PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

Aplicação de Redes Neurais no processo de análise dos movimentos contra a desigualdade de gênero nas redes sociais.

Nathalia Mariz de Almeida Salgado Inácio

PROJETO FINAL DE GRADUAÇÃO

CENTRO TÉCNICO CIENTÍFICO - CTC

DEPARTAMENTO DE INFORMÁTICA

Curso de Graduação em Ciência da Computação

Rio de Janeiro, novembro de 2021

Nathalia Mariz de Almeida Salgado Inácio

Aplicação de Redes Neurais no processo de análise dos movimentos contra a desigualdade de gênero nas redes sociais.

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: Hélio Côrtes Vieira Lopes

Rio de Janeiro

Novembro de 2021

AGRADECIMENTOS

Aos meus pais, Marcia Regina de Almeida Inácio e Aldo Salgado Inácio, e ao meu avô, José Mariz de Almeida, por estarem do meu lado nas horas mais difíceis e felizes da minha vida, por todas as oportunidades, incentivos, dedicação, confiança e amor. Vocês são o os meus pilares e me inspiram todos os dias a ser uma pessoa melhor.

À minha prima, Armanda Salgado Lopes, por ter sempre me incentivado a seguir meu caminho e ter sido, não só um exemplo a ser seguido, mas também minha mestra e conselheira de estudos.

Ao meu orientador, Hélio Côrtes Vieira Lopes, que durante esse ano, compartilhou seus conhecimentos, me acompanhando e apoiando no desenvolvimento deste projeto de conclusão de curso.

À minha universidade, a Pontifícia Universidade Católica do Rio de Janeiro, por ter fornecido todas as ferramentas, um ensino ímpar, oportunidades e experiências incomparáveis que me permitiram chegar ao final deste ciclo.

Aos meus professores, por toda paciência e dedicação ao longo dessa caminhada. Os ensinamentos de vocês transformaram por completo a minha vida e me guiaram na busca pelos meus sonhos.

Aos amigos que a universidade me deu de presente, em especial, à Júlia Affonso Figueiredo Rocha e ao Bruno Miranda Marinho, que compartilham de inúmeras memórias desses longos anos de estudos, amizade e cumplicidade. O apoio, força e conhecimento de vocês foram indispensáveis para a minha evolução profissional e pessoal, bem como para este projeto de conclusão de curso.

À todos os meus familiares e amigos, que sempre me incentivaram, aconselharam e apoiaram ao longo de toda minha jornada, não só acadêmica. A contribuição de vocês é, e sempre foi, fundamental para o meu crescimento.

Resumo

Mariz de Almeida Salgado Inácio, Nathalia. Côrtes Vieira Lopes, Hélio. Aplicação de Redes Neurais no processo de análise dos movimentos contra a desigualdade de gênero nas redes sociais. Rio de Janeiro, 2021.

As redes sociais têm ganhado bastante destaque nas últimas décadas e apesar delas serem muito conhecidas pelo seu potencial de sociabilização e entretenimento, são uma ferramenta poderosa de informação e protestos. Dentre elas, o Twitter têm sido um grande palco de mobilização por parte do movimento feminista, se transformando em um instrumento por onde mulheres podem expor suas reivindicações. Este projeto tem como objetivo apresentar uma análise acerca dos tweets coletados sobre luta contra a desigualdade de gênero, através de Redes Neurais Convolucionais e suas aplicações em processamento de linguagem natural (NLP) que, associadas aos dados extraídos, permitiram um desenvolvimento de um estudo detalhado sobre os mesmos.

Palavras-chave

Redes Neurais, Processamento de Linguagem Natural, Aprendizado de Máquina, Análise de Tweets, Movimentos de Desigualdade de Gênero

Abstract

Mariz de Almeida Salgado Inácio, Nathalia. Côrtes Vieira Lopes, Hélio. Application of Neural Networks in the process of analyzing movements against gender inequality in social networks. Rio de Janeiro, 2021.

Social networks have gained a lot of emphasis in the latest decades. Despite of their well known socialization and entertainment potential, they are also a powerful too for information and protests. Among them, Twitter has been a great way of mobilization by the feminist movement, being transformed into an instrument through which women can express their demands. This project aims to present an analysis of collected tweets about the fight against gender inequality. To that end, it was used Convolutional Neural Networks and their applications in natural language processing (NLP), which associated with the extracted data, allowed the development of a detailed study about them.

Keywords

Neural Networks, Natural Language Processing, Machine Learning, Tweet Analysis, Gender Inequality Movements

SUMÁRIO

Lista de Figuras

Lista de Tabelas

1. INTRODUÇÃO

As redes sociais têm ganhado bastante destaque e crescido muito nas últimas décadas, ao mesmo compasso que o número de seus usuários aumentam a cada ano, a suas utilidades também têm se mostrado cada vez mais diferenciadas. Apesar de serem vistas por muitos como um ambiente de sociabilização e entretenimento, onde se pode curtir e compartilhar conteúdo, as redes sociais têm sido uma ferramenta poderosa de informação que deram origem ao ciberativismo [1].

O ciberativismo [1] ganhou força em diversos movimentos no Brasil e no mundo, se fazendo presente na discussão e protestos de diversas causas sociais, raciais e de gênero. Entre tantas pautas marcadas pela influência das redes, não podemos deixar de notar o crescimento e popularidade que elas propiciaram ao movimento feminista, que trouxe as questões de desigualdade e violência de gênero à tona com diversos relatos, hashtags [4] e protestos no ambiente virtual. Algumas das redes mais famosas como Facebook, Twitter e Instagram [1] têm sido palco de grande mobilização por parte do movimento, pois fizeram ser possível a organização e exposição de suas pautas, além de um instrumento de inclusão onde mulheres poderiam expor suas inquietações.

Este projeto tem como objetivo apresentar uma análise acerca dos tweets sobre a luta contra a desigualdade de gênero, permitindo assim, extrair informações interessantes e precisas sobre este movimento. Para tanto, o projeto utilizará de aplicações em processamento de linguagem natural (NLP) [5] e Redes Neurais [5] que serão associadas aos dados extraídos desta principal rede social e permitirão o desenvolvimento de um estudo detalhado sobre os mesmos. Esse estudo será possível através dos dados recolhidos e representados por tabelas em um script desenvolvido na linguagem de programação Python, na plataforma Jupyter Notebook.

2. SITUAÇÃO ATUAL

Atualmente, vivemos um momento onde expandimos nosso espaço de interação para além do espaço fisico, cada vez mais, dominamos as redes sociais, transformando a forma que nos comunicamos. Essa transformação ocorre graças a internet e ao crescimento das redes sociais, que permitiram não só sociabilizarmos de formas diferentes mas também, protestarmos nesse novo ambiente que surgiu por meio da tecnologia.

Diante desse cenário, o movimento feminista entra em sua terceira onda: o feminismo contemporâneo ou o ciberativismo feminista [2 ,3]. Essa onda se

caracteriza principalmente pela utilização do ciberespaço [1] como um instrumento de protesto contra as desigualdades de gênero e um meio de compartilhar pautas de empoderamento feminino. Dessa forma, essa onda fez com que o movimento não se fizesse tão presente nas ruas, como nas manifestações do direito a voto, divórcio e a propriedade. Hoje, ele se faz presente nas mídias, gerando além de popularidade e inclusão, a amplificação do movimento a favor da igualdade de gênero.

Uma das ferramentas mais utilizadas para difundir o ciberativismo feminista [2 ,3] dentro das redes foram as famosas hashtags [4]. Utilizadas para marcar e agrupar postagens por tópicos de assunto, estas, têm ido além de seu propósito original e se tornaram um instrumento essencial nas mobilizações virtuais nos últimos anos. Especificamente dentro do movimento feminista, desde o ano de 2015, surgem diversas hashtags que evidenciam a necessidade do debate sobre essa pauta na internet. As que ficaram mais conhecidas como: #primeiroassedio, #agoraéquesãoelas e #meuamigosecreto [4], mostram a proporção gigantesca que essas reivindicações alcançaram dentro das redes sociais e como estas se tornaram um instrumento necessário para uma análise que procuraria obter conclusões sobre os movimentos contra a desigualdade de gênero.

Com consciência da extensão do debate acerca desse tópico nas redes, entende-se como é importante a mineração e análise de dados para um estudo que permitirá compreender não só, a narrativa desses movimentos como também, identificar características interessantes sobre este ciberativismo. O processo de mineração tem como propósito explorar e analisar dados para revelar padrões que antes não eram conhecidos. Esse processo tem se tornado cada vez mais conhecido pela sua utilização nas técnicas de Redes Neurais [5] com o objetivo de classificar e prever dados. Outra técnica que é muito utilizada no processo de mineração de dados são as aplicações de NLP [6], estas populares pela sua habilidade de entender e extrair informações da linguagem dos seres humanos.

Este projeto implementará um script desenvolvido em Python, com o objetivo de fazer uma extração, tratamento e posteriormente, uma análise [5,6] acerca dos dados relacionados ao tópico da luta contra a desigualdade de gênero, extraídos das uma das principais redes sociais utilizadas atualmente: o Twitter [1]. Esta análise abordará técnicas de classificação utilizando aplicações em Processamento de Linguagem Natural e de previsão através de Redes Neurais [5,6].

Dessa forma, conseguindo associar os picos dos protestos nessa rede social, aos fatores que teriam impulsionado os mesmos e permitindo uma análise exploratória desses tweets bem como, uma análise do conteúdo dos mesmo, o que tornaria possível identificar informações importantes sobre este protesto na rede social em questão.

3. OBJETIVOS

A proposta deste projeto final consiste em: (1) fazer a coleta de tweets relacionados a luta contra a desigualdade de gênero; (2) pré-processar e tratar os dados recolhidos; (3) realizar uma análise exploratória dos dados com o objetivo de extrair informações relevantes sobre os mesmos [7, 8]; (4) estudar algoritmos e possibilidades de análises utilizando NLP [6]; (5) realizar uma análise do conteúdo dos tweets coletados utilizando técnicas de NLP estudadas na etapa anterior. (6) estudar algoritmos e possibilidades de análises utilizando Redes Neurais. (7) realizar uma análise do conteúdo dos tweets coletados utilizando Redes Neurais conforme estudado na etapa anterior.

O objetivo deste projeto de conclusão de curso é estudar, tratar e explorar os tweets coletados bem como, pesquisar algoritmos e aplicações de NLP e Redes Neurais que permitam analisar, classificar e fazer previsões acerca do conteúdo das postagens.

Além disso, tem-se também como objetivo, documentar as etapas de forma didática a fim de deixar claro como o uso de Redes Neurais e Processamento de Linguagem Natural, podem ser úteis para analisar postagens de redes sociais.

Ao final, espera-se adquirir informações e detalhes com qualidade e precisão assim como, traçar associações e comparações a respeito dos protestos de desigualdade de gênero no Twitter Dessa forma, fazendo possível um estudo sobre esse tópico tão relevante e popular.

4. ATIVIDADES REALIZADAS

Nesta seção serão detalhadas as atividades realizadas para o desenvolvimento deste projeto. Além disso, serão apresentados os cronogramas idealizados para o realização das mesmas. Mais detalhes sobre a

implementação de cada uma das etapas listadas abaixo, serão abordados na seção referente a implementação do projeto (seção 6).

4.1 Estudos Preliminares

Inicialmente, foi necessário realizar estudos básicos relativos às aplicações de NLP, Redes Neurais e as ferramentas e bibliotecas que seriam necessárias para realizar o projeto. Dessa forma, conhecendo e ganhando experiência no ambiente tecnológico escolhido para o desenvolvimento do mesmo.

Paralelamente a esses estudos, foi desenvolvida uma pesquisa sobre as formas de extração de dados [7, 8] e possíveis fontes de datasets. Esses conhecimentos foram essenciais para que houvesse o entendimento dos desafios que seriam enfrentados durante a realização do projeto.

4.2. Coleta de Dados

Primeiramente, é importante entender que durante os estudos sobre as formas de extração de dados e possíveis fontes de datasets, tornou-se notória a facilidade e a popularidade da extração de dados do Twitter. Essa popularidade ocorre pelo Facebook e Instagram serem menos acessíveis através de suas APIs e grande parte de seus usuários manterem seus dados e interações privados enquanto por outro lado, o Twitter possui a vasta maioria de seus usuários em perfis públicos. Além disso, existem várias ferramentas que facilitam a extração de dados do Twitter, que facilitam a estruturação, o manuseio e a análise dos dados extraídos [7]. Devido a esses fatores, esta rede social se tornou o foco da coleta de dados deste projeto por apresentar características que facilitam sua análise e ser muito mais utilizado para pesquisas em comparação com as outras plataformas de rede social.

Dessa forma, a coleta de dados foi realizada por meio de um script desenvolvido na linguagem Python e rodado na plataforma Jupyter Notebook. Estas escolhas foram feitas pela estrutura e interface da plataforma Jupyter, que nos permite exibir os resultados polidamente permitindo uma organização que é muito útil na análise de dados e a linguagem Python por ser uma grande ferramenta que disponibiliza de diversas bibliotecas de dados . Portanto, através delas, foram coletados dados de postagens no Twitter através da Application Programming Interface (API) [9] disponibilizada pela plataforma. Com ela foi desenvolvido um método de busca de tweets através de palavras chaves [8] previamente estabelecidas, estas, fazendo parte do contexto de desigualdade de gênero além do uso de hashtags (termos iniciados com o caracter # e que

representam marcações utilizadas pelos usuários, servindo como identificação do assunto da postagem) que marcaram o movimento nas redes sociais.

