar para trabalhar com análises de dados?
- O que acha de usuários não especializados em análise de dados usando tais ferramentas?

O entrevistado especialista explicou que segue uma sequência de etapas para fazer suas análises, porém enfatizou que ao longo do processo se torna um trabalho de experimentação, dependendo da demanda, mesmo que siga um roteiro inicial de coletar os dados, realizar análises exploratórias e construir visualizações. Ele ressaltou ainda que, embora haja um pré-processamento semelhante para conjuntos de dados do mesmo tipo, esse processo pode variar para tipos diferentes.

Quanto às principais dificuldades enfrentadas na área de ciência de dados, o especialista destacou o desafio de mostrar que o projeto de ciência de dados (com aplicações de inteligência artificial e aprendizado automático) tem valor para o negócio.

A respeito da principal ferramenta utilizada, foi mencionada a linguagem de programação Python. O entrevistado defendeu que usuários não especializados não encontrariam muita dificuldade ao lidar com a linguagem para executar tarefas de análise e ciência de dados, tendo em vista que é possível encontrar uma documentação robusta, comunidades ativas de desenvolvedores e facilidade de uso. No entanto, ele ressaltou a importância de ter conhecimento em estatística e conceitos técnicos de ciência de dados.

2.3 Conclusão preliminar

A escolha de uma ferramenta apropriada para cumprir com os objetivos ao se trabalhar com ciência de dados, então, se torna um desafio, em especial para usuários não especializados. Observa-se, desse modo, que um sistema que não somente realize análises de ciência de dados, mas também demonstre explicações para as escolhas feitas poderia ser bastante útil, de forma a auxiliar e orientar, de forma didática, tais usuários.

3. Objetivo do Trabalho

O objetivo do presente trabalho foi projetar e desenvolver um software que apoie didaticamente usuários não especializados em ciência de dados, auxiliando-os a encontrar recursos e técnicas adequados para analisarem e compreenderem seus datasets, ao mesmo tempo em que explique as decisões que levaram à escolha das recomendações. Um sistema que auxilie na compreensão dessas tarefas e no aprendizado pode contribuir para que usuários sem muita experiência em ciência de dados consigam atender às suas demandas, tornando-os mais aptos em situações futuras semelhantes.

4. Plano de Ação

O plano de ação possui uma descrição de etapas e cronogramas propostos para as disciplinas de Projeto Final I e II.

4.1 Etapas

A Figura 1 a seguir apresenta um diagrama com as etapas a serem realizadas:

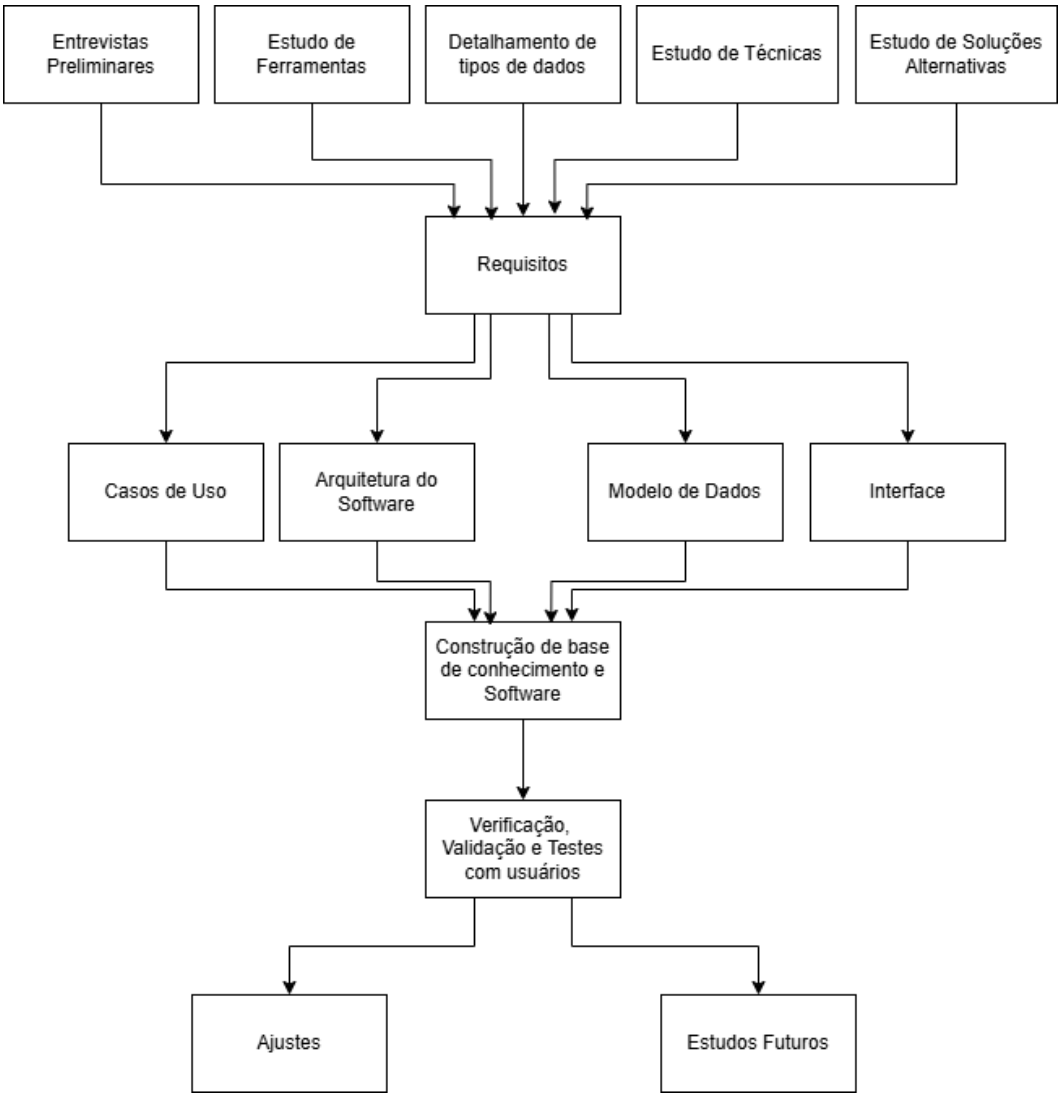


Figura 1: Diagrama de etapas do projeto

A primeira etapa inclui as seguintes tarefas:

- Entrevistas Preliminares: realizadas com dois tipos de

profissionais (um com experiência na área de ciência de dados e outro não) para melhor entendimento da situação atual;

- Estudo de ferramentas: explorar aplicações e sistemas que forneçam apoio para realizar tarefas de ciência de dados;
- Detalhamento de tipos de dados: descrição dos tipos de dados que vão estar presentes no escopo do sistema e suas características;
- Estudo de técnicas: pesquisa sobre técnicas de ciência de dados que podem ser aplicadas em datasets apresentando os tipos de dados descritos;
- Estudo de soluções alternativas: descrição de diferentes alternativas para a construção do software.

A partir dos estudos feitos na etapa anterior, são levantados requisitos:

- Requisitos: levantamento preliminar de requisitos funcionais para especificar os objetivos e funcionalidades do projeto.

Posteriormente:

- Casos de uso: descrever interações de usuários com as funcionalidades do sistema;
- Arquitetura de software: organização de módulos e de funções do sistema;
- Modelo de dados: representação e organização de dados do sistema;
- Interface: organização de telas do sistema.

Por fim, as etapas finais do projeto:

- Construção de base de conhecimento e software: executar a identificação de tipos de dados, sugestão de técnicas a serem aplicadas e possibilidade de executá-las no dataset;
- Verificação, validação e testes: averiguar o funcionamento do sistema com possíveis usuários finais e especialistas;
- Ajustes: correções a serem feitas no sistema;

- Estudos futuros: investigar outras tecnologias e abordagens para aprimorar o sistema.

4.2 Cronogramas

As Tabelas 1 e 2 contêm os cronogramas de planejamento de tarefas de Projeto Final 1 e Projeto Final 2, respectivamente.

Atividades	Março				Abril				Maio				Junho			
Proposta																
Estudo de ferramentas e plataformas que apoiam análise de dados																
Entrevistas iniciais para coleta de informações sobre tarefas de análise e ciência de dados																
Estudo de tipos de dados e técnicas aplicáveis																
Estudo de Soluções Alternativas para a implementação do software																
Definição de requisitos do sistema																
Relatório de Projeto Final 1																

Tabela 1: Cronograma de Projeto Final 1

Atividades	Agosto				Setembro				Outubro				Novembro			
Especificação de dados de funções e de interface																
Definição de casos de uso																
Construção do Software																
Testes e validação com profissionais e usuários finais																
Relatório de Projeto Final 2																

Tabela 2: Cronograma de Projeto Final 2

5. Tipos de Dados

Diversos tipos de dados são encontrados em tarefas recorrentes de ciência de dados, cada um tendo características específicas e técnicas apropriadas a serem aplicadas para que análises sejam feitas adequadamente.

Inicialmente, os tipos de dados abordados pela ferramenta serão numéricos, categóricos, geoespaciais, textuais e temporais. Dados de imagem, vídeo e áudio, por exemplo, podem vir a ser incluídos caso a aplicação seja extensível, permitindo, posteriormente, adicionar técnicas específicas para o tratamento e manipulação desses dados. Esses tipos de dados não estão presentes no escopo do projeto, uma vez que demandam maior complexidade de interpretação e tratamento.

A seguir são abordadas explicações e características relativas a alguns dos tipos de dados que podem ser encontrados, os quais serão os presentes no escopo da ferramenta. O estudo sobre essas informações baseou-se nas seguintes referências:

- Introdução à estatística, por Ranganathan e Gogtay [10], abordando sobre dados numéricos e categóricos;
- Técnicas e ferramentas de análise geoespacial no suporte de atendimentos a acidentes cardiovasculares, por Padgham et al (2019) [11];
- Mineração de big data com dados de turismo baseando-se em documentos (*corpus* de texto), por Li et al, (2019) [12], discorrendo a respeito de dados textuais;
- Visão geral de mineração de dados temporais, por Lin, Orgun e Williams (2002) [13].

5.1 Numéricos

Também chamados de dados quantitativos, podem ser divididos em 2 tipos: contínuos, indicando valores como temperatura, pressão, preço, quaisquer valores não inteiros; discretos, indicando números inteiros, como por exemplo número de visualizações em um vídeo ou quantidade de residentes em um apartamento.

Em muitas etapas de uma tarefa de ciência de dados é crucial trabalhar com dados estritamente numéricos, tendo em vista que determinadas técnicas suportam somente esse tipo de dado, não podendo receber dados textuais ou categóricos, por exemplo. Assim, a aplicação de técnicas de conversão para dados numéricos, como será discutido na seção 7, é bastante importante.

5.2 Categóricos

Dados categóricos, por sua vez, também podem ser divididos em dois grupos: ordinais, representando rankings ou uma ordem, como cargos e faixas de valores; nominais, representando dados sem hierarquia ou ordem, como raças de cachorro ou gêneros de filmes.

Vale ressaltar que é possível facilmente representar dados categóricos ordinais como numéricos discretos. Ao representar um atributo como uma ordem, por exemplo, baixo, médio ou alto, semanticamente seria possível fazer a representação como, respectivamente, 0, 1 e 2.

5.3 Geoespaciais

Durante o estudo realizado por Padgham et al (2019) [11] foram feitas análises de dados geoespaciais e como podem auxiliar no fornecimento de serviços médicos, mais especificamente no suporte de atendimentos a acidentes cardiovasculares. Na pesquisa, dados geoespaciais são definidos como: pontos específicos na superfície do globo terrestre, ou seja, latitude e longitude; conjuntos de pontos delimitando uma área; ou rotas, as quais são formadas pela delimitação de pontos e outros tipos de informações.

Demonstrando com um exemplo do mundo real os benefícios da utilização desse tipo de dado, o grupo de pesquisa ilustra como informações geoespaciais podem ser fundamentais para incrementar análises, como identificar a localização dos pacientes e os caminhos mais próximos até os centros de tratamento.

É importante considerar que dados que representam informações

geoespaciais podem ser de tipo numérico (ao conter dois atributos indicando latitude e longitude), categórico ou textual (ao conter um atributo indicando o nome de uma região ou um endereço).

5.4 Textuais

Texto é um formato de dado não estruturado que pode ser coletado de diversas fontes, como pesquisas com usuários, redes sociais e documentos escritos. Para executar análises em cima de datasets com dados textuais, existem diversas técnicas específicas para a realização de pré-processamento - como tokenização, stemização e lematização -, bem como outras que potencializam a compreensão sobre os dados, como análises de sentimento e modelagem de tópicos.

Na pesquisa de texto em mineração de Big Data de Li et al (2019) [12], são abordados os benefícios da aplicação de técnicas de processamento de linguagem natural em uma situação do mundo real. A aplicação dessas técnicas fornece apoio para que organizações compreendam como melhorar os serviços fornecidos a partir de opiniões armazenadas em dados textuais.

5.5 Temporais

Analisar a distribuição de dados ao longo do tempo é uma tarefa bastante importante, que pode ajudar a realizar predições a respeito dos valores. A utilização de dados temporais pode ser fundamental para realizar análises precisas. Esses dados podem vir em formatos informando somente a data de determinada ocorrência no dataset ou podem ser mais precisos, informando as horas, minutos e segundos, por exemplo.

Na pesquisa de Lin, Orgun e Williams (2002) [13], vários métodos de análises utilizando dados de tempo com Big Data são abordados, como predições em séries temporais e classificação temporal. Sobre as tarefas a serem resolvidas utilizando dados temporais, os pesquisadores comentam sobre o reconhecimento de sequências similares e a análise da periodicidade dos dados.

6. Ferramentas para Ciência de Dados

Múltiplas ferramentas voltadas para o auxílio de análises para ciência de dados foram exploradas e estudadas, visando a buscar entender o potencial de cada uma e como podem facilitar o trabalho com cada tipo de dados e de datasets. Tendo em vista que o intuito deste trabalho é a construção de um software que forneça apoio e tutoria para tarefas de ciências de dados, foram pesquisadas algumas ferramentas que fornecem recursos e funcionalidades relevantes para esse contexto:

6.1 Orange Data Mining

De maneira interativa, a aplicação Orange Data Mining fornece apoio para a aplicação de técnicas de ciência de dados e aprendizado de máquina sem a necessidade de código. A ferramenta permite uma fácil construção de um passo a passo a ser seguido para o tratamento de dados, visualização e aplicação de modelos preditivos, contando com um forte apelo educativo ao ilustrar cada etapa executada.

A ferramenta, assim, apresenta recursos apropriados para apoiar usuários que precisem executar tarefas de ciência de dados e fazer análises de datasets.

Dentre as muitas funcionalidades da ferramenta, podem ser destacadas a implementação de técnicas de:

- Visualização dos dados: gráficos de distribuições, de linha e de dispersão;
- Pré-processamento de dados: normalização e imputação;
- Aprendizado de máquina não supervisionado: algoritmos de redução de dimensionalidade e clusterização
- Aprendizado de máquina supervisionado: algoritmos de regressão e classificação

A Figura 2 ilustra o funcionamento da aplicação do Orange. A partir de um arquivo csv, é gerada uma tabela de dados, com a qual são extraídas informações com visualizações e aplicadas técnicas de transformação com pré-processamento de dados e redução de

dimensionalidade, para depois ser realizada uma predição sobre os valores. As técnicas usadas com a ferramenta serão abordadas na seção 7.

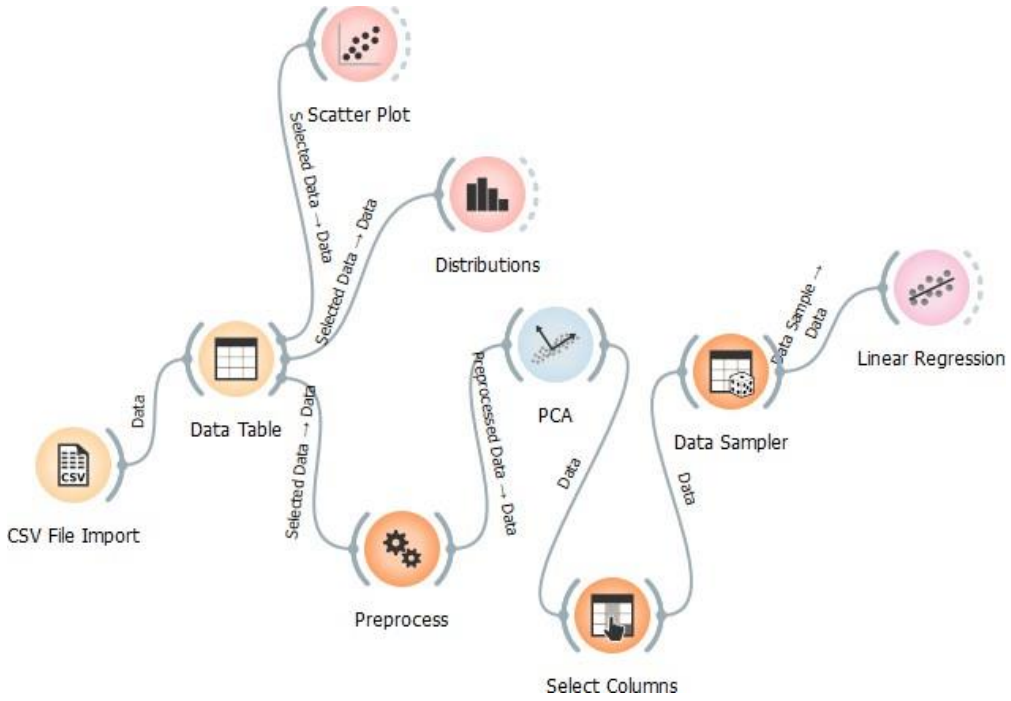


Figura 2: Exemplo de uso da aplicação Orange Data Mining para análise de dados, pré-processamento e aplicação de aprendizado de máquina - Fonte: O Autor.

6.2 ArcGIS

Uma das principais ferramentas para trabalhar com dados geográficos, o ArcGIS é uma plataforma com forte potencial para executar análises com base em informações geoespaciais.

Na construção dos mapas a aplicação permite o uso de múltiplas camadas, as quais são conjuntos de dados geográficos - podendo ser pontos, linhas ou polígonos - possuindo funcionalidades para poder filtrar essas camadas pelos seus atributos para facilitar a visualização e focar em informações específicas.

Além da geração de diversos tipos de mapas, o ArcGIS ainda fornece

a visualização de gráficos com base nos dados geoespaciais, fomentando o potencial de análises. ArcGIS ainda permite a execução da técnica de geocodificação, permitindo encontrar o ponto de localização de endereços e locais.

A Figura 3 ilustra a utilização de funcionalidades do ArcGIS para construir um mapa com informações de tempestades e tornados nos Estados Unidos, a partir da coleta de dados atualizados diariamente[4]:

Figura 3: Exemplo de uso do software ArcGIS para a construção de um mapa - Fonte: O Autor.

6.3 Lettria

A plataforma de inteligência artificial Lettria disponibiliza várias técnicas para trabalhar com dados textuais sem a necessidade de código, como, classificação de texto, análise de sentimento e modelagem de tópicos, sendo algumas das principais etapas realizadas em tarefas de processamento de linguagem natural.

A Figura 4 ilustra o funcionamento da funcionalidade AutoLettria presente na ferramenta Lettria, a qual permite o treinamento de modelos de classificação de texto de forma automática, demonstrando o desempenho dos experimentos feitos com os modelos.

[4] NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION (NOAA). Storm Reports. Disponível em: <https://www.spc.noaa.gov/climo/reports/>. Acesso em: 13 jun. 2024.

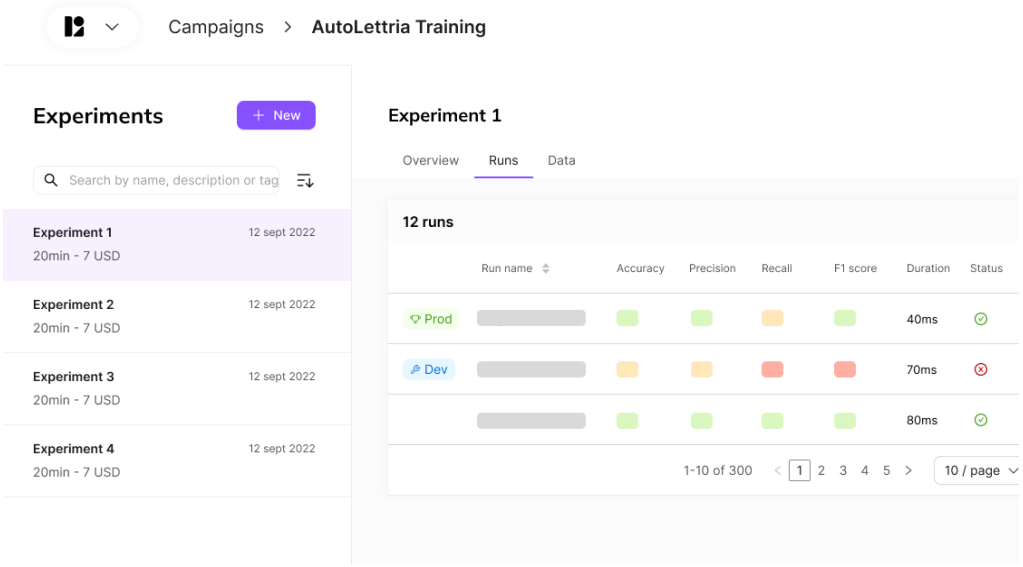


Figura 4: Exemplo de uso da plataforma Lettria para execução de tarefas de processamento de linguagem natural. Fonte: <https://www.lettria.com/features/autolettria>. Acesso em 20 jun. 2024.

