Análise de Sentimentos em Rede Social: Pandemia do Coronavírus no Brasil

William M. Rosa[1], **Eder Pazinatto**[1]

[1]Instituto de Ciências Exatas e Geociências – Universidade de Passo Fundo (UPF) – 99052-900 – Passo Fundo – RS – Brazil

{152096, ederpazinatto}@upf.br

Abstract. *Social networks are places with a lot of users. There, people from around the world share their feelings about most diverse subjects, which means the texts published on social media can be explored to try to understand what a certain public thinks about a specific subject. Sentiment Analysis arises from the need to understand what is the feeling in these texts in a productive way, due to the great amount of connected users. Through the data that is extracted from the texts, useful information for decision-making is generated. This article presents the sentiment analysis of Twitter, during the Coronavirus pandemic in Brazil, it was possible to perceive better reactions after the vaccination period started.*

Resumo. *As redes sociais são locais em que muitos usuários frequentam. Nelas, pessoas do mundo inteiro demonstram seus sentimentos sobre diversos assuntos. Dessa forma os textos publicados nas redes sociais podem ser explorados de forma a saber qual a reação de um público sobre algum assunto. A Análise de Sentimentos surge da necessidade de entender qual o sentimento nesses textos de forma produtiva devido ao grande número de usuários nas redes. Por meio dos dados extraídos dos textos, são geradas informações úteis para tomadas de decisões. Nesse contexto, este trabalho apresenta a análise de sentimentos de uma rede social, durante um período da pandemia do Coronavírus no Brasil, utilizando o método Naive Bayes, foi possível perceber o aumento de percepções positivas após o início do período das vacinas.*

1. Introdução

As redes sociais são espaços na internet que se expandiram ao decorrer dos últimos anos. Nas redes as pessoas compartilham ideias, opiniões e conhecimentos todos os dias. A rede social Twitter foi criada em 2006, popularizando-se como uma plataforma que permitia a expressão do dia a dia e as opiniões de seus usuários em mensagens curtas, surgindo assim um novo meio de comunicação. Opiniões são centrais para quase todas as atividades e são influenciadoras de nossos comportamentos [Liu 2012].

Com a popularização do formato, as empresas perceberam que um comentário em uma rede social tem uma capacidade de atingir um público grande e de afetar positivamente ou negativamente sua reputação no mercado[Padilha e Evangelista 2014]. Segundo [Marques e Vidigal 2018], 92% dos usuários acessam a Internet em busca de informação e 71% dos internautas estão presentes em redes sociais. A pesquisa também mostra que esses dados pertencem a internautas residentes em regiões metropolitanas do país e pertencentes a distintas classes sociais, estão presentes nessas redes, demonstrando o tamanho do público das redes sociais e assim justificando o maior interesse de organizações nesse formato.

Com o maior interesse das organizações nas redes sociais, era necessário poder extrair dados das redes sociais e tornar-los em informações úteis. Como a quantidade de dados é grande, foi necessário criar e automatizar o processo. O termo análise de sentimento surgiu pela primeira vez em 2003, com a intuição de extrair sentimentos associados positivamente ou negativamente para um assunto específico de um documento, ao invés de classificar o documento inteiramente como positivo ou negativo. A principal dificuldade da análise de sentimentos é identificar como os sentimentos são expressados em textos e se as expressões são positivas(favoráveis) ou negativaas(desfavoráveis)[Nasukawa e Yi 2003].

A análise de sentimentos é um processo complexo que envolve cinco passos principais para analisar os dados. Esses passos principais são:[D'Andrea et al. 2015]

- Coleta de Dados: o primeiro passo consiste em coletar dados de usuários de redes sociais, forums, blogs, entre outros. Esses dados são desorganizados, passiveis de manipulação pelos próprios usuários através de *bots*, usuários robôs automatizados, e usam diferentes vocabulários e escritas;
- Preparação dos dados: consiste em preparar os dados coletados, separando dados que não sejam texto e não relevantes para o assunto a ser analisado;
- Detecção de Sentimentos: as frases retiradas dos forums, redes sociais ou blogs são examinadas;
- Classificação de Sentimentos: após examinadas, as frases serão classificadas como positivas ou negativas, boas ou ruins, entre outras classificações. Essas classificações podem ser feitas utilizando vários pontos de análise;
- Apresentação dos resultados: o principal objetivo da análise de sentimento é converter textos sem classificação em informação útil. Quando a análise é terminada, o resultado pode ser apresentado em variados tipos de graficos, como barras, linha e pizza. O tempo também pode ser incluido na apresentação dos resultados.