4.3. Pré-processamento e Tratamento dos Dados

Essa atividade consistiu em preparar os dados para execução das tarefas de análise. Nela, foram realizadas as tarefas de estudar técnicas de NLP para pré-processar os dados coletados bem como, utilizar essas técnicas para tratar os tweets. Ao buscar pelos tweets em sua API, o Twitter retorna, para cada um deles, diversos atributos. Além do texto e data de criação da postagem, são retornados o número de vezes que a mensagem foi compartilhada, a quantidade de pessoas que marcaram a postagem como favorita, o idioma utilizado, entre outras [9]. Por conta dessa quantidade de informações, foi importante selecionar os atributos que seriam utilizados. Além disso, alguns atributos podem obter dados nulos portanto, foi necessário também fazer um tratamento dos casos desses dados, os removendo.

Também fez parte do pré-processamento dos dados, a remoção das chamadas “stop words” [5]. Essas são palavras que não contribuem para o sentido do texto e portanto, são consideradas descartáveis em uma frase. Através da remoção dessas palavras, é possível reduzir o tamanho do vocabulário, deixando o processo geral de treinamento da NLP mais rápido. Além disso, também foram removidos caracteres indesejados como links, pontos, virgulas, ponto e virgulas, parêntesis, símbolos e etc bem como também, linhas duplicadas da base de dados (algo comum pela funcionalidade do retweet).

Durante toda essa etapa de tratamento dos dados, os resultados obtidos eram apresentados em forma de gráficos e em tabelas, de modo que pudessem ser analisados e as mudanças acompanhadas.

4.4. Análise Exploratória

Além da coleta e tratamento dos dados, também foi realizada uma análise exploratória dos mesmos, com objetivo de extrair algumas informações básicas, como: (1) média de caracteres por tweet; (2) usuários que mais postaram sobre o tema; (3) o tweet mais curtido e o mais retweetado; (4) ECDF da quantidade de tweets, retweets e curtidas; (5) correlação entre as colunas de quantidade de curtidas, quantidade de retweets e tamanho do tweet; (6) nuvem das palavras mais utilizadas; (7) as hashtags mais utilizadas; (8) as fontes mais comuns onde

foram postadas os tweets; (9) índices de tweets por ano; (10) locais mais comuns de onde foram postados os tweets, gerando um heatmap.

A análise exploratória dos dados foi importante para entender melhor os dados coletados através da investigação dos mesmo com estatísticas e representações gráficas.

4.5. Análise do Conteúdo dos Tweets

Através dos estudos realizados sobre aplicações de NLP para análise de tweets, duas técnicas foram descobertas e escolhidas para serem implementadas neste projeto: (1) a extração dos n-grams mais frequentes, ou seja, as palavras co-ocorrentes mais utilizadas nas postagens; (2) a classificação do sentimento de cada tweet coletado.

Para a análise das n-grams, foi realizada a identificação dos bigramas (sequências de 2 palavras co-ocorrentes) e trigramas (sequências de 3 palavras co-ocorrentes) mais comuns nos tweets. Já para a classificação do sentimento, foi realizada uma análise utilizando Machine Learning, o treinamento do modelo foi feito usando a abordagem Bag of Words e o algoritmo Naive Bayes Multinomial.

4.6. Previsão de engajamento em postagens

Através dos estudos realizados sobre aplicações de Redes Neurais em possíveis classificações e previsões que poderiam ser feitas acerca dos tweets coletados. Com isso, foi feita uma análise preditiva utilizando uma rede neural de multicamadas para prever o engajamento que postagens poderiam ter, através da análise de seus títulos.

Nesta previsão, treina-se um modelo com a abordagem Bag of Words, passando as palavras mais utilizadas nos tweets e o engajamento (curtidas e retweets) que as postagens que as continham, tiveram. Desta forma, ao passar títulos de noticias, reportagens ou artigos, o modelo pode prever qual o engajamento que aquela postagem teria, baseando-se nas palavras utilizadas nos tweets que foram mais curtidos e retweetados.

4.7. Cronogramas do Projeto

Inicialmente, foi idealizado um cronograma inicial com as etapas listadas abaixo, como pode ser visto na Tabela 1, que foi apresentado no relatório do Projeto Final I.

Devido a adversidades encontradas durante a etapa de coleta de dados devido as restrições de acesso aos dados da API do Twitter, que levaram a extensão dessa tarefa, as etapas subsequentes listadas abaixo, tiveram que ser postergadas. Isso acarretou em um adiamento e remanejamento de algumas atividades propostas para o projeto como pode ser visto na Tabela 2.

As etapas previstas para o projeto foram as seguintes:

1) Fazer a coleta de dados da API do Twitter.
2) Pré-processar e tratar os dados recolhidos:
   a) Estudar técnicas de NLP para o pré-processamento e tratamento dos dados;
   b) Aplicar as técnicas estudadas;
   c) Visualizar o tratamento dos dados por meio da exibição em tabelas e gráficos;
   d) Avaliar os dados processados;
3) Realizar uma análise exploratória dos dados com o objetivo de extrair algumas informações básicas sobre os mesmos;
4) Registrar os resultados da análise exploratória e as informações extraídas em um documento.
5) Estudar algoritmos de aprendizado de máquina e possibilidades de análises utilizando técnicas NLP:
   a) Definir quais aspectos podem ser analisados e quais perguntas podem ser respondidas utilizando aplicações NLP;
   b) Definir quais algoritmos serão utilizados para coletar essas informações;
   c) De acordo com o(s) algoritmo(s) escolhidos e seus respectivos níveis de complexidade, analisar e definir mais etapas sobre o que será efetivamente implementado para obter os resultados desejados;
6) Estudar tipos de Redes Neurais e possíveis analises que podem ser feitas:
   a) Definir quais aspectos podem ser analisados e quais perguntas podem ser respondidas utilizando Redes Neurais;
   b) Definir quais algoritmos serão utilizados para coletar essas informações;
   c) De acordo com o(s) algoritmo(s) escolhidos e seus respectivos níveis de complexidade, analisar e definir mais etapas sobre o que será efetivamente implementado para obter os resultados desejados;
7) Extrair as informações desejadas acerca do tema proposto.
8) Criar um documento detalhando todo o projeto final.

Tabela 1 - Cronograma Projeto Final I

Etapas	Abr	Maio	Jun	Jul	Ago	Set	Out	Nov
Etapa 1	X	X						
Etapa 2		X	X					
Etapa 3			X	X				
Etapa 4				X				
Etapa 5					X	X		
Etapa 6					X	X	X	
Etapa 7							X	X
Etapa 8		X	X	X	X	X	X	X

Tabela 2 - Cronograma Projeto Final II

Etapas	Abr	Maio	Jun	Jul	Ago	Set	Out	Nov
Etapa 1	X	X	X					
Etapa 2			X					
Etapa 3				X	X			
Etapa 4					X			
Etapa 5				X	X	X		
Etapa 6					X	X	X	
Etapa 7							X	X
Etapa 8		X	X	X	X	X	X	X

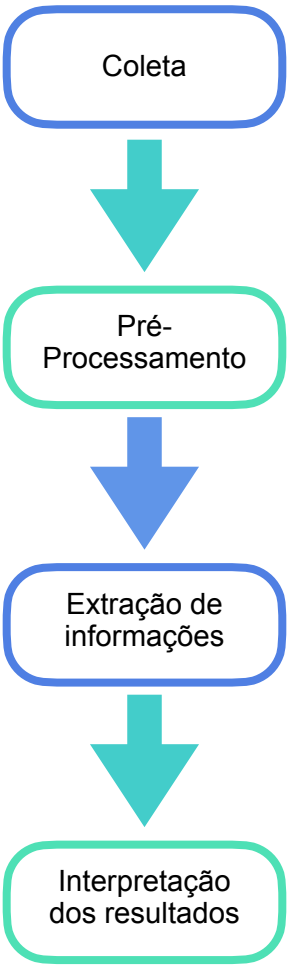
5. Projeto e Especificação do Sistema

Nesta seção serão apresentados os conceitos que foram abordados neste projeto, especificando como estes foram necessários para o desenvolvimento do sistema proposto.

5.1. Mineração dos Textos

Neste projeto, a mineração de textos, foi um conceito amplamente explorado já que esta, refere-se ao processo de obtenção de informações importantes de um texto. Através dela, podemos estabelecer padrões, tendências e relacionamentos afim de organizar e extrair informações desconhecidas e úteis de textos escritos em linguagem natural, no caso deste projeto, estes textos referem-se aos tweets coletados.

O processo de mineração dos textos pode ser dividido em diversas etapas [12], mas quatro delas foram fundamentais para este projeto: (1) coleta de dados; (2) pré-processamento dos dados coletados; (3) extração de informações; (4) interpretação dos resultados. Estas etapas são apresentadas abaixo, na Figura 1.



Fonte: Elaboração própria baseado em Martins (2003).

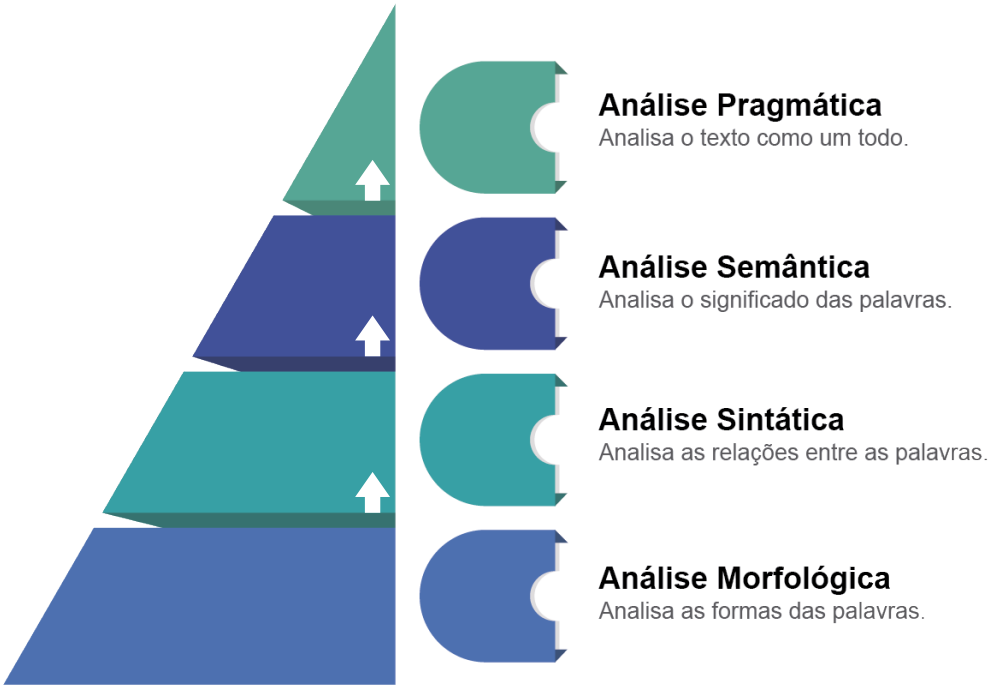
Figura 1 – Etapas do processo de mineração de dados.

Na etapa da coleta de dados, foi utilizada a API do Twitter para extrair os dados através de palavras-chave relacionadas ao tema de desigualdade de gênero. Na etapa de pré-processamento, foram utilizadas técnicas de processamento de linguagem natural para tratar padronizar os dados recebidos. Na etapa de extração de informações, foram identificados padrões afim de extrair informações relevantes sobre o tópico e estes foram apresentados por meio de tabelas e gráficos. Já na etapa da interpretação dos resultados, estes foram validados e analisados para as considerações finais do projeto.

5.2. Processamento de Linguagem Natural (NLP)

Processamento de Linguagem Natural, ou Natural Language Processing (NLP), é uma subárea da Inteligência Artificial que tem como objetivo fornecer ás máquinas a capacidade de entender textos escritos na linguagem dos seres humanos. Dessa forma, se torna possível interpretá-los, extrair informação, ou até mesmo analisar os sentimentos dos mesmos. Estas aplicações tem ganhado muita importância com aumento da geração de texto em plataformas online e redes sociais, já que se tem se tornado cada vez mais útil a interpretação dessas informações.

Para obter uma visão geral das análises textuais feitas através do Processamento de Linguagem Natural, apresenta-se a pirâmide de NLP e seus diferentes estágios de análise de um texto, na Figura 2.



Fonte: Elaboração própria baseado em Nlpforhackers (2021).

Figura 2 – Estágios da análise de um texto.

Apesar das diversas possibilidades de aplicações de NLP, esta foi explorada neste projeto nas etapas da análise exploratória, onde padrões foram identificados e informações foram extraídas de forma estruturada e apresentadas graficamente, e na etapa de análise de sentimentos, onde foi rastreada a importância e o sentimento das postagens coletadas através de algoritmos de aprendizado de máquina.

