UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Matheus Henrique Ferreira Protásio

Reforma da Previdência no Brasil: Uma análise a partir de dados do Twitter

Uberlândia, Brasil

2021

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Matheus Henrique Ferreira Protásio

Reforma da Previdência no Brasil: Uma análise a partir de dados do Twitter

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Ciência da Computação.

Orientador: Paulo Henrique Ribeiro Gabriel

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2021

Resumo

A reforma da previdência votada e aprovada no ano de 2019 foi uma das pautas mais relevantes na política brasileira nos últimos anos. Dada a relevância do tema, é importante analisar a opinião dos cidadãos sobre o mesmo. Isso pode ser feito utilizando dados provenientes de redes sociais que, com o advento da inclusão digital, cada vez são mais utilizadas pelos brasileiros com o passar dos anos. A rede social mais conhecida pelo compartilhamento rápido de notícias e opiniões na forma de curtos textos é o Twitter, onde esses pequenos textos são chamados tuítes. Neste trabalho foi feita uma análise qualitativa dos dados de uma base já montada com tuítes sobre a reforma da previdência no Brasil no ano de 2019. Foi realizada a classificação dos usuários da base, distinguindo usuários comuns de contas automatizadas (conhecidas como *bots*), esta análise foi feita utilizando a ferramenta *Botometer* que classificou 8.180 usuários como *bots* dentre os 237.894 usuários únicos da base. Com a classificação de usuários realizada, foi analisada a possibilidade de uma "rede" de propagação de mensagens ter sido utilizada para propagar uma determinada opinião a respeito do tema, tudo levou a crer que não houve utilização deste tipo de "rede", pois os *bots* produziram 40.151 tuítes dos 980.577 tuítes da base. Também foram realizadas classificações dos tuítes, por meio da análise de sentimentos, onde as classificações podiam ser: "positivo", "negativo" ou "neutro". Para a realização destas classificações foram utilizados quatro modelos: Naïve Bayes, Regressão Logística, Floresta Aleatória e Máquinas de Vetores de Suporte (SVM). Estes modelos possibilitaram uma comparação de resultados com os resultados obtidos pelo criador da base, pois ele os utilizou em seu trabalho para classificar os tuítes. Com relação à classificação dos tuítes produzidos pelos *bots*, para todos os modelos utilizados, a maioria dos tuítes recebeu classificação neutra. Algo que pode ter levado a este resultado é o fato de que muitos veículos de comunicação utilizam contas automatizadas para replicação de notícias que foram publicadas em seus sites, assim muitas contas de veículos de comunicação foram consideradas *bots* nesta análise. Na etapa de pré-processamento da base também houve o tratamento de *hashtags*, algo que não foi realizado na classificação feita pelo criador da base em seu trabalho. Em seguida foram comparados os novos resultados obtidos após o tratamento de *hashtags* com os resultados originais e foi possível observar que tanto para a classificação original, como para a nova classificação, Naïve Bayes classificou a maioria dos tuítes como positivos, já os demais modelos classificaram a maioria dos tuítes como neutros. Este comportamento se repetiu em outras análises realizadas na base, como por exemplo na classificação dos mil tuítes mais favoritados.

Palavras-chave: Análise de Sentimentos, Mineração de Opinião, Reforma da Previdência, Aprendizado de Máquina, Classificação de *bots*.

Abstract

The social security reform voted and approved in 2019 was one of the most relevant topics in Brazilian politics in recent years. Given the relevance of the topic, it is essential to analyze the opinion of citizens about it. This analysis can be done using data from social networks that Brazilians increasingly used over the years with the advent of digital inclusion. The social network best known for quick sharing of news and opinions in the form of short texts is Twitter, where these short texts are called tweets. In this final course assignment, we propose a qualitative analysis of the dataset already assembled with tweets about the social security reform in Brazil in 2019. The classification of users from the dataset was performed, distinguishing ordinary users from automated accounts (known as *bots*); we performed this analysis using the Botometer tool, which classified 8,180 users as *bots* among the 237,894 unique users from the dataset. With the completed classification of users, we analyze the possibility of a message propagation "network" had been used to propagate a certain opinion on the topic. Everything led to believe that there was no use of this type of "network", as the *bots* produced 40,151 tweets out of the 980,577 tweets. We also classify the tweets through sentiment analysis, where the possible outcomes could be: "positive", "negative" or "neutral". Four models were used to do these classifications: Naïve Bayes, Logistic Regression, Random Forest, and Support Vector Machines (SVM). These models allowed a comparison of results with the results obtained by the creator of the dataset, as he used them in his work to classify the tweets. Regarding the classification of tweets produced by the *bots*, most tweets received the neutral classification for all models used. Something that may have led to this result is that media generally use automated accounts to replicate news published on their websites; thus, many media accounts were considered *bots* in this analysis. In the pre-processing stage, we treated hashtags and compared the new results of this treatment with the initial results. It was possible to observe that both for the original classification and for the new classification, Naïve Bayes classified most of the tweets as positive. In contrast, the other models classified most tweets as neutral. We observed this behavior in other analyses performed in the dataset, such as the classification of the thousand tweets who received the most favorites.

Keywords: Sentiment Analysis, Opinion Mining, Brazilian Social Security Reform, Machine Learning, *bot* Classification.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

No ano de 2019, no Brasil, muito se discutiu sobre a Proposta de Emenda à Constituição (PEC) 6/2019, que tinha como objetivo modificar o sistema de Previdência Social no país. Em seu texto, foi proposta uma nova regra geral para aposentadorias futuras, regra esta que combinava idade mínima e tempo de contribuição. Com as novas regras (MACHADO; SILVEIRA, 2019), o tempo de contribuição mínimo passou a ser de 20 anos na iniciativa privada e 25 anos para servidores públicos; neste último caso, os trabalhadores precisarão atender outros dois pré-requisitos: cumprir pelo menos 10 anos na administração pública e cinco anos no cargo em que se aposentar. A idade mínima prevista para trabalhadores urbanos passou a ser de 65 anos para os homens e 62 anos para as mulheres; já no caso de trabalhadores do campo, a idade mínima para ambos os sexos passou a ser 60 anos.

Após sua tramitação e votação no plenário da Câmara dos Deputados, a PEC 6/2019 viria a ser aprovada, tornando-se a Emenda Constitucional 103/2019 (CÂMARA, 2019). Essa PEC foi de grande relevância no país, justamente devido à sua dimensão e graças às mudanças que provocaria na previdência social do Brasil, algo que afeta brasileiros da geração atual e, principalmente, de gerações futuras. Dada sua importância, esse tema ficou em alta nas redes sociais na época e, como a presença dos indivíduos na internet foi crescendo com o tempo — ao ponto de as redes sociais serem praticamente parte intrínseca da sociedade — torna-se interessante realizar um estudo mais detalhado sobre o "sentimento" das pessoas com relação à Reforma da Previdência. Esse estudo torna-se viável graças à grande quantidade de dados que é gerada diariamente nas redes sociais, nas quais pessoas publicam suas opiniões à respeito de diversos temas, facilitando a análise da opinião pública sobre um determinado assunto.

Este trabalho de conclusão de curso (TCC) tem foco na área de Análise de Sentimentos, também conhecida como Mineração de Opinião (LIU, 2010). Essa área une processos de análise de texto e processamento de linguagem natural, para que seja possível extrair informações relevantes à respeito de um conjunto de dados textuais. A Mineração de Opinião é uma área que pode ser utilizada de diversas maneiras, como por exemplo, empresas que procuram saber o *feedback* e aceitação do público com relação a um determinado produto lançado por ela, ou agências reguladoras que pretendem analisar o impacto de decisões políticas. Além disso, também é possível analisar a opinião das pessoas com relação a um tema considerado relevante para a sociedade (MEDURU et al., 2017; ANSARI et al., 2020; AYLIEN, 2021). Este último caso se encaixa neste TCC, onde o sentimento, a opinião dos usuários com relação à Reforma da Previdência será analisada a partir de uma base de dados.

Foram utilizados dados do Twitter obtidos por Ricci (2020) em seu TCC. O Twitter foi escolhido pois, além de ser uma das redes sociais mais utilizadas no mundo, é geralmente usada para expressar opiniões com relação à diversos temas. Isso acontece na forma de pequenos textos chamados de tuítes. Inicialmente cada tuíte poderia ter no máximo 140 caracteres, mas posteriormente isso foi modificado e hoje tuítes são formados por até 280 caracteres. Também existem os retuítes, que acontecem quando um usuário "compartilha" um tuíte de outro autor. Os retuítes podem ter comentários ou não. Em seu trabalho, Ricci (2020) montou um banco de dados contendo 980.577 tuítes relacionados à Reforma da Previdência no Brasil, os quais foram coletados entre os meses de janeiro de 2019 e novembro de 2019, e classificados em: "positivo", "negativo" ou "neutro". Essa classificação foi feita para representar o sentimento do autor do tuíte com relação à PEC 6/2019. O caso em que um tuíte da base recebe a classificação "neutro" significa a "ausência de sentimento" e, em muitos casos, isso está relacionado com alguma notícia postada por um veículo de comunicação.

1.1 Motivação

Em seu trabalho, Ricci (2020) classificou a base de dados coletada usando quatro algoritmos de Aprendizado de Máquina: Naïve Bayes, Regressão Logística, Florestas Aleatórias e Máquinas de Vetores de Suporte, os quais são comentados posteriormente neste trabalho. Ele comparou os resultados dos diferentes algoritmos e focou na classificação automatizada de toda essa base. No tratamento da base, Ricci (2020) retirou as *hashtags* dos tuítes; porém essas *hashtags* podem representar um componente importante do texto, podendo redefinir sua classificação. Por exemplo, é possível classificar um tuíte como "positivo" se o usuário incluiu *#ReformaJá* no mesmo.

Assim, além do pré-processamento feito antes das classificações obtidas pelo autor, é possível aplicar uma nova etapa, que leva em conta as *hashtags*, retuítes, contas falsas ou suspeitas de serem falsas, além de textos postados por robôs (os chamados *bots*). Com esse novo pré-processamento, a base pode ser reclassificada usando os mesmos algoritmos utilizados pelo autor, assim os novos resultados podem ser comparados com os antigos, proporcionando um estudo mais aprofundado sobre a base.

Dada a relevância do tema para o país e o quanto ele influenciou os noticiários na época, um estudo mais aprofundado dessa base de dados é importante para compreender o posicionamento dos usuários do Twitter com relação à Reforma da Previdência no Brasil e assim analisar a opinião de uma esfera da sociedade brasileira sobre o tema.

1.2 Objetivos

O principal objetivo deste TCC é analisar a base de dados criada por Ricci (2020) sob uma nova ótica, onde novos fatores são considerados, tais como: contas automatizadas e classificação de *hashtags*.