Diversas organizações têm utilizado da análise dos sentimentos nessas publicações em redes sociais como uma das atividades do processo de seus negócios. Times que acompanham os resultados das análises e times que respondem usuários em redes sociais são essenciais nos dias de hoje. Um exemplo na indústria automotiva é a comparação entre marcas como Mercedes, Audi e BMW. Segundo [Shukri et al. 2015], foi possível analisar

que na categoria *joy* a BMW esta a frente das outras duas marcas, enquanto na categoria *sadness* a Audi lidera. E essas são informações importantes para todas as organizações, por isso o aumento do interesse na análise de sentimentos.

Neste sentido, este trabalho tem como objetivo apresentar a análise de sentimentos aplicada a textos de uma rede social – Twitter – a fim de identificar os sentimentos da população brasileira a respeito da vacina e do Coronavírus.

2. Referencial Teórico

Nesta seção é apresentada uma visão geral sobre os conceitos abordados neste trabalho.

2.1. Redes Sociais

A maior forma de interação entre as pessoas na atualidade é a expressão de sentimentos em textos nas redes sociais. Segundo [Antunes et al. 2014] há um grande interesse nas redes sociais por parte do setor privado, principalmente por permitirem uma maior intereção entre as empresas e os clientes, além de permitirem o levantamento de dados relacionados aos produtos e serviços das próprias empresas. Uma das redes sociais, o *Twitter*, tem como diferencial as *hashtags*, indicando que o *Tweet*, forma com a qual as postagens são chamadas, faz referência a um tópico em específico, sendo assim um facilitador para obter feedbacks precisos.

As redes sociais têm sido bastante usadas para analisar a aceitação do público em relação a um produto, serviço ou acontecimento, uma vez que análise do sentimento de aprovação ou desaprovação expresso pelas frases que os clientes publicam nas rede sociais torna possível que organizações saibam sobre a reputação de produtos, serviços ou acontecimentos de acordo com os usuários das redes online [Silva 2016].

2.2. Análise de Sentimentos

Segundo [Silva 2016], a análise de sentimentos é um campo de estudo de mineração de dados com recente popularização ao crescimento da Internet e do conteúdo que é gerado por seus usuários, principalmente nas redes sociais, nas quais as pessoas publicam suas opiniõpes em uma linguagem coloquial e em muitos casos utilizando de artifícios gráficos para tornar ainda mais sucintos seus diálogos.

Dentre as aplicações, a mais frequente é a avaliação de produtos [Medhat et al. 2014], e os maiores desafios tem sido a dificuldade do processamento de dados em diferentes idiomas, devido a diversificação de caracteres e a classificação de ironia e sarcasmo.

2.3. Classificadores de Sentimentos

Existem várias formas de classificar sentimetnos em textos. Os principais meios utilizam algoritmos de aprendizado de máquina e classificadores, com base na semântica do texto [Duarte 2013].

2.3.1. Aprendizagem de Máquina

Segundo [Duarte 2013], esse tipo de classificador faz uso de algoritmos que são treinados com um conjunto de dados previamente rotulados e, assim, são capazes de classificar uma nova instância de acordo com o conhecimento adquirido.

2.3.1.1 Naive Bayes

O teorema de ***Bayes*** descreve a probabilidade de um evento, baseado em um conhecimento a *priori* que pode ou não estar relacionado ao evento escolhido para utilizar o teorema. O teorema mostra como alterar as probabilidades a *priori* tendo em vista novas evidências para obter probabilidades a posteriori [Bussab e Morettin 2010].

O teorema de ***Bayes*** diz que Posterior = probabilidade * proposição/evidência, como demonstrado na seguinte fórmula:

$P(A|B) = P(B|A) * P(A)/P(B)$

Baseado em um conhecimento anterior que pode ou não estar relacionado ao evento atual.