5.3. Aprendizado de Máquina

Aprendizado de Máquina, ou Machine Learning (ML), é outra subárea da Inteligência Artificial que tem como objetivo possibilitar que o computador aprenda a identificar padrões que existam em uma determinada base de dados [13]. Dessa forma, induzindo classificações e reconhecimentos através do aprendizado com soluções anteriores. Os algoritmos de Aprendizado de Máquina podem utilizar de diversos métodos de aprendizagem, entre eles, destacam-se:

Aprendizagem supervisionada: este método utiliza apenas entradas já rotuladas, ou seja, as quais já foram definidas previamente. Deste forma, o algoritmo aprende as regras que mapeiam as entradas para classificar os objetos da saída [14].

Aprendizagem semissupervisionada: este método classifica os objetos da saída combinando, em sua entrada, uma quantidade de dados rotulados com uma quantidade de dados não rotulados para o treinamento do algoritmo [14].

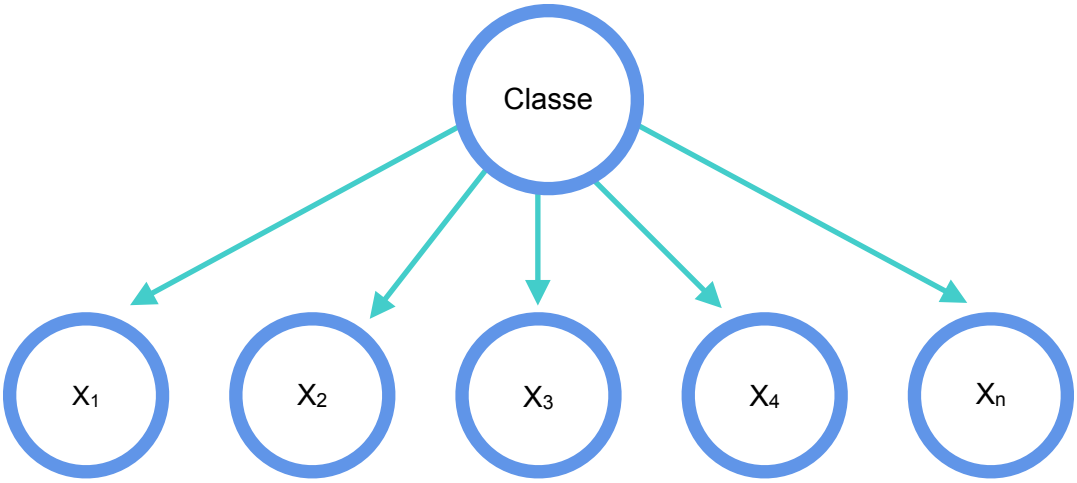
Aprendizagem não supervisionada: este método não utiliza entradas com rótulos previamente definidos para treinar o algoritmo. Dessa forma, este pode encontrar padrão ocultos nos dados e formar sua própria estrutura de classificação das entradas [14].

Através do Aprendizado de Máquina, diversos algoritmos podem ser abordados porém, neste projeto, foi abordado o algoritmo Naive Bayes para realizar a análise de sentimentos dos tweets coletados.

5.3.1 Naive Bayes

O Naive Bayes é um classificador do aprendizado de máquina supervisionado, baseado no teorema Bayes. Ele consiste em uma Rede Bayesiana de estrutura simples, que tem como base a probabilidade de cada evento ocorrer assumindo que todas variáveis de uma classe são independentes

entre si [15]. Essa característica que faz com que este algoritmo seja considerado “Naive”, ou seja, ingênuo, já que todas as variáveis contribuem de forma independente para seu cálculo de probabilidade. Esta independência pode ser vista através do grafo mostrado na Figura 3.



Fonte: Elaboração própria baseado em Santos (2007).

Figura 3 – Grafo do classificador Naive Bayes.

Podemos ver através da Figura 3, que todas as variáveis x_1, x_2, x_3, ..., x_n, possuem somente relação de dependência com a variável classe que seria a raiz, mas mantem a independência entre elas [15]. A classificação com o Naive Bayes utiliza como entrada um conjunto de dados já rotulados que serão utilizados como base para novas classificações. Dessa forma, sempre que o algoritmo tiver que classificar dados com rótulos desconhecidos, estes serão comparados com os que foram utilizados como entrada. O rótulo deste dado desconhecido será definido através do cálculo da probabilidade baseado no cálculo do Teorema de Bayes, definido na Equação 1 abaixo:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Equação 1 – Teorema de Bayes.

Por conta de sua regra de independência, este algoritmo possui um melhor resultado com problemas de classes múltiplas e é muito utilizado para a classificação de textos em análises de sentimentos, como foi utilizado este projeto.

5.4. Redes Neurais

Redes Neurais Artificiais (RNAs), são modelos matemáticos que lembram a estrutura do cérebro humano, tratando-se de uma metáfora da maneira como estas processam as informações utilizadas de forma parecida com a capacidade humana. Sua característica multidisciplinar possibilitou o desenvolvimento de diversos tipos de RNAs já a a expansão de suas aplicações deve-se a uma importante capacidade: aprendizado e generalização [16]. O aprendizado ocorre a partir dos dados de entrada e a generalização é a habilidade de prever resultados para dados desconhecidos, isto acontece por conta da conexão entre neurônios, que são utilizadas para armazenar o conhecimento adquirido na etapa do aprendizado da rede.

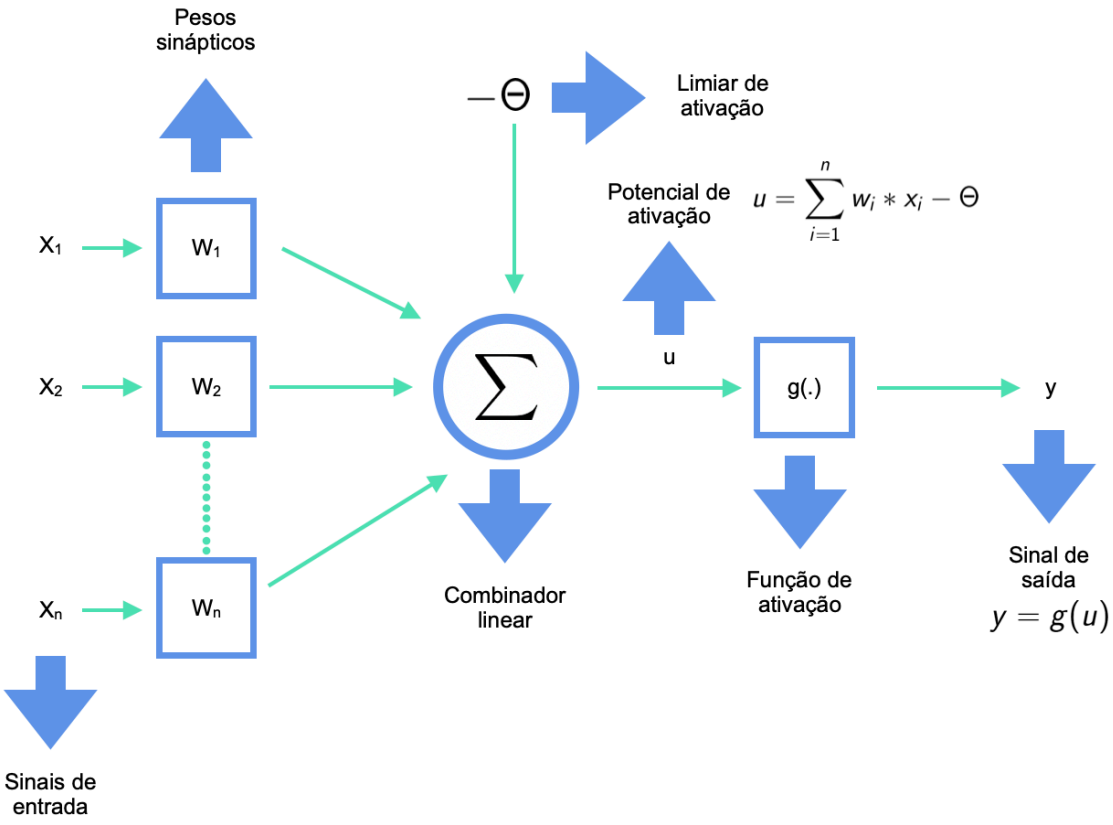
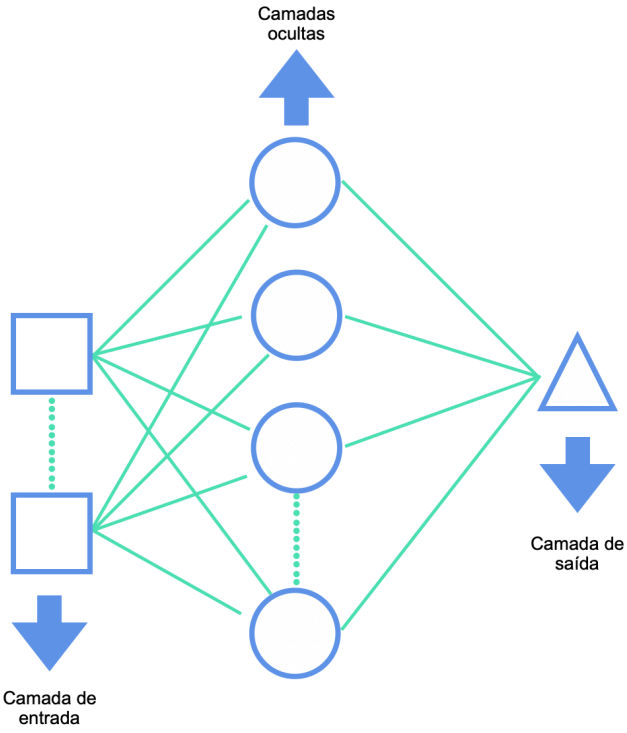


Figura 4 – Funcionamento de um neurônio artificial.

Existem diversas arquiteturas de Redes Neurais, isto porque, a arquitetura é definida pela maneira que esses neurônios, representados pela Figura 4, estão organizados e conectados entre si, ou seja, o número de camadas que a rede possui. Dessa forma, existem diversos modelos para implementação de uma Rede Neural como a SOM (*Self-organizing map*), RBF (*Radius Basis Function*), LMS (*Least Mean Square*), MPL (*Multi-Layer Perceptron*), entre outras [16,17]. Neste projeto foi abordado o uso de uma Rede Neural MLP para a previsão de engajamento de postagens a partir do engajamento obtido nos tweets coletados.

5.4.1 Perceptron Multicamadas (MLP)

A redes do tipo Perceptron Multicamadas (PMC ou MLP — *Multi Layer Perceptron*) possuem uma camada de entrada, uma ou mais camadas ocultas com um número indeterminado de neurônios e uma camada de saída, que contém os dados a serem preditos [16]. Esta arquitetura pode ser vista através da Figura 5.



Fonte: Elaboração própria baseado em Haykin (2007).

Figura 5 – Arquitetura de uma rede MLP.

Pode-se analisar na Figura 5, o diagrama com as disposições das diferentes camadas de uma rede MLP, cada uma delas sendo responsável por uma etapa para a previsão dos resultados:

Camada de entrada: responsável por receber os dados e passá-los para a camada seguinte.

Camadas ocultas: são responsáveis pelo processamento, através delas são transmitidos os dados por meio das conexões entre as camadas de entrada e saída. Seus pesos são uma codificação de características apresentadas nos padrões de entrada e serão multiplicados pelas mesmas, garantindo que a rede crie sua própria representação do problema.

Camada de saída: recebe os dados das camadas ocultas e constrói o padrão, fornecendo a resposta.

6. Implementação e Avaliação

Nesta seção serão apresentados as etapas detalhadas da implementação do projeto, avaliando os resultados obtidos e apresentando trechos do código fonte bem como gráficos e tabelas gerados. Para visualizar o código fonte por completo, este pode ser acessado através do repositório deste projeto final.

6.1. Coleta de Dados

6.1.1. Forma da Coleta

Inicialmente, foi criada a conta de desenvolvedor para acessar a API do Twitter e a partir disso, foi desenvolvido um script em Python para extrair e tratar os dados da API (Application Programming Interface), conforme relatado na Seção 4. Durante esse processo, surgiu a primeira adversidade: ao tentar filtrar para coletar tweets desde 2015 (no no qual o movimento ganhou força e emergiu nas redes sociais) foi identificado que não era possível buscar por tweets com mais de 7 dias do momento da requisição para a API [9]. Isso ocorria já que a conta gratuita de desenvolvedor criada no Twitter faz esta restrição de acesso aos seus dados por data e a solução seria assinar uma conta Premium de desenvolvedor que daria acesso a todos os dados da API, sem limitação de data. A partir da assinatura de um ambiente premium para ter um acesso maior aos dados, foi possível buscar por tweets mais antigos e fazer mais *requests*

para a API, o que possibilitou uma coleta maior de tweets através de um método automatizado de coleta de dados para busca e download dos mesmos.

Este método foi implementado através das bibliotecas *SearchTweets* [10] e *Tweepy* [11] e consistiu nas seguintes etapas: (1) configurar via código o *endpoint* e credenciais de forma a ter acesso ao ambiente criado para desenvolvimento do projeto; (2) definir as palavras-chaves, ano, e idioma das *queries* a serem feitas (como há uma limitação de quantidade de tweets por *request*, foram feitas mais de uma requisição a API, mantendo as palavras-chaves e mudando as datas); (3) armazenar os dados obtidos através das *queries* em um json (pela mesma limitação descrita anteriormente, cada *query* obteve seu próprio json, e todos foram concatenados em um único json denominado: *tweetsData*); (4) transformar o json em um *dataframe* para melhor visualização e manipulação dos dados.