Para a realização desta análise foram removidos os tuítes produzidos por contas automatizadas (os chamados *bots*), bem como novas etapas foram adicionadas ao pré-processamento, desta vez levando em conta as *hashtags*, as quais, como dito anteriormente, são de grande importância no contexto de um tuíte. Também foi feita uma análise das mensagens que receberam mais retuítes e das mais favoritadas.

Neste trabalho foi utilizada a linguagem de programação Python para desenvolvimento do algoritmo de pré-processamento e classificação. Essa linguagem foi escolhida por possuir uma ampla gama de bibliotecas focadas na manipulação e análise de dados. Python é, também, uma linguagem muito documentada e utilizada em pesquisas que envolvem a área de Ciência de Dados (em inglês, *Data Science*) e aprendizado de máquina (em inglês, *Machine Learning*), onde a Mineração de Opinião está inserida.

A verificação de contas automatizadas foi feita por meio da ferramenta *Botometer*, que oferece uma API para isso. Essa ferramenta fornece uma classificação no intervalo [0,5], sendo que quanto mais próximo de 5, maior a chance de a conta ser um *bot*. Se a classificação ficar próxima da metade, a ferramenta está com dificuldade de discernir se a determinada conta é ou não um *bot*. Neste trabalho, para considerar uma conta como automatizada, foram considerados diferentes parâmetros, a fim de verificar qual classificação ficou mais adequada. Os retuítes também foram analisados pois, como a base possui muitas informações a respeito de cada instância, é possível acessar de maneira simples e descobrir quais foram às contas que publicaram os tuítes que mais tiveram retuítes em cada mês e, consequentemente, descobrir quais foram as mensagens que mais receberam retuítes durante todo o período.

A base de tuítes foi classificada novamente por meio dos quatro algoritmos analisados por Ricci (2020), ou seja, Naïve Bayes, Regressão Logística, Florestas Aleatórias, Máquinas de Vetores de Suporte (SVM). Esses algoritmos classificam cada tuíte presente na base em "positivo", "negativo" ou "neutro" de acordo com o sentimento do autor do tuíte com relação à Reforma da Previdência, onde em casos classificados como positivos, o indivíduo estava falando à favor da PEC 6/2019, em casos negativos o indivíduo não concordava ou criticava o projeto, já casos neutros, como citado anteriormente, são caracterizados pela "ausência de sentimento", isto é, tuítes imparciais. Para ser classificado, cada tuíte teve suas palavras de interesse, junto com as suas respectivas *hashtags* armazenadas em um dicionário Python. A frequência de cada palavra será analisada e o contexto das *hashtags* também, essas "métricas" serão utilizadas para classificação individual de

cada tuíte.

1.3 Organização da Monografia

Esta monografia está organizada da seguinte maneira. No próximo capítulo (Capítulo 2) é apresentada a fundamentação teórica e os algoritmos utilizados para classificar os tuítes da base são descritos. Também são discutidos alguns trabalhos relacionados a este TCC. No Capítulo 3, são apresentadas escolhas de projeto, além do desenvolvimento geral e resultados obtidos no trabalho, tais como o processo de classificação dos usuários da base e a reclassificação de tuítes com posterior comparação de resultados. Por fim, o Capítulo 4 apresenta algumas conclusões obtidas através de diferentes análises realizadas no decorrer do trabalho. Também são propostas ideias de trabalhos futuros.

2 Referencial Teórico

Neste capítulo é apresentado o referencial teórico com uma visão geral sobre a área da Análise de Sentimentos, aprendizado de máquina, classificadores utilizados, ferramentas utilizadas para realização deste trabalho, além de trabalhos correlatos desenvolvidos na área.

2.1 Aprendizado de Máquina

A técnica de aprendizado de máquina (do inglês, *Machine Learning*) é um processo computacional que busca atingir um resultado desejado sem ser literalmente programado para produzir esta saída, basicamente esses algoritmos se adaptam e se tornam melhores com o tempo e mais aptos para a obtenção de melhores resultados (NAQA; MURPHY, 2015). Essa técnica "simula" o comportamento de um ser humano que vai aprendendo com seus erros e melhorando. O aprendizado de máquina pode ser aplicado em reconhecimento de imagens, texto, etc.

A fase de treinamento é a fase de aprendizado desses algoritmos, onde eles recebem dados para "aprenderem". Um algoritmo que utiliza a técnica de aprendizado de máquina, por exemplo, é um algoritmo que recebe várias imagens de cachorros e gatos, o algoritmo passa pelo treinamento e vai aprendendo quando deve classificar uma imagem recebida como gato ou como cachorro. Assim, após o treinamento, é passada para o algoritmo uma imagem ou um conjunto de imagens e ele deve realizar uma classificação destes dados. É necessário encontrar um equilíbrio ao treinar o modelo, pois se pouco treinado, ele pode sofrer com o *underfitting*, que é o caso de o modelo apresentar desempenho ruim já no treinamento. Já se o modelo for treinado excessivamente, ele sofre com *overfitting*, que é o caso de o modelo ter um desempenho bom no treinamento, mas ruim nos testes. No caso de *overfitting* o modelo perde a capacidade de generalização e começa a "decorar" o que tem que ser feito.

O aprendizado de máquina é utilizado em diversas áreas da computação, a principal é a área de Inteligência Artificial, com reconhecimento de padrões, onde são utilizados algoritmos refinados capazes de resolverem problemas de classificação e regressão simulando o comportamento de um ser humano. Esses algoritmos também estão presentes no cotidiano de um usuário comum, seja em recomendações de filmes baseadas nos filmes assistidos em uma plataforma de *streaming*, ou mesmo recomendando sugestões de produtos que podem ser do interesse do consumidor, com base em compras e pesquisas anteriores em uma determinada loja online.

2.2 Análise de Sentimentos

A Análise de Sentimentos une conceitos de aprendizado de máquina com processamento textual e visa classificar um determinado dado de acordo com o sentimento da pessoa que o escreveu. É uma área em crescimento na computação e que possui grande margem para novas pesquisas, justamente devido ao fato de as redes sociais serem um importante componente na sociedade, que gera grandes quantidades de dados diariamente a respeito de diversos assuntos.

Esta é uma das áreas de pesquisa que recebe maior destaque em processamento de linguagem natural. Desde o início dos anos 2000, ela se tornou uma área muito ativa devido a explosão das redes sociais (GOMES, 2019a). O principal objetivo da Análise de Sentimentos é classificar instâncias de um determinado conjunto de dados em "positivo", "negativo" ou "neutro", dependendo do sentimento emitido naquela instância. Esse processo é feito por meio do processamento textual, retirando as palavras desnecessárias e separando as palavras mais importantes para a classificação. Como exemplos de frases que representam as classificações, temos: "Eu gosto muito de futebol.", "Aquele filme é péssimo." e "Hoje é quinta-feira.", sendo essas frases respectivamente correspondentes à "positivo", "negativo" e "neutro". Isso ocorre porque na primeira frase, "gosto muito" transmite o sentimento positivo, na segunda frase a palavra "péssimo" é muito importante para a classificação negativa desta instância, e por último, a última frase representa uma constatação, sem palavras que atribuem um sentimento a ela, o que a caracteriza como uma instância neutra.

Praticamente tudo o que acontece nas redes sociais, opiniões sobre determinados assuntos, seja esse assunto algo comercial ou mesmo político, pode gerar uma boa análise de sentimentos. O objetivo deste trabalho é analisar o tema da Reforma da Previdência graças a sua importância, e com a Análise de Sentimentos é possível extrair muitas informações relevantes à respeito da opinião do brasileiro sobre ele.

2.3 Classificadores

Esta seção apresenta os algoritmos que foram utilizados por Ricci (2020) na classificação dos tuítes da base e consequentemente utilizados neste trabalho, para que assim fosse possível obter uma comparação de resultados.

2.3.1 Naïve Bayes

Naïve Bayes é um algoritmo probabilístico baseado no *Teorema de Bayes*. Ele recebe o nome de *naive* (ingênuo) porque desconsidera a correlação entre as variáveis. É

um algoritmo que lida bem com a resolução de problemas textuais, além de ser aplicado em problemas de análise de sentimento em redes sociais (GOMES, 2019b).

O Teorema de Bayes é dado pela Equação 2.1 que serve de base para calcular a probabilidade de um evento A ocorrer dada a ocorrência de um evento B, ou seja, calcular a probabilidade condicional:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{2.1}$$

Onde:

- $P(A|B)$ é a probabilidade de A ocorrer dado que B já ocorreu;
- $P(B|A)$ é a probabilidade de B acontecer dado que A já ocorreu;
- $P(A)$ é a probabilidade de A ocorrer;
- $P(B)$ é a probabilidade de B ocorrer, tal que $P(B) \neq 0$.

2.3.2 Regressão Logística

Regressão Logística (do inglês, *Logistic Regression*) é um algoritmo que trabalha com os conceitos de estatística e probabilidade e lida com problemas de classificação. Esse algoritmo mede a relação entre a variável dependente categórica e uma ou mais variáveis independentes, estimando as probabilidades usando uma função logística (GATEFY, 2021).

O modelo da regressão logística é dado pela Equação 2.2:

$$P(Y) = \frac{1}{1 + e^{-f(x)}} \tag{2.2}$$

sendo $f(x)$:

$$B_0 + B_1X_1 + B_2X_2 + ... + B_kX_k \tag{2.3}$$

Onde:

- $\{X_1,\ X_2,\ ...\ ,\ X_k\}$ é o conjunto de variáveis independentes;
- Y é a variável dependente;
- $P(Y)$ é a probabilidade do evento ocorrer;
- $\{B_0,\ B_1,\ B_2,\ ...\ ,\ B_k\}$ são estimados de acordo com o conjunto de dados de treino.

Para casos de classificação, onde deve ser determinado se um evento ocorre ou não, caso $P(Y) \geq 0{,}5$ o evento ocorre, caso contrário não.

2.3.3 Florestas Aleatórias

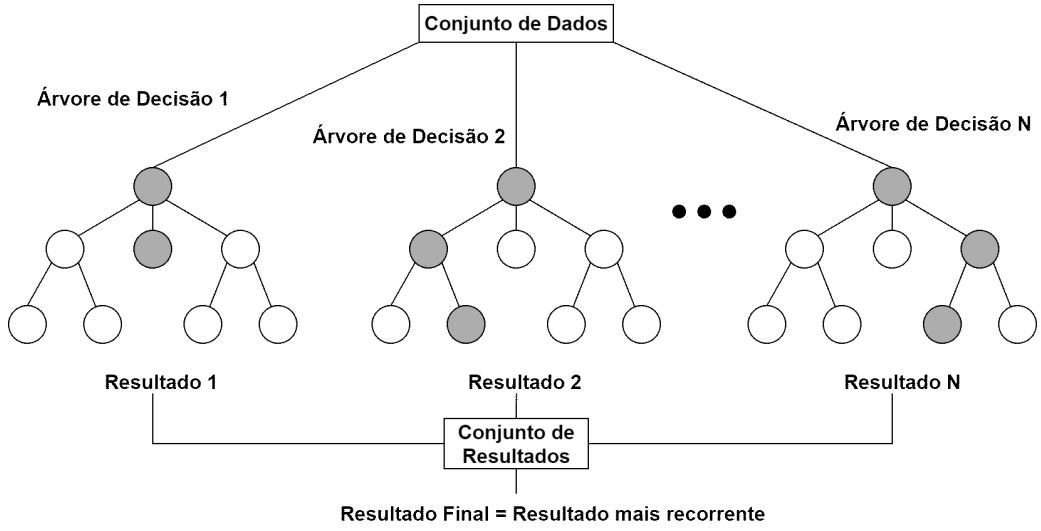
Floresta Aleatória (do inglês, *Random Forest*) é um algoritmo de classificação que utiliza um conjunto de árvores de decisão, cada uma dessas árvores dá uma classificação para o problema, a classificação mais recorrente torna-se a classificação do modelo (YIU, 2019).