Utilizando alguns dados ficticios, temos que:

- 100 pessoas realizaram um teste;
- 20% das pessoas que realizaram o teste possuíam a doença;
- 90% das pessoas que possuíam a doença, receberam positivo no teste;
- 30% das pessoas que não possuíam a doença, receberam positivo no teste.

Utilizando o teorema de ***Bayes*** temos que:

- $P(doença|positivo) = 20\% * 90\%$
- $P(doença|positivo) = 0,2 * 0,9$
- $P(doença|positivo) = 0,18$
- $P(semdoença|positivo) = 80\% * 30\%$
- $P(semdoença|positivo) = 0,8 * 0,3$
- $P(semdoença|positivo) = 0,24$

Após isso é necessário normalizar os dados, para que a soma das duas probabilidades seja igual ou próximo a 1, porém sem ser maior que 1.

- $P(doença|positivo) = 0,18/(0,18 + 0,24) = 0,4285$
- $P(semdoença|positivo) = 0,24/(0,18 + 0,24) = 0,5714$
- $0,4285 + 0,5714 = 0,9999$ ou aproximadamente 1

Com isso é possivel concluir que se o resultado do teste de uma nova pessoa for positivo, ela possui aproximadamente 43% (0,4285) de chance de estar doente.

O algoritmo ***Naive Bayes*** é um classificador probabilístico baseado no **Teorema de Bayes**. Por ser um método simples de classificação, é muito utilizado. Sua principal característica é poder ter uma base de treinamento totalmente descorrelacionada ao assunto com o qual o algoritmo será utilizado.

2.3.1.2 Máquina de Vetores de Suporte

A máquina de vetores suporte, também chamada por *Support Vector Machine*(SVM), é um outro classificador, desenvolvido por Vapnik, e têm a capacidade de resolver problemas de classificação e regressão, diferentemente do ***Naive Bayes*** que não consegue lidar com regressão, adquirindo com o aprendizado na etapa de treinamento a capacidade de generalização.

Uma SVM Constrói um classificador de acordo com um conjunto de

padrões por ele identificados nos exemplos de treinamento, onde a classificação é conhecida[Junior 2010], e este conjunto de padrões é colocado em um hiperplano, como demonstra a Figura 1.

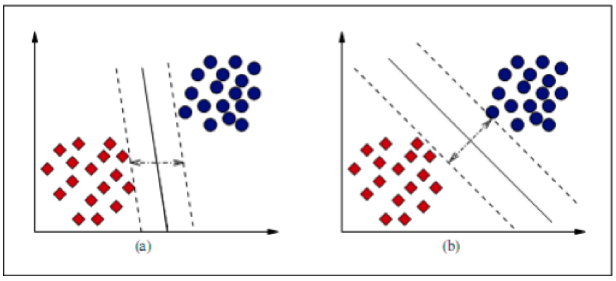

Figura 1. Exemplo de uma Máquina de Vetores de Suporte [Junior 2010]

2.3.2. Classificadores Semânticos

Classificadores semânticos atribuem sentimentos usando dicionários e recursos léxicos que contêm palavras previamente rotuladas. Os vários sentidos de uma mesma palavra são levados em consideração para fornecer uma classificação mais específica [Duarte 2013].

Alguns métodos que utilizam classificadores semânticos:

- SentiWordNet, como demonstra a Figura 2;
- SentiLex;
- OpLexicon.

POS	Offset	PosScore	NegScore	SynsetTerms
a	01150475	0	0.625	sorry#1
a	02273643	0.5	0	secure#5
a	01838253	0.625	0	fine#2
n	03931044	0	0	image#3
n	03931044	0	0	picture#1
v	01824736	0.125	0	like#1

Figura 2. Estrutura do classificador SentiWordNet [Amarouche et al. 2015]

3. Metodologia

Para a realização deste trabalho, com base nos conceitos de análise de sentimentos e redes sociais, foi definido como amostra um conjunto de tweets relacionados a pandemia do Coronavírus no Brasil. Para isso, foram utilizadas *Application Programming Interfaces*, APIs, de extração de dados e de análise de sentimentos.