6.1.2. Dados Coletados

Foram coletados tweets entre os dias 01/01/2015 e 30/09/2021, período em que os movimentos começaram a ganhar destaques nas redes sociais, até o momento final da parte correspondente a implementação do script deste projeto. No total, foram coletados e analisados 9.436 tweets, sendo que neste total não se incluem os retweets já que, estes foram removidos durante o tratamento dos dados.

6.1.3. Palavras-Chaves Utilizadas

A busca dos tweets foi feita através de filtros por palavras-chave (Tabela 3) com termos relacionados a desigualdade de gênero e hashtags que representavam o movimento. A definição desses termos foi feita através de pesquisas em artigos e noticias onde foram identificadas as palavras mais utilizadas para definir e buscar este movimento.

As hashtags [4] escolhidas para servirem como filtro de busca, representam dois movimentos que representaram especificamente a luta pela desigualdade de gênero:

#DesafioDalgualdade: campanha criada pela Plan International Brasil [18], ONG responsável pela promoção da igualdade para meninas. Esta, promove uma reflexão e debate sobre igualdade de gênero, com o objetivo de questionar nossos hábitos e comportamentos.

#DeixaElaTrabalhar: manifesto lançado nas redes sociais por jornalista, defendendo o trabalho e a representação das mulheres no meio esportivo.

Tabela 3 - Termos utilizados para coleta de tweets

Tipo	Termos
Palavras-chave	desigualdade de gênero, divisão sexual do trabalho, diferença salarial, equidade de gênero, igualdade de gênero.
Hashtags	#DesafioDaIgualdade, #DeixaElaTrabalhar

Além disso, os termos listados acima na Tabela 3 foram todos buscados apenas na língua portuguesa para a redução do escopo do projeto e para evitar tweets em outros idiomas que poderiam comprometer uma análise mais focada no movimento dentro do Brasil.

6.2. Tratamento dos Dados

A qualidade e normalização da base de dados é um fator muito importante para a Mineração de Textos e para execução das tarefas de análises. Dessa forma, se faz necessário realizar alguns tratamentos para limpar, padronizar, tratar dados ausentes e organizar a base de dados de uma maneira geral. Para tal, algumas técnicas de pré-processamento foram utilizadas:

6.2.1. Ajustes de Atributos

A base de dados dos tweets coletados, possui dezenas colunas de atributo. Isto porque, a API do Twitter, retorna para todo tweet, diversas informações como: o texto da publicação, quantidade de curtidas e retweets, data e fonte da postagem bem como, os dados do usuário como localização, nome, número de seguidores e por ai vai.

Para as análises realizadas neste projeto, precisou-se ajustar alguns atributos por conta do formato que eles estavam sendo retornados pela API. Por exemplo, as fontes de postagens estavam acompanhadas de um link e as datas das postagens estavam em um formato diferente do desejado para os experimentos. Para ajustar esses dados, o link foi removido da coluna *source,* mantendo os atributos com apenas a fonte da postagem e as datas de publicação referentes a coluna *created_at* foram formatadas da maneira

necessária. Estes ajustes feitos no *dataframe* foram possíveis através da biblioteca *Pandas* [19].

6.2.2. Padronização do Texto

Uma técnica importante para padronizar os textos coletados de redes sociais é a remoção de caracteres indesejados, ou seja, pontuações e caracteres especiais. Além disso, é útil padronizar o texto em apenas letras minúsculas para evitar que a mesma palavra seja indexada de mais de uma maneira por conta dos caracteres iniciais diferentes.

Dessa forma, foi implementada uma função *cleanTweet*, para a limpeza dos tweets coletados, nela, remove-se todos os tipos de pontuações, parêntesis, aspas, @ e links dos textos a serem analisados. Abaixo, na Tabela 4, pode-se visualizar um exemplo da aplicação da padronização de textos descrita acima em um texto extraído de um dos tweets coletados.

Tabela 4 - Exemplo da Aplicação da Padronização nos Tweets

Tweet Original	Tweet Padronizado
Como a igualdade de gênero fez da Suécia um país mais rico https://t.co/NJM7L7WbmY	como a igualdade de gênero fez da suécia um país mais rico
@Camy_Dragen exatamente! a tão famigerada igualdade só vai existir quando os papéis de gênero ñ existirem +, e isso serve pro homem tbm	camy_dragen exatamente a tão famigerada igualdade só vai existir quando papéis gênero ñ existirem e isso serve pro homem tbm

6.2.3. Remoção de Stopwords

Além da padronização de texto, a função *cleanTweet* também faz a remoção das *stopwords.* Estas são palavras que não acrescentam nenhum sentido para o texto e são bastante frequentes nos mesmos, como preposições e artigos. Além disso, essa remoção é uma técnica muito relevante para os experimentos e análises que o projeto propõe a realizar, visto que em uma possível exibição de termos mais frequentes dos tweets, por exemplo, as *stopwords* seriam exibidas como os termos mais frequentes, já que são muito

utilizadas ao longo dos textos apesar de não agregarem valor para o mesmo. Dessa forma, os termos realmente relevantes, iriam se tornar pouco frequentes e não seria exibidos nesta possível análise.

Neste contexto, a biblioteca *NLTK (Natural Language Toolkit)* [20] foi utilizada para implementar essa remoção, como podemos ver no Exemplo de Código 1, na função *cleanTweet*. Esta biblioteca contém pacotes para fazer com que as máquinas entendam a linguagem humana e contém registradas as *stopwords* da língua portuguesa.

```
def cleanTweet(tweet):
    tweet = re.sub(r"http\S+", "", tweet).lower()
    tweet.replace('.','').replace(';','').replace('-','').replace(':','').replace(')','')
    tweet.replace('(','').replace('@','').replace(',','').replace('RT','').replace('rt','')
    tweet.replace('!','').replace('?','').replace("'",'')
    stopwords = set(nltk.corpus.stopwords.words('portuguese'))
    words = [i for i in tweet.split() if not i in stopwords]
    return (" ".join(words))
```

Exemplo de Código 1 - Função para o tratamento dos tweets

Abaixo, na Tabela 5, pode-se visualizar um exemplo da remoção das *stopwords* nos tweets exemplificados na Tabela 4, após a padronização de textos.

Tabela 5 - Exemplo da Remoção das Stopwords nos Tweets

Tweet Anteriormente Padronizado	Tweet com as Stopwords Removidas
como a igualdade de gênero fez da suécia um país mais rico	igualdade gênero fez suécia país rico
camy_dragen exatamente a tão famigerada igualdade só vai existir quando papéis gênero ñ existirem e isso serve pro homem tbm	camy_dragen exatamente tão famigerada igualdade vai existir papéis gênero ñ existirem + serve pro homem tbm

6.2.4. Remoção de Linhas Duplicadas

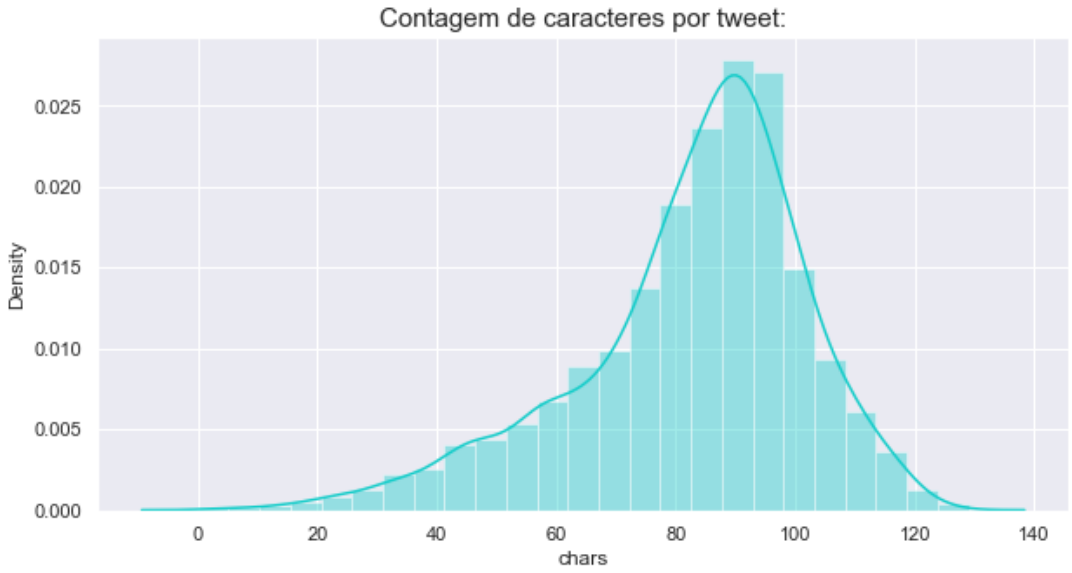
No Twitter, o retweet de uma publicação é muito comum, ou seja, a repostagem daquele post por outro usuário. Com isso, a base de dados pode ter uma grande quantidade de tweets replicados. Desta forma, é importante a

remoção de tweets iguais, para isto, deve-se analisar a coluna *text*, onde está o texto dos tweets. Através da função *drop_duplicates,* fornecida pela biblioteca *Pandas* [19], pode-se comparar e os tweets e remover da base de dados aqueles que estiverem duplicados.

Dessa forma, a base de dados que anteriormente estava com 21.000 tweets, ficou com um total de 9.436 tweets finais para serem analisados e expostos aos experimentos previstos pelo projeto.

6.3. Análise Exploratória

Foi realizada uma análise exploratória com objetivo de extrair algumas informações básicas, que irão ser apresentadas ao longo desta seção. Para dar início a este processo de analise, foi inspecionada a quantidade de caracteres contidos nos tweets, como podemos ver através da Figura 6. Para tal, utilizou-se da biblioteca *Pandas* [19] para criar uma coluna denominada *chars* com a quantidade de caracteres de cada tweet, ou seja, com o tamanho de cada *string* da coluna *text*.



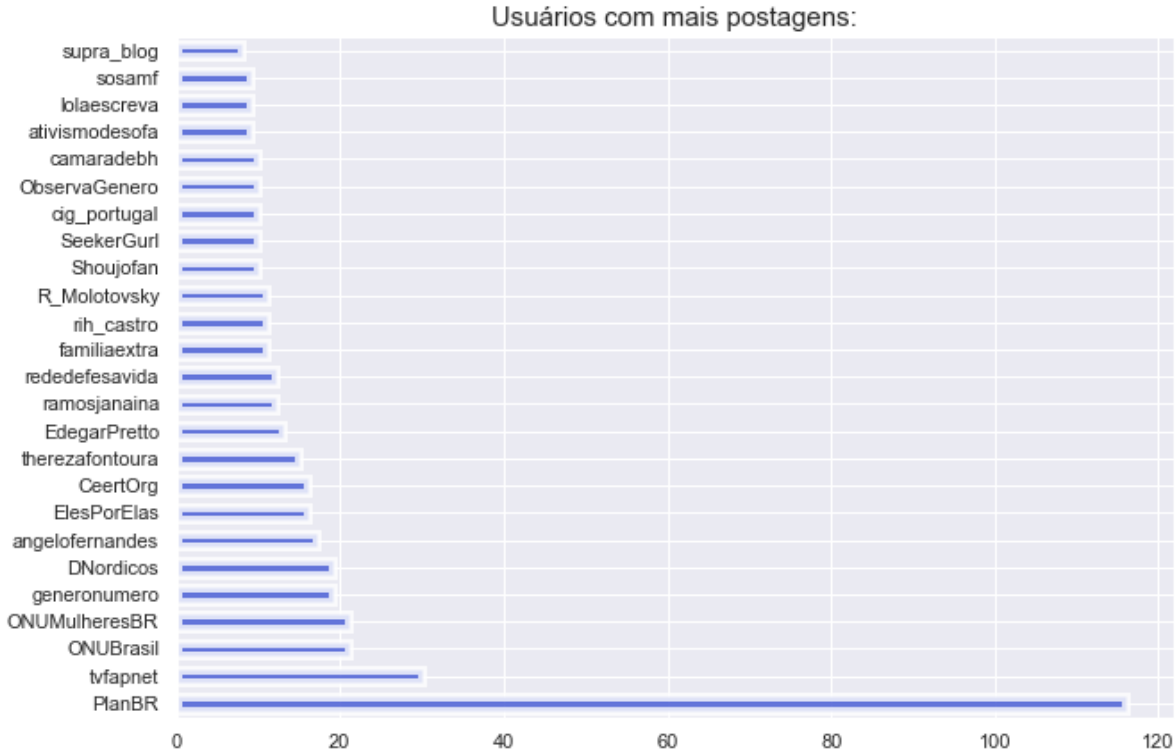
Fonte: Elaboração própria a partir do script desenvolvido em Python.

Figura 6 – Quantidade de caracteres por tweet.

Além disso, houve uma contagem da quantidade mínima, máxima e média de caracteres por tweet, chegando aos números: (1) contagem mínima: 0; (2) contagem máxima: 129; (3) contagem média: 83. Lembrando que, os tweets que continham 0 caracteres são aqueles que correspondem ao compartilhamento de

fotos ou links (estes que foram removidos no processo de tratamento dos dados).