2.3.3.1 Árvore de Decisão

Uma árvore de decisão é um classificador composto por um nó raiz, nós de decisão e nós folha. Este classificador é a base para uma floresta aleatória, pois é levada em conta a classificação individual de várias árvores de decisão para que o resultado da floresta aleatória seja obtido.

A Figura 1 apresenta a estrutura de uma Floresta Aleatória, que conta com a classificação de várias árvores de decisão para obter a classificação final.

Figura 1 – Estrutura de uma Floresta Aleatória.



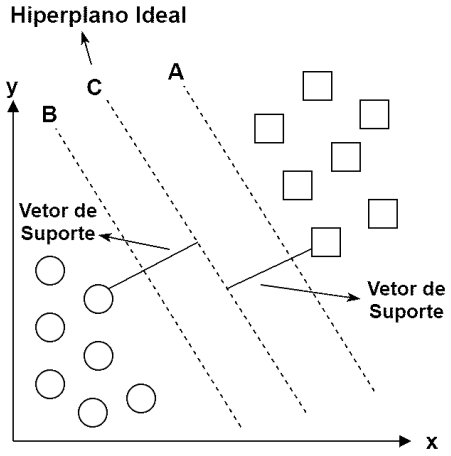
Fonte: O autor.

2.3.4 Máquinas de Vetores de Suporte

Uma Máquina de Vetores de suporte (do inglês, *Support Vector Machine*) é um método de aprendizado de máquina que classifica um problema, pode ser usada tanto para problemas de classificação como para problemas de regressão. O algoritmo cria um hiperplano que separa os dados em classes (OLIVEIRA JUNIOR, 2010).

Como exemplo de uma SVM, temos um conjunto de dados em que devemos classificar os quadrados e os círculos, para isto procura-se o hiperplano ideal que separa esses dados, como é mostrado na Figura 2.

Figura 2 – Separação de dados por meio de um hiperplano ótimo usando Máquina de Vetores de Suporte (SVM).



Fonte: O autor.

2.4 Ferramentas

Esta seção apresenta as ferramentas que possibilitaram a realização deste TCC.

2.4.1 pandas

pandas[1] é uma biblioteca *open source* escrita para a linguagem Python, ela é amplamente utilizada em trabalhos realizados na área de *Data Science*. Essa biblioteca possui estruturas flexíveis para o tratamento de dados, como por exemplo o *DataFrame*, que é uma estrutura que lida com conjuntos de dados e os trata em forma de tabela, o que facilita a visualização e manipulação de dados.

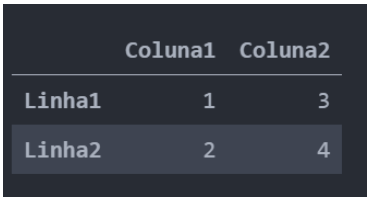
A biblioteca também oferece funções para manipulação de arquivos, como por exemplo a função *read_csv* que abre um arquivo csv e o armazena em um *DataFrame*. Também é possível manipular arquivos de diferentes extensões (json, xls, etc). Por sua ampla documentação e simplicidade para tratar problemas que contém uma grande quantidade de dados, *pandas* se torna uma excelente ferramenta para manipulação e análise de dados. A Figura 3 mostra como é um *DataFrame* criado com a biblioteca *pandas*.

2.4.2 Jupyter Notebook

Neste trabalho foi utilizado o ambiente *Jupyter Notebook*[2] com a linguagem Python. *Jupyter* foi escolhido graças à sua capacidade de rodar o código em partes, chamadas células. Neste ambiente é possível documentar o código visualmente de maneira simples, utilizando a linguagem de marcação *Markdown*. Como foram utilizados *DataFrames* e

[1] Disponível em: <https://pandas.pydata.org/>
[2] Disponível em: <https://jupyter.org/>

Figura 3 – Exemplo de um *DataFrame*.

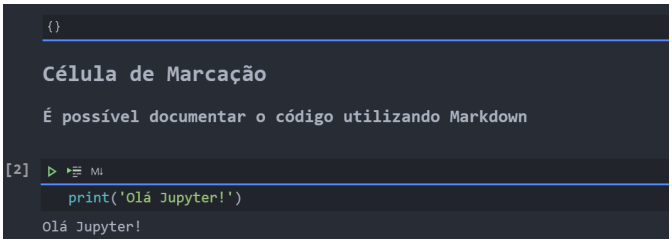


	Coluna1	Coluna2
Linha1	1	3
Linha2	2	4

Fonte: O autor.

foram gerados gráficos, com o *Jupyter* esses elementos aparecem no código, a medida que sua célula é executada. Isso facilita na visualização dos dados, o que é uma grande vantagem de se utilizar o *Jupyter* neste tipo de projeto. A aparência do *Jupyter* está diferente daquela encontrada no site oficial, pois o *Jupyter* foi utilizado diretamente no editor de código *VSCode*.

Figura 4 – Exemplo de células do *Jupyter Notebook*.



Fonte: O autor.

2.4.3 scikit-learn

scikit-learn[3] é uma biblioteca de aprendizado de máquina criada para a linguagem Python. Com ela é possível criar modelos de classificação, regressão e agrupamento de dados (*Clustering*). É uma das bibliotecas mais utilizadas quando se trata de *machine learning* na linguagem Python.

Como esta biblioteca possui a implementação de diversos algoritmos utilizados para aprendizado de máquina, foi possível utilizar funções prontas que implementam os classificadores, isso permitiu a utilização dos mesmos algoritmos que Ricci (2020) utilizou, o que torna a comparação de resultados justa, pois se fossem diferentes implementações, haveria diferença de desempenho entre esses classificadores. Para conseguir utilizar os modelos criados por ele, foi necessário usar a versão 0.21.3 da biblioteca, pois versões mais atualizadas perdem a compatibilidade com algumas funções necessárias para o funcionamento dos modelos.

[3] Disponível em: <https://scikit-learn.org/>

2.4.4 Botometer

Botometer[4] é um projeto criado em parceria pelo *Observatory on Social Media* (OSoMe) e pelo *Network Science Institute* (IUNI) da Universidade de Indiana. O principal objetivo do *Botometer* é a verificação de contas do Twitter, diferenciando contas autênticas de contas automatizadas.

Existe uma API do *Botometer* que dado um usuário, é retornado um objeto json que contém diversas informações sobre este usuário, tais como língua primária e classificações sobre ele. O objeto json contém classificações tanto considerando o inglês como considerando usuários que são de outros países (classificações *english* e *universal*). Com o *Botometer* é feita uma série de análises sobre a conta e verificada a probabilidade de aquele usuário ser ou não um *bot*. As principais variáveis presentes no objeto json são o *overall*, que representa uma classificação geral para o usuário e a *Complete Automation Probability* (CAP) , que corresponde à probabilidade de uma conta com score superior ao recebido em *overall*, ser automatizada. É possível visualizar os valores normais (de 0 a 5) e os valores brutos (de 0 a 1).

Como exemplo podemos supor que uma conta possui um *overall* de 4.8 de 5 e o CAP de 90%, isso significa que 90% das contas com esse score ou superior são reconhecidas como *bots* de acordo com os modelos do *Botometer*.

A Figura 5 apresenta um objeto json de uma conta analisada utilizando a API do *Botometer*.

Figura 5 – Objeto json retornado pela API do *Botometer* para uma determinada conta.


```
{
"cap":
    {"english": 0.7965971518887632, "universal": 0.7089830786080018},

"display_scores":
    {"english": {"astroturf": 1.2, "fake_follower": 1.8, "financial": 0.1, "other": 3.2, "overall": 3.2, "self_declared": 0.0, "spammer": 0.3},
     "universal": {"astroturf": 1.2, "fake_follower": 1.2, "financial": 0.0, "other": 2.0, "overall": 1.0, "self_declared": 0.1, "spammer": 0.4}
    },

"raw_scores":
    {"english": {"astroturf": 0.23, "fake_follower": 0.37, "financial": 0.02, "other": 0.63, "overall": 0.63, "self_declared": 0.01, "spammer": 0.06},
     "universal": {"astroturf": 0.23, "fake_follower": 0.25, "financial": 0.0, "other": 0.39, "overall": 0.2, "self_declared": 0.02, "spammer": 0.07}
    },

"user":
    {"majority_lang": "pt", "user_data": {"id_str": "229339679", "screen_name": "MiguelCampos12"}}


}
```


Fonte: O autor.

2.4.5 Ferramentas Auxiliares

Esta seção contém algumas ferramentas auxiliares utilizadas para o desenvolvimento do TCC, tais como ferramentas utilizadas para criar gráficos.

[4] Disponível em: <https://botometer.osome.iu.edu/>

***Natural Language Toolkit* (NLTK)**[5]: NLTK é um pacote Python utilizado para processamento de linguagem natural. Oferece muitas ferramentas úteis para códigos que utilizam processamento textual, neste trabalho foi utilizada a função *tokenize* para separar tuítes em tokens e posteriormente obter palavras de interesse para a análise e classificação individual de cada tuíte.

matplotlib[6]: Biblioteca Python utilizada para criação de gráficos. Utilizada neste trabalho para gerar os gráficos de resultados obtidos nas novas classificações da base.

numpy[7]: Biblioteca matemática de alto nível para a linguagem Python, muito utilizada para processamento de grandes matrizes, possui uma vasta coleção de funções matemáticas prontas para uso. Foi utilizada neste trabalho em conjunto com a biblioteca *pandas* para manipulação de uma grande quantidade de dados e para algumas manipulações matemáticas.

re[8]: Módulo *built-in* da linguagem Python utilizado para manipulação de expressões regulares. Utilizado neste trabalho para remoção de URL e menções na fase de pré-processamento da base de dados.

pickle[9]: Módulo *built-in* da linguagem Python utilizado para serialização de objetos. Foi utilizado neste trabalho para serializar o dicionário de *hashtags* criado na etapa de pré-processamento e disponibilizá-lo em forma de arquivo *pkl*. Também foi utilizado para acessar arquivos *pkl* disponibilizados por Ricci (2020) em seu trabalho.