3.1. Coleta e Tratamento de Dados

Os dados utilizados foram obtidos por meio da API da rede social *Twitter*, selecionados pela palavra chave Coronavírus no periodo de setembro de 2020 a maio de 2021.

A escolha da rede social twitter se deu pelo fator quantidade, pois é a rede social onde os usuários, principalmente jovens e jovens adultos, opinam sobre diversos temas diariamente de forma abundante.

Após a finalização da coleta de dados, que resultou em mais de 13000 *Tweets*, foi necessário tratar os dados para retirar o que não era relevante para a análise e também foi identificado que muitos dos dados eram repetidos devido a contas bots, contas automáticas que normalmente possuem o intuito de repetir várias vezes o mesmo *Tweet* par que assim apareçam nas páginas dos outros usuários. Também foi necessário a tradução manual, devido a gírias e abreviações, para o inglês para que os dados pudessem ser comparados a uma outra base de dados previamente rotulada.

Para a preparação do texto, a tarefa foi dividida em quatro fases, como demonstra a Figura 3:

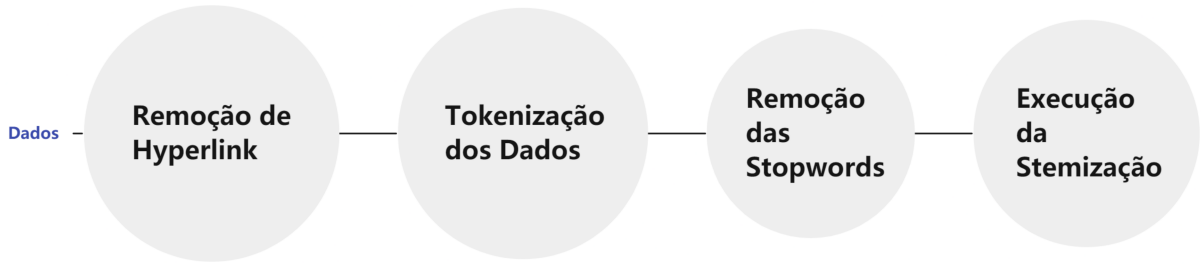

Figura 3. Fluxograma da preparação de dados

- Remoção de hyperlinks e marcação: fase inicial, na qual foi feita a remoção de partes irrelevantes dos dados, como hyperlinks e marcações da própria rede social;
- Tokenização dos dados: fase em que o texto foi separado palavra por palavra;
- Remoção das *Stopwords*: fase em que foi feita a retirada de palavras que podem ser consideradas irrelevantes para a análise e polarização do texto;
- Stemização: fase final que constitui em processar e reduzir as palavras flexionadas e ou derivadas ao seu tronco(raiz) sendo que o tronco não necessariamente precisa ser idêntico a palavra original.

4. Aplicação

Nesta seção serão apresentados os métodos utilizados para a realização do trabalho.

4.1. Linguagem e API

A escolha de ***Python*** como linguagem deve-se a diversos fatores, como:

- Comunidade madura e solidária: *Python* foi criado há mais de 30 anos e muitos alunos são iniciados em ciência da computação com *Python*, portanto é uma linguagem que teve tempo para amadurecer e é solidária devido a como boa parte das pessoas são introduzidas ao mundo da computação via *Python*;
- Versatilidade e Bibliotecas: *Python*, muito por sua popularidade, é uma linguagem que possui muitas bibliotecas e *frameworks* desenvolvidos;
- *Python* possui sua codificação de fácil leitura e é bastante utilizada para desenvolvimento web e *Machine Learning*

Para a extração de dados foi utilizada a API fornecida pela própria rede social *Twitter*, após um pequeno processo de cadastro como desenvolvidor, que possui algumas limitações, devido a ferramenta utilizada ser gratuita. A API disponibilizada para python possui limitações relacionadas a quantidade de *Tweets* que podem ser resgatados por vez. Também possui limitações com a forma que os *Tweets* podem ser extraídos: do presente (momento em que a API foi chamada) para o passado. Devido a essas barreiras impostas pela API, foi necessário encontrar um valor de *requests* na API que garantisse a não duplicação de *Tweets* e que não excedesse o limite da própria API. Os parâmetros utilizados para a busca na API foram os seguintes:

- **Q**: parâmetro de busca, com as palavras-chave variando entre ”coronavírus”e ”vacina”;
- **Count**: Quantidade de *Tweets* a serem resgatados, 200 por coleta;
- **Tweetmode**: *Tweets* completos ou encurtados, foram buscados completos;
- **Resulttype**: tipos de *Tweets* como populares, comuns ou ambos, foi escolhido buscar ambos;
- **Lang**: parâmetro para a escolha de linguagem, no caso, a portuguesa.

Após cada execução do programa, era adicionado em uma base de dados os novos *Tweets* recebidos pela API para o caso de estudo. As seguintes tabelas 1 e 2 apresentam o volume de *tweets*, tanto da base pré-rotulada como para a base criada para o estudo.

	Treinamento	Análise	Pré pre-processamento
Tweets	3650	15527	16420

Tabela 1. Volume total de *tweets*

Como demonstrado na Tabela 1, 16420 *Tweets* foram extraídos e após o pré-processamento, 15521 foram destinados a análise de sentimentos e outros 893 foram descartados. Outros 3650 *Tweets* foram utilizados para o treinamento.

	Positivos	Negativos
Tweets	2140	1510

Tabela 2. Volume de dados da base de treinamento

Para a base de treinamento, os tweets foram manualmente rotulados em 2 tipos:

- Positivo: quando o *Tweet* era julgado como positivo pelo autor;
- Negativo: quando o *Tweet* era julgado como negativo pelo autor.

A figura 4 exemplifica *Tweets* utilizados para a base de treinamento.

```
Positive Example:
My beautiful sunflowers on a sunny Friday morning off :) #sunflowers #favourites
Negative Example:
okay she doesnt want to talk to me then I will stop :(
```

Figura 4. Exemplo de *Tweets* pré-rotulados

4.2. Preparação e Desenvolvimento

Para a realização da preparação, primeiro foi necessário fazer um pré-processamento. Devido a popularidade de contas *bots*, foi necessário criar um processo para retirar os tweets repetidos e também retirar alguns manualmente, pois a quantidade de gírias dificultava o processo automático.

Após a remoção, foi necessário traduzir cuidadosamente os tweets para o inglês, o que foi feito manualmente devido a limitação de requests encontrada nas APIs de tradução e a quantidade de palavras não formais encontradas nos textos. Foi criado um dicionário, de acordo com a experiência do autor, com as palavras não formais encontradas e seus significados em português.

Foi necessário traduzir para o inglês devido ao fato de que a rede é majoritariamente utilizada por jovens de outros países, que utilizam principalmente o inglês para interagir, facilitando assim a extração de dados para a criação de uma outra base de dados pré rotulada manualmente.

Além disso, há maior quantidade e disponibilidade de ferramentas para o pré-processamento e a Análise de Sentimento para a língua inglesa. A partir dessa decisão, foi garantido que a base de dados para este estudo fosse grande, tanto para a parte de treinamento do método quanto para a análise do caso de estudo.

Começando a preparação dos dados para serem analisados, foram utilizadas operações em expressões regulares, comumente chamadas de *regex*, para a remoção de *hyperlinks* e marcações da rede social, como *hashtaghs* e *retweets*, como demonstra a Figura 5:

```
def removeHyperlink(tweet):

    new_tweet = re.sub(r'^RT[\s]+', '', tweet)#rt
    new_tweet = re.sub(r'https?:\/\/.*[\r\n]*', '', new_tweet)#links
    new_tweet = re.sub(r'#', '', new_tweet)#

    return new_tweet
```