Buscou-se também realizar uma análise exploratória dos dados em relação aos usuários, buscando quais que postaram mais tweets sobre desigualdade de gênero e analisando as (os) possíveis influenciadoras (es), como exibido na Figura 7. Esta foi feita através da coluna *user,* que é composta por dicionários com as informações dos usuários, dessa forma, as informações que iriam ser utilizadas foram extraídas e atribuídas a colunas novas dentro do dataframe. As informações necessárias foram relacionas aos nomes dos usuários, coluna *username* e as informações sobre a localização dos mesmo, coluna *derivated*. Esta última será útil para uma analise futura em relação as regiões das quais esses tweets foram postados.



Fonte: Elaboração própria a partir do script desenvolvido em Python.

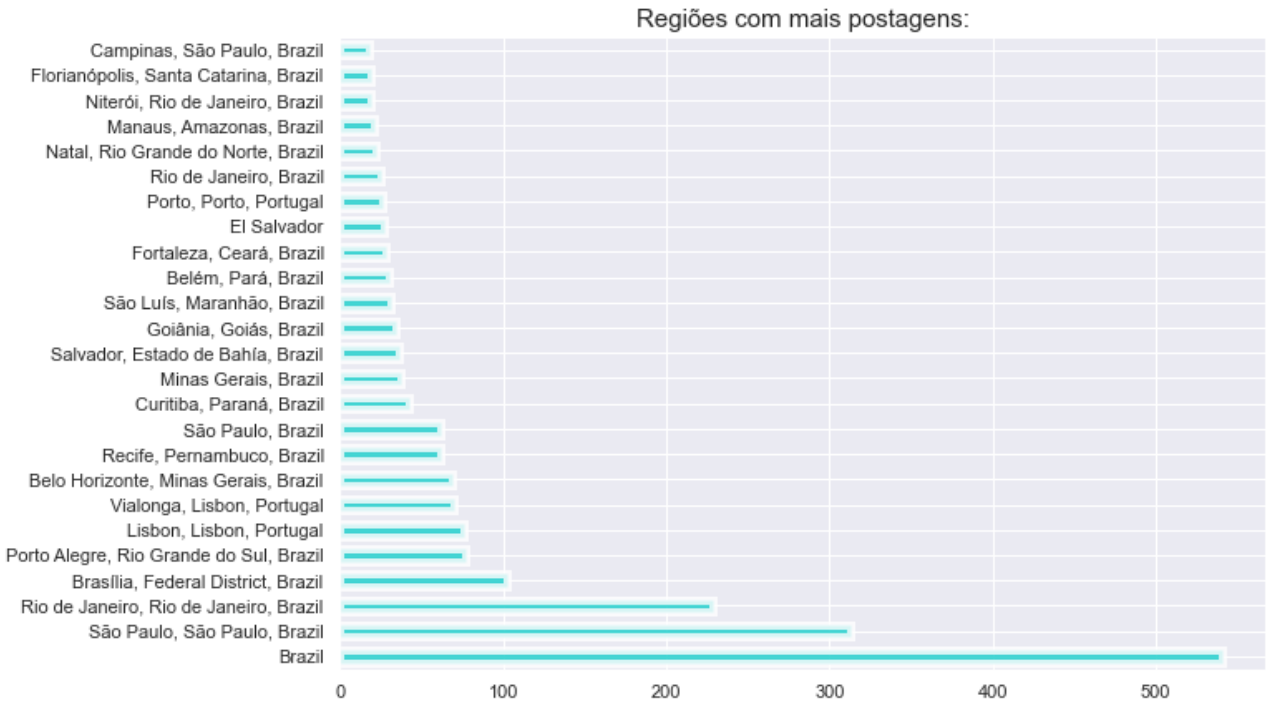
Figura 7 – Usuários com mais postagens.

Pode-se analisar pela figura acima, que o usuário que postou, disparadamente, mais tweets sobre o tópico de desigualdade de gênero, foi o perfil da Plan International Brasil [18], que foi apresentada anteriormente como uma ONG responsável pela promoção da igualdade para meninas. Já o segundo usuário que segue a ONG nesta quantidade de postagens é referente ao

Programa Diferente [21], um programa jornalístico que aborda pautas diferenciadas. Esta é seguida por duas contas da ONU, uma referente a entidade das Nações Unidas no Brasil, a ONU Brasil [22], e a ONU Mulheres Brasil [23], entidade das Nações Unidas para a igualdade de gênero e empoderamento feminino.

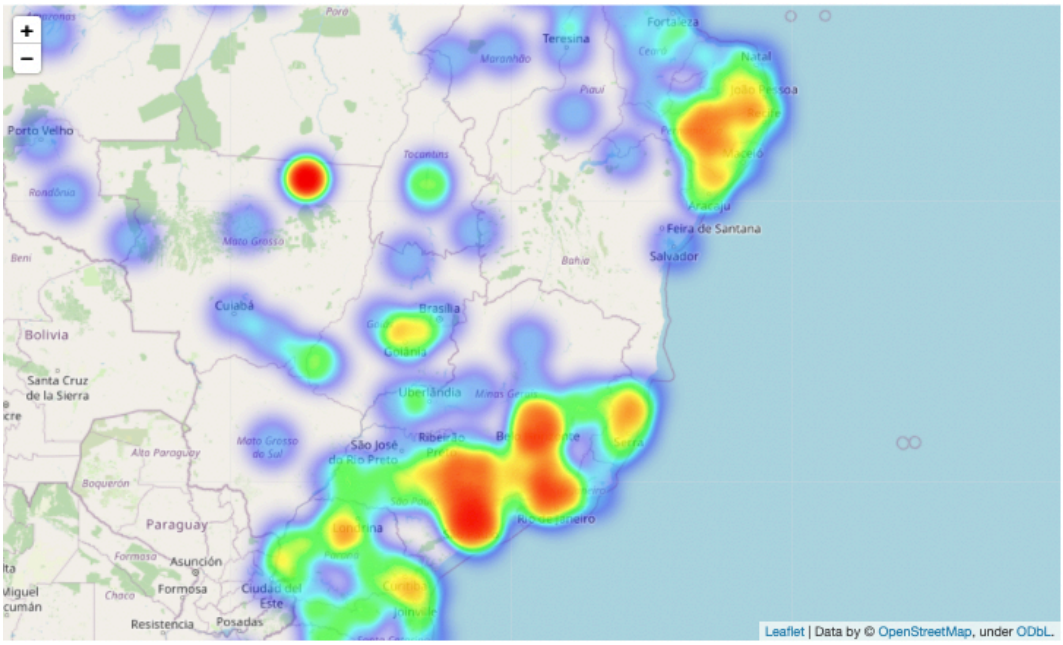
Ainda explorando as informações dos usuários, foi feita uma análise da regiões das quais foram postados os tweets, como apresentado na Figura 8. Dessa forma, foi possível gerar também um mapa de calor em relação a essas regiões, exibido na Figura 9, ambas as representações são úteis para analisar quais as regiões do país estão mais envolvidas no movimento

Esta implementação foi feita através da coluna *derivated*, a qual continha as informações dos locais de publicação dos tweets e da onde foram extraídos os endereços dos mesmos. Importante lembrar que, apenas os usuários que permitiram o acesso a sua localização, vinham com esta informação não-nula nesta coluna.



Fonte: Elaboração própria a partir do script desenvolvido em Python.

Figura 8 – Regiões com mais postagens.



Fonte: Elaboração própria a partir do script desenvolvido em Python.

Figura 9 – Regiões com mais postagens representadas pelo mapa de calor.

Além disso, a representação pelo mapa de calor foi possível através das bibliotecas *GeoPy* e *Folium,* responsáveis, respectivamente, pela busca das latitudes e longitudes dos endereços extraídos e pela representação dos mesmos em um mapa de calor. Através do gráfico e do mapa, pode-se identificar que a região mais envolvida no debate no Twitter é as região sudeste, seguida pela capital Brasília. O envolvimento maior dessa região pode estar diretamente relacionado ao fato de, segundo uma publicação do IBGE (Instituto Brasileiro de Geografia e Estatística), a maior desigualdade salarial do Brasil está na região Sudeste [25, 26].

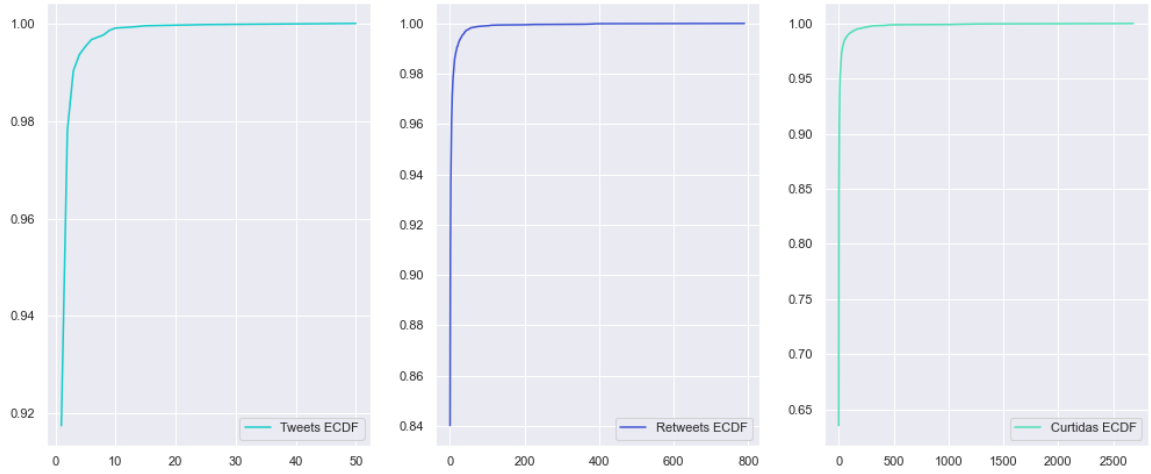
Explorando a quantidade de curtidas e retweets das publicações, outras análises interessantes foram realizadas. Estas duas informações referentes ao número de curtidas e retweets já são retornadas pela API através das colunas *favorite_count* e *retweet_count*. Portanto, buscando em todo dataframe quais tweets tem a maior contagem nessas respectivas colunas, foi possível descobrir qual tweet que teve maior número de curtidas e retweets, ou seja, o que teve mais engajamento dentro da nossa base de dados. Na Tabela 6, é apresentado o resultado dessa busca, bem como as respectivas quantidades de curtidas e retweets da publicação.

Tabela 6 - Resultado do tweet com mais engajamento

Tweet	Número de curtidas	Número de retweets	Ano da postagem
"Marta saiu do interior de Alagoas. É mais fenômeno que qualquer macho batendo bola. Por que não se aposenta milionária? Porque é mulher. Isso se chama desigualdade de gênero no mercado de trabalho"	2.683 curtidas	790 retweets	2018

Ainda explorando esses dados, pode-se apresentar a ECDF (função de distribuição empírica) dos tweets em relação a suas quantidades de curtidas, retweets e publicações por usuário. Esta análise pode nos fornecer bem mais informações sobre os dados coletados já que, modela dados empíricos, ou seja, observados.

A ECDF é uma distribuição de probabilidade que se obtém através da análise de uma amostra, em vez de toda a população de dados. No eixo x estará a quantidade da variável que vai estar sendo medida e o valor de y é a fração daquele ponto do dado correspondente ao seu valor do eixo x. Esta função e seus resultados obtidos podem ser representados através da Figura 10, exibida abaixo.



Fonte: Elaboração própria a partir do script desenvolvido em Python.

Figura 10 – ECDF da quantidade de tweets, retweets e curtidas.

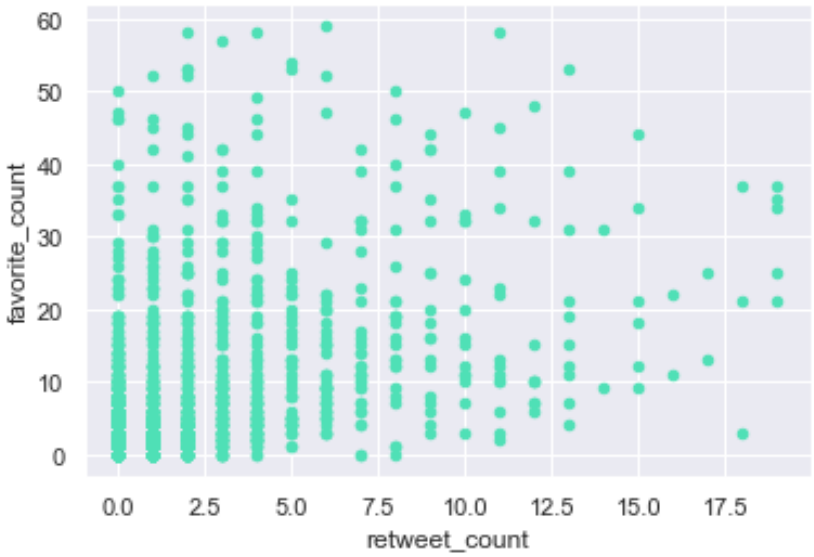
Através destes gráficos pode-se concluir que praticamente toda a base de usuários possui até 10 tweets porém existem *outliers* de usuários com algo próximo aos 20 tweets. Já sobre a quantidade de retweets, percebe-se que praticamente 99% da base tem menos de 100 retweets, a maioria ficando com 0 retweets porém, com *outliers* de usuários com mais de 400 retweets. Já sobre a quantidade de curtidas, nota-se que praticamente 99% da base tem menos de 200 curtidas mas também, com *outliers* de usuários com tweets com mais de 1000 curtidas.