6.4 DataPrep by Trifacta

O serviço em nuvem produzido pela empresa Trifacta em colaboração com a Google, DataPrep, é uma ferramenta capaz de identificar problemas presentes em um dataset e sugerir transformações para corrigi-los. Assim, DataPrep se torna uma plataforma de forte apoio à limpeza e ao pré-processamento de dados.

Devido à grande importância da etapa de pré-processamento, o estudo dessa ferramenta pode ser bastante útil para o desenvolvimento do sistema proposto. O uso de técnicas de aprendizado automático para a identificação de problemas presentes no dataset - como descrito na visão geral do produto em sua documentação sobre transformação preditiva[5] - fornece apoio ao usuário da ferramenta para decidir quais passos podem ser tomados na realização de transformações em cima dos dados.

[5] TRIFACTA. Overview of Predictive Transformation. Disponível em: <https://docs.trifacta.com/Dataprep/en/trifacta-application/concepts/feature-overviews/overview-of-predictive-transformation.html#overview-of-predictive-transformation>. Acesso em: 9 jun. 2024.

A Figura 5 ilustra o uso da ferramenta DataPrep para fazer a transformação de dados, aplicando a técnica de imputação para substituir dados faltantes em uma coluna pela média dos valores.

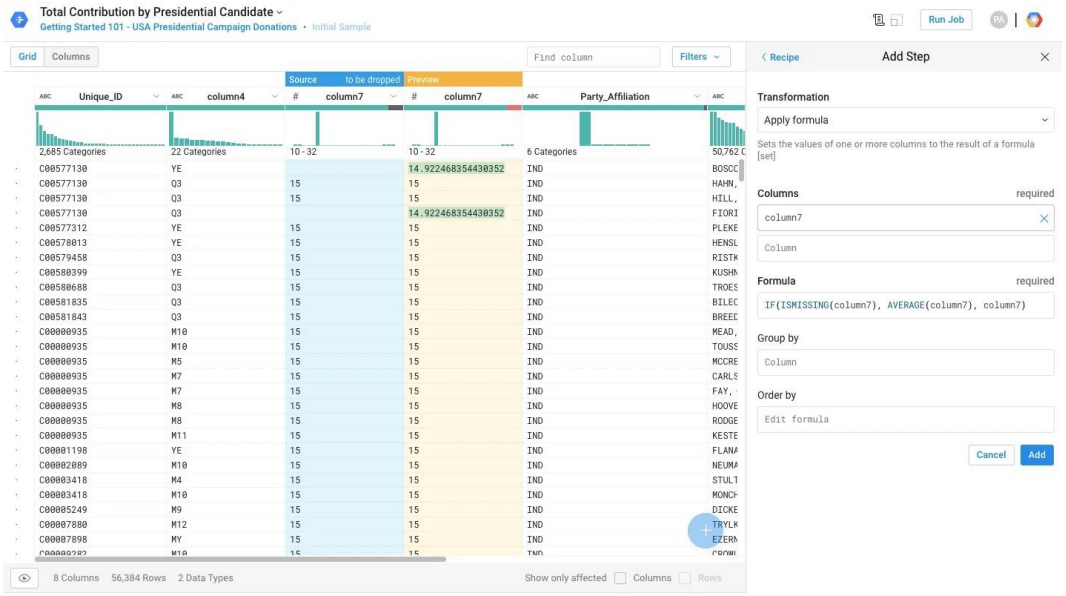


Figura 5: Exemplo de uso da ferramenta Data Prep preparo de dados (data cleaning) - Fonte: <https://blog.searce.com/data-pre-processing-with-google-cloud-dataprep-dd2a50d23f19>. Acesso em 25 jun. 2024.

6.5 Python

Uma das principais linguagens de programação utilizadas no campo de ciência de dados devido à grande quantidade de desenvolvedores ativos participando de suas atualizações, Python conta com diversos frameworks e bibliotecas que viabilizam múltiplos tipos de análises, destacando-se:

- Numpy [14]: computação científica;
- Pandas [15]: armazenamento e manipulação de datasets;
- Scikit-learn [16]: múltiplas técnicas de aprendizado automático e ciência de dados;
- Seaborn [17]: visualização de dados com gráficos.

Ademais, ainda inclui bibliotecas que permitem fazer análises com dados geoespaciais, como:

- Folium [18]: construção de mapas;
- Geopy [19]: localização das coordenadas de cidades, endereços e regiões geográficas.

6.6 R

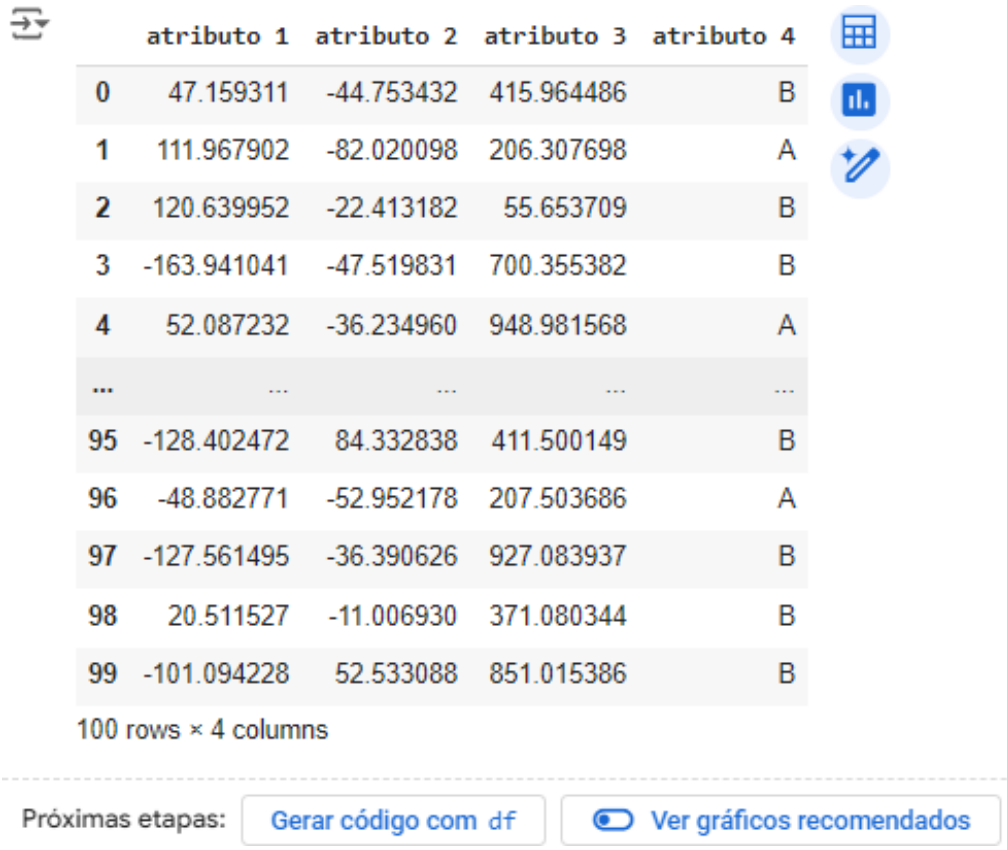
Além de Python, a linguagem de programação R [20] se encontra como um recurso bastante utilizado entre cientistas e analistas de dados devido ao grande conjunto de técnicas estatísticas e gráficas, possuindo um alto número de pacotes que facilitam trabalhar com ciência de dados.

Uma vez que a ferramenta é propícia para a realização de cálculos em dados vetoriais e matriciais, a aplicação de técnicas de ciência de dados e aprendizado de máquina se torna bastante viável.

6.7 Google Colab

Um serviço da Google que permite armazenamento e compartilhamento de documentos de código - em especial da linguagem Python - conta com o uso de inteligência artificial que, além de facilitar a geração de código, executa análises em cima de dados, podendo gerar gráficos ilustrando distribuições (com gráficos de linha e histogramas) e relações entre atributos (com gráficos de dispersão).

As Figuras 6, 7 e 8 ilustram um exemplo da geração de visualizações a partir de um dataset gerado por simulação, utilizando a biblioteca numpy, com três atributos numéricos contínuos e um atributo categórico nominal:



	atributo 1	atributo 2	atributo 3	atributo 4
0	47.159311	-44.753432	415.964486	B
1	111.967902	-82.020098	206.307698	A
2	120.639952	-22.413182	55.653709	B
3	-163.941041	-47.519831	700.355382	B
4	52.087232	-36.234960	948.981568	A
...	...	...	...	...
95	-128.402472	84.332838	411.500149	B
96	-48.882771	-52.952178	207.503686	A
97	-127.561495	-36.390626	927.083937	B
98	20.511527	-11.006930	371.080344	B
99	-101.094228	52.533088	851.015386	B

100 rows × 4 columns

Figura 6: dataset gerado por simulação, com as opções - à direita do resultado - de converter para tabela, sugerir gráficos e gerar código a partir do conjunto de dados. Fonte: O Autor.

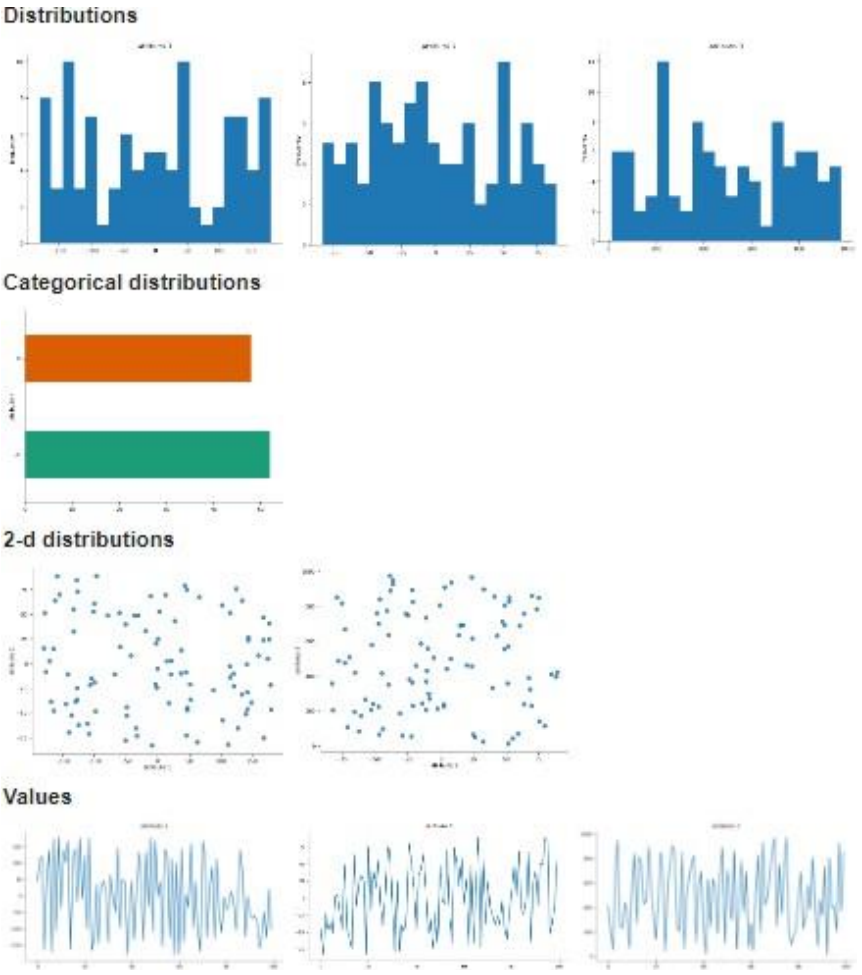


Figura 7: visualizações geradas após selecionar a opção de sugerir gráficos. Fonte: O Autor.

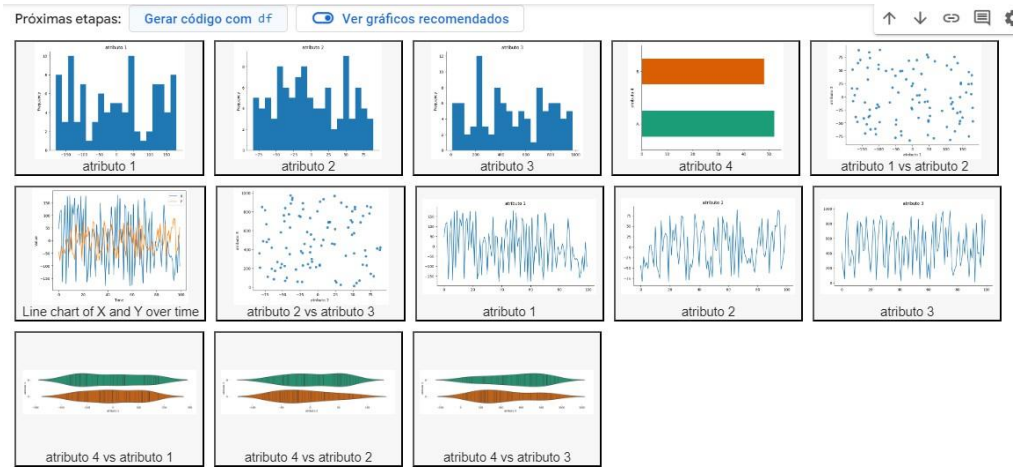


Figura 8: visualizações geradas após selecionar a opção de ver gráficos recomendados. Fonte: O Autor.

6.8 Conclusão sobre Ferramentas

Dados os potenciais de cada uma das ferramentas citadas, é notável que múltiplas funcionalidades poderiam ser implementadas na aplicação objeto do presente trabalho.

O Orange Data Mining apresenta fortes características didáticas no seu formato de construção de etapas a serem feitas com as técnicas. Python, devido à vasta quantidade de bibliotecas que auxiliam na execução de tarefas de ciência de dados e de desenvolvimento de software, será a ferramenta mais aproveitada para o projeto, utilizando-a diretamente no código fonte. O Google Colab, por sua vez, faz algumas sugestões de visualizações que podem auxiliar na análise exploratória, sendo possível fazer uso de inteligência artificial para apoiar na decisão de sugestões.

A transformação preditiva utilizada na DataPrep pode ser útil na identificação de características do dataset fornecido, permitindo o uso de sua API para detectar problemas no dataset, como dados faltantes.

As ferramentas ArcGIS e Lettria apresentam múltiplas técnicas e funcionalidades para lidar com, respectivamente, dados geoespaciais e textuais, de forma que seria possível usar suas APIs ao invés de implementar as técnicas sobre esses tipos de dados manualmente para potencializar as análises.

Devido ao forte potencial estatístico e matemático da linguagem, fazer uso da linguagem R [20] também poderia ser vantajoso na construção do sistema.

7. Técnicas de Ciências de Dados

Diferentes tipos de dados e de datasets recebem tratamentos diferentes ao se trabalhar com ciência de dados, sendo importante saber quais técnicas utilizar dependendo das perguntas que se deseja responder e das demandas a serem atendidas com o dataset, em busca de compreender adequadamente os dados.

A Figura 9 a seguir (Jain et al, 2022) [21] ilustra um simples ciclo de etapas de projetos de ciência de dados:

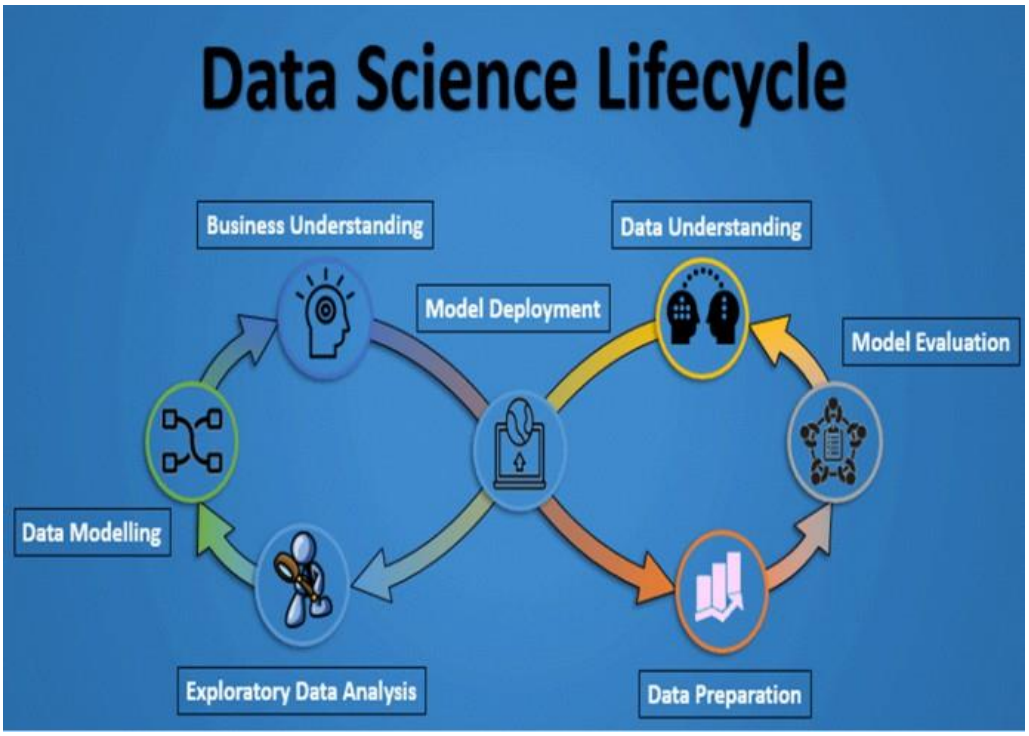


Figura 9: Ciclo de etapas de ciência de dados. Fonte: Jain et al, 2022.

O propósito do projeto é dar foco na etapa de entendimento de dados, avaliando os tipos de dados presentes para identificar as técnicas aplicáveis e promovendo análises exploratórias e de preparo de dados, sugerindo etapas de pré-processamento e engenharia de atributos.

A seguir, algumas técnicas utilizadas por cientistas de dados serão abordadas:

7.1 Visualização de Dados

Em uma análise exploratória, a visualização se torna um passo crucial para melhor compreender distribuições e correlações dos dados. Dessa forma, saber o tipo correto de visualização e escolha de gráficos e mapas para realizar a análise de um dataset é fundamental para permitir que os próximos passos sejam executados da forma adequada.

Gráficos de dispersão, de linha, histogramas e mapas de calor, por exemplo, são algumas formas de visualização que auxiliam na análise de dados.

A seguir, são apresentadas essas visualizações a partir do conjunto de dados simulados da Figura 6 (seção 6.7). Os gráficos são gerados utilizando a biblioteca Seaborn, enquanto o mapa é construído utilizando folium.

A Figura 10 apresenta um gráfico de dispersão, com o eixo X representando os valores do atributo 1, o eixo Y representando os do atributo 2 e as cores o atributo 4. Esse gráfico auxilia na análise da relação entre esses atributos, demonstrando como variam entre si.

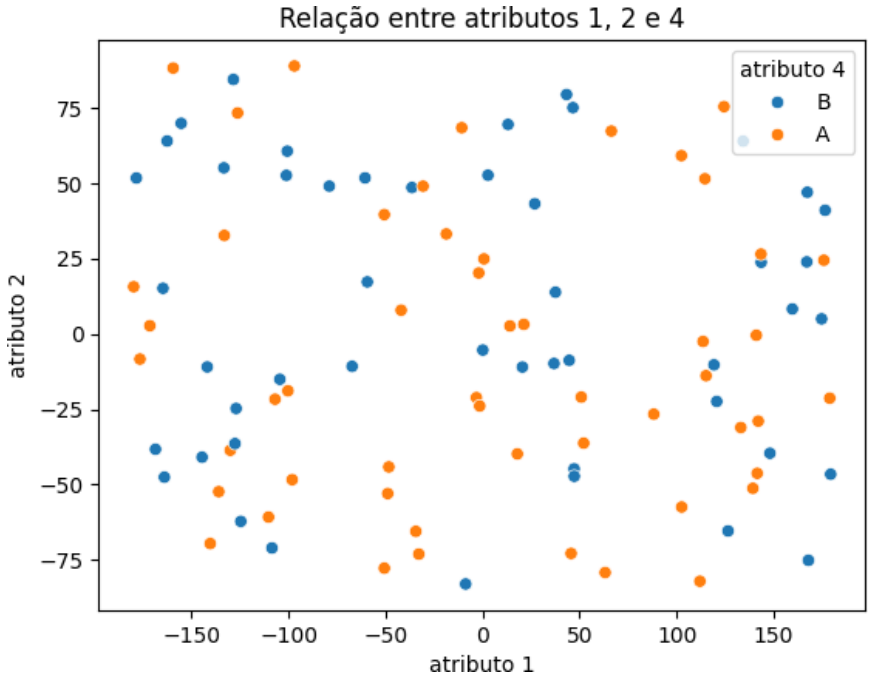


Figura 10: Gráfico de dispersão ilustrando a relação entre os atributos numéricos 1 e 2 e o atributo categórico. Fonte: O Autor.