Figura 5. Removendo marcações e *hyperlinks*

Em seguida, foi necessário tokenizar os textos, diminuindo o tamanho de palavras com muitas letras repetidas e, não necessariamente, transformando todas as letras em letras minúsculas, para em seguida remover as palavras que não dão ou não alteram o sentido da frase. Para isso foi utilizado o conjunto de bibliotecas ***NLTK***, ou *Natural Language Toolkit*, e sua função TweetTokenizer, para separar todas as palavras de todos os textos, deixando todas as letras minúsculas e reduzindo o tamanho das palavras quando eram encontradas letras repetidas e sem sentido. Para isso, foram utilizados os parâmetros **preserve_case=False**, para deixar todas as letras minúsculas, os parâmetros **strip_handles=True** e **reduce_len=True** para reduzir o tamanho das palavras com letras repetidas, como demonstra a Figura 6:

```
tokenizer = TweetTokenizer(preserve_case=False, strip_handles=True,
                           reduce_len=True)

def tokenizeTweet(tweet):

    tweet_tokens = tokenizer.tokenize(tweet)

    return tweet_tokens
```

Figura 6. Exemplo de código para reduzir palavras e as separar

Após a tokenização, foi feita a retirada das *stopwords*, utilizando um pacote da biblioteca **NLTK** que já possui todas as stopwords em várias linguas diferentes, e das pontuações, como demonstra a Figura 7:

```
nltk.download('stopwords')
stopwordsEnglish = stopwords.words('english')
punctuations = string.punctuation

def removeStopwords(tweet_tokens):

    tweets_clean = []

    for word in tweet_tokens:
        if (word not in stopwordsEnglish and word not in punctuations):
            tweets_clean.append(word)

    return tweets_clean
```

Figura 7. Remoção das *stopwords*

Na última etapa da preparação, a stemização, foi utilizada uma função da biblioteca **NLTK**, *PorterStemmer*, para reduzir as palavras em suas formas radicais. Embora a biblioteca tenha módulos para a execução do *Naive Bayes*, os mesmos não foram utilizados e o algoritmo foi implementado manualmente como demonstra o item **4.2.1 Treinamento**.

4.2.1. Treinamento

Para iniciar o treinamento do algoritmo utilizando a base de dados para treino ***Naive Bayes***, primeiro foi necessário criar um dicionário com a frequência das palavras. Após criada, o algoritmo para treinar o método usou três parâmetros, sendo eles:

- Frequência;
- Lista de *Tweets*;
- Lista dos rótulos(positivo/negativo) dos *Tweets*.

Sua saída será a probabilidade da palavra ser positiva ou negativa e a probabilidade dela estar em um *Tweets* positivo ou negativo.

Para o treinamento, foi necessário separar o dicionário de frequência de palavras em palavras únicas positivas ou negativas, calcular o logaritmo da probabilidade da palavra ser positiva ou negativa e calcular o logaritmo da probabilidade da palavra estar em um *Tweet* positivo ou negativo, sendo este último salvo em um vetor devido a quantidade

grande de palavras que podem existir nos dados. É necessário utilizar logaritmo pois as operações são feitas com vários números decimais pequenos e isso pode levar a um *underflow* de precisão numérica, por tanto, é boa prática utilizar logaritmos para evitar esse possível problema. Para efetuar o cálculo, foi utilizada a biblioteca *NumPy* como demonstra a Figura 8:

```
#calcula logprior
logprior = np.log(dPos) - np.log(dNeg)

for word in unique_words:

    #frequencia negativa e positiva da palavra
    freqPos = freqs.get((word, 1), 0)
    freqNeg = freqs.get((word, 0), 0)

    #probabilidade da palavra ser positiva ou negativa
    probPos = (freqPos + 1) / (nPos + V) #V = quantidade de palavras únicas
    probNeg = (freqNeg + 1) / (nNeg + V) #V = quantidade de palavras únicas

    #>0 = positivo , <0 = negativo
    loglikelihood[word] = np.log(probPos / probNeg)
```

Figura 8. Cálculo das probabilidades.

4.2.2. Execução

Para executar o algoritmo treinado, também foi necessário criar uma função que necessitará três parâmetros para seu funcionamento, sendo eles:

- Texto a ser analisado;
- Probabilidades das palavras serem positivas ou negativas;
- Probabilidade da palavra estar em um *Tweet* positivo ou negativo.