2.5 Trabalhos Correlatos

Esta seção apresenta alguns trabalhos relacionados com este trabalho de conclusão de curso. A Análise de Sentimentos é uma área em crescimento na computação e a maioria das pesquisas que acontecem nessa área acontecem no exterior, porém ainda é possível ampliar esse número de pesquisas, graças à grande quantidade de dados que é gerada diariamente nas redes sociais, sendo assim, é possível gerar pesquisas tanto para o campo acadêmico como para o campo econômico, neste último caso analisando algo que seja do interesse de uma empresa. Os trabalhos a seguir foram relevantes e serviram de base para este TCC.

Uma visão geral sobre a Análise de Sentimentos no contexto do Twitter é fornecida por Gokulakrishnan et al. (2012), eles afirmam a importância da Mineração de Opinião dos usuários da plataforma, pois essa mineração pode ser benéfica tanto para empresas como para consumidores, no primeiro caso é possível analisar a opinião pública com relação

[5] Disponível em: <https://www.nltk.org/>
[6] Disponível em: <https://matplotlib.org/>
[7] Disponível em: <https://numpy.org/>
[8] Disponível em: <https://docs.python.org/3/library/re.html>
[9] Disponível em: <https://docs.python.org/3/library/pickle.html>

a esta empresa, já no segundo caso é falado que é possível um consumidor consultar a opinião de outras pessoas com relação a algum serviço ou produto que o interesse.

Todo o processo de Análise de Sentimentos é explicado por eles, desde a coleta dos tuítes, pré-processamento de dados, até a classificação dos tuítes em "positivo", "negativo" e "neutro/irrelevante" por meio de diferentes classificadores: Naïve Bayes, Florestas Aleatórias, Máquinas de Vetores de Suporte (SVM), etc.

Ricci (2020) propôs em sua monografia uma análise com relação à Reforma da Previdência que ocorreu em 2019 no Brasil. Ele construiu uma base de tuítes ao longo do período de interesse e analisou a opinião dos usuários do Twitter sobre este tema de grande relevância para o país. Para coletar os dados com tuítes de janeiro a novembro de 2019 e construir a base foi utilizada a ferramenta (*getoldtweets*)[10]. A coleta de dados, o pré-processamento e a classificação foram feitas utilizando a linguagem Python. Os tuítes foram pré-processados passando pelo processo de *tokenização*, que corresponde à separação das palavras e *emoticons* (também conhecidos como *emojis*) de cada tuíte, isso foi feito utilizando o *TweetTokenizer*[11], um módulo do *NLTK*. Após o processo de *tokenização* foram retiradas as palavras que não possuíam valor semântico para os classificadores, tais como: "da", "de", "na", etc. Como abreviações são comumente utilizadas no ambiente virtual, isso também foi tratado na etapa de pré-processamento, juntamente com *emoticons*, que foram separados em *emoticons_positivos*, *emoticons_negativos* e *emoticons_neutros*. Por fim, cada tuíte foi armazenado em um dicionário Python, removendo *hyperlinks*, *urls*, *hashtags* e menções, já que esses não possuem valor semântico para os classificadores.

A técnica de aprendizado por transferência (do inglês, *Transfer Learning*) foi utilizada por ele para classificar sua base, isto é, foram utilizadas bases já classificadas na língua portuguesa com sentimentos "positivos", "negativos" e "neutros" que serviram para a etapa de treinamento e validação do modelo (RICCI et al., 2021). Por fim, após a coleta e classificação dos tuítes da base, foi feita uma comparação dos resultados obtidos pelos diferentes classificadores utilizados pelo autor: Naïve Bayes, Florestas Aleatórias, Regressão Logística e Máquinas de Vetores de Suporte (SVM). Os resultados mostram que os classificadores obtiveram desempenho adequado, sendo Naïve Bayes o de mais baixa acurácia entre eles, com aproximadamente 84% (RICCI, 2020; RICCI et al., 2021).

Como as redes sociais compõe um ambiente de debate e mobilização política, em sua dissertação Costa (2020) investigou o impacto causado pelo Twitter no âmbito legislativo, no contexto da Reforma da Previdência, analisando se haveria mudança de comportamento dos senadores graças às estratégias adotadas por dois grupos, um favorável a reforma e o outro contrário a ela.

A coleta de dados foi feita utilizando a linguagem Python com a ferramenta *Twit-*

[10] Disponível em: <https://pypi.org/project/GetOldTweets3/>
[11] Documentação disponível em: <https://www.nltk.org/api/nltk.tokenize.html>

terSearch que consultava a API do Twitter. Também foi feito um mapeamento dos grupos que eram a favor e contra a proposta da Reforma de Previdência. Para a realização deste mapeamento foi utilizado o pacote *NetworkX*[12] do Python que se baseia no método de otimização de modularidade descrito por (BLONDEL et al., 2008), para assim promover a detecção de comunidades.

O autor analisou o conteúdo e o comportamento de cada grupo, estudando a abordagem utilizada por eles. Foram identificados os principais perfis publicadores de cada grupo, que em geral eram jornalistas, influenciadores, políticos e veículos de mídia. Além disso também foram analisadas datas importantes com o objetivo de verificar picos de tuítes em momentos cruciais, como nos dias de votação da Reforma. Ele também analisou as principais *hashtags* que foram levantadas no debate, já que com as *hashtags* é possível deixar um assunto em alta no Twitter. As *hashtags* mais utilizadas pelo grupo pró-Reforma foram: *#previdenciaoumorte*, *#reformadaprevidencia* e *#reformadaprevidenciaja*, já o grupo que era contra a proposta da Reforma da Previdência utilizou mais a *hashtag*: *#reformadaprevidencia*, que é uma *hashtag* "neutra" usada para se referir ao assunto, tanto que também foi uma das mais utilizadas pelo grupo pró-Reforma. Uma das *hashtags* utilizadas que demonstrava o descontentamento do grupo contra a Reforma foi a *#reformaperversa*, mas esta foi bem pouco utilizada se comparada com aquelas utilizadas pelo outro grupo. A conclusão foi de que os senadores votaram de acordo com a comunidade no qual se encaixavam e que não houve mudanças notáveis de comportamento dos senadores em decorrência das estratégias tomadas por cada grupo. Além disso foi observado que o grupo pró-Reforma publicava mais sobre o assunto, enquanto o grupo que era contra, focava em criticar o mérito da proposta, sem estratégia de *hashtags* relevante.

Em seu trabalho, Silva (2018) propôs uma análise do cenário político do Brasil utilizando Análise de Sentimentos. Ele utilizou técnicas de aprendizado de máquina em conjunto com técnicas de processamento de linguagem natural para explorar a rede de usuários do Twitter e analisar o sentimento destes usuários com relação ao cenário político do Brasil na época. Foram coletados tuítes de julho de 2016 a julho de 2017 sobre assuntos e pessoas relacionadas à política brasileira, foram eles: Reforma da Previdência, Lei da Terceirização, PEC do Teto de Gastos, além dos políticos Luiz Inácio Lula da Silva, Dilma Rousseff e Aécio Neves. Assim, foi montada uma base de 3645 tuítes e posteriormente houve a classificação manual de 2586 instâncias, com o objetivo de criar um classificador de tuítes de política.

Para facilitar este processo de classificação manual surgiu a ideia de desenvolver um classificador que permite que as pessoas colaborem na classificação, onde cada usuário informa a quantidade de tuítes que quer classificar e consegue classificá-los de

[12] Disponível em: <https://networkx.org/>

forma simples, este classificador recebeu o nome de CLAM[13] (Classificador Manual) e sua implementação está disponível no *Github* do autor.

O CLAM foi desenvolvido em Python com o framework *Django* e permite classificar tuítes e gerar um csv com os dados classificados. Também há um recurso muito interessante no CLAM que é a capacidade de marcar um tuíte como irônico, já que a ironia é uma dificuldade na área de Análise de Sentimentos, assim a pessoa que está classificando um determinado tuíte de política pode marcar a presença de ironia nele, além de classificá-lo como "positivo", "negativo" ou "neutro".

Além da construção do CLAM, foram implementados os algoritmos SVM e Regressão Logística, tanto versões para o caso binário como para casos com várias classes e comparadas com as implementações existentes no *scikit-learn*, os modelos apresentaram acurácia de 70,4% para o caso binário e 68,3% para o caso de três classes, acurácias próximas daquelas obtidas pela implementação do *scikit-learn*. No fim, o autor concluiu o trabalho com diferentes implementações de classificadores que obtiveram um bom resultado na classificação de tuítes de política, algo relevante, pois existe muita ironia em tuítes sobre este assunto, o que acaba dificultando o processo de Análise de Sentimentos.

Por fim, França e Oliveira (2014) propuseram a análise de tuítes relacionados aos protestos que ocorreram no Brasil em 2013, no período de junho a agosto. Nesta data o Brasil passou por diversos protestos e a opinião de muitos usuários foi colocada nas redes sociais na época, incluindo o Twitter. Foi montada uma base com mais de 300 mil tuítes e o objetivo dos autores era analisar a polaridade das opiniões a respeito do tema, investigando a quantidade de mensagens de apoio ou repúdio aos protestos.

A base de tuítes foi montada a partir da busca de *hashtags* referentes ao tema, como por exemplo *#vemprarua* e *#acordabrasil*. Essas *hashtags* de interesse foram utilizadas na consulta realizada pela API do Twitter. Em posse dos dados, foi realizado o pré-processamento textual, removendo *links*, menções a outros perfis, *stopwords* (palavras sem valor para a análise do classificador), etc. Assim, foi mantido somente o conteúdo necessário para a realização da classificação. O algoritmo escolhido para classificar os tuítes foi o de Naïve Bayes, pois os autores verificaram a satisfatoriedade dos resultados de trabalhos encontrados na literatura que utilizaram este algoritmo. Os textos foram analisados como *bag-of-words* assim como no trabalho de Ricci (2020), isto é, as posições exatas das palavras são ignoradas pelo classificador, além de que também foi utilizado o pacote *NLTK* com a linguagem Python para lidar com o processamento de linguagem natural.

Para treinar o classificador foi utilizada uma base rotulada por 3 humanos, construída a partir de amostras de mensagens da base coletada, essas amostras foram classifica-

[13] Disponível em: <https://github.com/romaolucas/manual-classifier-helper>

das em positivas (apoio aos protestos) e negativas (repúdio aos protestos), não foi utilizada classificação neutra para as instâncias da base. Ao final foram criados dois conjuntos de 100 instâncias cada, um com instâncias positivas e outro com instâncias negativas. Uma parte de cada conjunto (70%) foi utilizada para treinamento e 30% para testes.