A Figura 11 apresenta um gráfico de linha, também utilizado para auxiliar na análise de relação entre atributos numéricos. Nesse gráfico, o eixo X representa os valores do atributo 1, enquanto o eixo Y os do atributo 3.

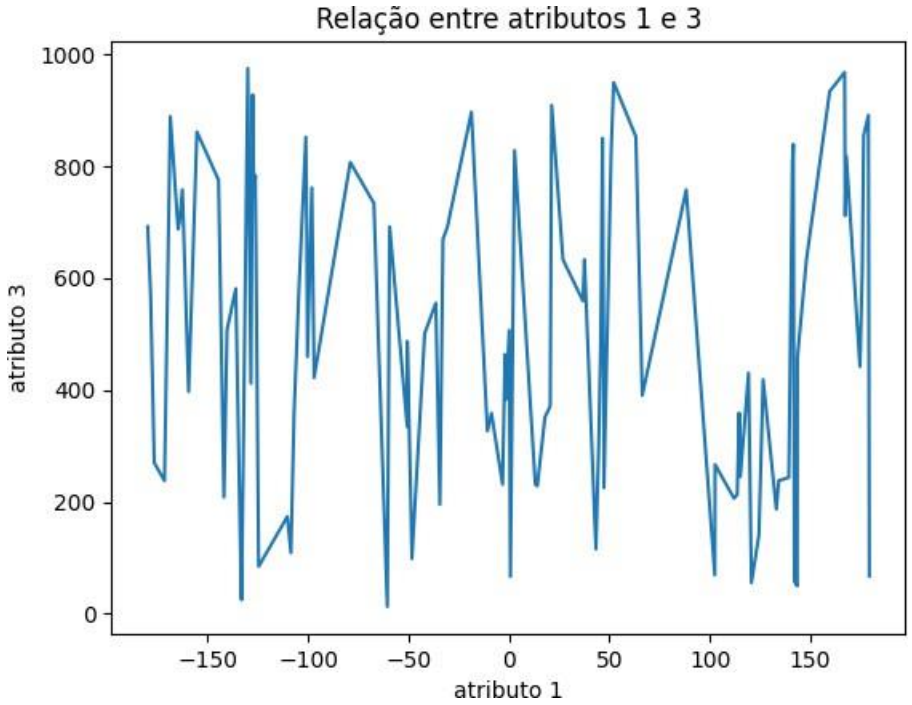


Figura 11: Gráfico de linha ilustrando a relação entre os atributos numéricos 1 e 3. Fonte: O Autor.

A Figura 12 é de um gráfico de histograma que demonstra a distribuição do atributo 1 em diferentes intervalos. A partir do histograma é possível analisar que não há uma concentração muito alta do atributo 1 em valores mais próximos de seu limite inferior, seu limite superior ou sua média, por exemplo.

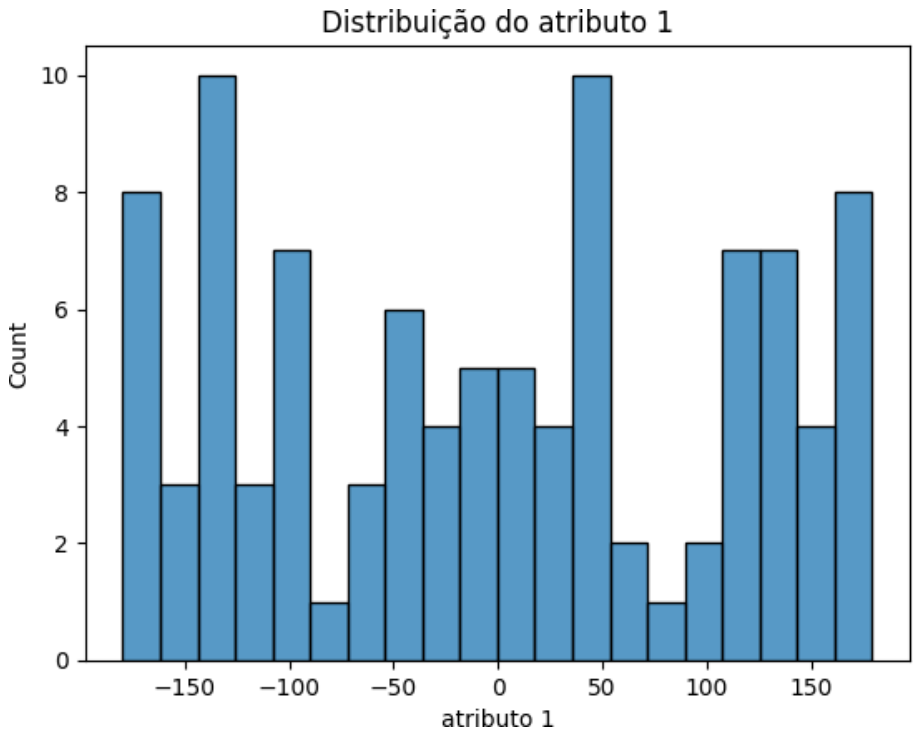


Figura 12: Gráfico de Histograma ilustrando a distribuição do atributo 1.
Fonte: O Autor.

A Figura 13 apresenta um mapa de calor que demonstra para cada um dos 100 elementos do dataset a incidência do atributo 3 em cada ponto, localizado geograficamente com longitude baseada no atributo 1 e latitude no atributo 2. Dessa forma, pontos do mapa com coloração mais avermelhada apresentam um maior valor para o atributo 3, enquanto pontos de cor mais próxima do azul indicam um valor menor para esse atributo. Dessa forma, um mapa de calor pode auxiliar a analisar a variação de um atributo numérico de acordo com sua localização geográfica.

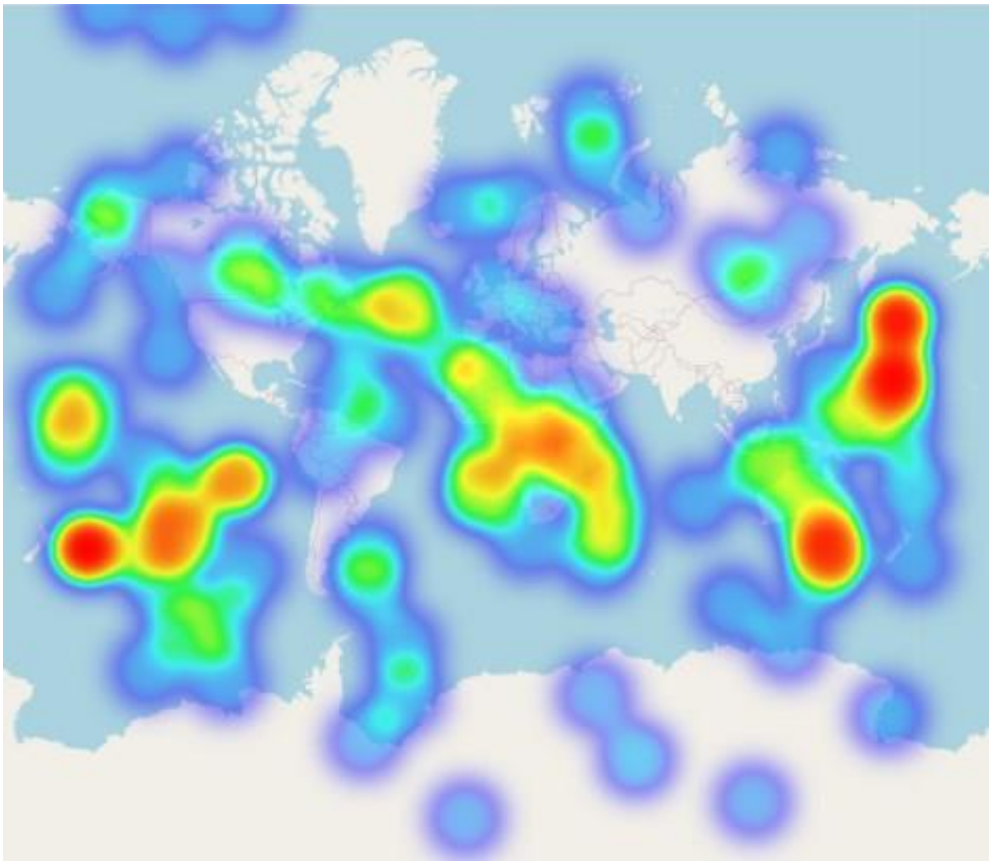


Figura 13: mapa de calor ilustrando a incidência do atributo 3 utilizando como localização de pontos os atributos 1 (longitude) e 2 (latitude). Fonte: O Autor.

Existem ainda múltiplos outros tipos de gráficos e de mapas que podem auxiliar na compreensão da relação entre os dados, como: boxplots, que auxiliam na análise de métricas estatísticas e de distribuição de dados contínuos; mapas coropléticos, aplicados quando os dados geográficos vêm em formato poligonal como um conjunto de dados para avaliar a distribuição de um dado numérico; e mapas de calor temporais, caso além de dados geográficos e numéricos, o dataset apresente também um dado informando o tempo.

7.2 Pré-processamento de Dados

De acordo com sua distribuição ou seu formato, alguns dados, por vezes, precisam passar por um pré-processamento antes da execução de etapas seguintes de análise. As técnicas seguintes são algumas das principais realizadas nessa etapa de transformação dos dados:

7.2.1 Imputação de Dados

Por diversos motivos, um dataset pode apresentar dados faltantes, não apresentando nenhum valor para determinado atributo de algum elemento. Em determinadas ocasiões, quando um atributo com dados faltantes não é essencial ou a ocorrência de dados faltantes é muito baixa, descartar linhas (amostras) ou até colunas (atributos) poderia resolver a ausência de informações. Em outros casos, manter essas informações pode ser importante para a tarefa desejada.

Com isso, a imputação de dados se torna uma técnica importante para contornar essa questão. Dados categóricos faltantes podem ser substituídos pela moda (elemento de maior frequência), enquanto dados numéricos podem ser substituídos por 0, média ou mediana, por exemplo.

7.2.2 Padronização de Dados

Ao lidar com datasets com múltiplos atributos numéricos, as escalas podem ser distintas. Isso se torna prejudicial a modelos sensíveis a escala, como determinados modelos lineares de predição, algoritmos de redução de dimensionalidade e de clusterização, uma vez que podem considerar atributos em escalas maiores com uma relevância maior do que atributos em escalas menores. Esse é um passo crucial para ser realizado antes de passar para alguns dos principais modelos de aprendizado de máquina como Modelos Lineares, Redes Neurais e PCA.

A técnica de padronização é uma forma de fazer com que esses dados passem a ter o mesmo peso e, com isso, melhorar a performance de modelos. A padronização funciona de forma que para cada elemento em um espaço amostral (de treino por exemplo) do atributo numérico, remove-se a diferença da média de todos os elementos desse mesmo espaço e divide-se tal diferença pelo desvio padrão.

Para ilustrar a diferença, foi utilizado o mesmo conjunto de dados simulados da Figura 6. Os atributos se apresentam inicialmente em escalas diferentes: o atributo 1 varia entre -180 e 180; o atributo 2 entre -90 e 90; e o atributo 3 entre 0 e 1000. Os gráficos da Figura 14 demonstram a simulação, antes e depois da aplicação da técnica de

padronização:

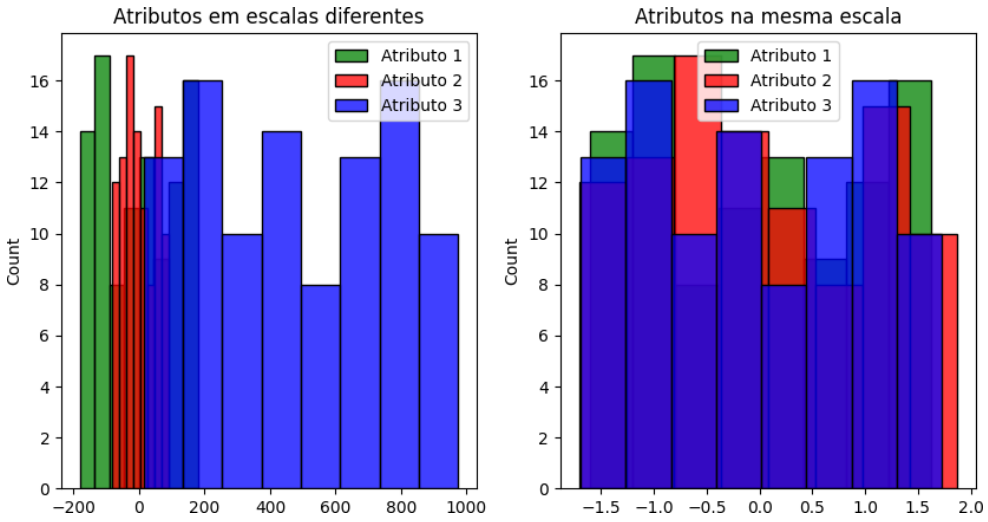


Figura 14: Gráficos representando variação de escalas antes e depois da padronização. Fonte: O Autor.

7.2.3 Encodings

Encodings são técnicas para transformar dados não numéricos em números. One Hot Encoding e Ordinal Encoding são alguns dos principais métodos usados para transformar dados categóricos em números, sendo o primeiro mais apropriado para dados nominais e o segundo para ordinais.

Como determinados modelos não possuem suporte para lidar com dados categóricos diretamente, utilizar técnicas de encoding para convertê-los para numéricos se torna um passo importante durante o pré-processamento de determinadas tarefas.

Um exemplo de aplicação de Encoding seria a transformação de um atributo categórico com valores "A", "B" e "C" em novas colunas, em que cada coluna criada indica a presença do respectivo valor em cada linha. Nesse caso, seria realizado o One Hot Encoding, de modo que, para cada linha, apenas uma das colunas recebe o valor 1 (indicando o valor correspondente da linha), enquanto as demais colunas recebem 0.

7.2.4 Tokenização, Stemização e Lematização

Essas três técnicas são utilizadas como forma de pré-processamento de texto.

Segundo os autores Müller e Guido (2016, Cap. 7) [22], a técnica de tokenização é um passo importante, no qual é definido o que consiste uma palavra, dividindo cada documento nas palavras presentes (*tokens*), utilizando espaços em branco e pontuações, por exemplo. Isso auxilia técnicas de extração de atributos, que seriam uma forma de encontrar uma melhor representação dos dados (Müller e Guido, 2016, Cap. 3) [22].

Já stemização e lematização são técnicas de normalização de texto, podendo ser bastante úteis em tarefas de processamento de linguagem natural. Ao se trabalhar com um dataset contendo dados textuais, flexões (de gênero, número e grau) de uma palavra podem não trazer um significado adicional, sendo mais vantajoso considerar um plural e um singular, por exemplo, como a mesma informação, sendo stemização e lematização técnicas possíveis para isso.

A stemização funciona transformando uma palavra em seu radical ou forma base, enquanto a lematização, por outro lado, leva em conta o contexto em que as palavras estão inseridas ao invés de simplesmente a reduzirem à forma base.

7.3 Correlação

Correlação é a métrica estatística utilizada para avaliar o quanto dois atributos estão relacionados entre si. Segundo Grus (2019, Cap. 5) [23], a correlação varia entre -1 e 1, de forma que, para dados numéricos:

- Uma correlação forte negativa (próxima de -1) indica que o aumento de um dos atributos está relacionado à redução do outro;
- Uma correlação forte positiva (próxima de 1) indica que o aumento de um dos atributos está relacionado ao aumento do outro, enquanto a redução de um está relacionada à redução de outro;

- Uma correlação de 0 indica que não há uma relação direta entre os atributos.

Seguindo com o dataset simulado, foi gerada uma matriz de correlação entre os atributos numéricos do dataset, utilizando a correlação de Pearson[6]. A Figura 15 é uma gráfico de mapa de calor que ilustra essa matriz:

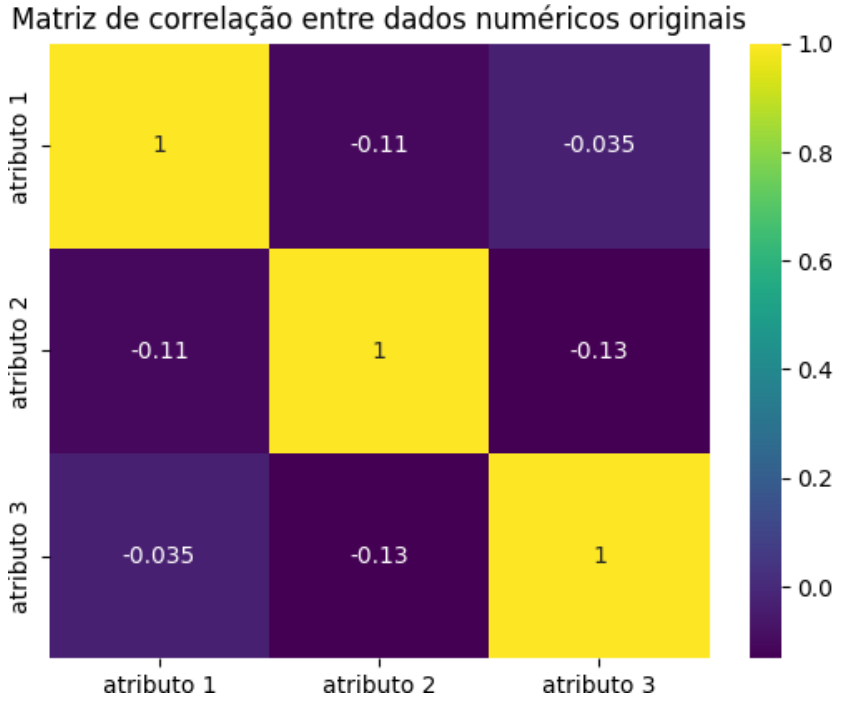


Figura 15: matriz de correlação com base nos dados simulados.
Fonte: O Autor.

Devido à geração dos três atributos ter sido de forma aleatória, não apresentam forte correlação entre si, tendo em vista que todas as correlações (com exceção de uma atributo com si próprio) tendem a 0.

Ao adicionar um atributo numérico a partir do segundo atributo (multiplicando-o por 10 e adicionando um pequeno ruído aleatório simulado), obteve-se uma matriz de correlação representada pelo gráfico da Figura 16:

[6] **Kent State University Library.** Pearson Correlation using SPSS Statistics. Disponível em: https://libguides.library.kent.edu/SPSS/PearsonCorr. Acesso em: 09 jun. 2024.

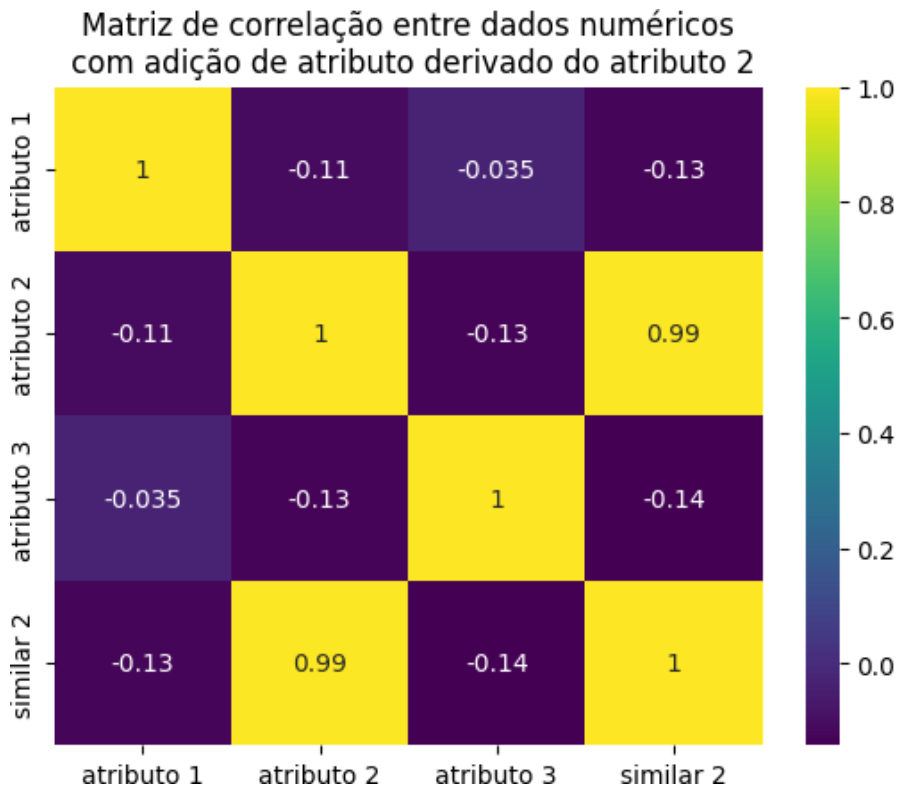


Figura 16: matriz de correlação com base nos dados simulados com adição de novo atributo derivado de um dos atributos. Fonte: O Autor.

Uma vez que o novo dado (atributo similar 2) foi gerado utilizando outro dado, multiplicando e adicionando uma leve distorção, a correlação entre o dado original e o dado derivado é bastante alta, muito próxima de 1 (0,99). Com isso, poderia ser avaliado que o novo dado não acrescenta tanta informação, visto que é diretamente correlacionado a outro atributo, e poderia ser descartado antes da aplicação de um modelo de aprendizado de máquina, por exemplo, de forma que, reduzindo o número de atributos utilizados, a performance do modelo poderia tornar-se mais rápida e mantendo resultados similares.