Como demonstrado na Figura 9:

```
def bayesPredict(tweet, logprior, loglikelihood):
    processedTweet = processTweet(tweet)

    p = 0

    p += logprior

    for word in processedTweet:
        if word in loglikelihood:
            p+= loglikelihood[word]

    return p
```

Figura 9. Naive Bayes

Para cada *Tweet*, serão somados as polaridades das palavras e as polaridades da probabilidade das palavras estarem em *Tweets* positivos ou não. Com isso, o resultado será negativo, menor que zero, caso o *Tweet* seja considerado negativo, ou positivo, maior que zero, caso o *Tweet* seja considerado positivo, como demonstrado na Tabela 3:

Tweet Original	*Score*
3 billions for buying tractors, but can't buy vaccines. Bolsonaro should fall	-3.58
My dad was vaccinated, thank god I'm so relieved	0.57
More vaccines are coming to my city happy	2.15
So many are dying to COVID daily sad	-3.23

Tabela 3. Tabela com exemplos de resultados de análise para *Tweets* positivos e negativos, já em inglês.

4.3. Resultados

Nesta seção serão apresentados os gráficos gerados após efetuar a análise dos dados.

Durante o periodo de coleta, de setembro a maio, tiveram muitos altos e baixos com relação a pandemia do Coronavírus no Brasil. Após a finalização do método, os resultados foram somados e armazenados por mês. O resultado final do mês deve-se a soma de todas as análises feitas nos *Tweets* do mesmo mês como demonstra a Figura 10:

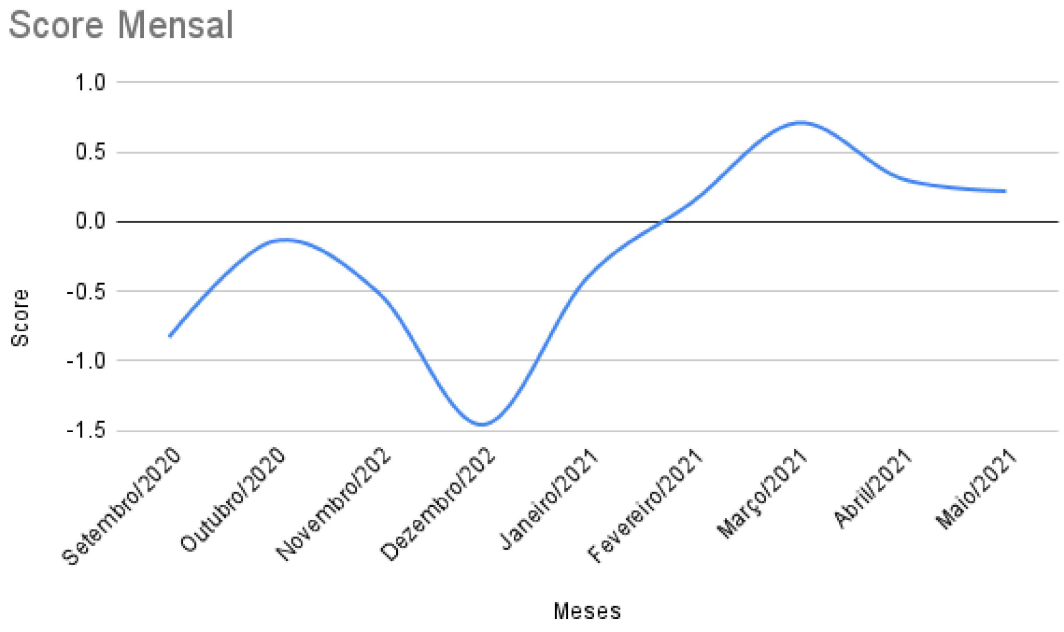

Figura 10. Gráfico demonstrando a variação dos *scores* ao longo da coleta de dados, sendo Dezembro o pior mês e Março o melhor.

O gráfico de dispersão da Figura 11 representa o *Score* associado aos dias do mês janeiro, mês o qual foi iniciado o processo de vacinação no país. O gráfico demonstra um crescimento na quantidade de tweets positivos próximos ao dia do início da vacinação e uma diminuição dos tweets negativos.