Após o processo de treino e teste do algoritmo foi feita a análise da polaridade dos tuítes relacionados aos protestos, onde se obteve o resultado de que houve maior incidência de tuítes que apoiavam os protestos considerando todo o período analisado, apesar de que, no final do período analisado, o número de tuítes que foram escritos em repúdio aos protestos foi maior do que os tuítes que eram a favor, provocando uma leve alteração de polaridade. Também foi feita pelos autores uma separação de tuítes por regiões, apesar de que a maioria dos tuítes da base (quase 300 mil) não possuíam localização, já que para coletar a localização, o indivíduo que postou o tuíte deveria estar com o GPS do dispositivo ativado e também deveria permitir o Twitter coletar esta informação. Os tuítes que possuíam localização foram divididos em "Estados", "Internacional" e "Outros", este último contém tuítes publicados no Brasil, mas nenhum estado ou cidade estava presente nas informações desses tuítes, impossibilitando de agrupá-los em outro grupo. A região que menos publicou tuítes sobre os protestos foi a região Norte, já a região que mais publicou sobre o tema foi a região Sudeste.

Os resultados finais obtidos eram esperados (maior número de tuítes que expressavam algum tipo de apoio aos protestos), isso era notável no comportamento da população brasileira na época que, em grande parte, se mobilizou positivamente com relação ao tema. Além disso, na hora de montar a base, os autores utilizaram *hashtags* relacionadas aos protestos e foi constatado que não foram utilizadas *hashtags* de repúdio a eles (nenhum levantamento foi feito para verificar se essas *hashtags* existiram ou não).

3 Desenvolvimento e Resultados

Neste capítulo, são apresentados os resultados deste TCC a partir de novas classificações, bem como decisões tomadas no decorrer do desenvolvimento deste trabalho. Conforme já mencionado, todo o desenvolvimento foi feito utilizando a linguagem Python. O código-fonte está disponível de maneira aberta no repositório[14] do autor. É importante enfatizar que este trabalho considerou apenas a base de tuítes coletada por Ricci (2020) em seu TCC e não foram feitas novas coletas de tuítes sobre o assunto. Portanto, toda análise descrita neste capítulo se baseia nesses dados.

3.1 Identificação de Contas Automatizadas

Esta seção descreve algumas escolhas de projeto tomadas com relação à classificação dos usuários da base. O processo de coleta de dados dos usuários e a posterior identificação dos *bots* também é apresentado.

3.1.1 Amostra Inicial

Inicialmente, foi selecionada uma amostra aleatória contendo mil tuítes da base de dados original (RICCI, 2020; RICCI et al., 2021). Essa amostra continha apenas atributos relevantes para a realização de algumas análises iniciais, como o identificador do tuíte, o nome do usuário que publicou a mensagem, o texto original do tuíte, além das classificações correspondentes de cada algoritmo para aquele tuíte. Graças ao tamanho reduzido da amostra (se comparado com o tamanho da base original) foi possível avaliar diferentes alternativas de classificação dos usuários, para que os mesmos fossem diferenciados entre *bots* e usuários comuns. Além disso, também foram realizadas análises que posteriormente foram replicadas na base original, como, por exemplo, a verificação de quantos usuários distintos foram responsáveis pelos mil tuítes. Foi observado que esses mil tuítes foram gerados por 974 usuários distintos e, a partir dessa informação, foram utilizadas três ferramentas diferentes para classificação desses usuários: *Botometer*, *PegaBot*[15] e *Bot Sentinel*[16].

A primeira análise foi realizada com a ferramenta *Botometer*. Com ela, foram experimentados diferentes valores de CAP, variável que serve como um *threshold* para separar usuários em *bots* e usuários comuns, sugerida na documentação da ferramenta

[14] Código-fonte utilizado no desenvolvimento e na obtenção de resultados deste trabalho está disponível em: <https://github.com/matheushfp/opinion-mining>
[15] Disponível em: <https://pegabot.com.br/>
[16] Disponível em: <https://botsentinel.com/>

para análises do tipo "é um *bot*", "não é um *bot*". A tarefa de classificar uma conta em automatizada ou não automatizada é uma tarefa complexa, uma vez que vários fatores devem ser levados em consideração, como: spam de mensagens e links, nome de usuário, foto de perfil, entre outros. A variável CAP é sugerida pois leva em conta toda a análise feita para aquele usuário e atribui a ele um valor entre 0 e 1. Outra sugestão encontrada na documentação da ferramenta é utilizar 95% como *threshold*, isto é, usuários que recebem um valor de 0,95 ou superior para essa variável são considerados *bots*. Porém, dependendo da aplicação, o valor de CAP pode ser mais flexível. Com isso em mente, os 974 usuários foram classificados utilizando diferentes valores de *threshold*: desde 90% até 95%. Observa-se que, com CAP mais próximo a 90%, mais usuários são considerados *bots*, mas a taxa de erro na classificação tende a ser maior.

Considerando *bots* os usuários com CAP maior que 95%, dos 974 usuários da amostra apenas 10 foram considerados *bots*; já com CAP 90%, esse número subiu para 36 *bots*. Em seguida esses 36 usuários foram acessados e classificados manualmente em *bots* e contas comuns. Como os dados foram coletados ao longo do ano de 2019, algumas das contas que publicaram tuítes na época foram excluídas, banidas ou suspensas. Dentre essas 36 contas, ao acessar o perfil individual de cada usuário no Twitter, é possível observar que três contas possuem criação após o período de coleta para montar a base, isto é, três contas foram excluídas e posteriormente criadas novamente utilizando os mesmos nomes de usuário. Com esse detalhe em mente, para analisar a precisão de classificação da ferramenta esses três usuários foram desconsiderados, pois tratam-se de novos usuários e não os mesmos usuários que publicaram os tuítes contidos na base.

Para as 10 contas que receberam CAP acima de 95%, duas foram desconsideradas pois se enquadram no caso de contas criadas após 2019. Assim, oito contas foram analisadas e todas foram consideradas *bots* na análise manual. Já considerando o CAP 90%, das 33 contas, 27 foram consideradas *bots* na análise manual. Mesmo que a precisão foi de 100% considerando o CAP 0,95, muitos *bots* foram desconsiderados, pois passando a usar 0,90 o número aumentou de oito *bots* para 27 *bots* na análise manual.

Os resultados com CAP 0,90 e 0,91 foram parecidos, com diferença que para o CAP 0,90 a ferramenta Botometer considerou um *bot* a mais. Analisando os resultados, optou-se por uma flexibilização maior, pois com valores de CAP mais altos muitos *bots* eram desconsiderados. Assim, empregou-se o valor de CAP 90% para a classificação da base completa. Na Tabela 1 é possível observar os resultados dessa análise inicial.

Após a classificação de *bots* na amostra inicial, foram utilizadas outras ferramentas citadas anteriormente para classificação dos *bots*. Porém, nem o *PegaBot* ou o *Bot Sentinel* tinham APIs no momento de realização deste trabalho; assim a classificação teria que ser feita utilizando o próprio site das ferramentas e, como a base original, possui muitos usuários, isso se tornaria inviável de ser feito em tempo hábil. Com essa limitação e com

Tabela 1 – Análise com a ferramenta Botometer e análise manual à respeito de *bots* na amostra.

CAP	**Botometer**	**Contas criadas após 2019**	**Classificação manual**
90%	36	3	27 *bots* em 33 contas
91%	35	3	27 *bots* em 32 contas
92%	21	2	18 *bots* em 19 contas
93%	15	2	13 *bots* em 13 contas
94%	10	2	8 *bots* em 8 contas
95%	10	2	8 *bots* em 8 contas

Fonte: O autor.

um conhecimento prévio maior na ferramenta *Botometer*, optou-se por utilizá-la para classificação da base original neste trabalho.

3.1.2 Coleta de Dados

Tomadas as escolhas citadas anteriormente, havia a necessidade de classificar os usuários da base original em *bots* e usuários comuns. Para isso ser feito, foi utilizada a API do *Botometer* para coletar dados relevantes sobre cada usuário único na base. A resposta da API se dá na forma de um arquivo json contendo informações sobre o usuário solicitado, como mostrado anteriormente no capítulo Ferramentas, seção *Botometer*. A base original possui 980.577 tuítes produzidos por 237.894 usuários. Foi feito um *script* em Python que se comunica com a API para a coletar e armazenar dos dados de cada usuário presente na base. Esta coleta de dados foi iniciada no dia 27/05/2021 e finalizada no dia 21/06/2021, foi necessário praticamente um mês para coletar os dados de todos os usuários da base, pois a API tinha um limite de requisições diárias.

Ao longo do período de coleta, um total de 885 chamadas para a API retornaram erro. No *script* feito para coleta dos dados, quando a API retornava um erro, o nome do usuário correspondente era armazenado em uma lista. Posteriormente esses usuários foram submetidos novamente para a API e seus resultados obtidos com sucesso. É importante ressaltar que alguns usuários da base, atualmente, têm em suas contas uma função ativada de “proteção de tuítes”, sendo que somente os seguidores dessas contas têm acesso aos tuítes e detalhes delas. Para essas contas que hoje tem tuítes inacessíveis, a API retorna um erro “Not authorized” no json e esses usuários posteriormente recebem valor de CAP equivalente a NaN (tipo de dado da biblioteca *Numpy* que significa *Not a Number*). Com essa representação, é possível diferenciá-los dos outros usuários da base.

3.1.3 Classificação de Usuários

Uma vez que os dados de cada usuário foram coletados e armazenados, torna-se possível realizar a classificação de todas as contas únicas da base, podendo, assim, diferenciá-las em: contas automatizadas (*bots*) e contas não automatizadas. As contas inacessíveis são descartadas na análise de *bots* da base, pois não é possível classificar esses usuários.

Para realizar a classificação, foi utilizado o *threshold* mostrado anteriormente, isto é, usuários com CAP maior ou igual a 0,90 foram considerados *bots*, os que receberam valor de CAP diferente de NaN e menor que 0,90 foram considerados usuários comuns e os demais, inacessíveis. O resultado nesta classificação foi de que das 237.894 contas únicas que produziram tuítes para a base, 8.180 eram automatizadas e foram responsáveis pela produção de 40.151 tuítes dos 980.577 pertencentes na base. Dentre todas as contas presentes na base, 49.917 não permitiram o acesso da API, pois usam a função de "proteção de tuítes". Na Tabela 2 é possível observar esses resultados.

Tabela 2 – Classificação de usuários da base.

Tipo de conta	**Quantidade**	**Tuítes**
Bot	8.180	40.151
Inacessível	49.917	184.054
Comum	179.797	756.372

Fonte: O autor.

3.2 Pré-processamento dos Dados

Com o objetivo de classificar novamente a base de Ricci (2020) e obter novos resultados, torna-se necessário fazer um novo pré-processamento dos dados. Nesse contexto, além das etapas aplicadas no trabalho original (conversão do texto dos tuítes em minúsculo, remoção de URLs e menções, *tokenização* e remoção de *stopwords*), houve o tratamento de *hashtags*, de forma que ao final do pré-processamento fossem geradas duas bases, uma exatamente igual a base pré-processada de Ricci (2020) (base sem *hashtags*) e outra com *hashtags* para a realização de uma nova classificação. Todas as etapas de pré-processamento estão resumidas abaixo e, em seguida, a Figura 6 mostra alguns tuítes antes e depois da realização de todo o processo.