Existem outras métricas de correlação apropriadas para outras combinações de dados, sendo a métrica de Pearson mais apropriada para avaliar se existe uma relação linear entre dados numéricos.

7.4 Aprendizado de Máquina

Aprendizado de máquina ou aprendizado automático consiste em

fornecer dados para modelos, treiná-los e, a partir do que o modelo “aprendeu”, extrair informações ou realizar predições. Dois dos principais tipos de aprendizado de máquina são aprendizado não supervisionado e supervisionado.

Algoritmos supervisionados, como regressão e classificação, são treinados com o intuito de realizar a predição de um ou mais atributos, chamados de *target*, com base nos demais atributos do dataset. Já os algoritmos não supervisionados não apresentam um *target* a se predizer, como redução de dimensionalidade e clusterização.

Os quatro tipos de algoritmos a seguir foram pesquisados com base na documentação da biblioteca scikit-learn, a qual fornece, em Python, múltiplas implementações. Os exemplos de dataset também são provenientes da biblioteca.

7.4.1 Redução de Dimensionalidade

Algoritmos bastante importantes para realizar as análises, buscam simplificar conjunto de dados de alta dimensionalidade, ou seja, com muitos atributos.

Seus propósitos incluem facilitar a visualização dos dados - permitindo visualizar a distribuição dos dados em duas ou três dimensões - e melhorar o tempo de execução de algoritmos supervisionados e não supervisionados, tornando-os mais eficientes para casos de datasets muito complexos.

Alguns exemplos de algoritmos de redução dimensional são: PCA (Análise de Componentes Principais), um algoritmo de redução linear que busca realizar a projeção do dataset em um espaço dimensional menor; t-SNE (*T-Distributed Stochastic Neighbor Embedding*), o qual facilita a visualização de dados com grande dimensionalidade em um espaço bidimensional; e Latent Dirichlet Allocation, modelo utilizado para a descoberta de tópicos em um conjunto de dados textuais.

7.4.2 Clusterização

Clusterização consiste em separar os elementos do dataset em grupos (clusters) com base nos seus atributos, de forma que cada grupo apresente características similares.

Existem múltiplos algoritmos de clusterização, como KMeans e DBSCAN, cada um com suas características específicas e com possibilidade de aplicação em diferentes conjuntos de dados dependendo de suas particularidades.

O KMeans funciona melhor para casos com clusters de tamanhos semelhantes e pior para casos em que os clusters são alongados ou com formatos irregulares, sendo ainda necessário definir previamente o número de clusters. Já o DBSCAN, por outro lado, não requer uma definição prévia do número de clusters, performando melhor para encontrar clusters de densidades (tamanho) distintos e formatos irregulares.

7.4.3 Algoritmos de Regressão

Modelos que, a partir de um conjunto de treino, recebem um target para predizer e os atributos que serão utilizados como base para os cálculos e encontrar o *target*. O objetivo torna-se, na validação do modelo, reduzir a diferença entre o valor predito e o valor real do dado de *target*.

Diferentes modelos de regressão possuem diferentes características e são apropriados para lidar com problemas específicos de regressão. Para um dataset mais simples, a aplicação de regressão linear pode ser o suficiente para entender como os dados se relacionam e realizar predições adequadas, enquanto que para um mais complexo, com mais atributos e linhas, pode ser necessário um modelo mais robusto, como uma rede neural ou floresta aleatória.

Uma aplicação no mundo real seria buscar predizer o preço de uma

casa[7] a partir de dados como latitude e longitude, média de população da região, média de idade das casas da região, dentre outros. Com base nos demais atributos, um algoritmo de regressão poderia ser aplicado para encontrar o *target*, o qual seria o preço.

7.4.4 Algoritmos de Classificação

Similarmente aos algoritmos de regressão, na classificação os modelos são treinados e buscam encontrar um *target* categórico. Múltiplos tipos de métricas podem ser selecionadas para avaliar se um modelo apresenta bons resultados, sendo a escolha de métrica mais apropriada dependendo da tarefa.

Alguns modelos de classificação são: Regressão Logística, o qual, apesar do nome, é um modelo linear de classificação; Árvore de Decisão de Classificação; modelos de ensemble de classificação, os quais são agrupamentos de modelos base (geralmente árvores de decisão); e Support Vector Machines.

Vale ressaltar que a classificação de texto funciona utilizando atributos textuais para realizar a predição, sendo um exemplo classificar notícias[8] em categorias com base no texto dessas notícias.

7.5 Geocodificação

Como relatado na pesquisa de Padgham et al (2019), a técnica de geocodificação consiste em converter um endereço ou região em coordenadas geográficas. É possível também fazer uma geocodificação reversa, em que coordenadas são convertidas em um endereço.

Com a geocodificação é possível, então, converter dados textuais que representem uma localização para dados geográficos, em pontos de latitude e longitude.

---

[7] Dataset "*california housing*": https://inria.github.io/scikit-learn-mooc/python_scripts/datasets_california_housing.html. Acesso em: 9 jun. 2024.
[8] Dataset "*20 news group*": https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_20newsgroups.html#sklearn.datasets.fetch_20newsgroups. Acesso em: 9 jun. 2024.

7.6 Modelagem de Tópicos

De acordo com Müller e Guido (2016, Cap. 7) [22], é uma técnica de processamento de linguagem natural para agrupar elementos de um dataset com dados textuais em um mesmo tópico, em que cada um desses grupos possui documentos e palavras mais representativos.

O algoritmo de redução dimensional Latent Dirichlet Allocation (abordado na subseção 7.4.1) performa buscando grupos de palavras que possam definir um tópico. Assim, essa técnica pode ser aproveitada para encontrar informações relevantes sobre quais temas são abordados em um dataset com dados textuais.

7.7 Engenharia de Atributos

De acordo com Müller e Guido (2016, Cap. 4) [22], a engenharia de atributos é o processo de criação de novas variáveis a partir dos dados do conjunto. Múltiplas técnicas abordadas nessa seção podem ser utilizadas para promover essa criação de novos atributos, como: Clusterização, em que pode ser criado como novo atributo o cluster ao qual o dado pertence; Modelagem de Tópicos, cujo novo atributo indica o tópico identificado pertencente; e One Hot Encoding, que gera novos atributos binários representando cada valor categórico do atributo original.

Esses novos atributos podem potencializar bastante as análises ao demonstrar novas informações presentes nos dados e podem contribuir para a execução de modelos preditivos.

7.8 Análise de Séries Temporais

De acordo com Nielsen (2019, Cap. 1) [24], a análise de uma série temporal consiste em extrair informação de dados organizados em ordem cronológica, buscando aplicar um diagnóstico para os dados históricos do passado e buscar predizer os dados do futuro.

Múltiplas aplicações de séries temporais são listadas, sendo responsáveis pelo incentivo ao desenvolvimento da técnica, como na medicina - com o uso de equipamentos como eletrocardiogramas e

eletroencefalogramas para a aplicação de diagnósticos dos pacientes - e na meteorologia - para fazer previsões do tempo com base em dados coletados de estações climáticas -, demonstrando como a utilização dessa técnica pode potencializar as análises sobre um dataset contendo dados temporais.

7.9 Conclusão sobre Técnicas

Nessa seção foram abordadas múltiplas técnicas que podem ser aplicadas entre os diferentes tipos de dados listados na seção 5. Existem ainda diversas outras técnicas que não foram discorridas e poderiam incrementar as análises em cima de um dataset, como seleção de atributos, transformações polinomiais, detecção de anomalias, geração de grafos de conhecimento e análise de sentimento, por exemplo.

Dadas as técnicas expostas, o planejamento para o software foi, a partir dos tipos dos atributos do dataset e suas características (como distribuição e incidência de dados faltantes), sugerir quais técnicas poderiam ser aplicadas, buscando auxiliar o usuário a aprender sobre tarefas de ciências de dados e compreender suas informações disponíveis.

8. Tipos de Datasets

Um mesmo conjunto de dados pode apresentar múltiplos tipos de dados diferentes, o que faz com que técnicas diferentes possam ser necessárias para fazer as análises corretamente. Além disso, algumas técnicas são apropriadas para lidar com múltiplos atributos de um mesmo tipo de dados.

Na Tabela 3 a seguir, são comentadas sobre algumas técnicas - apresentadas e discutidas na seção 7 - possíveis de serem aplicadas dependendo de combinações de tipos dados presentes em um conjunto, incluindo técnicas de visualização de dados, pré-processamento, engenharia de atributos e aplicação de aprendizado de máquina:

Tipos de dados presentes	**Técnicas possíveis**
Numérico	Histograma Imputação Padronização
Categórico	Gráfico de Barra Imputação Encoding
Múltiplos atributos numéricos	Gráfico de Dispersão Gráfico de Linha Imputação Padronização Correlação Regressão Redução de dimensionalidade Clusterização
Múltiplos atributos categóricos	Imputação Encoding Classificação Redução de dimensionalidade Clusterização

Numérico e categórico	Gráfico de Linha Imputação Padronização Encoding Redução de dimensionalidade Clusterização Regressão (*target* numérico) Classificação (*target* categórico)
Geoespaciais	Construção de mapas Geocodificação
Numérico e geoespacial	Mapa de calor Mapa coroplético
Categórico e geoespacial	Mapa categórico
Textual	Stemização Lematização Modelagem de Tópico
Categórico e textual	Classificação de texto
Geoespacial e textual	Geocodificação
Numérico e temporal	Gráfico de Linha Padronização Encoding Série temporal
Categórico e temporal	Gráfico de Dispersão Gráfico de Linha Encoding Série Temporal
Numérico, geoespacial e temporal	Mapa de calor temporal

Tabela 3: Técnicas aplicáveis para diferentes combinações de tipos de dados.

O software proposto no trabalho, portanto, buscaria sugerir as técnicas possíveis a partir da identificação dos tipos de dados presentes no dataset fornecido pelo usuário.

9. Soluções Alternativas

Nessa seção serão abordados alguns tópicos de formas alternativas para executar a implementação do software proposto.

9.1 Profiling dos Tipos de Dados

Uma forma de promover a identificação de seus dados seria com apoio da biblioteca do pandas, usando suas funcionalidades de exploração, auxiliando na identificação das características dos dados.

De acordo com Couto et al (2022) [25], algumas informações podem ser extraídas com o profiling aplicado a big data como, por exemplo, estatísticas, qualidade e padrões de um conjunto de dados. No artigo, é comentada sobre a possibilidade de realizar o profiling com o apoio de metadados que descrevam o dataset. Tendo em vista que esses metadados podem auxiliar na identificação de características do dataset fornecido, uma alternativa de implementação do sistema seria permitir que os usuários adicionassem metadados e utilizá-los para identificar os tipos de dados presentes e, assim, melhorar a sugestão de técnicas.

Uma outra opção ainda seria fazer uso de um LLM para apoiar na identificação dos tipos de dados, integrando-o, por meio de sua API, com o sistema, o que poderia auxiliar na análise sobre as características dos dados.

Para a primeira versão do sistema, o uso do pandas será utilizado para promover o Profiling, sem a integração de apoio de metadados ou LLM.

9.2 Técnicas Sugeridas

Uma forma de verificar quais técnicas podem ser aplicadas seria utilizando uma base de conhecimento, baseando-se nas características do conjunto fornecido pelo usuário. A base funcionaria como um agente lógico treinado para identificar, por exemplo, que a partir de um dataset apresentando colunas com dados numéricos e geoespaciais, seria possível construir um mapa de calor para ampliar as análises ao

identificar esses tipos de dados, como ilustrado na Tabela 3 (seção 8), a qual seria utilizada como uma referência para promover a definição das técnicas que podem ser aplicadas.

De forma alternativa, ainda seria possível integrar um LLM, fornecendo ao modelo o conjunto de dados ou detalhando os tipos de dados identificados em um prompt. Tendo esses dados, o LLM poderia retornar as técnicas acompanhadas de uma explicação sobre o motivo dessas recomendações.

Inicialmente, a alternativa de implementação para a identificação será somente o uso da base de conhecimento.

9.3 Integração com Outras Ferramentas

Ao identificar determinadas características de um dataset, seria possível promover as sugestões de algumas técnicas não implementadas diretamente no sistema, recomendando também ferramentas que forneçam tal implementação. Por exemplo, a ferramenta ArcGIS, como abordado anteriormente, apresenta suporte para algumas técnicas de estatística espacial, como a análise de padrões geográficos. Caso o sistema identificasse que tal técnica seria valiosa para a análise de um dataset, poderia ser sugerido o ArcGIS como referência para poder aplicá-la no conjunto de dados fornecido.

Além disso, seria possível fazer uso direto de funcionalidades de algumas outras ferramentas abordadas na seção 6, integrando-as no sistema por meio de plugins ou APIs. Por exemplo, ao invés de desenvolver com código uma técnica de análise de dados geoespaciais, poderia ser feito o uso direto da ferramenta ArcGIS e utilizar de sua grande gama de funcionalidades para lidar com esse tipo de dado.

A alternativa escolhida para integrar, inicialmente, seria a de sugerir ferramentas que executem as técnicas não implementadas diretamente.

9.4 Ferramenta Extensível

Seria possível, todavia, montar uma solução extensível para

aplicação das técnicas a serem utilizadas, de forma que um usuário utilizasse um arquivo de base de conhecimento para adicionar novas técnicas a serem sugeridas, podendo incluir a sugestão de ferramentas que as executem. Dessa forma, não seria necessário atualizar diretamente o código fonte para que mais técnicas pudessem ser sugeridas.

Além disso, ainda existe a alternativa de tornar o sistema um projeto *Open Source*, o que possibilitaria que desenvolvedores alterassem o código fonte e adicionassem mais funcionalidades sobre técnicas e tipos de dados aceitos pela ferramenta.

A alternativa inicial escolhida para tornar o sistema extensível será a do uso de arquivo de base de conhecimento.

10. Requisitos

Para melhor entendimento dos objetivos do sistema, a seguir são listados os requisitos do software:

[Req01] O sistema deve permitir que os usuários façam upload de arquivos de extensão csv.

[Req02] O sistema deve sugerir técnicas de ciência de dados para os usuários com base no dataset fornecido.

[Req03] O sistema deve listar informações sobre as técnicas sugeridas, como seu funcionamento, o motivo da sugestão e resultados esperados ao aplicar a técnica.

[Req04] O sistema deve identificar os tipos de dados dos atributos como numérico, categórico, geoespacial, textual ou temporal.

[Req05] O sistema deve relatar ao usuário caso não tenha conseguido identificar o tipo de dados de alguns dos atributos no dataset fornecido.

[Req06] O sistema deve fornecer a opção de o usuário alterar o tipo e o subtipo de dado caso seja identificado de forma incorreta.

[Req07] O sistema deve permitir que o usuário faça download de um arquivo de extensão csv, json ou txt com metadados identificados após o Profiling.

[Req08] O sistema deve permitir que o usuário faça download de imagens geradas nas visualizações de dados.

[Req09] O sistema deve permitir que o usuário expanda a base de conhecimento, incluindo a sugestão de novas técnicas aplicáveis a datasets.

[Req10] O sistema deve informar ao usuário caso o arquivo fornecido não tenha sido processado corretamente pelo sistema.

[Req11] O sistema deve apresentar uma mensagem de erro caso a base de conhecimento não seja válida.

11. Modelo de Dados

Para essa versão inicial, a ferramenta desenvolvida não armazena os dados do dataset fornecidos pelo usuário, nem suas informações, pois não possui funcionalidade de cadastramento (uma vez que não é um requisito). A única persistência existente é para a base de conhecimento (de regras) e para o arquivo com informações sobre as técnicas, conforme descrito, posteriormente, na seção 16.3.

Em decorrência disso, não houve a necessidade de definir um modelo de dados para armazenar as informações dos usuários para essa versão e, assim, o sistema não necessita de um banco de dados, simplificando sua arquitetura.

12. Esboço de Casos de Uso

Nessa seção são abordados casos de uso planejados para o sistema desenvolvido, servindo como documentação para o funcionamento. A Figura 18 ilustra um diagrama de casos de uso representando funcionalidades do sistema e a interação com os usuários.

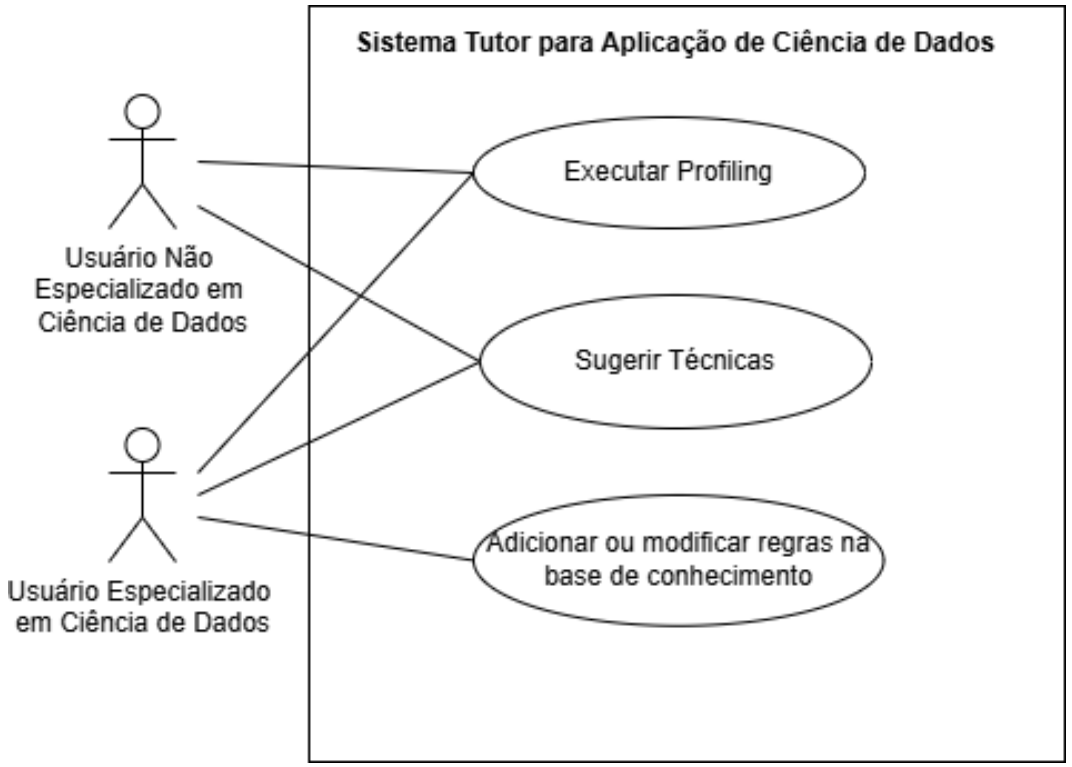


Figura 18: Diagrama de Casos de Uso.

Caso de Uso UC01

Nome:	Executar Profiling
Objetivo:	Permitir que um usuário verifique resultado do Profiling
Atores:	Usuário
Pré-condição:	Usuário possui conjunto de dados
Pós-condição (cenário de sucesso):	Usuário consegue visualizar resultados do Profiling

Pós-condição (cenário de insucesso):	Usuário não consegue visualizar resultados do Profiling
Trigger:	Usuário fornece arquivo csv para o sistema
Fluxo Principal:	1. Sistema apresenta a opção de fornecer arquivo de extensão csv. [Req01] 2. Usuário fornece arquivo csv para o sistema. 3. Sistema processa o arquivo. [E1] 4. Sistema mostra tabela de tipos e subtipos identificados para as colunas que foram identificadas e informa as colunas que não foram identificadas. [Req04][Req05][A1] 5. Usuário seleciona opção de “Baixar Metadados” em um dos três formatos de arquivo (csv, json ou txt). [Req07]
Fluxos Alternativos:	[A1] Usuário deseja corrigir o tipo ou subtipo de alguma coluna: 1. Usuário seleciona opção do tipo ou subtipo correto 2. Usuário seleciona a opção de “Corrigir”; [Req06] 3. Sistema retorna para pass 4 do Fluxo Principal
Fluxos de Exceção	[E1] O sistema não consegue processar o arquivo. 1. O sistema apresenta a mensagem “Não foi possível realizar a leitura de arquivo do dataset fornecido”. [Req10] 2. O sistema retorna para o passo 1 do fluxo principal.

Caso de Uso UC02

Nome:	Sugerir Técnicas
Objetivo:	Permitir que um usuário verifique técnicas sugeridas
Atores:	Usuário
Pré-condição:	Usuário possui conjunto de dados
Pós-condição (cenário de sucesso):	Usuário consegue verificar técnicas sugeridas
Pós-condição (cenário de insucesso):	Usuário não consegue verificar técnicas sugeridas

Trigger:	A opção “Sugestão de Técnicas” na tela de resultados do Profiling é clicada
Fluxo Principal:	1. Sistema apresenta visualizações iniciais. [Req02][E1] 2. Usuário seleciona expansão de informações sobre técnicas de visualização sugeridas. [Req03] 3. Sistema apresenta técnicas sugeridas (além das técnicas de visualização). [Req02] 4. Usuário seleciona expansão de informações sobre as demais técnicas sugeridas. [Req03]
Fluxos Alternativos:	
Fluxos de Exceção	[E1] O arquivo de base de conhecimento apresenta erros. 1. O sistema apresenta a mensagem “Base de conhecimento inválida”. [Req11] 2. O sistema retorna para a tela inicial.