Outra analise que pode ser feita através desses dados, é checar a correlação entre eles. Isto pode ser útil para entender se a quantidade, por exemplo, de caracteres, curtidas e retweets impactam um ao outro, ou seja, se a quantidade de retweets afeta a quantidade de curtidas ou até mesmo se o tamanho de um tweet afeta seu engajamento. Uma forma de identificar possíveis relações entre essas colunas é utilizando do método de correlação *.corr()* da biblioteca *Pandas* [19]. Este, devolve um número de -1 a 1, onde quanto mais próximo das extremidades (-1 e 1), mais forte é a correlação entre as duas colunas e, quanto mais próximo de zero, mais fraca a correlação. Pode-se visualizar os resultados obtidos através deste método na Tabela 7.

Tabela 7 - Resultado da correlação entre as colunas

	favorite_count	**retweet_count**	**chars**
favorite_count	1.000000	0.820257	0.014986
retweet_count	0.820257	1.000000	0.010254
chars	0.014986	0.010254	1.000000

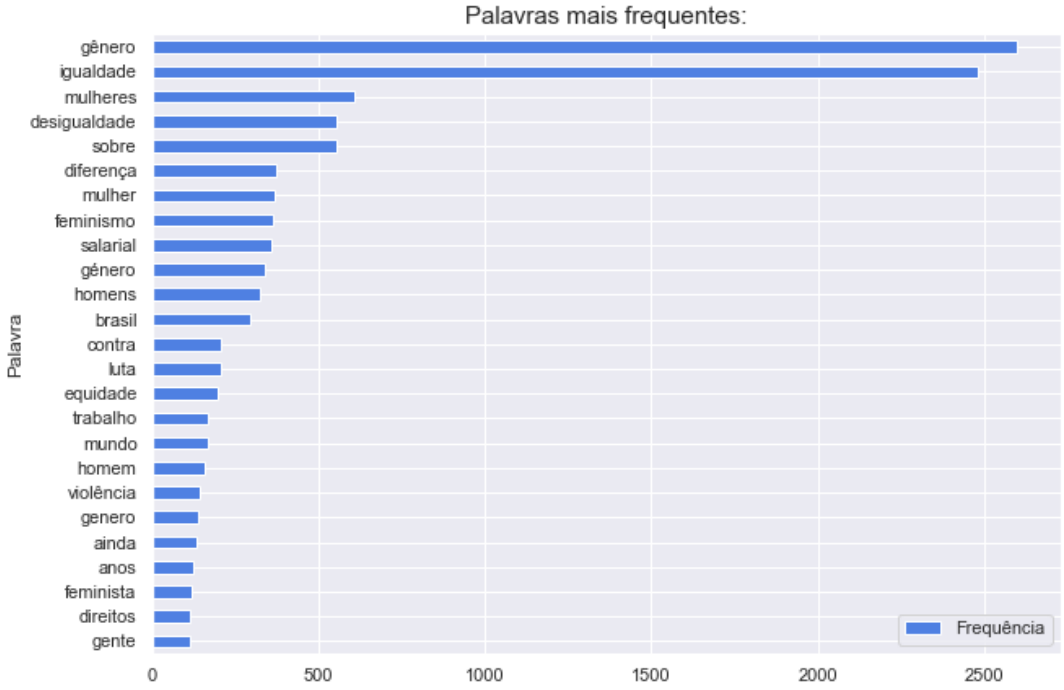
Com a tabela acima, pode-se perceber que as colunas de favoritos e retweets tem uma grande correlação (0.82). Portanto, pode-se filtrar esses valores que possuem mais ocorrências para possibilitar uma melhor observação do gráfico. Onde tem-se uma distribuição “diretamente proporcional” entre os valores das duas colunas, vista na Figura 11.



Fonte: Elaboração própria a partir do script desenvolvido em Python.

Figura 11 – Correlação da quantidade de retweets e curtidas.

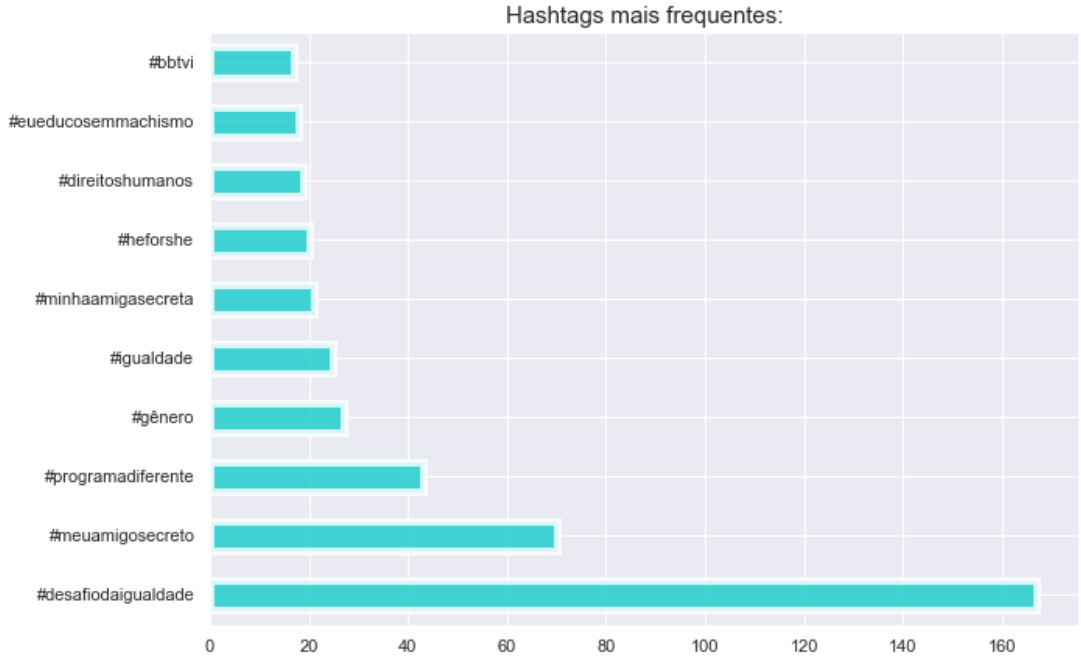
Também foram exploradas as palavras mais frequentes de todas as postagens dessa forma, apresentando o que os usuários estão mais dizendo sobre o tema nos tweets coletados, conforme a Figura 12. Esta abordagem utilizou a biblioteca *NLTK (Natural Language Toolkit)* [20], com a qual pode-se verificar facilmente as palavras mais frequentes dos tweets que tinham mais de 3 caracteres, listando as 25 que tinham mais frequência.



Fonte: Elaboração própria a partir do script desenvolvido em Python.

Figura 12 – Palavras mais frequentes nos tweets.

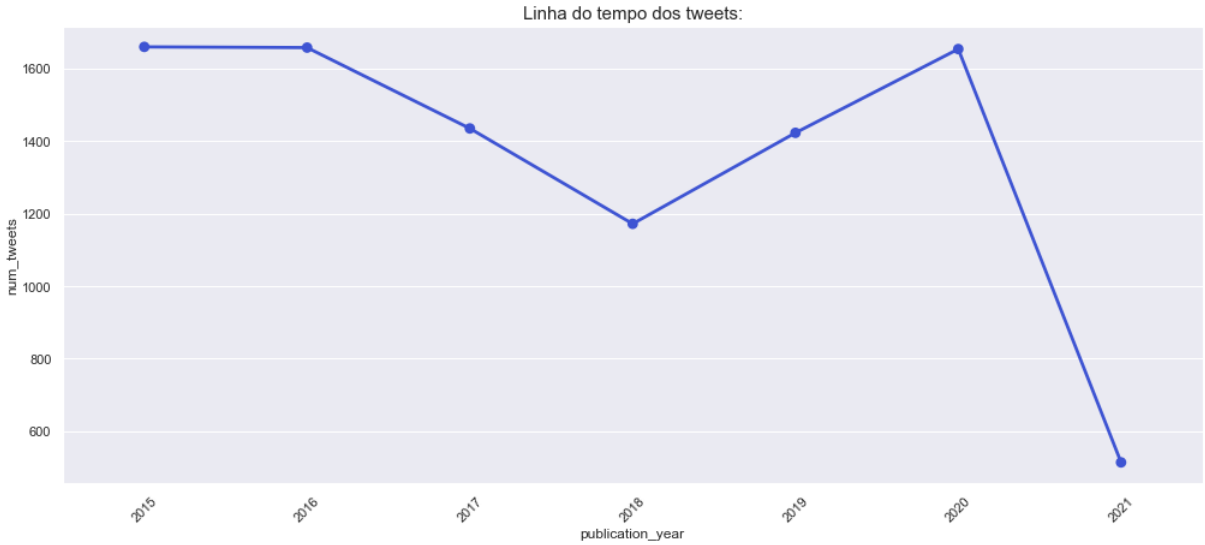
Além disso, também foram coletadas as *hashtags* que mais ocorreram na base de dados, dessa forma, possibilitando uma analise sobre as campanhas que estão relacionadas ao tema de desigualdade de gênero, conforme apresentado na Figura 13. É notório que a *hashtag* com maior ocorrência nos tweets coletados foi a *#DesafioDaIgualdade* já que esta serviu como palavra-chave para a busca dos mesmos na API do Twitter. Além dela, outras hashtags conhecidas apareceram: (1) #MeuAmigoSecreto, campanha que viralizou em 2015 que tinha como objetivo denunciar o machismo; (2) #HeForShe (ou em português #ElesPorElas), esta sendo uma campanha de solidariedade pela igualdade de gênero criada em 2014 pela ONU Mulheres [23].



Fonte: Elaboração própria a partir do script desenvolvido em Python.

Figura 13 – Hashtags mais frequentes nos tweets.

Por último, foi feita uma análise das postagens ao longo dos anos para identificar o comportamento da quantidade de tweets sobre o tema durante dos anos coletados de 2015 a 2021, observando sua evolução. Desta forma, foi feita a analise da coluna *publication_year* para construir uma linha do tempo, onde pode-se observar os anos que mais obtiveram postagens e os que tiveram menos, conforme a Figura 14. Uma análise futura poderia indicar quais fatores contribuíram para o aumento ou queda da discussão sobre o tema nas redes sociais, associando a eventos divulgados pela mídia ou até mesmo aqueles que viralizaram nas redes sociais.



Fonte: Elaboração própria a partir do script desenvolvido em Python.

Figura 14 – Linha do tempo de tweets.

6.4. Extração das n-grams

Uma aplicação comum de NLP é análise do conteúdo do texto onde seus termos são analisados dentro do contexto em que são utilizados para extrair-se significados das mensagens, uma dessas possíveis análises é a extração dos *n-grams.* Um *n*-grama é uma sequência de *n* itens (fonemas, silabas, letras, palavras) contínuos de um texto, neste caso, de um tweet.

Para demonstrar a aplicação dessa técnica, foi realizada uma análise dos bigramas, referente ao *n-grams* de duas palavras e dos trigramas, referente ao n-grams de três palavras. Para isto, foi necessária a tokenização do textos dos tweets, este é o processo de dividir um texto em uma lista de palavras, como exemplificado na Tabela 8.

Tabela 8 - Exemplo da Tokenização de um Tweet

Antes	Depois
'igualdade gênero fez suécia país rico'	['igualdade', 'gênero', 'fez', 'suécia', 'país', ‘rico']

Para isto utilizou-se novamente a biblioteca de linguagem natural, *NLTK* [20], que fornece um pacote específico para a tokenização de tweets chamado *TweetTokenize* além do pacote *ngrams*. A partir desses pacotes, pode-se listar

as sequências de palavras co-ocorrentes mais frequentes, obtendo as 5 bigramas e trigramas e suas respectivas ocorrência da base de dados, como apresentado na Tabela 9 e na Tabela 10.

Tabela 9 - Bigramas e suas ocorrências.

Bigramas	Ocorrências
(igualdade, gênero)	3251
(desigualdade, gênero)	736
(diferença, salarial)	516
(sobre, igualdade)	416
(homens, mulheres)	238

Tabela 10 - Trigramas e suas ocorrências.

Trigramas	Ocorrências
(sobre, igualdade, gênero)	322
(diferença, salarial, homens)	145
(salarial, homens, mulheres)	143
(luta, igualdade, gênero)	139
(fórum, econômico, mundial)	93

6.5. Análise de Sentimentos

A Análise de Sentimentos tem como objetivo classificar e um texto com sentimento de positivo, negativo ou neutro de forma automática a partir da base de dados pré-processada e padronizada. Dessa forma, pela execução de um algoritmo de aprendizado de máquina, torna-se possível classificar cada tweet coletado quanto à sentimento do seu autor.

Para realizar esta análise, um modelo de classificação de sentimentos foi treinado, teve sua acurácia avaliada e esta, ao se mostrar satisfatória, permitiu que este modelo fosse aplicado na base de dados para a detecção dos sentimentos acerca dos tweets sobre o tema deste projeto.