Transformação de texto do tuíte em minúsculo. Esse processo é realizado para padronizar o texto; assim palavras escritas totalmente em maiúsculo ou com inicial maiúscula podem agora ser comparadas com palavras escritas de maneira totalmente minúscula. Esse procedimento é simples e a própria linguagem de programação já tem um método *built-in* que o realiza, este método é o `lower()`.

Remoção de links, menções e *hashtags*. Esses procedimentos foram feitos com o auxílio de expressões regulares. Em Python foi utilizado o módulo *re* para manipulação de expressões regulares e consequentemente a remoção dos dados desejados. Para cada tipo de remoção foi criada uma função, e no caso de remoção de *hashtags*, essa função é utilizada para montar a base sem *hashtags*, mas para a base que leva em conta as *hashtags*, posteriormente são utilizadas outras funções para o tratamento delas.

Tokenização. Nesta etapa os tuítes são transformados em listas de palavras, dessa maneira torna-se mais fácil a remoção de palavras que não possuem valor para a análise final, pontuação, etc. Para isso foi utilizado o *TweetTokenizer* presente no módulo *NLTK*, que gera um objeto a partir de cada tuíte.

Padronização de abreviações e tratamento de *emoticons*. Para esta etapa e para a próxima, foram utilizados arquivos disponibilizados por Ricci (2020) em seu repositório[17]. Com relação à padronização de abreviações, este procedimento é feito com o auxílio de um dicionário, onde a chave é uma abreviação e o valor é a palavra escrita de maneira composta, este dicionário é utilizado para substituição de palavras abreviadas, tais como "vc" que é substituída por "você".

Já para o tratamento de *emoticons*, é utilizado um dicionário que mapeia um sentimento à um *emoticon* específico. Esse dicionário possui como chaves: "emoticon_positivo", "emoticon_negativo" e "emoticon_neutro". O valor de cada uma dessas chaves é uma lista com diversos *emoticons* que melhor se enquadram naquela classificação. Com o auxílio deste dicionário, os *emoticons* são substituídos nos tuítes pelas chaves equivalentes, assim eles são rotulados.

Remoção de *stopwords* e transformação de tuítes *tokenizados* em frases. Nesta etapa foi utilizada uma lista de *stopwords*, isto é, palavras que não possuem valor para a análise de sentimentos. Esta lista foi formada utilizando as *stopwords* do *NLTK* para a língua portuguesa, em conjunto com todos os tipos de pontuação (a lista de pontuação é acessada por meio da biblioteca *string*). Ela contém palavras como "de", "da", além de pontos como "!", "." e os termos "reforma" e "previdência". Todas estas palavras foram removidas dos tuítes. Após a remoção de *stopwords*, foi realizado o processo inverso ao da tokenização, onde os tokens são transformados novamente em frases. Isso foi feito com uma função criada pelo autor chamada *untokenize*, esta função utiliza o método *built-in* `join()` para transformar os tokens em tuítes novamente.

Padronização de *hashtags*. Esta e a próxima etapa são feitas para montar a base pré-processada que possui *hashtags*. Inicialmente são coletadas todas as *hashtags*

[17] Disponível em: <https://github.com/RafaelDRicci/PythonSentimentAnalysis>

presentes na base. Assim que coletadas, é criado um dicionário onde a chave é a *hashtag* da maneira original que foi escrita no tuíte e o valor é esta *hashtag* após um processo de padronização, onde a *hashtag* é colocada em texto minúsculo e os acentos são removidos. Este dicionário é utilizado para substituir as *hashtags* dos tuítes por suas versões padronizadas. Esse processo é feito pois existem casos de *hashtags* como: #reformadaprevidencia, #ReformaDaPrevidencia e #ReformaDaPrevidência, que basicamente são a mesma coisa, mas sem esta padronização previamente descrita, elas são consideradas *hashtags* diferentes e isso prejudicaria a posterior contagem para saber quais foram as *hashtags* mais utilizadas na base, já que no exemplo citado anteriormente, uma *hashtag* seria considerada como três, onde cada uma teria sua contagem de aparições em tuítes da base.

Tratamento de *hashtags*. Para esta etapa houve um levantamento de quais foram as 150 *hashtags* mais utilizadas na base e, posteriormente, uma classificação manual das que eram relevantes para o tema da reforma da previdência e poderiam ser classificadas em *hashtags* positivas e negativas. Dentre essas 150 *hashtags*, algumas foram ignoradas, como por exemplo: #lulalivre, #bolsonaro e #globolixo.

Foram consideradas apenas *hashtags* que apoiavam a reforma ou que eram contra ela. Como exemplo de *hashtags* positivas temos: #reformadaprevidenciaja e #euapoionovaprevidencia. Já para negativas temos: #reformadaprevidencianao e #reformanao. Após a classificação manual destas *hashtags* consideradas mais relevantes para a análise de sentimentos, foi criado um dicionário com chaves: "hashtag_positiva" e "hashtag_negativa", e o valor para cada chave era uma lista com *hashtags* que melhor se encaixavam naquela classificação. Não foram inseridas *hashtags* neutras no dicionário, pois com esse tratamento pretendia-se analisar o comportamento dos modelos, verificando se alguns tuítes que eram considerados neutros, quando eram classificados sem considerar as *hashtags*, passariam a ser positivos ou negativos. Este dicionário foi disponibilizado no repositório do autor via arquivo *pkl* por meio do uso da biblioteca *Pickle*. Depois da criação deste dicionário, as *hashtags* que receberam classificação positiva e negativa foram substituídas no tuíte original por "hashtag_positiva" e "hashtag_negativa". Por fim a base processada foi criada para novas classificações.

3.3 Novas Classificações e Resultados

Nesta seção serão apresentadas novas classificações realizadas, além de gráficos e tabelas com resultados e análises gerais sobre cada classificação. Como dito anteriormente, foram utilizados os modelos criados por Ricci (2020) para reclassificar a base após o novo pré-processamento. Esses modelos foram disponibilizados por ele em seu repositório.

Figura 6 – Cinco tuítes antes e depois do pré-processamento.

(a) Antes do pré-processamento

```
Achou um absurdo!! Apoie a Reforma da Previdência, ela acaba com esse tipo de coisa #reformadaprevidencia #ReformadaPrevidenciaJa
@geraldoalckmin Reforma da Previdência Não! #ReformaDaPrevidenciaNao
Não a reforma da previdência! #NaoAReformaDaPrevidencia #ReformaDaPrevidenciaNAO #ReformaPeNaCova
Boa noite Presidente eu apoio a reforma da previdência 👍😂👏👏😂👍
Reforma da previdência já! #ReformaDaPrevidenciaJa #PrevidenciaOuMorte
```

(b) Depois do pré-processamento

```
achou absurdo apoie acaba tipo coisa #reformadaprevidencia hashtag_positiva
hashtag_negativa
hashtag_negativa hashtag_negativa hashtag_negativa
boa noite presidente apoio emoticon_positivo emoticon_positivo emoticon_positivo emoticon_positivo emoticon_positivo emoticon_positivo
hashtag_positiva hashtag_positiva
```

Fonte: O autor.

3.3.1 Classificação dos tuítes de *bots*

A primeira classificação foi feita com o intuito de ver qual seria a classificação dos tuítes de *bots* para o criador da base. Assim, foi utilizada a primeira base pré-processada (na qual não há tratamento de *hashtags*), para atingir exatamente os mesmos resultados dele, desta forma é possível saber qual seria a classificação de cada um de seus modelos, caso houvesse uma separação entre contas automatizadas e contas comuns na época em que seu trabalho foi realizado. Esta replicação da base pré-processada original também foi usada futuramente para comparação de resultados, onde foi possível comparar os resultados da nova classificação (pós tratamento de *hashtags*) com o modelo inicial que foi classificado sem nenhuma *hashtag*.

Submetendo os 40.151 tuítes produzidos por contas automatizadas da base para todos os modelos, foi atingido o resultado observado na Tabela 3. A partir dos resultados dessa classificação de tuítes de contas automatizadas, foi criado o gráfico em escala logarítmica apresentado na Figura 7.

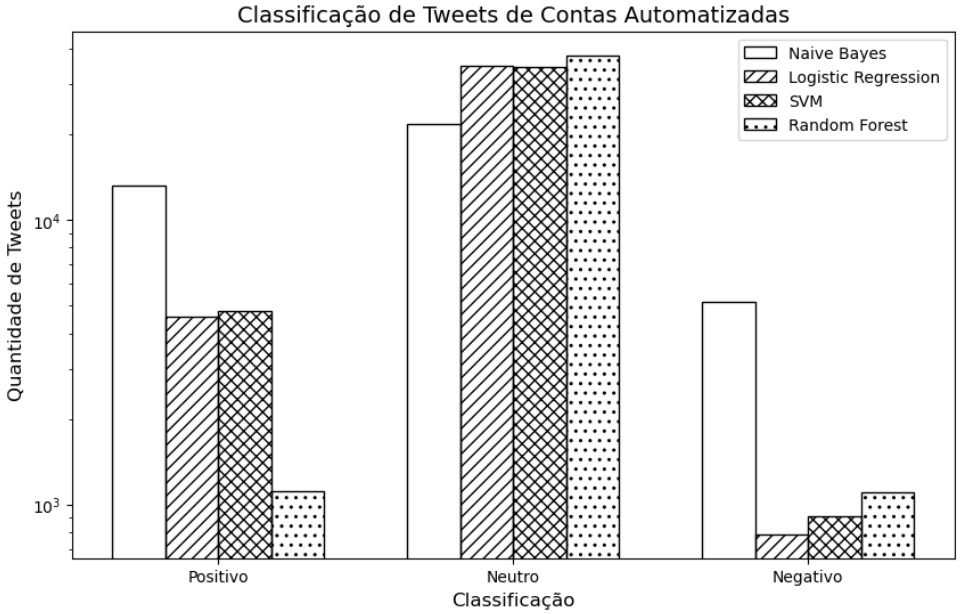
Tabela 3 – Classificação de tuítes de *bots*.

Modelo	Positivo	Negativo	Neutro
Naïve Bayes	13.199	5.138	21.814
Regressão Logística	4.577	788	34.786
Floresta Aleatória	1.115	1.102	37.934
SVM	4.814	911	34.426

Fonte: O autor.

Com os resultados obtidos nesta classificação é possível observar que dentre os 40.151 tuítes, a maioria recebeu a classificação neutra por todos os algoritmos. Isto pode se dar graças ao fato de que contas com qualquer tipo de automatização são consideradas

Figura 7 – Classificação dos tuítes de contas automatizadas utilizando quatro modelos distintos. O gráfico está em escala logarítmica.