Caso de Uso UC03

Nome:	Adicionar ou modificar regras na base de conhecimento
Objetivo:	Permitir que um usuário faça alterações na base de conhecimento
Atores:	Usuário
Pré-condição:	Usuário
Pós-condição (cenário de sucesso):	Usuário consegue modificar base de conhecimento
Pós-condição (cenário de insucesso):	Usuário não consegue modificar base de conhecimento
Trigger:	A opção “Editar base de regras” na tela inicial é clicada
Fluxo Principal:	1. Sistema apresenta a base de conhecimento original. [Req02] 2. Usuário modifica base, modificando ou adicionando alguma regra como em um editor de texto. [Req09][A1] 3. Usuário confirma as modificações feitas selecionando a opção “Confirmar”. [E1]

Fluxos Alternativos:	[A1] Usuário verifica a base e não deseja realizar modificações: 1. Usuário seleciona a opção “Cancelar”
Fluxos de Exceção	[E1] Alguma modificação do usuário é inválida: 1. O sistema apresenta a mensagem “Base de conhecimento inválida”. [Req11] 2. O sistema retorna para a tela inicial.

13. Esboço de Interface

Essa seção aborda sobre esboços de telas planejadas para a ferramenta desenvolvida para o projeto.

A Figura 19 apresenta a tela inicial do sistema, a qual apresenta uma opção para fornecer um arquivo de extensão csv para promover as análises:

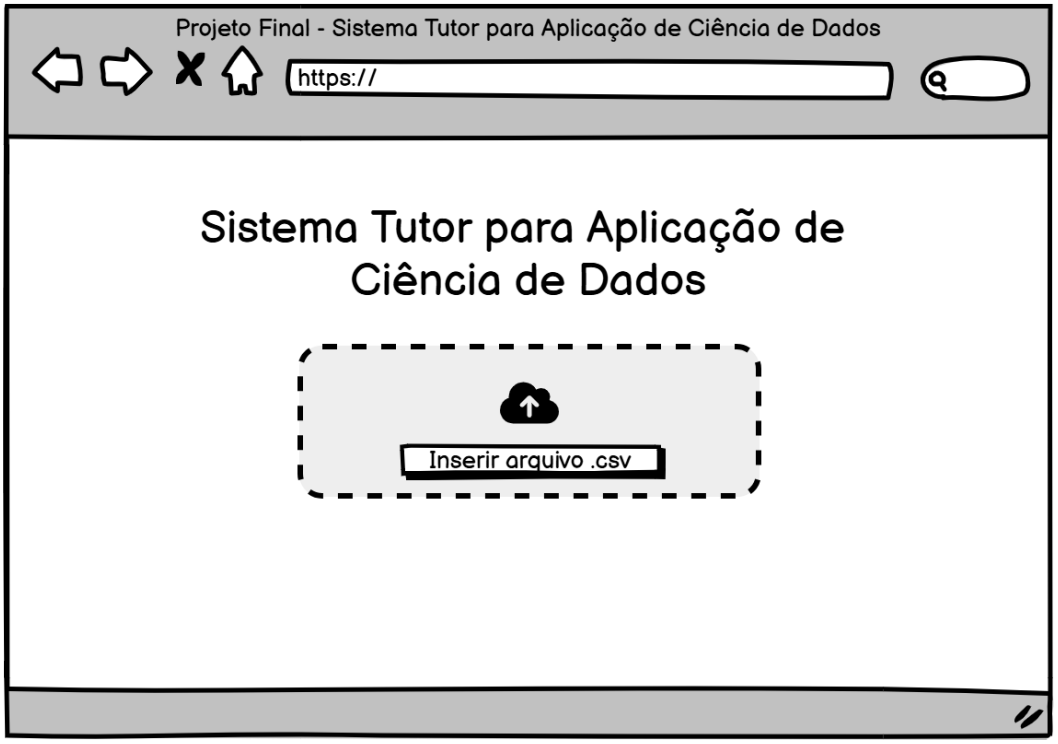


Figura 19: Tela Inicial do Sistema.

Em seguida, a Figura 20 apresenta a tela de resultado de Profiling, com a identificação de tipos e subtipos de colunas para o arquivo fornecido, permitindo que o usuário faça correções:

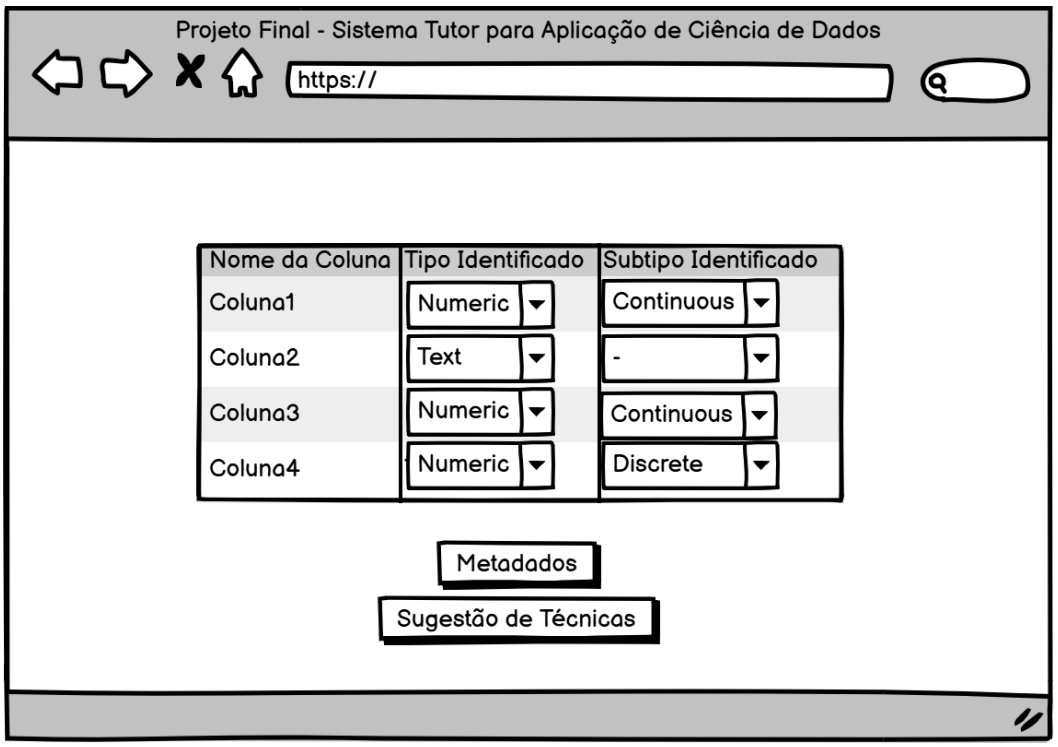


Figura 20: Tela de Resultados do Profiling.

Selecionando a opção de “Sugestão de Técnicas”, é apresentada a tela da Figura 21, exibindo as técnicas de visualizações iniciais, permitindo que o usuário expanda as informações sobre a técnica, como na Figura 22:

Figura 21: Tela de Visualizações Iniciais apresentadas após o Profiling.

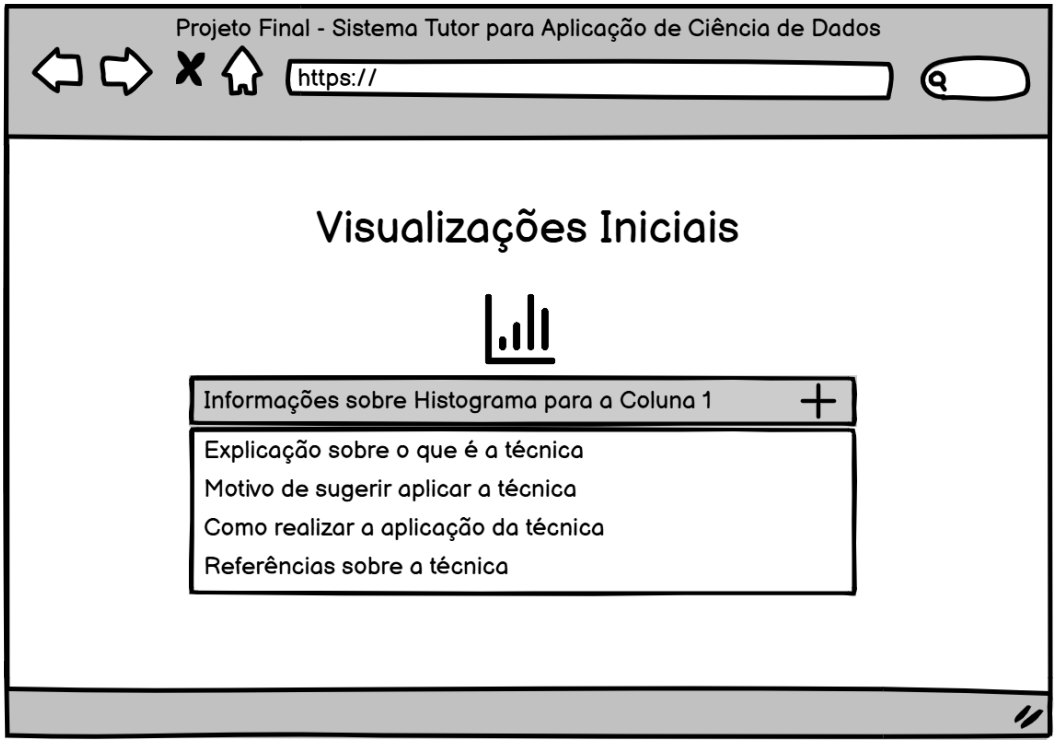


Figura 22: Tela com expansão de informações sobre uma técnica de visualização.

Por fim, após apresentadas as visualizações iniciais, são expostas as demais técnicas sugeridas, como na tela da Figura 23, a qual também apresenta a opção de expandir para mais informações sobre a técnica, como na Figura 24:

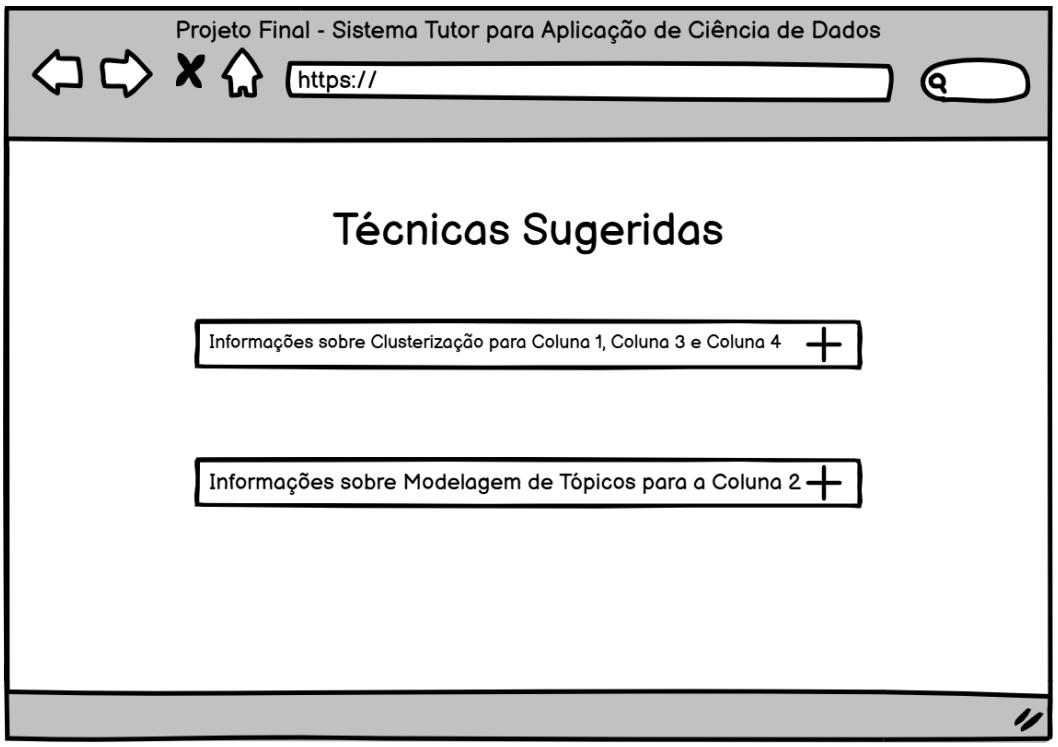


Figura 23: Tela de Técnicas Sugeridas.

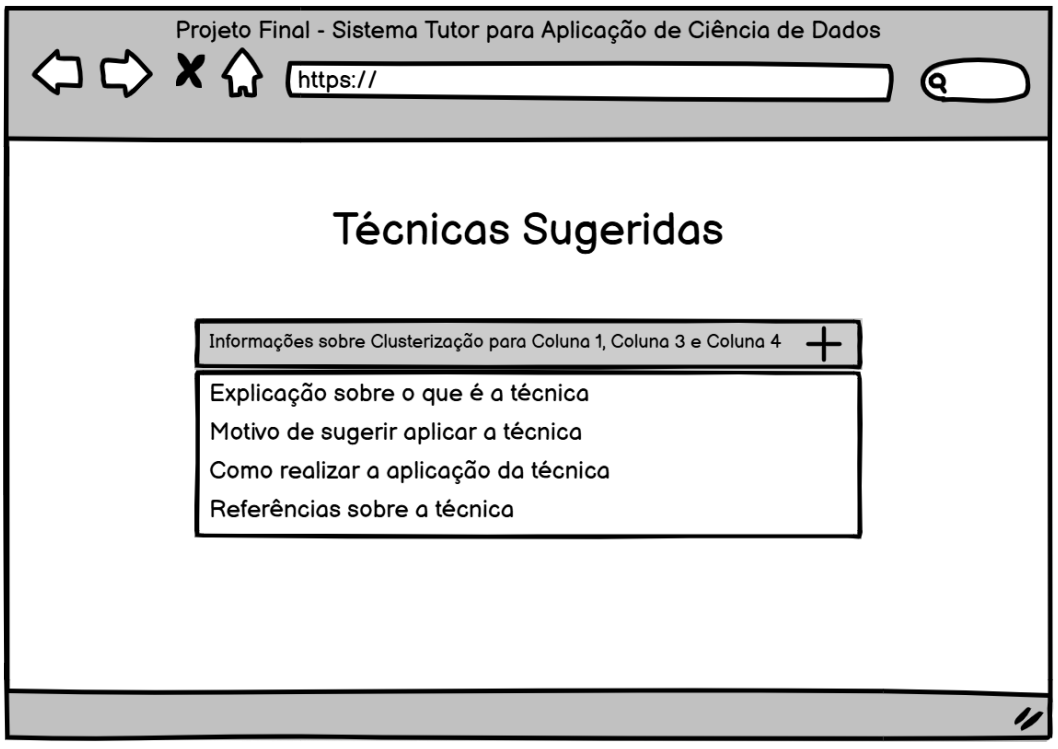


Figura 24: Tela com expansão de informações sobre uma técnica sugerida.

14. Tidy Data

Com os avanços em pesquisas e desenvolvimento na área de ciência de dados, torna-se necessário o surgimento de boas práticas ao realizar tarefas nesse campo. Uma ideia sugerida é a de Tidy Data, propondo um formato de dados que seja limpo e adequado para as análises.

Para o projeto, é fundamental adotar um modelo mais padronizado para os conjuntos de dados, a fim de viabilizar sugestões de análises adequadas e a execução automática de algumas delas. Dado o objetivo didático do sistema, a padronização dos arquivos é necessária para garantir um melhor entendimento dos usuários.

Vale ressaltar, contudo, que atualmente, devido à grande quantidade de fontes de dados distintas, surge também uma grande variedade de dados, dificultando encontrar um padrão. Dessa forma, por mais formatos que o sistema aceite, de certa forma haverá limitações a respeito de como os dados podem ser fornecidos para a ferramenta.

Segundo Tierney e Cook [26], as boas práticas de Tidy Data são limitadas para dados com valores faltantes, com múltiplos softwares voltados para a visualização e análise de dados apenas descartando-os. Entretanto, como abordado na seção 7.2.1, apenas descartar dados faltantes pode levar à perda de informações valiosas sobre o conjunto. Tendo isso em vista, vale considerar a importância de promover um tratamento adequado para dados faltantes, como, inicialmente, analisar visualmente os dados faltantes (caso presentes no dataset) e, posteriormente, tratar esses dados com ferramentas adequadas, dependendo de suas características.

Para seguir em um formato apropriado de Tidy Data, os autores defendem que algumas premissas devem ser feitas sobre os dados, como:

- Cada variável deve apresentar sua própria coluna;
- Cada dado observado deve apresentar sua própria linha;
- Cada valor deve estar em sua própria célula (ou seja, sua própria linha e própria coluna).

Essas premissas sobre a estrutura do conjunto são importantes para permitir a utilização de ferramentas de análise e visualização dos dados.

Tendo isso em vista, destaca-se que atributos multivalorados violariam essas premissas, uma vez que conteriam mais de um valor por célula. Na versão inicial do sistema, portanto, será exposto um disclaimer sobre essa informação e dados multivalorados serão tratados como strings (podendo ser tratados dados categóricos ou textuais). Para passos futuros, pode ser feita uma análise mais aprofundada para atributos multivalorados e separá-los em diferentes colunas.

15. Prolog

A linguagem de programação Prolog, utilizada em áreas de inteligência artificial, sistemas de recomendação e sistemas de apoio à decisão, apresenta um formato de incluir lógica em programação de forma simples e direta. Como exemplo, o trabalho de Dallos Parra et al. (2021) [27] descreve sobre o desenvolvimento de um sistema com recomendações baseadas em Prolog para a gestão de lixo eletrônico. O sistema, assim, faz uso de regras em Prolog para promover, a partir das informações fornecidas pelo usuário, a recomendação mais adequada para lidar com os resíduos.

Dada a sua facilidade de implementar a definição de regras, fatos e relacionamentos de variáveis e constantes, o Prolog foi escolhido para implementar a sugestão de técnicas propostas para o conjunto de dados fornecido pelo usuário do sistema. Uma das características valiosas do Prolog é permitir verificar quais instância de termos nos fatos que tornam determinado predicado verdadeiro, o que contribui para a proposta do sistema.

15.1 Funcionamento da Linguagem

Segundo Davis (1985) [28], na programação lógica, são declarados fatos e regras, definindo como termos na linguagem, sendo variáveis ou constantes, podem estar relacionados. Isso permite verificar se determinada informação sobre um termo é verdadeira e realizar *queries* baseadas nas relações.

Para ilustrar esse funcionamento, os exemplos de código a seguir demonstram o Prolog usando como exemplo a relação entre alguns personagens da obra *Game of Thrones*.

Inicialmente, são definidos no Código 1 como verdadeiros os fatos de o termo cersei ser *parent* dos termos joffrey, myrcella e tommen, além de o termo tywin ser *parent* do termo cersei:

Código 1: definição de fatos.

```
%                Facts
parent(cersei, joffrey)
parent(cersei, myrcella)
parent(cersei, tommen)
parent(tywin, cersei)
```

Em seguida, no Código 2, são definidas como regras que dois termos são *siblings* caso sejam diferentes e tenham o mesmo *parent* e que uma termo X é *grandparent* de um termo Y caso X seja *parent* de um termo que seja *parent* de Y:

Código 2: definição de regras.

```
% Rules
siblings(X, Y) :- parent(Z, X), parent(Z, Y), X \= Y
grandparent(X, Y) :- parent(X, Z), parent(Z, Y)
```

Com esses elementos definidos, podem ser executadas verificações, como a do Código 3, para verificar se o termo tywin é *grandparent* do termo joffrey (cujo, resultado é verdadeiro):

Código 3: verificar se resultado de uma *query* é verdade.

```
?- grandparent(tywin, joffrey).
true.
```

Também é possível executar *queries*, para averiguar, como no exemplo do Código 4, os *siblings* do termo tommen, apresentando como resultado os termos joffrey e myrcella:

Código 4: *query* para encontrar termos que sejam *siblings* do termo tommen

```
?- siblings(X, tommen).
X = joffrey;
X = myrcella .
```

15.2 Prolog no Sistema

Ao identificar os tipos e demais características dos dados no conjunto

fornecido, podem ser definidos os fatos. Junto com uma base pré-definida de regras, a utilização do Pyswip [29] - interface de Prolog em Python - permitiria a execução de *queries* e verificações de fatos, permitindo, assim, retornar as sugestões de forma prática, sem a necessidade de múltiplos loops e condições no código, tornando-o mais legível.

15.3 Extensibilidade das Sugestões

Como abordado na seção 9.4 sobre a extensibilidade de sugestões, o uso de Prolog permite uma fácil inclusão de novas regras das sugestões de técnicas, dado que precisaria somente incluir mais regras em um arquivo de base, permitindo expandir as sugestões promovidas para o conjunto.