6.5.1. Treino do modelo

O método utilizado para esta análise foi o aprendizado supervisionado, ou seja, o algoritmo aprende o modelo a partir de entradas já rotuladas, com as quais este, aprende as regras que as mapeiam para classificar novos dados. Para isso, foi utilizado um *dataset,* disponibilizado pelo governo de Minais Gerais, que continha 8199 tweets em português já com seus respectivos sentimentos detectados. Esta base de treino já possuía 3300 tweets classificados como positivos, 2446 classificados como negativos e 2453 classificados como neutros.

Através deste *dataset*, treina-se o modelo utilizando a abordagem *Bag of Words* e o algoritmo *Naive Bayes*, disponíveis pela biblioteca *NLTK* [20], já mencionada anteriormente. A abordagem Bag of Words treina o modelo com um vetor que contém as palavras da base e suas respectivas quantidades de ocorrências. Para esta abordagem, deve-se novamente tokenizar os textos dos tweets porém esta tokenização foi feita através do pacote *CountVectorizer* que além de tokenizar o texto, criar um vocabulário com as palavras conhecidas. Além disso, também foi utilizado o método *fit_transform* para ajustar o modelo, aprender o vocabulário e transformar os dados de treinamento em vetores com frequência das palavras. A implementação dos mesmos pode ser vista através do Exemplo de Código 2, abaixo.

```
1   tweets = training_dataset['Text'].values
2   classification = training_dataset['Classificacao'].values
3   tweet_tokenizer = TweetTokenizer()
4
5   vectorizer = CountVectorizer(analyzer="word", tokenizer=tweet_tokenizer.tokenize)
6   freq_tweets = vectorizer.fit_transform(tweets)
7   type(freq_tweets)
8   freq_tweets.shape
9
10  # Treinamento do modelo
11  model = MultinomialNB()
12  model.fit(freq_tweets,classification)
```

Exemplo de Código 2 - Treino do modelo para classificação dos sentimentos.

Com o objetivo de propor uma melhoria no modelo de classificação foi criado um método para acrescentar uma tag de negação nos textos dos tweets após a identificação da palavra “não”. Essa abordagem tem como objetivo dar mais peso para o modelo identificar uma inversão de sentimentos da frase, o que auxiliaria a classificar um texto com o sentimento negativo. Essa função, apresentada no Exemplo de Código 3, foi aplicada nos tweets e os resultados

foram comparados para analisar qual modelo foi melhor para a classificação esperada.

```
def applyNegTag(tweet):
    neg = ['não']
    neg_tag = False
    result = []
    words = tweet.split()
    for word in words:
        word = word.lower()
        if neg_tag == True:
            word = word + '_NEG'
        if word in neg:
            neg_tag = True
        result.append(word)
    return (" ".join(result))
```

Exemplo de Código 3 - Função para aplicar a tag de negação.

6.5.2. Métricas de Avaliação

Em problemas de classificação, deve-se saber se o modelo esta classificando da maneira desejada, para isso, deve-se avaliar o modelo. Dessa forma, é possível utilizar de algumas métricas para avaliar modelos de classificação, dentre elas, algumas foram exploradas neste projeto. Abaixo, exibe-se a função implementada para aplicar todas as métricas que serão abordadas nos tópicos seguintes através do Exemplo de Código 4.

```
def giveMetrics(model, tweets, classification):
    results = cross_val_predict(model, tweets, classification, cv=10)
    print('Acurácia do modelo: {}'.format(metrics.accuracy_score(classification,results)))
    print(metrics.classification_report(classification,results,all_sentiments))
    print(pd.crosstab(classification, results, rownames=['Real'], colnames=['Predito'], margins=True))
```

Exemplo de Código 4 - Função para avaliação do modelo.

6.5.2.1. Acurácia

Métrica responsável por revelar a performance do modelo, ou seja, indica quantas, dentre todas as classificações, o modelo treinado classificou corretamente. Esta informação foi possível ser extraída através da biblioteca *Scikit-Learn* [27], uma biblioteca de aprendizado de máquina, que disponibiliza de um pacote chamado *metrics.* Este pacote possui o método *accuracy_score* que retorna o valor da acurácia do modelo treinado, os resultados obtidos nos dois modelos são apresentados na Tabela 11.

Tabela 11 - Resultados de acurácia dos modelos

Sem tag de negação	Com tag de negação
0.8767	0.7087

6.5.2.2. Matriz de confusão

A matriz de confusão condiz em uma tabela que indica as frequências de classificação para cada classe do modelo comparando com os resultados esperados pelo mesmo, ou seja, ela é responsável por indicar a quantidade de erros e de acertos que o modelo teve. Abaixo seguem as Tabelas 12 e 13 com as matrizes de confusão respectivamente dos modelos sem tag de negação e com a tag de negação, ambas foram geradas através da biblioteca *Pandas* [19]. Nelas, são indicados os valores preditos, correspondentes aos sentimentos que o modelo classificou e os valores reais, correspondentes ao sentimento que é esperado, ou seja, como o modelo deveria ter classificado.

		Valores preditos		
		Negativo	**Neutro**	**Positivo**
Valores reais	**Negativo**	2245	192	9
	Neutro	258	2060	135
	Positivo	49	368	2883

Tabela 12 - Matriz de confusão modelo sem tag de negação.

		Valores preditos		
		Negativo	**Neutro**	**Positivo**
Valores reais	**Negativo**	1620	386	440
	Neutro	408	1443	602
	Positivo	115	437	2748

Tabela 13 - Matriz de confusão modelo com tag de negação.

Analisando as tabelas acima, pode-se facilmente identificar que o modelo sem a tag de negação obteve resultados melhores em suas classificações já que, somente 9 tweets que eram de fato negativos foram classificados como positivos e apenas 49 tweets que deveriam ser classificados como positivos foram classificados pelo programa como negativos. Apesar de uma grande quantidade de tweets acabarem sendo classificados como neutros apesar de possuirem outra polaridade, ainda assim este modelo apresenta classificações melhores do que o possui a tag de negação. Além disso, pode ser observado que no modelo com tag, o número de tweets classificados como negativos de forma correta, diminuiu bastante. O que significa que, nesse caso, o uso das tags de negação não ajudou a melhorar o modelo.

6.5.2.3. Precisão, Revocação e F1-score

Outras métricas interessantes de serem exploradas são as métricas de precisão, revocação e F1-score do modelo desejado. Estas podem nos permitira analisar os seguintes fatores:

Precision (Precisão): dentre todas as classificações de classe Positivo que o modelo fez, quantas estão corretas. Determina o quão bom o modelo é em prever valores positivos.

$$precision = \frac{TP}{TP + FP}$$

Equação 2 – Fórmula de cálculo da precisão do modelo.

Recall (Revocação): dentre todas as situações de classe Positivo como valor esperado, quantas estão corretas. Determina o quão bem o modelo trabalhou.

$$recall = \frac{TP}{TP + FN}$$

Equação 3 – Fórmula de cálculo da revocação do modelo.

F1-score: média harmônica entre *precision* e *recall*.

$$2 * \frac{precision * recall}{precision + recall}$$

Equação 4 – Fórmula de cálculo do F1-score do modelo.

Novamente foi utilizado o pacote *metrics* da biblioteca *Scikit-Learn* [27] pois esta possui uma função muito útil chamada *classification_report*, que entrega todas as métricas descritas acima prontas em uma tabela. baixo seguem as Tabelas 14 e 15 com os resultados respectivamente dos modelos sem tag de negação e com a tag de negação.

	precision	recall	f1-score	support
Positivo	0.95	0.87	0.91	3300
Negativo	0.88	0.92	0.90	2446
Neutro	0.79	0.84	0.81	2453
accuracy			0.88	8199
macro avg	0.87	0.88	0.87	8199
weighted avg	0.88	0.88	0.88	8199

Tabela 14 - Métricas do modelo sem tag de negação.

	precision	recall	f1-score	support
Positivo	0.73	0.83	0.78	3300
Negativo	0.76	0.66	0.71	2446
Neutro	0.64	0.59	0.61	2453
accuracy			0.71	8199
macro avg	0.71	0.69	0.70	8199
weighted avg	0.71	0.71	0.71	8199

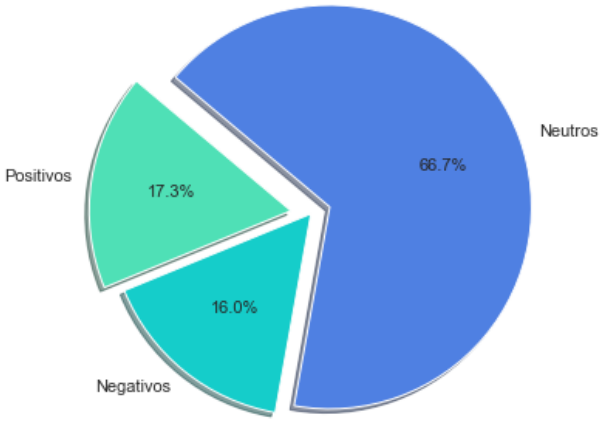
Tabela 15 - Métricas do modelo com tag de negação.

6.5.3. Resultados obtidos

Através das métricas expostas acima, pode-se perceber que o modelo que não possuía a tag de negação teve um melhor desempenho, não só com uma

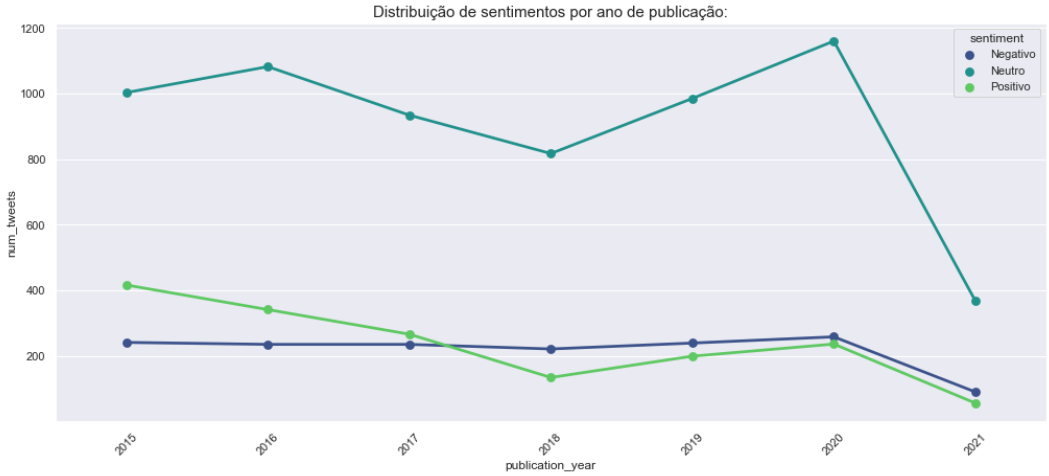
acurácia mais alta mas também com resultados melhores na matriz de confusão e nas métricas de precisão, revocação e F1-score.

Portanto, este modelo foi aplicado na base de dados de tweets sobre desigualdade de gênero para classificar os sentimentos dos mesmos, obtendo-se os seguintes resultados: (1) 1648 tweets classificados como positivos; (2) 1519 tweets classificados como negativos; (3) 6349 tweets classificados como neutros. Através destes foram gerados os gráficos correspondentes a porcentagem de cada classificação de sentimento e a distribuição desses sentimentos ao longo dos anos coletados, como pode-se observar respectivamente na Figura 15 e na Figura 16.



Fonte: Elaboração própria a partir do script desenvolvido em Python.

Figura 15 – Porcentagem da classificação dos sentimentos nos tweets.



Fonte: Elaboração própria a partir do script desenvolvido em Python.

Figura 16 – Linha do tempo da classificação dos sentimentos nos tweets.

Pode-se perceber que a maior parte dos tweets foram classificados como neutros, isto pode ocorrer por uma alta frequência de tweets classificados como neutros na base de dados utilizada para testar o modelo e também por alguns tweets demonstrarem um caráter informativo, como reportagens ou dados estatísticos. Já na linha do tempo representada acima, além do pico de tweets classificados como neutros em 2020, podemos analisar uma leve queda dos tweets positivos em 2018, sendo que estes se mantinham mais frequentes que os negativos em 2015, 2016 e 2017. Uma análise futura poderia indicar quais fatores contribuíram para o aumento ou queda de tweets com estes respectivos sentimentos.

6.6. Previsão de Engajamento de Postagens

A última análise proposta por este projeto é uma previsão de engajamento de postagens através do treino de uma Rede Neural a partir dos tweets coletados sobre desigualdade de gênero e seus respectivos engajamentos. Nota-se que é considerado engajamento, a quantidade de interação dos usuários com um tweet, ou seja, o número de curtidas e de retweets que a postagem possui. Através desses e das palavras que continham os tweets com mais engajamento, o modelo prevê qual a possibilidade de outras publicações serem populares nas redes a partir de seus respectivos títulos. Esta implementação se deu a partir das seguintes etapas:

6.6.1. Bag of Words

As redes neurais operam em números, dessa forma, precisa-se transformar os tweets em números para passá-los para a rede. Para isso, utiliza-se a abordagem *Bag of Words*, onde cada palavra é definida como 1 se estiver presente no tweet ou 0 caso não esteja presente. Além disso, para melhorar a precisão, foi feito um pequeno pré-processamento no texto chamado *stemming*. Este, é o processo de reduzir palavras a um ancestral comum portanto, “falar”, “falou” e “falando” são representados por um único nó nas entradas da rede, ao invés de três e é implementado através do pacote *LancasterStemmer* da biblioteca *NLTK* [20]. Isso reduz o número de nós na rede, tornando a previsão mais rápida e mais precisa. Para ambas as etapas descritas acima, foram implementadas funções para aplicar o processo de *stemming* e criar o *bag of words.* Estas estão representadas no Exemplo de Código 5, abaixo.

```python
def prepareTexts(text):
    stemmer = LancasterStemmer()
    stopWords = set(nltk.corpus.stopwords.words('portuguese'))
    text = re.sub('[^A-Za-z ]+', '', text)
    words = nltk.word_tokenize(text.lower())
    return [stemmer.stem(w.lower()) for w in words if w not in stopWords]

def transformInBagOfWords(text, words):
    bag = []
    for word in words:
     bag.append(1) if word in text else bag.append(0)
    return bag
```

Exemplo de Código 5 - Funções para processo de Stemming e abordagem Bag of Words.