Fonte: O autor.

bots e, não necessariamente, contas que possuam algum comportamento tóxico. Tendo isso em mente, vários veículos de imprensa utilizam mecanismos de automatização para compartilhar notícias em suas redes sociais, assim, logo que um novo post (como por exemplo uma notícia sobre a reforma) é publicado no site, a conta do Twitter daquele veículo de imprensa posta um link para divulgar esta matéria. Foi observado pelo autor que várias contas que postavam somente links, como contas de veículos de imprensa, muitas vezes acabaram recebendo a classificação de conta automatizada.

3.3.2 Classificação dos tuítes de contas inacessíveis

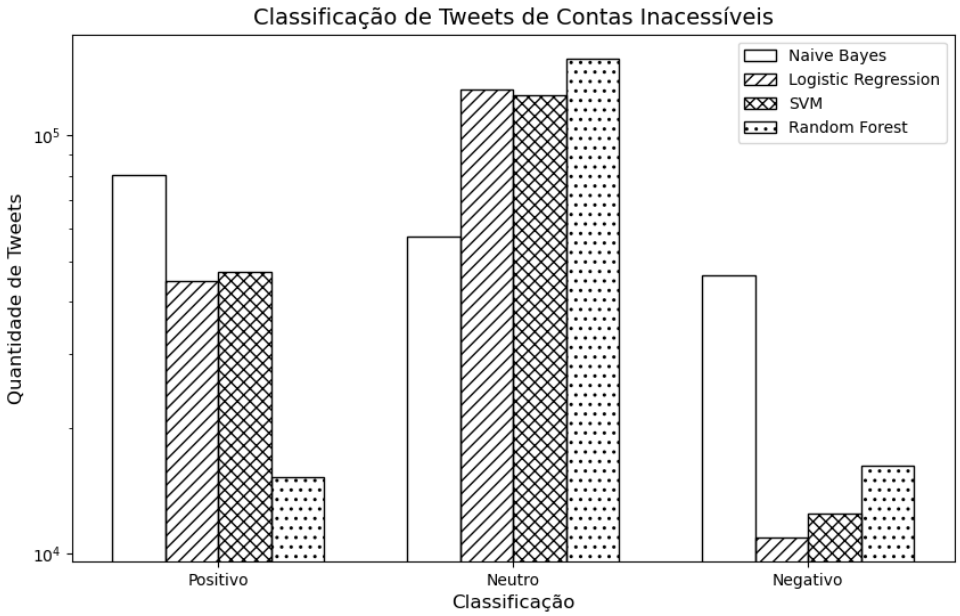
Neste trabalho também houve uma classificação das contas que eram inacessíveis para o *Botometer* e, consequentemente, não se enquadravam na classificação de *bots* nem de usuários comuns. A Tabela 4 mostra os resultados dessa classificação. Em seguida também é possível visualizar os resultados por meio do gráfico da Figura 8.

Tabela 4 – Classificação de contas inacessíveis.

Modelo	Positivo	Negativo	Neutro
Naïve Bayes	80.591	46.275	57.188
Regressão Logística	44.818	10.962	128.274
Floresta Aleatória	15.238	16.271	152.545
SVM	47.222	12.479	124.353

Fonte: O autor.

Figura 8 – Classificação dos tuítes de contas inacessíveis. O gráfico está em escala logarítmica.



Fonte: O autor.

É possível observar que para o caso da classificação usando o modelo de Naïve Bayes há maior equilíbrio entre as três diferentes classes, já para os demais modelos há uma maior disparidade, com a maioria de tuítes classificados como neutros. Para praticamente todos os modelos, a classe que menos recebeu tuítes foi a negativa, exceto para o modelo que utiliza Floresta Aleatória, onde a classe que recebeu menos tuítes foi a classe positiva.

3.3.3 Nova classificação da base original

Após a realização do novo pré-processamento, a base original foi submetida a uma nova etapa de classificação, desta vez contando com o tratamento de *hashtags*. A nova classificação da base permitiu uma comparação entre os antigos resultados, de quando a base foi classificada desconsiderando as *hashtags* dos tuítes, com os novos resultados, agora contendo elas. Na Tabela 5, é possível visualizar a comparação de resultados, onde na esquerda está a tabela com a classificação original e na direita a tabela com a classificação pós tratamento de *hashtags*. Abaixo dessas duas, encontra-se uma tabela que mostra o aumento ou diminuição de cada classe, comparando os antigos resultados com os novos.

Com os resultados apresentados, é possível observar que, para a classificação original, a maioria dos tuítes presentes na base foram rotulados como positivos utilizando o modelo de Naïve Bayes, já para os demais modelos, a maioria dos tuítes recebeu classificação neutra. Adicionando o tratamento de *hashtags* foi possível diminuir os tuítes neutros

Tabela 5 – Comparação de diferentes classificações da base.

(a) Classificação sem hashtags

Modelo	Positivo	Negativo	Neutro
Naïve Bayes	409.855	212.387	358.335
Regressão Logística	209.452	46.425	724.700
Floresta Aleatória	69.066	68.789	842.722
SVM	222.860	53.462	704.255

(b) Classificação com hashtags

Modelo	Positivo	Negativo	Neutro
Naïve Bayes	409.592	212.545	358.440
Regressão Logística	209.871	46.562	724.144
Floresta Aleatória	69.122	68.874	842.581
SVM	223.430	53.504	703.643

(c) Aumento ou diminuição relativa pós tratamento de hashtags

Modelo	Positivo	Negativo	Neutro
Naïve Bayes	- 0,06%	+ 0,07%	+ 0,03%
Regressão Logística	+ 0,2%	+ 0,3%	- 0,08%
Floresta Aleatória	+ 0,08%	+ 0,1%	- 0,02%
SVM	+ 0,3%	+ 0,08%	- 0,09%

Fonte: O autor.

de praticamente todos os modelos, isso só não ocorreu para Naïve Bayes que, como dito anteriormente, tinha classificação positiva como maioria.

Para os casos onde houve diminuição de tuítes que receberam classificação neutra quando adicionado o tratamento de *hashtags* observa-se que para o classificador que utiliza Regressão Logística, 556 tuítes que eram neutros passaram a receber classificação diferente. Desses, 419 passaram a receber a classificação positiva e 137 a negativa. No caso da classificação utilizando Floresta Aleatória, apenas 141 tuítes neutros receberam classificação diferente após o tratamento de *hashtags*, dentre eles 56 passaram a receber classificação positiva e 85 classificação negativa. Já para o SVM houve o maior número de tuítes que eram neutros e passaram a receber outra classificação. Foram 612 tuítes com nova classificação, dentre eles, 570 novos positivos e 42 negativos.

Após a nova classificação, foram retiradas as contas que eram consideradas *bots* da base e foi observado que apesar de a maioria das contas automatizadas terem classificação neutra, após a remoção delas da base, a mesma proporcionalidade encontrada antes foi mantida, isto é, Naïve Bayes continuou com maioria de tuítes classificados como positivos e os demais algoritmos com maioria de tuítes neutros. A classificação da base sem *bots* pode ser vista na Tabela 6.

Tabela 6 – Classificação da base sem *bots*.

Modelo	**Positivo**	**Negativo**	**Neutro**
Naïve Bayes	396.418	207.399	336.609
Regressão Logística	205.293	45.770	689.363
Floresta Aleatória	68.007	67.772	804.647
SVM	218.606	52.593	669.227

Fonte: O autor.

3.3.4 Análise de Contas Individuais

Por fim, foi realizada uma análise dentre os mil tuítes que mais receberam retuítes e dos mil tuítes que mais receberam a marca de "favorito" da base. Nessa análise foi possível observar que o tuíte mais retuitado da base, na época da coleta, contava com 29.367 retuítes e 68.769 "likes"; isso fez com que esse tuíte não fosse apenas o mais retuitado da base, mas o segundo mais favoritado. Esse tuíte em questão foi produzido por uma conta de um cidadão anônimo, ou seja, não foi escrito por uma pessoa pública.

Dentre os 10 tuítes que mais receberam retuítes, três foram produzidos por figuras públicas: o então presidente da república Jair Bolsonaro, a então deputada estadual do Estado de São Paulo, Janaina Paschoal e o jornalista Alexandre Garcia. Os 10 tuítes que mais receberam retuítes foram produzidos por 10 contas distintas, sendo que no momento da escrita desta monografia, três foram excluídas e uma suspensa do Twitter.

Já analisando os 10 tuítes que mais foram favoritados na base, oito foram produzidos por figuras públicas. Dentre esses oito tuítes, três foram publicados por Jair Bolsonaro, dois por Janaina Paschoal, um por Alexandre Garcia, um pelo deputado federal Eduardo Bolsonaro e outro pela deputada federal Tabata Amaral. As figuras 9 e 10 mostram informações dos 10 tuítes mais retuitados e mais favoritados da base, respectivamente.

Figura 9 – Informações dos dez tuítes mais retuitados.

username	retweets	favorites	text
jozecoms	29367	68769	morrer de estudar pra passar em uma federal (q...
vitornasc	16459	39671	TRIGÉSIMO OITAVO PRESIDENTE DA REPÚBLICA FEDER...
alexandregarcia	13882	65581	Ouvi palestra de Paulo Guedes. Ele é genial. S...
okmatheuso	11621	39931	Se a reforma da previdência for aprovada cenas...
pdrmuriel	11151	30521	- extermínio e violência contra a população ne...
jairbolsonaro	10672	78530	O auxílio-reclusão ultrapassa o valor do salár...
JanainaDoBrasil	10533	52941	Quando o Presidente da Câmara ameaça deixar a ...
mah13MC	10068	27017	A Marinha não quer entrar na Reforma da Previd...
newbips	9258	28129	holy shit, is this a motherfucking reforma da ...
AFS_Andrada	9013	30236	Reforma da previdência de Jair Bolsonaro e Pau...

Fonte: O autor.

Figura 10 – Informações dos dez tuítes mais favoritados.

username	retweets	favorites	text
jairbolsonaro	10672	78530	O auxílio-reclusão ultrapassa o valor do salár...
jozecoms	29367	68769	morrer de estudar pra passar em uma federal (q...
alexandregarcia	13882	65581	Ouvi palestra de Paulo Guedes. Ele é genial. S...
JanainaDoBrasil	10533	52941	Quando o Presidente da Câmara ameaça deixar a ...
jairbolsonaro	6547	48734	Os avanços que o Brasil precisa dependem da ap...
tabataamaralsp	4907	45477	Meu voto pela Reforma da Previdência não foi v...
jairbolsonaro	7203	41756	Nenhuma outra proposta de reforma foi tão firm...
JanainaDoBrasil	6563	41116	Depois da Reforma da Previdência (que há de pa...
okmatheuso	11621	39931	Se a reforma da previdência for aprovada cenas...
BolsonaroSP	7705	39883	O pessoal que fica falando que você é gado, bo...

Fonte: O autor.