16. Desenvolvimento do Sistema

O sistema desenvolvido na linguagem Python, faz uso de bibliotecas e interfaces como:

- pandas: manipulação dos conjuntos de dados;
- streamlit [30]: framework para a implementação de frontend do sistema;
- Pyswip: integração com Prolog;
- folium, wordcloud [31] e plotly [32]: visualizações de dados - mapas interativos, nuvens de palavras e múltiplos gráficos, respectivamente.

O diagrama na Figura 25 demonstra o funcionamento do sistema com os seus principais componentes:

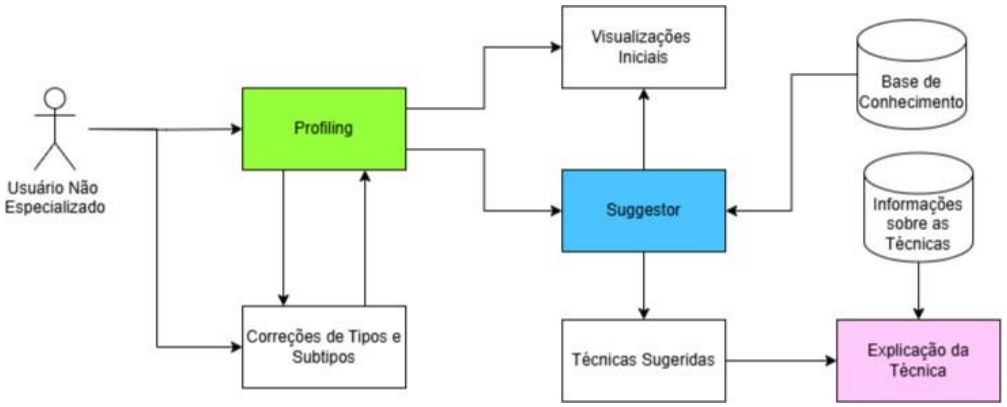


Figura 25: Diagrama de componentes do sistema.

Com cores destacadas, se encontram os três principais componentes do sistema: o módulo de Profiling, responsável por promover as identificações sobre as colunas, como abordado na seção 16.2; o módulo Suggestor, responsável por promover as sugestões de técnicas com base nos resultados do Profiling e nas regras definidas na base de conhecimento (Apêndice A), como discorrido na seção 16.3; e a funcionalidade de explicação de técnicas sugeridas (com as informações presentes no Apêndice B), explorando o intuito didático da ferramenta, também exposta na seção 16.3.

Dessa forma, um usuário utiliza o sistema fornecendo um arquivo csv, o qual é analisado pelo módulo de Profiling. A partir disso, o usuário pode efetuar uma correção de tipos ou subtipos, caso o sistema tenha atribuído-os de forma incorreta para alguma coluna do dataset. Em seguida, o usuário pode ver as técnicas sugeridas, tanto as de visualizações iniciais quanto demais técnicas - que envolvam pré-processamentos ou engenharia de atributos, por exemplo -, podendo então, expandir as informações para ter acesso a explicações sobre as técnicas sugeridas.

Esses módulos e funcionalidades do sistema serão abordados com mais detalhes a seguir. Para ilustrar o funcionamento da ferramenta desenvolvida, será utilizado o dataset *Earthquake Data Overview*[9], descrevendo dados sísmicos em múltiplas regiões do mundo, seguindo com premissas de Tidy Data e não apresentando atributos multivalorados. A Figura 26 demonstra as colunas do dataset e suas cinco primeiras linhas:

Place	Latitude	Longitude	Country	Continent	Magnitude
Bamako	12.6354	-8.0023	Mali	Africa	4.7
Niamey	13.513	2.1151	Niger	Africa	5.7
Southern Chile	-39.8234	-73.0691	Chile	South America	4.9
Freetown	8.4815	-13.2315	Sierra Leone	Africa	4.8
Bamako	12.6422	-7.999	Mali	Africa	5.3

Figura 26: Tabela ilustrando as colunas e as cinco primeiras linhas do Dataset com dados sísmicos.

16.1 Recebimento do Conjunto de Dados

Nessa versão inicial do sistema, são aceitos apenas arquivos com extensão csv. Embora a inclusão de arquivos nos formatos json e xml tenha sido considerada, muitos dos arquivos usados para teste estavam em formatos variados, o que dificultou seu processamento e a automatização da identificação de tipos. Além disso, alguns arquivos json e xml apresentavam hierarquias complexas, resultando em atributos

[9] "Earthquake Data Overview": https://www.kaggle.com/datasets/vizeno/earthquake-data-overview. Acesso em 24 out. 2024.

multivalorados, os quais - para essa primeira versão da ferramenta - ainda não são analisados.

Dado o objetivo didático da ferramenta, buscou-se priorizar uma implementação que leve em conta o Tidy Data para permitir a sugestão de análises de uma forma automática. Dessa forma, datasets mais limpos apresentam melhores resultados nas sugestões e análises.

Ao entrar na página inicial do sistema, será exibido um display para selecionar um arquivo csv, como na Figura 27:

Figura 27: Página inicial do sistema, antes de fornecer um dataset.

16.2 Módulo de Profiling dos Dados

O Profiling dos dados funciona de forma a averiguar as características dos dados: o percentual de dados faltantes, os tipos identificados - bem como os subtipos se houver - e a proporção de valores únicos.

16.2.1 Identificações

Com apoio do pandas e utilizando decisões condicionais, os tipos de cada atributo são identificados pelo sistema.

Como discutido anteriormente na seção 5, os cinco tipos de dados no escopo são: numérico, categórico, geoespacial, textual e temporal. Nesse trabalho, os subtipos especificados são os de dados numéricos e geoespaciais: os dados numéricos podem ser classificados como contínuos ou discretos, enquanto os dados geoespaciais podem ser representados por um par de latitude e longitude ou por nomes que indicam um local geográfico. Essa escolha para os dados geoespaciais

está alinhada com os princípios do Tidy Data. Embora seja possível considerar um atributo que represente um conjunto de pontos ou uma única coordenada, isso exigiria uma análise específica para lidar com atributos multivalorados, o que, para preservar a simplicidade e seguir as premissas do Tidy Data, ficará fora do escopo desta versão do trabalho.

Como discorrido na seção 5.2, dados categóricos ainda podem ser subdivididos em ordinais ou nominais. Entretanto, durante o desenvolvimento notou-se uma dificuldade de diferenciar automaticamente essa diferença e, por isso, não foram estabelecidos subtipos para colunas categóricas.

As identificações são feitas utilizando a proporção de valores únicos (não nulos) de um atributo, seu nome e sua categoria na estrutura de dados do Pandas após fazer o processamento do arquivo. Essas identificações, assim, seguem o formato do diagrama da Figura 28:

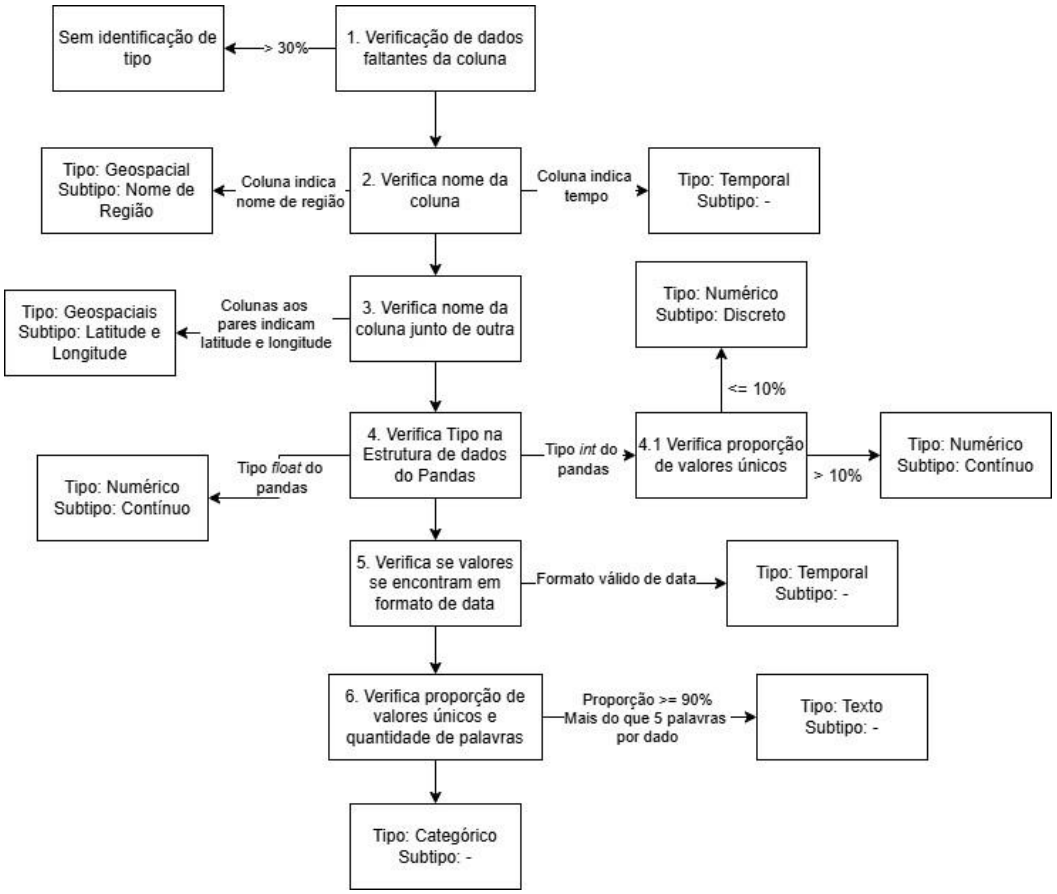


Figura 28: Diagrama de etapas do módulo de Profiling para a identificação de tipos e subtipos.

Essas etapas são executadas para cada atributo que ainda não tenha sido identificado:

- Etapa 1: inicialmente é verificado o percentual de dados faltantes da coluna:
  - Se esse percentual for maior que 30%, o módulo não identifica seu tipo ou subtipo;
  - Caso contrário, o módulo passa para a Etapa 2.
- Etapa 2: módulo verifica se é possível identificar o tipo de um atributo a partir de seu nome:
  - Se o nome indica temporalidade (como *day*, *month*, *year* ou *date*, por exemplo), o tipo atribuído à coluna é temporal;
  - Se o nome indica um espaço geográfico (como *country*, *city* ou *continent*, por exemplo), o tipo atribuído à coluna é geoespacial e o subtipo é nome de região;
  - Se o módulo não tiver identificado a partir do nome, passa para a Etapa 3.
- Etapa 3: módulo identifica se os nomes de colunas aos pares indicam o tipo:
  - Se o nome da coluna indica latitude e o nome de outra indica longitude, o tipo atribuído a ambas é geoespacial, com uma tendo o subtipo latitude e a outra o subtipo longitude;
  - Caso contrário, o módulo passa para a Etapa 4.
- Etapa 4: módulo verifica o tipo da estrutura de dados do Pandas:
  - Se o tipo for *float* o tipo atribuído é numérico e o subtipo é contínuo;
  - Se o tipo for *int* , o módulo passa para a sub-etapa 4.1:
    - Se a proporção de dados únicos for menor que 10% o tipo é numérico e o subtipo é discreto;
    - Caso contrário, o tipo é numérico e o subtipo é contínuo.
  - Caso contrário, o módulo passa para a etapa .
- Etapa 5: módulo verifica se os dados da coluna apresentam

um formato válido de datas (como dia/mês/ano e ano/mês/dia, por exemplo):

- Se coluna apresenta formato válido, o tipo é temporal;
- Caso contrário, passa para a etapa 6.

- Etapa 6: módulo verifica percentual de dados únicos e quantidade de palavras nos valores:
  - Se o percentual de dados únicos for maior que 90% e todos os valores não nulos apresentarem 5 ou mais palavras, o tipo atribuído é textual;
  - Caso contrário, o tipo atribuído é categórico.

Para o exemplo, o sistema conseguiu identificar tipos para todos os atributos - uma vez que não apresenta dados faltantes em nenhuma de suas colunas -, como demonstrado na Figura 29, identificando os seguintes tipos e subtipos:

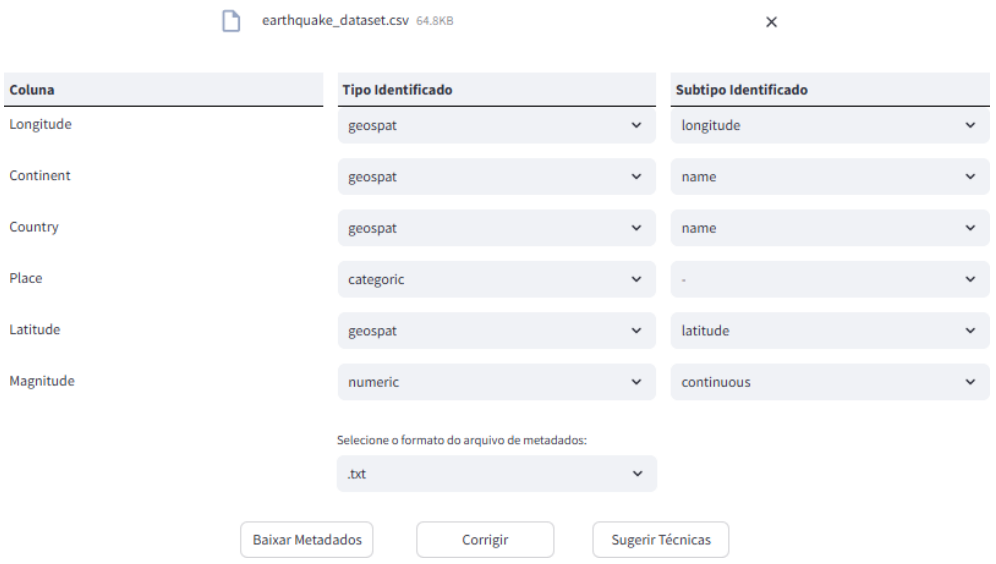


Figura 29: Resultado do Profiling de dados para o conjunto de dados.

Essas identificações ocorreram porque:

- Os atributos *Country* e *Continent*, devido a seu nome, são atribuídos ao tipo geoespacial, com o subtipo nome da região;
- Os atributos *Latitude* e *Longitude*, identificados pelo nome aos pares, são atribuídos ao tipo geoespacial, com subtipo latitude

e longitude, uma vez que o par dessas colunas indica a coordenada de ocorrência de cada linha;

- O atributo *Magnitude* foi identificado pelo pandas como um tipo *float*, por isso o tipo atribuído é numérico e o subtipo é contínuo;
- O atributo *Place* foi identificado como categórico por não se encaixar nos demais tipos na identificação do Profiling.

Como observado na Figura 29, há 3 botões após resultado do Profiling: “Baixar Metadados”, em que o sistema realiza o download de um arquivo no formato selecionado contendo os resultados do Profiling; ”Corrigir”, que efetua as modificações realizadas pelo usuário; e “Sugerir Técnicas”, ativando o módulo Suggestor do sistema.

16.2.2 Correções

O sistema exibe os tipos e subtipos identificados em *select boxes*, permitindo ao usuário corrigi-los manualmente caso o sistema tenha feito alguma identificação incorreta para o dataset, de forma a corrigir possíveis erros sem comprometer as análises e sugestões realizadas.

Para ilustrar essa correção, o tipo do atributo *Continent* foi alterado de geoespacial para categórico. Embora esse atributo represente o nome de uma região, a coluna também pode ser mais tratada como uma variável categórica ordinal. A Figura 30 apresenta essa modificação:

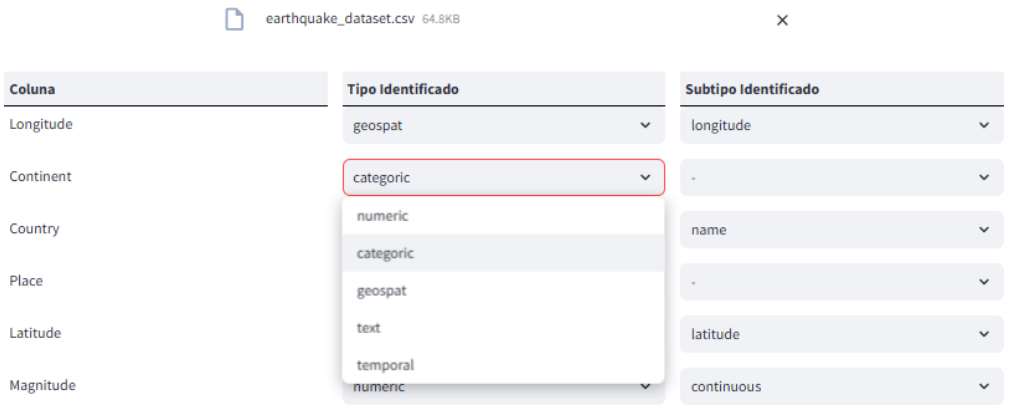


Figura 30: Correção de tipo do atributo *Continent* para categórico.

16.3 Módulo Suggestor

Fazendo uso da interface Pyswip, foi utilizada a lógica do Prolog para executar as sugestões de técnicas de ciência de dados e formas de visualização com base nos dados, tendo como base algumas das técnicas listadas na Tabela 3 para diferentes datasets e combinações de tipos. As regras do Prolog são advindas de uma base pré-definida (a qual poderia ser aumentada com a extensibilidade), presente no Apêndice A. Os fatos são gerados a partir dos resultados do Profiling. Devido à aplicação de algumas técnicas sobre múltiplos atributos numéricos e categóricos (como as de correlação e clusterização, por exemplo), o módulo também gera como fatos listas com os atributos desses tipos.

Dessa forma, utilizando como exemplo o dataset, tendo como resultado os dados do Profiling, seriam gerados os seguintes fatos:

Código 5: resultado de fatos de tipos e subtipos gerados pelo módulo Suggestor após Profiling do dataset e correção de tipo.

```
numeric('Magnitude')
continuous('Magnitude')
categoric('Place')
categoric('Continent')
geospat('Latitude')
latitude('Latitude')
geospat('Longitude')
longitude('Longitude')
geospat('Country')
name('Country')
```

Utilizando o arquivo pré-definido de regras (Apêndice A), o módulo promove as sugestões das técnicas, seguindo os seguintes formatos:

- suggest_single_col(X, *technique*): *technique* é aplicada somente à coluna X;
- suggest_pair_col(X, Y, *technique*): *technique* é aplicada ao par de colunas X e Y;
- suggest_trio_col(X, Y, Z, *technique*): *technique* é aplicada ao

trio de colunas X, Y e Z;

- suggest_all_cols(X, *technique*): *technique* é aplicada a todas as colunas de uma lista X;
- suggest_all_pair_cols(X, Y, *technique*): *technique* é aplicada a todas as colunas de uma lista X e uma lista Y.

Como exemplo, há algumas das seguintes regras de visualização, exibidas no Código 6:

Código 6: regras pré-definidas de visualização na base de regras.

```
suggest_single_col(X, 'histogram') :- numeric(X), continuous(X)
suggest_single_col(X, 'barplot_cat') :- categoric(X)
suggest_pair_col(X, Y, 'choropleth') :- numeric(X), geospat(Y), name(Y)
suggest_trio_col(X, Y, Z, 'heatmap') :- numeric(X), geospat(Y), latitude(Y),
        geospat(Z), longitude(Z)
```

Essas regras indicam o seguinte:

- Sugerir a aplicação de histograma na coluna X caso a coluna X seja do tipo numérico e do subtipo contínuo;
- Sugerir a aplicação de barplot (especificando o barplot categórico) na coluna X caso a coluna X seja do tipo categórico;
- Sugerir a aplicação de mapa coroplético no par de colunas X e Y caso X seja do tipo numérico, Y seja do tipo geoespacial e do subtipo nome de uma região;
- Sugerir a aplicação de mapa de calor no trio de colunas X, Y e Z caso X seja do tipo numérico, Y seja do tipo geoespacial com subtipo latitude e Z seja do tipo geoespacial com subtipo longitude.

A partir das regras demonstradas e dos fatos resultantes do módulo de Profiling, o sistema obtém as seguintes técnicas de visualização:

- Aplicar *histogram* à coluna *Magnitude*;
- Aplicar *barplot_cat* à coluna *Place*;
- Aplicar *barplot_cat* à coluna *Continent*;
- Aplicar *choropleth* ao par de colunas *Magnitude* e *Country*;

- Aplicar *heatmap* ao trio de colunas *Magnitude*, *Latitude* e *Longitude*.

Há ainda outras regras abaixo, como as de engenharia de atributos listadas no Código 7, referentes às diferentes formas em que se poderia aplicar a técnica de clusterização, dependendo dos tipos de atributos presentes no dataset:

Código 7: regras pré-definidas de engenharia de atributos na base de regras.

```
suggest_all_pair_cols(X, Y, 'clustering_num') :- numeric_list(L1),
        categoric_list(L2), length(L1, Len1), length(L2, 0), Len1 > 1, X = L1,
        Y = L2
suggest_all_pair_cols(X, Y, 'clustering_cat') :- numeric_list(L1),
        categoric_list(L2), length(L1, 0), length(L2, Len2), Len2 > 1, X = L1,
        Y = L2
suggest_all_pair_cols(X, Y, 'clustering_num_cat') :- numeric_list(L1),
        categoric_list(L2), length(L1, Len1), length(L2, Len2), Len1 > 0,
        Len2 > 0, X = L1, Y = L2
```

Assim, a técnica de clusterização também é recomendada devido à presença de atributos tanto numéricos quanto categóricos no dataset. Para promover uma melhor explicação ao usuário sobre as observações e a justificativa da técnica, foi destacada a distinção entre os tipos de atributos na base de regras. Assim, quando a base contém atributos numéricos e categóricos, a clusterização é sugerida com informações específicas que incluem os dois tipos. No caso de haver apenas múltiplos atributos numéricos e nenhum categórico, as informações serão direcionadas exclusivamente para o dado presente. Por outro lado, caso haja múltiplos atributos categóricos e nenhum numérico, as orientações serão voltadas exclusivamente sobre os categóricos.