Para compor o *bag of words*, a função *transformInBagOfWords*, recebe um tweet e uma lista das palavras mais relevantes das postagens. Estas, foram selecionadas através da sua contagem na base de dados, caso possui-se mais de 5 ocorrências, seria adicionada na lista das palavras com mais importância. Esta lista obteve um total de 2185 palavras que tinham mais de 5 ocorrências na base de dados.

6.6.2. Treinamento da rede neural

Para treinar o modelo, percorre-se os tweets e cria-se duas matrizes: uma para as entradas, os *inputs*, composta pelo *bag of words* e outra para saídas os *outputs*, composta pelo engajamento do tweet, denominado de pontuação. A pontuação é determinada pelo número de curtidas e retweets de cada tweet recebido. Para a rede neural, usa-se o pacote *MLPRegressor* da biblioteca *Scikit-learn* [27], que implementa uma rede *perceptron* multicamadas onde pode-se definir o número de camadas ocultas. Define-se o número de nós ocultos como 50% das entradas, o conjunto discreto de palavras relevantes do *bag of words* e em seguida, 25%, o que resulta em uma rede com 4 camadas, uma de entrada, 2 ocultas e uma de saída. A implementação desse treino pode ser vista no Exemplo de Código 6.

```python
inputs = []
outputs = []

tweets_df['engagement'] = tweets_df['favorite_count'] + tweets_df['retweet_count']

for tweet in tweets_df['text']:
    text = prepareTexts(tweet)
    bag = transformInBagOfWords(text, relevant_words)
    inputs.append(bag)

for engagement in tweets_df['engagement']:
    score = min(engagement,1)
    outputs.append(score)

neural_network = MLPRegressor(activation='relu', alpha=0.0001, hidden_layer_sizes=(int(len(final_words)*0.5),int(len(final_words)*0.25)),solver='adam', max_iter=400)
neural_network.fit(inputs, outputs)
```

Exemplo de Código 6 - Treino da rede neural.

6.6.3. Resultados obtidos

Com a Rede Neural treinada, pode-se fazer previsões passando um pacote de palavras para o método de previsão. Através da biblioteca *Feedparser* [28], busca-se por feeds RSS e passa-se os títulos das postagens para serem analisados e categorizados pelas rede, esta, analisará o título das postagens dos feeds e fará a previsão se as respetivas postagens irão ter engajamento ou não. Foram 6 feeds passados para a rede, onde 4 são relacionados ao tema de treinamento, ou seja, feeds com publicações apenas sobre desigualdade de gênero e os outros 2, são sobre notícias gerais retiradas do site da BBC [29]. A rede retorna o titulo da postagem e a pontuação da mesma, quanto maior esta pontuação, maior a probabilidade de engajamento na mesma. Foram exibidas as 5 postagens previstas com maior engajamento e as 5 postagens previstas com menor engajamento, estas são exibidas respectivamente na Tabela 16 e na Tabela 17.

Título da reportagem	Pontuação
Gloria Steinem: “O autoritarismo começa com o controle sobre o corpo das mulheres”.	1.050453378460511
Belo Horizonte elege sua primeira vereadora trans, Duda Salabert, que faz história com votação recorde.	0.9024637571849576
O que cabe ao gênero é coisa de todos.	0.8860916787416923
Edina Alves, a única árbitra na elite do futebol sonha apitar outra Copa do Mundo.	0.8165052401339323
A Copa do despertar feminista de Marta: “O futebol feminino depende de vocês para sobreviver”.	0.7958150366687816

Tabela 16 - Reportagens previstas com maior engajamento.

Título da reportagem	Pontuação
Japão anuncia que voltará a caçar baleias.	-0.03916179738235226
Gari londrino ganha R$ 17,5 mi na loteria e vai trabalhar no dia seguinte.	-0.024917585896333416
Refém francês é decapitado na Argélia por grupo ligado ao ‘EI'.	-0.018677761165333567
Papa ordena 1ª prisão dentro do Vaticano de acusado de pedofilia.	-0.01641330820896375
Catalunha avança para plebiscito, mesmo com “Não” escocês.	-0.013312710144493756

Tabela 17 - Reportagens previstas com menor engajamento.

Pode-se perceber através das tabelas exibidas acima que, o grupo de postagens que foram previstas terem maior interação a partir de seus títulos, foram aquelas relacionadas ao tema de treino da Rede Neural utilizada. Enquanto, o grupo de postagens previstas para terem menor interação, são relacionadas a títulos de assuntos mais gerais, passados pelos feed de notícias da BBC [29]. Este comportamento é o esperado pela rede, já que foi treinada para prever o engajamento baseado nos tweets coletados sobre desigualdade de gênero.

7. Considerações Finais

Este projeto de conclusão de curso teve como objetivo explorar as possibilidades de uso de dados de redes sociais para uma análise de movimentos sobre a desigualdade de gênero. Para isso, foi realizado um estudo e a implementação de um script em Python com diversas análises sendo realizadas a partir de tweets coletados sore o tema em questão. Os resultados mostraram que muitas informações úteis podem ser extraídas de maneira eficiente a partir de métodos computacionais, fornecendo informações uteis de maneira automatizada. O objetivo do projeto é exploratório e para fins de

aprendizado, com o propósito de prover um entendimento a respeito das tecnologias utilizadas em todo o processo. Dado o escopo da pesquisa, foram utilizadas algumas técnicas de Mineração de Textos, Análises Exploratórias, e Análise de Sentimentos bem como aplicações de Aprendizado de Máquina , Processamento de Linguagem Natural e Redes Neurais para viabilizar o desenvolvimento do projeto.

Apesar dos bons resultados e aprendizado satisfatório adquirido durante o desenvolvimento do projeto, alguns desafios durante o mesmo se destacaram:

Textos curtos: cada tweet tem limite máximo de 280 caracteres, este fator dificulta a dedução do significado das postagens e faz com que seja necessário um número maior de tweets para a extração de informações úteis.

Linguagem informal: também relacionado a este limite de caracteres, é natural que os usuários façam uso de uma linguagem informal, com uso de gírias, abreviações e expressões que são características da comunicação em redes sociais.

Além disso, também foram identificados alguns pontos de melhoria e sugestões de trabalhos futuros para aperfeiçoar as análises feitas, como por exemplo:

Adição de um dicionário de *emojis*: nas redes sociais é muito comum utilizar os famosos símbolos *emojis*, para a comunicação. Além disso, um *emoji* pode ser fundamental para identificar a percepção e o sentimento do usuário no tweet escrito pelo mesmo. A aplicação de um dicionário de emojis, aperfeiçoaria a analise de sentimentos proposta pelo projeto, tornando sua classificação mais precisa.

Integração com outras redes sociais: atualmente utiliza-se como base de dados apenas postagens extraídas do Twitter. No entanto, seria interessante que outras redes sociais também pudessem compor esta base de dados, como por exemplo o Facebook *e o* Instagram*,* assim aumentando a abrangência e diversidade da coleta de dados.

8. Referências Bibliográficas

[1] QUEIROZ, Eliani de Fátima Covem. Ciberativismo: a nova ferramenta dos movimentos sociais. **Revista Panorama-Revista de Comunicação Social**, v. 7, n. 1, p. 2-5, 2017. Acesso em Abril/2021.

[2] DUTRA, Zeila Aparecida Pereira. A Primavera das mulheres: Ciberfeminismo e os Movimentos Feministas. **Revista Feminismos**, v. 6, n. 2, 2018. Acesso em Abril/2021.

[3] REIS, Josemira Silva; NATANSOHN, Graciela. Com quantas hashtags se constrói um movimento? O que nos diz a "Primavera Feminista" brasileira. **Tríade: Revista de Comunicação, Cultura e Mídia**, v. 5, n. 10, 2017. Acesso em Abril/2021.

[4] WITTEKIND, Milena. Empoderamento feminino: estudo de manifestações feministas nas redes sociais por meio de hashtags. 2017. Acesso em Abril/2021.

[5] CARNEIRO, Alvaro. Redes Neurais Convolucionais para processamento de linguagem natural.
Disponível em: <https://medium.com/data-hackers/redes-neurais-convolucionais-para-processamento-de-linguagem-natural-935488d6901b>.
Acesso em Abril/2021.

[6] TAKE BLIP. Tudo sobre NLP: o que é processamento de linguagem natural e seus desafios na Inteligência Artificial. Disponível em: <https://www.take.net/blog/tecnologia/nlp-processamento-linguagem-natural/>. Acesso em Abril/2021.

[7] ARAUJO, Gabriela Denise. Análise de sentimento de mensagens do twitter em português brasileiro relacionadas a temas de saúde. 2014.
Acesso em Maio/2021.

[8] XAVIER, Fernando et al. Análise de redes sociais como estratégia de apoio à vigilância em saúde durante a Covid-19. **Estudos Avançados**, v. 34, n. 99, p. 261-282, 2020. Acesso em Maio/2021.

[9] TWITTER API. Disponível em: <https://developer.twitter.com/en/products/twitter-api>. Acesso em Maio/2021.

[10] PYTHON TWITTER SEARCH API. Disponível em: <https://twitterdev.github.io/search-tweets-python/ >. Acesso em Junho/2021.

[11] TWEEPY. Disponível em: <https://www.tweepy.org>.
Acesso em Junho/2021.

[12] MARTINS, Claudia Aparecida et al. Uma experiência em mineração de textos utilizando clustering probabilístico clustering hierárquico. **Instituto de Ciências Matemáticas e de Computação. São Carlos: Universidade de São Paulo**, 2003. Acesso em Agosto/2021.

[13] MITCHELL, Tom M. et al. Machine learning. 1997. **Burr Ridge, IL: McGraw Hill**, v. 45, n. 37, p. 870-877, 1997. Acesso em Agosto/2021.

[14] ROSSI, Rafael Geraldeli. **Classificação automática de textos por meio de aprendizado de máquina baseado em redes.** 2016. Tese de Doutorado. Universidade de São Paulo. Acesso em Agosto/2021.

[15] SANTOS, Edimilson Batista dos. **A ordenação das variáveis no processo de otimização de classificadores bayesianos: Uma abordagem evolutiva.** 2007. Acesso em Agosto/2021.

[16] HAYKIN, Simon. **Redes neurais: princípios e prática**. Bookman Editora, 2007. Acesso em Setembro/2021.

[17] SILVA, A. M. Utilização de Redes Neurais Artificiais para Classificação de SPAM. **Masters, Centro Federal De Educação Tecnológica De Minas Gerais**, 2009. Acesso em Setembro/2021.

[18] PLAN INTERNACIONAL BRASIL. Disponível em: <https://plan.org.br>. Acesso em Outubro/2021.

[19] PANDAS - PYTHON DATA ANALYSIS LIBRARY. Disponível em: <https://pandas.pydata.org>. Acesso em Outubro/2021.

[20] NLTK - NATURAL LANGUAGE TOOLKIT. Disponível em: <https://www.nltk.org>. Acesso em Julho/2021.

[21] PROGRAMA DIFERENTE. Disponível em: <https://www.programadiferente.com>. Acesso em Outubro/2021.

[22] ONU BRASIL. Disponível em: <https://brasil.un.org/pt-br>. Acesso em Outubro/2021.

[23] ONU MULHERES BRASIL. Disponível em: <http://www.onumulheres.org.br>. Acesso em Outubro/2021.

[24] GEOPY'S. Disponível em: <https://geopy.readthedocs.io/en/stable/>. Acesso em Outubro/2021.

[24] FOLIUM. Disponível em: <https://geopy.readthedocs.io/en/stable/>. Acesso em Outubro/2021.

[25] CNN BRASIL. Disponível em: <https://www.cnnbrasil.com.br/business/mulheres-ganham-77-7-dos-salarios-dos-homens-no-brasil-diz-ibge/>. Acesso em Novembro/2021.

[26] AGÊNCIA DE NOTÍCIAS IBGE. Disponível em: <https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/30172-estatisticas-de-genero-ocupacao-das-mulheres-e-menor-em-lares-com-criancas-de-ate-tres-anos >. Acesso em Novembro/2021.

[27] SCIKIT-LEARN. Disponível em: <https://scikit-learn.org/stable/>. Acesso em Outubro/2021.

[28] FEEDPARSER. Disponível em: <https://pypi.org/project/feedparser/>. Acesso em Outubro/2021.

[29] BBC NEWS BRASIL. Disponível em: <https://www.bbc.com/portuguese>. Acesso em Outubro/2021.