Analisando novamente os mil tuítes mais favoritados e os mil tuítes que mais receberam retuítes, observa-se que os primeiros foram produzidos por 276 usuários e os mais retuitados foram produzidos por 314 usuários. Nenhum destes usuários foram considerados *bots*, levando em conta a análise de *bots* na base de dados citada anteriormente. Já com relação à classificação desses tuítes, é possível observar que a proporção da classificação da base original se manteve.

Tabela 7 – Classificação dos mil tuítes mais retuitados e favoritados da base.

(a) Postagens mais retuitadas

Modelo	Positivo	Negativo	Neutro
Naïve Bayes	419	187	394
Regressão Logística	250	33	717
Floresta Aleatória	91	48	861
SVM	259	40	701

(b) Postagens mais favoritadas

Modelo	Positivo	Negativo	Neutro
Naïve Bayes	434	184	382
Regressão Logística	262	30	708
Floresta Aleatória	96	53	851
SVM	274	37	689

Fonte: O autor.

4 Conclusões

Este trabalho de conclusão de curso teve como objetivo a realização de uma análise qualitativa dos dados que pertencem à base criada por Ricci (2020) sobre a Reforma da Previdência no Brasil no ano de 2019. Foram feitas novas classificações e algumas comparações de novos resultados obtidos com os resultados obtidos por ele em seu trabalho.

A ideia inicial foi separar os usuários comuns dos *bots*, com o intuito de analisar o impacto das contas automatizadas na análise de sentimentos realizada nos tuítes da base. A classificação feita para determinar se um usuário é ou não um *bot*, é um processo complexo, onde diversos fatores devem ser levados em conta, tais como: fotos genéricas de perfil, spam de links nos tuítes, spam de várias mensagens iguais sobre um determinado assunto em um curto período de tempo, etc. Apesar de algumas contas terem uma identificação de várias destas características citadas anteriormente, nem sempre é possível classificar uma conta de maneira "fácil", seja com uma análise manual ou utilizando ferramentas de detecção de *bots*. Utilizando a ferramenta *Botometer* vários fatores são analisados para atribuir um *score* para uma conta, mas mesmo contas com *score* alto podem na verdade não ser *bots*, que são os casos de falsos positivos. Porém, levando em conta os prós e contras, acabou-se optando por utilizar a ferramenta para esta análise, pois esse problema de falsos positivos é comum se levarmos em conta o estado da arte da classificação de *bots*.

Após a classificação das contas e diferenciação dos usuários da base, mesmo removendo os tuítes produzidos por contas que foram consideradas automatizadas e reclassificando a base, notou-se que a proporcionalidade observada nos resultados originais foi mantida, ou seja, para o modelo de Naïve Bayes a maioria dos tuítes continuaram sendo positivos e para os demais a maioria dos tuítes da base continuaram sendo neutros. Com o resultado obtido nesta reclassificação da base, tudo leva a crer que não houve uma "rede" de propagação de mensagens, que iria mudar a análise de sentimentos realizada, já que com esse tipo de "rede" a análise iria mostrar uma percepção manipulada dos usuários com relação ao tema. Porém, é importante ressaltar que 184.054 tuítes, o que corresponde a aproximadamente 18,77% da base, foram produzidos por contas inacessíveis pela API, isto é, essas contas não puderam ser classificadas. Esse número de quase 20% da base é um número bem relevante, pois se todas essas contas pudessem ser diferenciadas em *bots* e usuários comuns, a conclusão sobre a existência de uma "rede" de propagação de mensagens no contexto dos tuítes da base poderia ser diferente.

Também foi possível observar que mesmo após o tratamento de *hashtags* realizado no pré-processamento dos dados da base, ao reclassificar todos os tuítes utilizando

os modelos, não houve grande diferença ao comparar os novos resultados com aqueles obtidos por Ricci (2020) em seu trabalho, onde as *hashtags* foram retiradas antes da classificação. Para Naïve Bayes a maioria dos tuítes continuou recebendo classificação positiva enquanto para Regressão Logística, Floresta Aleatória e SVM, a maioria dos tuítes continuou recebendo classificação neutra.

4.1 Trabalhos Futuros

Neste trabalho foi utilizada a ferramenta *Botometer* para classificação de *bots*. Seria interessante reclassificar os usuários da base utilizando outras ferramentas, ou mesmo desenvolver um classificador de *bots* próprio. Para montar este classificador seriam utilizados diferentes *datasets*, contendo tuítes produzidos tanto por *bots* como por pessoas comuns. Também poderia haver uma classificação manual de uma amostra da base, visitando perfis de usuários desta amostra, identificando os *bots* e procurando um padrão entre os tuítes produzidos por eles.

Como neste trabalho foram utilizados os mesmos modelos disponibilizados pelo criador da base, para comparação de resultados pós tratamento de *hashtags* com os resultados originais, além de outras comparações "diretas", é interessante criar novos classificadores, que serão montados utilizando novos dados de treino, a fim de obter novos resultados. Com esses novos resultados obtidos, pode-se observar se a classificação dos tuítes da base permanece parecida ou se muda drasticamente. Além disso, também pode ser feita uma classificação "simplificada", onde tuítes que contenham *hashtags* já são rotulados de acordo com a classificação da *hashtag*. No caso de várias *hashtags*, o tuíte recebe a classificação igual da maioria das *hashtags* que ele contém, assim, os tuítes que possuem *hashtags* teriam uma classificação mais direta, pois não precisariam passar por todo o processo que os demais são submetidos.

Por fim, também poderiam ser classificadas manualmente amostras mais "relevantes" da base, como por exemplo, os mil tuítes mais retuitados. Algumas novas análises poderiam ser feitas a partir dessa amostra citada como, por exemplo, verificar dentre esses mil tuítes quantos foram produzidos por figuras públicas, ou qual foi a classificação majoritária destes tuítes.

Referências

ANSARI, M. Z.; AZIZ, M.; SIDDIQUI, M.; MEHRA, H.; SINGH, K. Analysis of political sentiment orientations on twitter. *Procedia Computer Science*, Elsevier BV, v. 167, p. 1821–1828, 2020. Citado na página 8.

AYLIEN. Using NLP and text mining to understand how media coverage influenced the US presidential election. *Aylien*, 2021. Disponível em: <https://bit.ly/3gIDESJ>. Acesso em: 27 abr. 2021. Citado na página 8.

BLONDEL, V. D.; GUILLAUME, J.-L.; LAMBIOTTE, R.; LEFEBVRE, E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics Theory and Experiment*, IOP Publishing, v. 2008, n. 10, p. P10008, out. 2008. Citado na página 21.

CÂMARA DOS DEPUTADOS. *PEC 6/2019*: Proposta de emenda à constituição. Brasília, DF, 2019. Disponível em: <https://bit.ly/3u7Ju44>. Acesso em: 24 abr. 2021. Citado na página 8.

COSTA, P. A. *Reforma da previdência no Twitter: a relação entre o debate e a tramitação da PEC 06/2019 no Senado Federal*. 53 p. Monografia (Trabalho de Conclusão de Curso de Pós-graduação) — Centro Universitário de Brasília, Brasília, DF, 2020. Citado na página 20.

FRANÇA, T. C. de; OLIVEIRA, J. Análise de sentimento de tweets relacionados aos protestos que ocorreram no Brasil entre junho e agosto de 2013. In: *Anais do III Brazilian Workshop on Social Network Analysis and Mining*. Porto Alegre, RS: Sociedade Brasileira de Computação, 2014. p. 128–139. Citado na página 22.

GATEFY. O que é regressão logística e como a utilizamos para classificar e-mails. *Gatefy*, mar. 2021. Disponível em: <https://bit.ly/32TJMzs>. Acesso em: 25 abr. 2021. Citado na página 14.

GOKULAKRISHNAN, B.; PRIYANTHAN, P.; RAGAVAN, T.; PRASATH, N.; PERERA, A. Opinion mining and sentiment analysis on a Twitter data stream. In: *International Conference on Advances in ICT for Emerging Regions*. EUA: IEEE, 2012. p. 182–188. Citado na página 19.

GOMES, P. C. T. Análise de sentimentos com machine learning. *Data Geeks*, mar. 2019. Disponível em: <https://bit.ly/3yxr8vZ>. Acesso em: 20 mai. 2021. Citado na página 13.

_____. Classificação com naive bayes. *Data Geeks*, fev. 2019. Disponível em: <https://bit.ly/3ueptsF>. Acesso em: 25 abr. 2021. Citado na página 14.

LIU, B. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing, 2nd Ed.*, 2010. Citado na página 8.

MACHADO, R.; SILVEIRA, W. Reforma da Previdência prevê idade mínima de 65 anos para homens e 62 para mulheres. *Agência Câmara de Notícias*, fev. 2019. Disponível em: <https://bit.ly/32WH6B5>. Acesso em: 24 abr. 2021. Citado na página 8.

MEDURU, M.; MAHIMKAR, A.; SUBRAMANIAN, K.; PADIYA, P.; GUNJGUR, P. N. Opinion mining using Twitter feeds for political analysis. *International Journal of Computer (IJC)*, v. 25, n. 1, p. 116–123, maio 2017. Citado na página 8.

NAQA, I. E.; MURPHY, M. J. What is machine learning? In: NAQA, I. E.; LI, R.; MURPHY, M. J. (Ed.). *Machine Learning in Radiation Oncology: Theory and Applications*. Cham: Springer International Publishing, 2015. p. 3–11. ISBN 978-3-319-18305-3. Citado na página 12.

OLIVEIRA JUNIOR, G. M. de. *Máquina de Vetores Suporte: estudo e análise de parâmetros para otimização de resultado*. 41 p. Monografia (Trabalho de Graduação) — Universidade Federal de Pernambuco, Recife, PE, 2010. Citado na página 15.

RICCI, R. D. *Análise de sentimentos no Twitter sobre a Reforma da Previdência no ano de 2019*. 70 p. Monografia (Trabalho de Conclusão de Curso) — Universidade Federal de Uberlândia, Uberlândia, MG, 2020. Citado 13 vezes nas páginas 9, 10, 13, 17, 19, 20, 22, 24, 27, 28, 29, 36 e 37.

RICCI, R. D.; FARIA, E. R.; MIANI, R. S.; GABRIEL, P. H. R. Social security reform in Brazil: A twitter sentiment analysis. In: KÖ, A.; FRANCESCONI, E.; KOTSIS, G.; TJOA, A. M.; KHALIL, I. (Ed.). Cham: Springer International Publishing, 2021, (Lecture Notes in Computer Science, v. 12926). p. 143–154. Citado 2 vezes nas páginas 20 e 24.

SILVA, L. R. *Análise de Sentimentos Aplicada à Política*. 64 p. Monografia (Trabalho de Formatura Supervisionado) — Universidade de São Paulo, São Paulo, SP, 2018. Citado na página 21.

YIU, T. Understanding random forest. *Towards Data Science*, jun. 2019. Disponível em: <https://bit.ly/3u5Dzw4>. Acesso em: 25 abr. 2021. Citado na página 15.