16.3.1 Resultado de Visualizações Iniciais

Após o resultado das queries de Prolog, serão exibidas para o usuário algumas visualizações iniciais, demonstrando distribuições dos dados e correlações entre atributos, por exemplo.

Para o dataset de exemplo, com as regras pré-definidas do Apêndice A, são apresentadas as visualizações de histograma, gráficos de barra, mapas coroplético e de calor. As figuras a seguir ilustram algumas dessas visualizações, representando o resultado para uma coluna única (Figura 31), para um par de colunas (Figura 32) e para um trio de colunas (Figura 33).

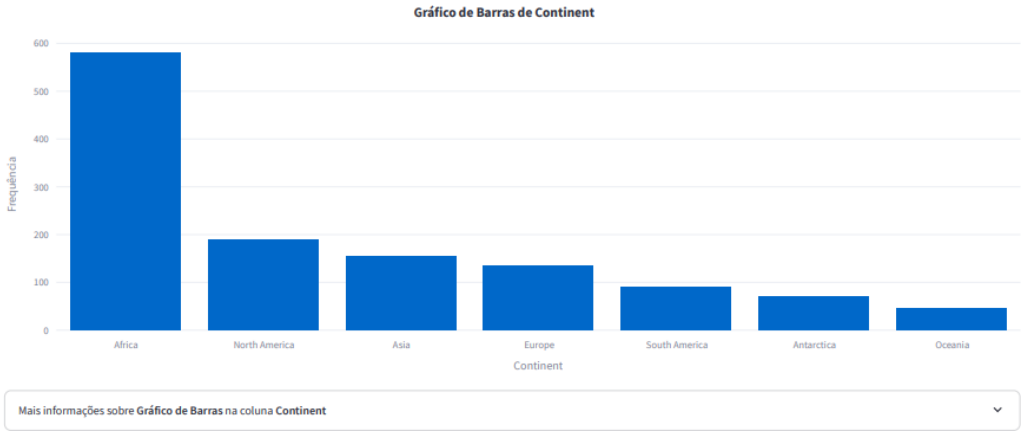


Figura 31: Gráfico de barras gerado na visualização inicial para demonstrar a frequência de cada categoria no atributo *Continent*.

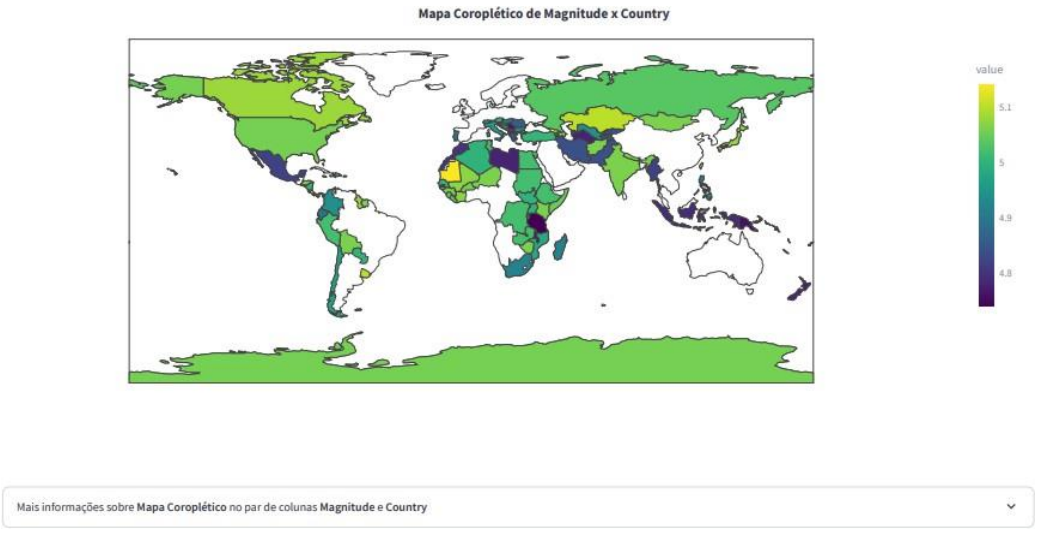


Figura 32: Mapa Coroplético gerado na visualização inicial para demonstrar a média de magnitudes em diferentes países.

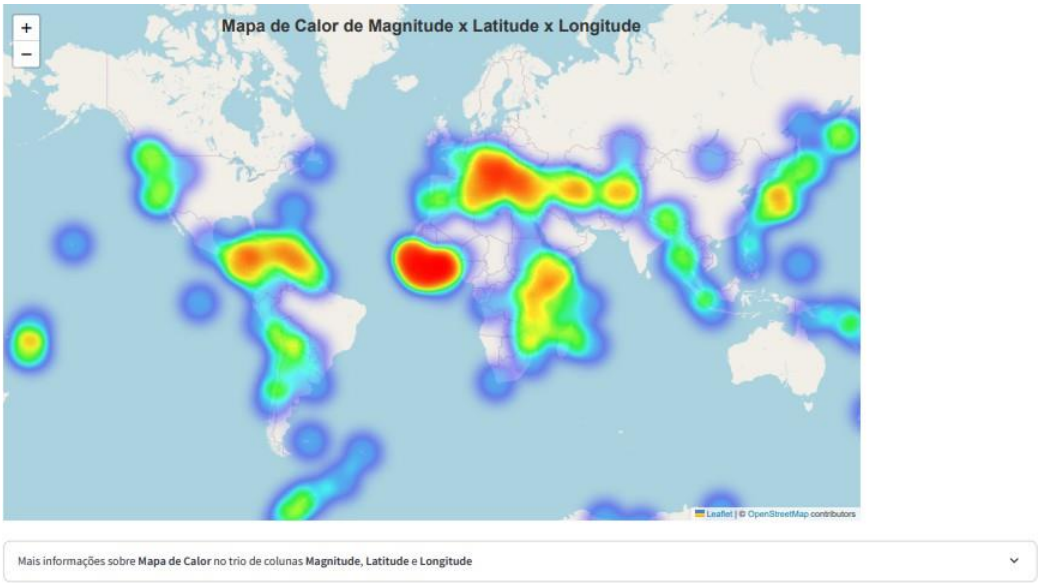


Figura 33: Mapa de calor gerado na visualização inicial para demonstrar a incidência da magnitude em diferentes coordenadas.

A Figura 34 a seguir demonstra o resultado da expansão da explicação sobre a técnica de Gráfico de Barras aplicada na coluna *Continent*, com uma descrição sobre o que é a técnica, a justificativa da sugestão, formas de implementá-la, o resultado esperado com a implementação e referências que abordem a sugestão. Essas informações se encontram presentes no Apêndice B.

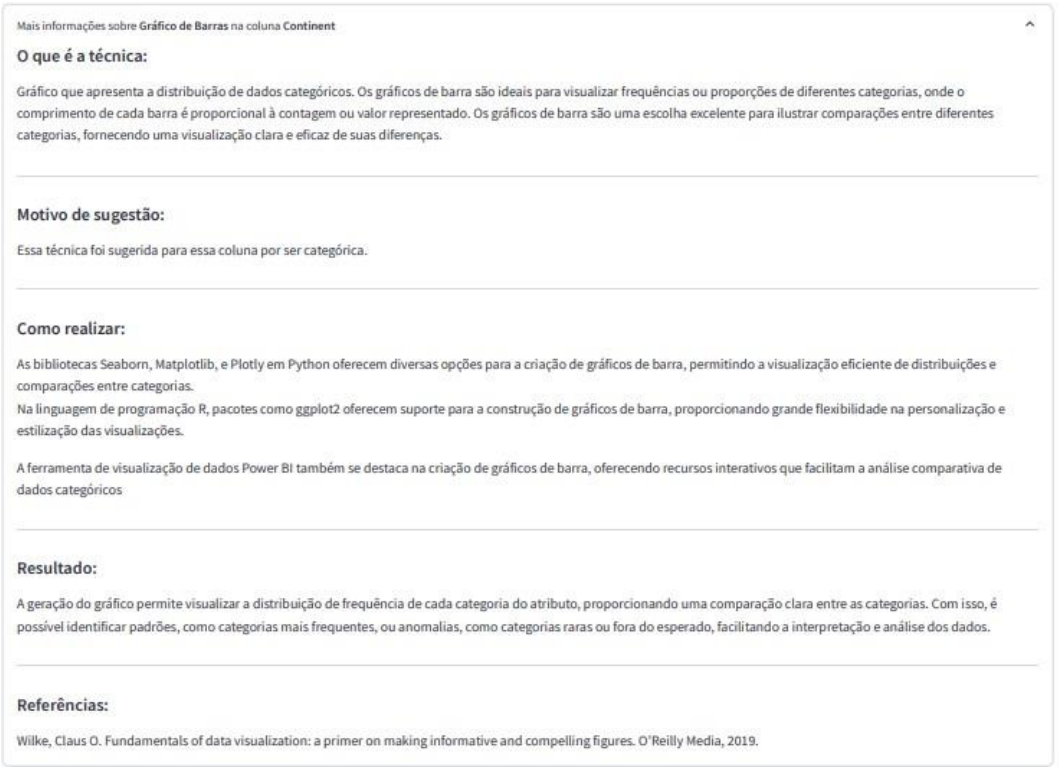


Figura 34: Expansão de informações sobre a aplicação de Gráficos de barras no atributo *Continent*.

16.3.2 Resultado de Sugestões de Técnicas para o Usuário

Além das visualizações propostas, existem também as técnicas de pré-processamento e engenharia de atributos sugeridas. Na base, há a técnica de clusterização, aplicada nos grupos de colunas que sejam numéricas ou categóricas.

Dessa forma, para o dataset de exemplo - após sugeridas as técnicas de visualização do 16.3.1 - seria sugerida também a técnica de clusterização, aplicada às colunas numéricas (para o caso apenas a *Magnitude*) e categóricas (*Place* e *Continent*). Assim, é demonstrada uma explicação do que é a técnica, como executá-la, resultados esperados na aplicação da técnica, referências e observações, como na Figura 35:

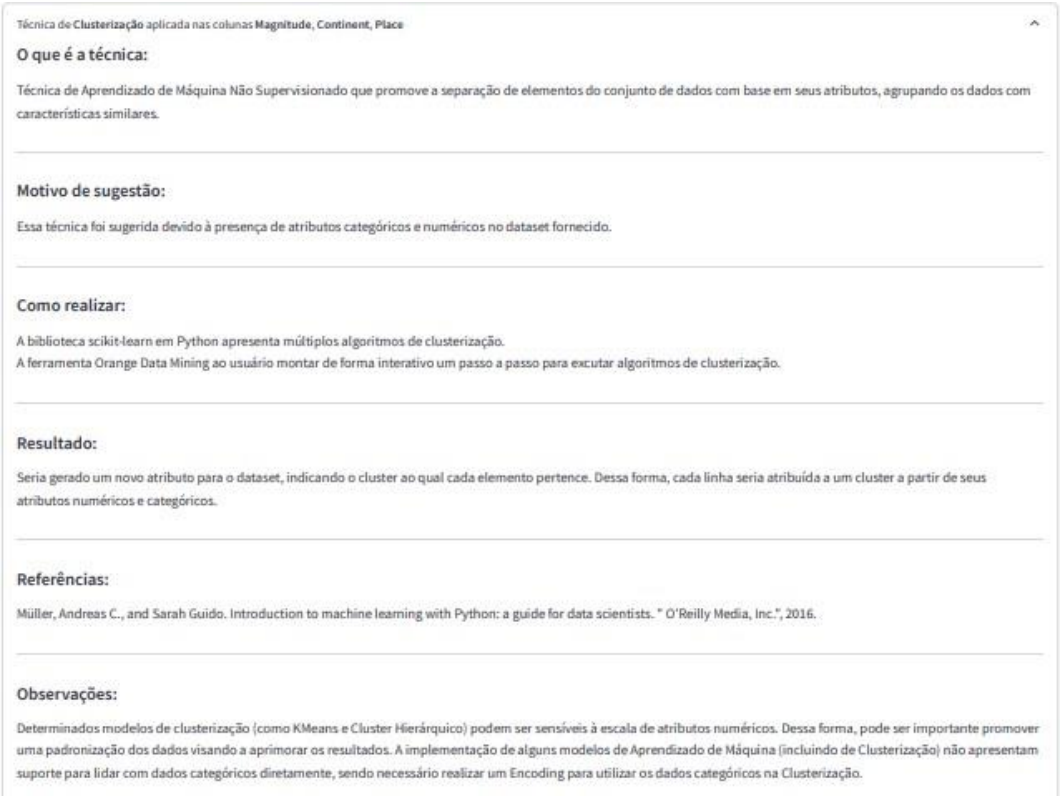


Figura 35: explicação sobre a sugestão da técnica de Clusterização aplicada nas colunas numéricas e categóricas do dataset.

Foi também incluída a imagem da Figura 36 para ilustrar para o usuário sobre a técnica de Clusterização:

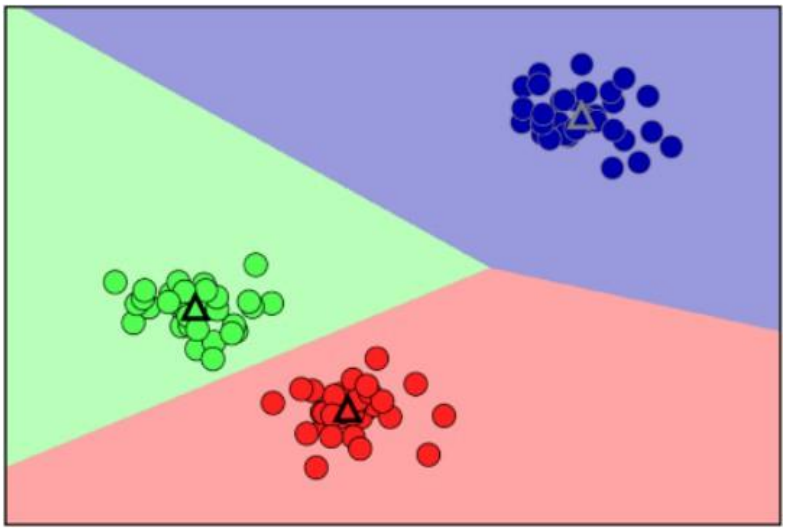


Figura 36: ilustração da técnica de Clusterização, adicionada ao final de sua explicação. Fonte: MÜLLER; GUIDO, 2016, fig. 3-23.

16.4 Metadados resultantes do Profiling

Após o resultado do Profiling (e feitas as devidas correções pelo usuário, caso necessárias), são obtidos, para cada uma das colunas que o sistema conseguiu identificar:

- Tipo atribuído (numérico, categórico, geoespacial, textual ou temporal);
- Subtipo atribuído (contínuo ou discreto se numérico; latitude, longitude ou nome se geoespacial; nenhum para demais tipos);
- Percentual de dados faltantes;
- Proporção de valores únicos.

Além disso - para as demais colunas, que não tiveram seu tipo identificado -, ainda tem-se o percentual de dados faltantes.

O usuário poderia, então, baixar uma versão de metadados com essas informações resultantes do Profiling, contendo esses dados podendo ser na extensão csv, txt ou json.

Ao selecionar a opção de baixar metadados como na Figura 37, o sistema baixa um arquivo informando os resultados do Profiling, que pode ser observado na Figura 38:

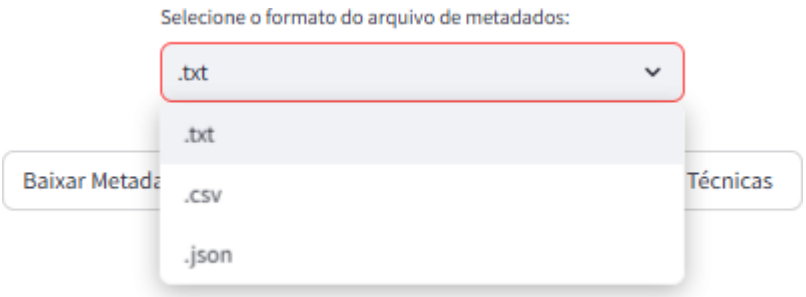


Figura 37: selecionando a extensão de txt para baixar metadados.

Resultados identificados:
- Continent:
    Tipo identificado: categoric
    Percentual de dados faltantes: 0.000
    Proporção de valores únicos: 0.006
- Longitude:
    Tipo identificado: geospat
    Subtipo identificado: longitude
    Percentual de dados faltantes: 0.000
    Proporção de valores únicos: 0.964
- Magnitude:
    Tipo identificado: numeric
    Subtipo identificado: continuous
    Percentual de dados faltantes: 0.000
    Proporção de valores únicos: 0.017
- Latitude:
    Tipo identificado: geospat
    Subtipo identificado: latitude
    Percentual de dados faltantes: 0.000
    Proporção de valores únicos: 0.911
- Country:
    Tipo identificado: geospat
    Subtipo identificado: name
    Percentual de dados faltantes: 0.000
    Proporção de valores únicos: 0.097
- Place:
    Tipo identificado: categoric
    Percentual de dados faltantes: 0.000
    Proporção de valores únicos: 0.132

Figura 38: Arquivo txt de metadados resultantes do Profiling.

16.5 Extensibilidade

Para adicionar mais regras, existe uma opção na página inicial de modificar o arquivo de base de regras do Prolog e digitar a regra no formato da linguagem. Assim, por exemplo, caso o usuário queira promover análises entre um atributo de nome geográfico e um atributo categórico para montar um mapa categórico ou sugerir um *boxplot* para atributos numéricos contínuos, por exemplo, poderia incluir as regras do Código 8 à base:

Código 8: regras adicionadas pelo usuário.

```
suggest_single_col(X, 'boxplot') :- numeric(X), continuous(X)
suggest_pair_col(X, Y, 'categoric_map') :- categoric(X), name(Y)
```

Dessa forma, ao selecionar a opção de exibir e editar o arquivo de base de regras, o usuário pode ver as regras da base pré-definida, como na Figura 39, além de adicionar novas regras, como na Figura 40:

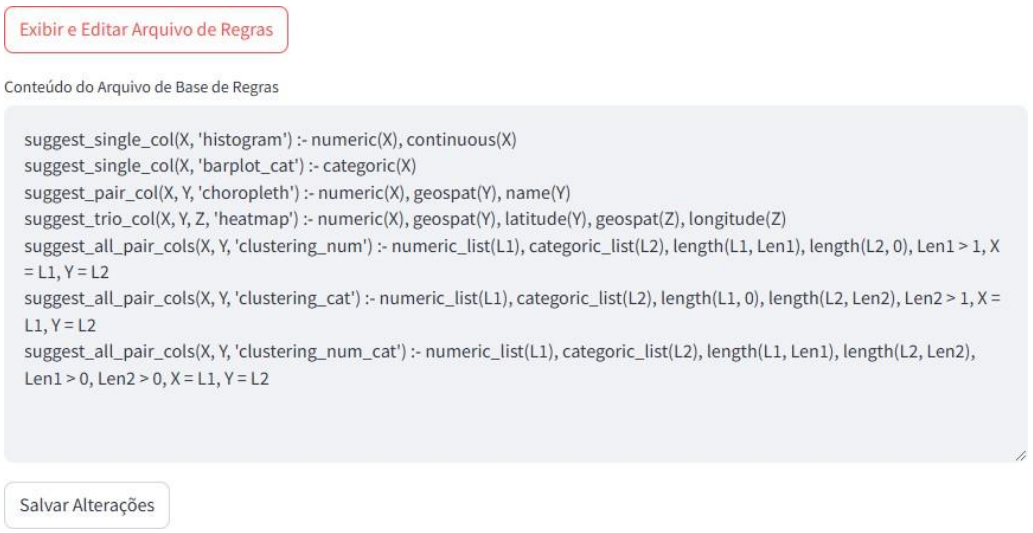


Figura 39: Exibição do arquivo de base de regras antes de modificações listando as regras presentes nos Códigos 6 e 7.

Figura 40: Exibição do arquivo de base de regras após adição de novas técnicas sugeridas utilizando as regras do Código 8.

Por fim, podem ser observadas, após as técnicas pré-existentes na base, o resultado das *queries* para essas técnicas e as colunas em que foram aplicadas, como na Figura 41:

Figura 41: Resultado de técnicas sugeridas com a extensão do usuário ao adicionar as novas regras do Código 8.

17. Validação e Verificação com Usuário

Após o desenvolvimento de uma versão inicial do sistema, foram realizadas verificações de funcionalidades com dois perfis de usuários distintos: um não especializado[10] em ciência de dados - utilizando um conjunto de dados próprio - e outro especializado[11] - utilizando o dataset de dados sísmicos que foi demonstrado na seção 16:

17.1 Validação com Usuário Não Especializado

Para o teste com o usuário não especializado em ciência de dados, foi utilizado o seguinte roteiro como base:

- Explicação Inicial sobre o sistema: O sistema desenvolvido é o meu trabalho de Projeto Final com o objetivo de funcionar como uma ferramenta auxiliar na tutoria de ciência de dados, contribuindo para que usuários não especializados possam lidar com seus conjuntos de dados com análises mais adequadas e diversificadas.
- Você tem alguma familiaridade com ciência de dados ou análises de dados?
- Você poderia explicar sobre o seu conjunto de dados utilizado para o teste?:
  - O que é?
  - Em que situação usa o conjunto?
  - Para que serve?
  - Como é utilizado?
- Pedir para o usuário fornecer os comandos para utilizar a ferramenta.
- Perguntar se o usuário utilizaria a ferramenta.

[10] Estudante de enfermagem, 21 anos de idade, Tempo de experiência de 10 meses como assistente de enfermagem. Graduando em Enfermagem.
[11] Cientista da computação, 23 anos de idade. Formado em Ciência da Computação, Tempo de experiência de 1 ano e meio em Ciência de Dados. Mestrando em Informática.

- Pedir sugestões de melhoria.

O usuário não especialista afirmou não ter qualquer experiência com ciência de dados ou análise de dados. Sobre o seu dataset, o conjunto é utilizado em seu trabalho e é referente a uma planilha para fazer a conexão entre pessoas, juntá-las aos pares, com o objetivo de encontrar dias e horas em que podem se encontrar. O dataset apresenta 119 linhas e 8 colunas (previamente 10, porém 2 colunas foram removidas em prol da anonimização dos dados).

O conjunto tem o objetivo de ajudar a verificar as preferências de cada pessoa para permitir que possam ser pareadas adequadamente. Quando questionado sobre a forma de utilizar, o usuário declarou não utilizar gráficos ou quaisquer outras análises, apenas investigando manualmente para identificar pares que fazem sentido.

Ao ser mostrado o sistema, o usuário guiou os comandos para a sua utilização. O usuário, então, indicou para fazer o upload de seu dataset para promover as análises.

Após uma breve explicação sobre os tipos e subtipos durante a verificação, o usuário sugeriu modificar um dos atributos - *Year in program* - de temporal para categórica. Apesar de o nome indicar temporalidade, a coluna indica o ano de escolaridade, justificando assim a mudança de tipo.

Com exceção de uma das colunas que não pode ser identificada devido ao percentual de dados faltantes, todos os demais atributos (sete) foram atribuídos ao tipo categórico (e sem subtipo). Para esses, nenhuma correção foi feita.

Selecionando a opção de “Sugerir Técnicas” (após confirmar a correção de tipos e subtipos), foram apresentadas as visualizações

iniciais e suas explicações ao usuário. O usuário acreditou que as explicações estavam claras e condizentes com os dados de seu conjunto.

Quando questionado se utilizaria a ferramenta, o usuário comentou que sim, para ter um maior entendimento sobre frequências e dados do conjunto, afirmando que o software auxilia nas análises e é intuitivo.

Como sugestão, o usuário sugeriu adicionar uma personalização aos gráficos, como mudança de cor, por exemplo, além de incluir de uma explicação sobre os tipos e subtipos de dados para deixar mais claros tanto os resultados relativos às identificações (do Profiling) quanto para proporcionar ao usuário um maior entendimento sobre possíveis correções.

17.2 Validação com Usuário Especializado em Ciência de Dados

Similarmente à entrevista com o usuário não especializado, foi seguido o roteiro a seguir:

- Explicação Inicial sobre o sistema: O sistema desenvolvido é o meu trabalho de Projeto Final com o objetivo de funcionar como uma ferramenta auxiliar na tutoria de ciência de dados, contribuindo para que usuários não especializados possam lidar com seus conjuntos de dados com análises mais adequadas e diversificadas.
- Qual a sua experiência com ciência de dados?
- Pedir para o usuário fornecer os comandos para utilizar a ferramenta.
- Comentar sobre a extensibilidade da ferramenta.
- Perguntar se o usuário acha válida a ferramenta para alguém inexperiente em ciência de dados.
- Pedir sugestões de melhoria.

O usuário especialista disse ter como experiência em ciência de dados trabalhos em sua graduação, seu projeto de iniciação tecnológica e é sua área de estudo na pós graduação.

Ao ser mostrado o sistema, o usuário ditou os comandos para a utilização de suas funcionalidades. O usuário, então, indicou para fazer o upload do dataset de dados sísmicos para promover as análises. O usuário não sugeriu nenhuma correção para os tipos e subtipos identificados (como na Figura 29).

Após selecionar a opção "Sugerir Técnicas", o sistema apresentou ao usuário visualizações iniciais acompanhadas de explicações. O usuário explorou as informações detalhadas de cada técnica sugerida, fornecendo feedback e propostas de melhoria. Ele ainda destacou como positivo o uso da linguagem R como sugestão para executar técnicas de visualização.

Como sugestão, o usuário apresentou as seguintes observações:

- Melhor explicação sobre o que está presente nos metadados e como poderia ser utilizado, uma vez que o público-alvo do sistema são pessoas leigas em ciência de dados;
- Melhor explicação sobre tipos e subtipos no escopo do sistema, incluindo *tooltips*;
- Utilização mais adequada de cores nas visualizações para melhor representar informações;
- Maior inclusão de informações sobre técnicas - em especial para as técnicas que não são executadas pelo sistema, como a de clusterização -, para explorar mais o objetivo didático do sistema;
- Inclusão de um exemplo prático no sistema, para facilitar o entendimento do usuário sobre como utilizá-lo. Isso poderia

ser feito adicionando uma página de guia demonstrando o uso do sistema;

- Explicação sobre a extensibilidade e demonstração prática de como adicionar novas regras. Além disso, foi sugerido melhorar a forma de estender a base, necessitando de menor uso de Prolog por parte do usuário.

Quando questionado se recomendaria o sistema para alguém interessado em aprender mais sobre ciência ou análise de dados, o usuário especialista respondeu que, para a versão atual, não recomendaria. No entanto, destacou que, com os ajustes sugeridos implementados em versões posteriores, recomendaria o sistema.

17.3 Conclusão de Validações

As validações com os usuários evidenciaram a necessidade de determinados ajustes para permitir que o sistema desenvolvido fosse mais adequado para os usuários, buscando explicar melhor sobre as características dos dados e de técnicas sugeridas.

Visando a deixar o sistema mais informativo, alguns dos ajustes sugeridos pelos usuários na validação foram, então, implementados: inclusão de *tooltip* de informações sobre tipos e subtipos na tabela ilustrando resultados do Profiling - como demonstrado nas Figuras 42, 43 e 44 - e de outro *tooltip* informando sobre os metadados, como na Figura 45.

Figura 42: Adição de *tooltips* na tabela de resultados do Profiling nas colunas de tipo e subtipo identificados.

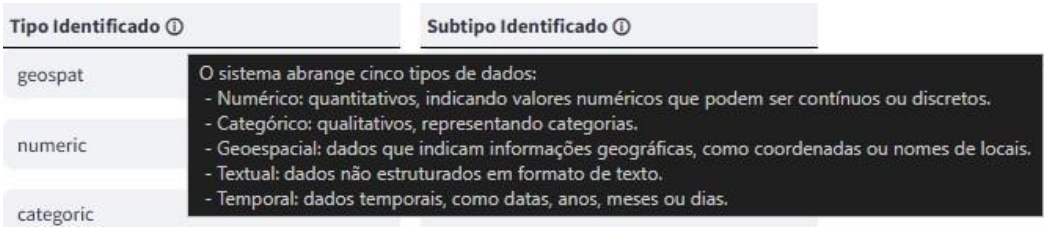


Figura 43: Texto do *tooltip* informando sobre os tipos na tabela de resultados do Profiling.

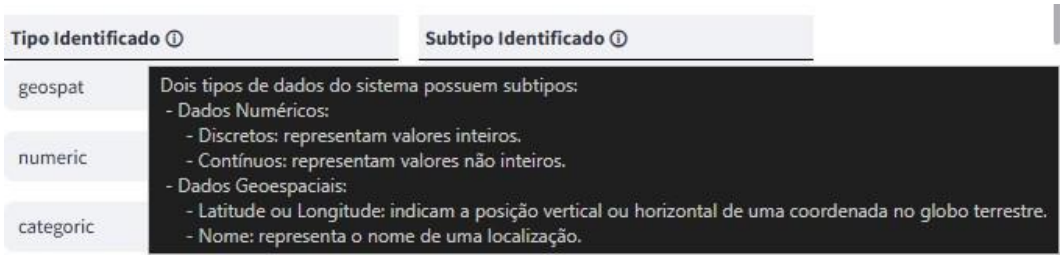


Figura 44: Texto do *tooltip* informando sobre os subtipos na tabela de resultados do Profiling.

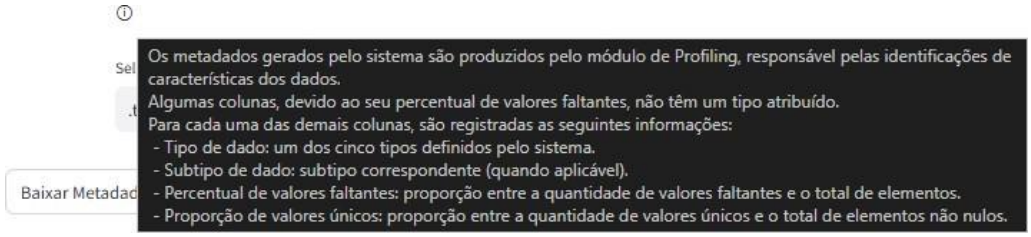


Figura 45: Texto do *tooltip* informando sobre os metadados gerados pelo Profiling.

18. Considerações Finais

Nesta seção são abordadas as considerações finais sobre o trabalho:

18.1 Conclusão

Este trabalho teve como objetivo a especificação e o desenvolvimento de uma ferramenta que pudesse didaticamente auxiliar no entendimento de técnicas de análise e ciência de dados, buscando promover apoio aos usuários para encontrarem técnicas e ferramentas apropriadas para que possam analisar e compreender seus datasets.

A ferramenta atende à proposta e ao objetivo iniciais, uma vez que fornece exemplos de técnicas aplicáveis ao conjunto, listando os motivos da aplicação, como realizá-la, referências e resultados esperados, o que pode contribuir para que os usuários alvos do sistema, não especializados em ciência de dados, aprendam sobre múltiplas técnicas que possam ser incluídas em suas análises futuras. Entretanto, mais alguns ajustes se fazem necessários para explorar - de fato - o intuito didático do sistema, além de realizar novas validações após com usuários após tais ajustes.

18.2 Possibilidades de Trabalhos Futuros

- Utilização metadados para contribuir no Profiling: como discorrido na seção 9.1, a realização do Profiling poderia ser incrementada com o uso de metadados pré-existentes, o que poderia permitir fazer identificações mais precisas;
- Expansão do arquivo de base de regras pré-definidas: expandir as técnicas que poderiam ser sugeridas potencializaria o objetivo didático da ferramenta ao listar mais visualizações e demais técnicas de ciência de dados possíveis, bem como expandir as explicações das técnicas já existentes fornecendo mais detalhes para os usuários.

- Melhor utilização de dados do Profiling nas sugestões: poderiam ser mais utilizados para a construção dos fatos da base, incluindo o percentual de dados faltantes e a proporção de valores únicos para permitir mais análises e sugestões de técnicas;
- Possibilidade de execução de técnicas sugeridas de pré-processamento de dados, engenharia de atributos e demais técnicas que não sejam de visualização: seria possível expandir a ferramenta para que as técnicas sugeridas que fossem selecionadas pelo usuário - após a explicação do porquê das sugestões e quais seriam os resultados esperados -, ao invés de somente recomendá-las e propor ferramentas que as executem;
- Uso de múltiplos datasets: o sistema atualmente permite somente a análise de datasets unitários, a inclusão de múltiplos datasets que estejam relacionados poderia potencializar as análises;
- Inclusão de mais extensões de arquivos: nessa versão inicial, somente dados em csv são suportados pelo sistema, podendo ser expandido para a inclusão de mais formatos como xml, json e geojson;
- Melhor separação de tipos e subtipos: para outras versões do sistema poderia ser realizada uma melhor separação entre subtipos, como expandir as hierarquias e granularidades de dados temporais (incluindo como subtipo o formato de data) ou de dados geoespaciais de nomes de região (diferenciando se indica uma cidade, um país ou um continente, por exemplo);
- Incremento na extensibilidade: em versões futuras fornecer um formulário para que o usuário preencha demais informações das técnicas adicionadas, e não somente as

sugestões com base no tipo e subtipo;

- Inclusão de Explainable AI: integrar modelos supervisionados com foco na explicabilidade permitiria análises preditivas mais transparentes, potencializando o uso do conjunto de dados;
- Por fim, algumas propostas apresentadas nas validações com os usuários poderiam ser implementadas, como a adição de uma página com exemplos práticos e uma maior personalização para visualizações iniciais.

19. Referências Bibliográficas

[1] IBM. **What is Data Science?** Disponível em: https://www.ibm.com/topics/data-science. Acesso em: 29 abr. 2024.

[2] Demsar J, Curk T, Erjavec A, Gorup C, Hocevar T, Milutinovic M, Mozina M, Polajnar M, Toplak M, Staric A, Stajdohar M, Umek L, Zagar L, Zbontar J, Zitnik M, Zupan B (2013) Orange: Data Mining Toolbox in Python, Journal of Machine Learning Research 14(Aug): 2349−2353.

[3] Scott, L.M., Janikas, M.V. (2010). Spatial Statistics in ArcGIS. In: Fischer, M., Getis, A. (eds) Handbook of Applied Spatial Analysis. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-03647-7_2

[4] Lettria. **Lettria: Text Analysis Tools.** Disponível em: https://www.lettria.com/. Acesso em: 29 abr. 2024.

[5] Python Software Foundation. **Python Language Reference, version 3.10.** Disponível em: https://www.python.org/. Acesso em: 30 abr. 2024.

[6] Google. **Google Colab.** Disponível em: https://colab.research.google.com/. Acesso em: 29 abr. 2024.

[7] Google Cloud. **Dataprep by Trifacta.** Disponível em: https://cloud.google.com/dataprep. Acesso em: 29 abr. 2024.

[8] Microsoft Corporation. **Microsoft Excel.** Disponível em: https://www.microsoft.com/pt-br/microsoft-365/excel. Acesso em: 29 abr. 2024.

[9] Microsoft Corporation. **Microsoft Project.** Disponível em: https://www.microsoft.com/pt-br/microsoft-365/project/project-management-software. Acesso em: 29 abr. 2024.

[10] Ranganathan, Priya, and Nithya J. Gogtay. "An introduction to statistics–data types, distributions and summarizing data." Indian journal of critical care medicine: peer-reviewed, official publication of Indian Society of Critical Care Medicine 23.Suppl 2 (2019): S169.

[11] Padgham M, Boeing G, Cooley D, Tierney N, Sumner M, Phan TG and Beare R (2019) An Introduction to Software Tools, Data, and Services for Geospatial Analysis of Stroke Services. Front. Neurol. 10:743. doi: 10.3389/fneur.2019.00743

[12] Li, Qin, et al. "A review of text corpus-based tourism big data mining." *Applied Sciences* 9.16 (2019): 3300.

[13] Lin, Weiqiang, Mehmet A. Orgun, and Graham J. Williams. "An Overview Of Temporal Data Mining." *AusDM* (2002): 83-90.

[14] HARRIS, Charles R. et al. Array programming with NumPy. Nature, v. 585, n. 7825, p. 357-362, 2020.

[15] Team, The Pandas Development. "pandas-dev/pandas: Pandas." *Zenodo, February* (2020).

[16] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

[17] Waskom, M. L., (2021). seaborn: statistical data visualization. Journal of Open Source Software, 6(60), 3021, https://doi.org/10.21105/joss.03021.

[18] python-visualization. (2020). Folium. Disponível em: https://python-visualization.github.io/folium/. Acesso em: 9 jun. 2024.

[19] **GEOPY.** geopy. Disponível em: https://pypi.org/project/geopy/. Acesso em: 9 jun. 2024.

[20] R CORE TEAM. *R: A Language and Environment for Statistical Computing*. Vienna: R Foundation for Statistical Computing, 2021. Disponível em: https://www.R-project.org/. Acesso em: 9 jun. de 2024.

[21] Jain, Shaveta. "Comprehensive survey on data science, lifecycle, tools and its research issues." *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*. Vol. 1. IEEE, 2022.

[22] Müller, Andreas C., and Sarah Guido. *Introduction to machine learning with Python: a guide for data scientists*. " O'Reilly Media, Inc.", 2016.

[23] Grus, Joel. *Data science from scratch: first principles with python*. O'Reilly Media, 2019.

[24] Nielsen, Aileen. *Practical time series analysis: Prediction with statistics and machine learning*. O'Reilly Media, 2019.

[25] Couto, Júlia Colleoni, et al. "New trends in big data profiling." Science and Information Conference. Cham: Springer International Publishing, 2022.

[26] Tierney, Nicholas J., and Dianne H. Cook. "Expanding tidy data principles to facilitate missing data exploration, visualization and assessment of imputations." arXiv preprint arXiv:1809.02264 (2018).

[27] Dallos Parra, David Leonardo, Maryury Julieth Carvajal Camargo, and Juan Manuel Sánchez Céspedes. "Design of a Rules-Based Recommendation System Implemented in Prolog for the Management of Electronic Waste from ICT." *Journal of Ecological Engineering* 22.11 (2021).

[28] Davis, Ruth E. "Logic programming and Prolog: A tutorial." *IEEE Software* 2.5 (1985): 53.

[29] TEKOL, Yüce; PySwip contributors. **PySwip v0.3.0**. 2024. Disponível em: https://pyswip.org. Acesso em: 19 out. 2024.

[30] STREAMLIT, Inc. *Streamlit: The fastest way to build and share data apps*. 2024. Disponível em: https://streamlit.io/. Acesso em: 24 out. 2024.

[31] MUELLER, Andreas. *Wordcloud: A little word cloud generator in Python*. Disponível em: https://github.com/amueller/word_cloud. Acesso em: 24 out. 2024.

[32] PLOTLY TECHNOLOGIES Inc. *Plotly: Modern analytics for Python*. 2024. Disponível em: https://plotly.com/python/. Acesso em: 24 out. 2024.

20. Apêndices

APÊNDICE A - Arquivo de base de regras comentado

```
/* sugestão de histogram para colunas numéricas contínuas */
suggest_single_col(X, 'histogram') :- numeric(X), continuous(X)

/* sugestão de barplot_discrete para colunas numéricas discretas */
suggest_single_col(X, 'barplot_discrete') :- numeric(X), discrete(X)

/* sugestão de barplot_cat para colunas categóricas */
suggest_single_col(X, 'barplot_cat') :- categoric(X)

/* sugestão de wordcloud para colunas textuais */
suggest_single_col(X, 'wordcloud') :- text(X)

/* sugestão de scatterplot_num_num para pares de colunas numéricas contínuas */
suggest_pair_col(X, Y, 'scatterplot_num_num') :- numeric(X), continuous(X), numeric(Y), continuous(Y), X \= Y

/* sugestão de choropleth para pares de colunas (uma numérica e outra geoespacial indicando o nome de uma região) */
suggest_pair_col(X, Y, 'choropleth') :- numeric(X), geospat(Y), name(Y)

/* sugestão de heatmap para um trio de colunas (uma numérica, uma de latitude e uma de longitude */
suggest_trio_col(X, Y, Z, 'heatmap') :- numeric(X), geospat(Y), latitude(Y), geospat(Z), longitude(Z)

/* sugestão de correlation  para todos os atributos numéricos caso tenha mais de um atributo numérico */
suggest_all_cols(X, 'correlation') :- numeric_list(L1), length(L1, Len1), Len1 > 1, X = L1

/* sugestão de clustering_num para todos os atributos numéricos caso tenha mais de um atributo numérico e não tenha nenhum atributo categórico */
suggest_all_pair_cols(X, Y, 'clustering_num') :- numeric_list(L1), categoric_list(L2), length(L1, Len1), length(L2, 0), Len1 > 1, X = L1, Y = L2

/* sugestão de clustering_cat para todos os atributos categóricos caso tenha mais de um atributo categórico e não tenha nenhum atributo numérico */
suggest_all_pair_cols(X, Y, 'clustering_cat') :- numeric_list(L1), categoric_list(L2), length(L1, 0), length(L2, Len2), Len2 > 1, X = L1, Y = L2

/* sugestão de clustering_num_cat para todos os atributos numéricos e categóricos caso tenha no mínimo um atributo numérico e no mínimo um categórico */
suggest_all_pair_cols(X, Y, 'clustering_num_cat') :- numeric_list(L1), categoric_list(L2), length(L1, Len1), length(L2, Len2), Len1 > 0, Len2 > 0, X = L1, Y = L2
```

APÊNDICE B - Link para o arquivo techniques.json completo

https://drive.google.com/file/d/1umYCOgES-U_yBfiF0J8ZP08-L8s4d1Q0/view?usp=drive_link