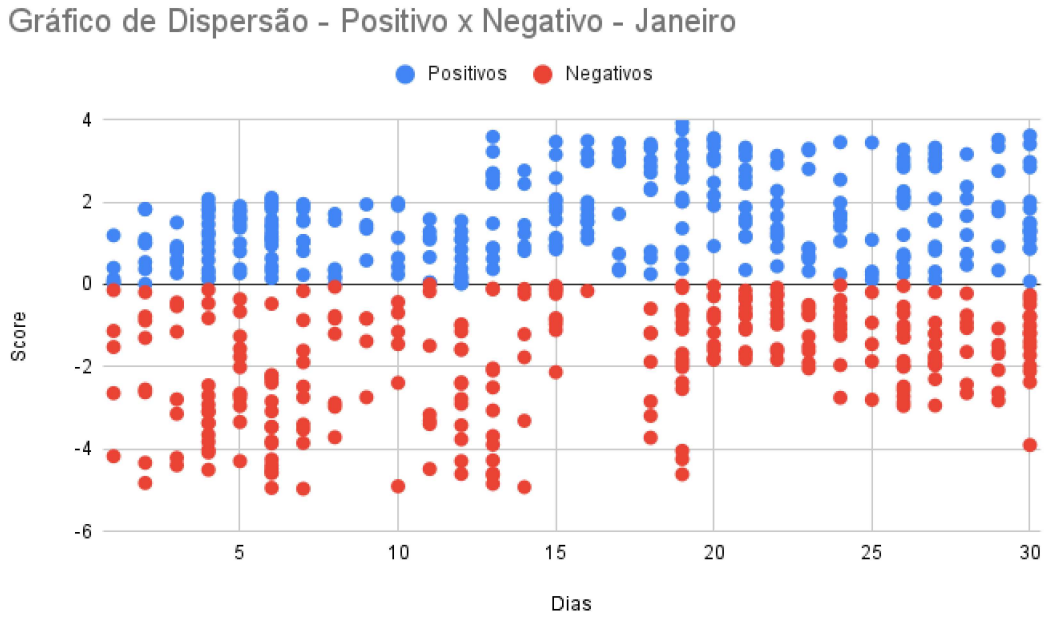

Figura 11. Gráfico demonstrando a melhora nos *scores* aproximadamente ao dia onde as vacinação começaram.

O gráfico da Figura 12 apresenta o montante total positivo e negativo dos *Tweets* extraídos ao longo do período de coleta, com um número negativo expressivamente maior comparado ao positivo. Isso se deve principalmente aos meses de 2020, quando os *Tweets* coletados eram majoritariamente negativos, principalmente no mês de dezembro, como demonstrado na figura 13.

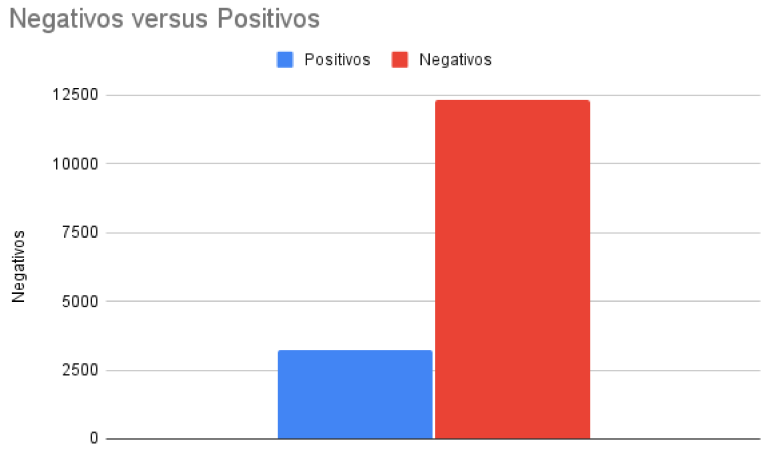

Figura 12. *Tweets* negativos superam em quase 5 vezes os *tweets* positivos.

O gráfico de dispersão abaixo representa o mês com menor score entre todos e muito está relacionado às datas comemorativas, como é possível perceber analisando a figura 13 nos dias próximos ao Natal e Ano novo, pois as pessoas não podiam celebrar as datas pois ainda estavam em uma pandemia.

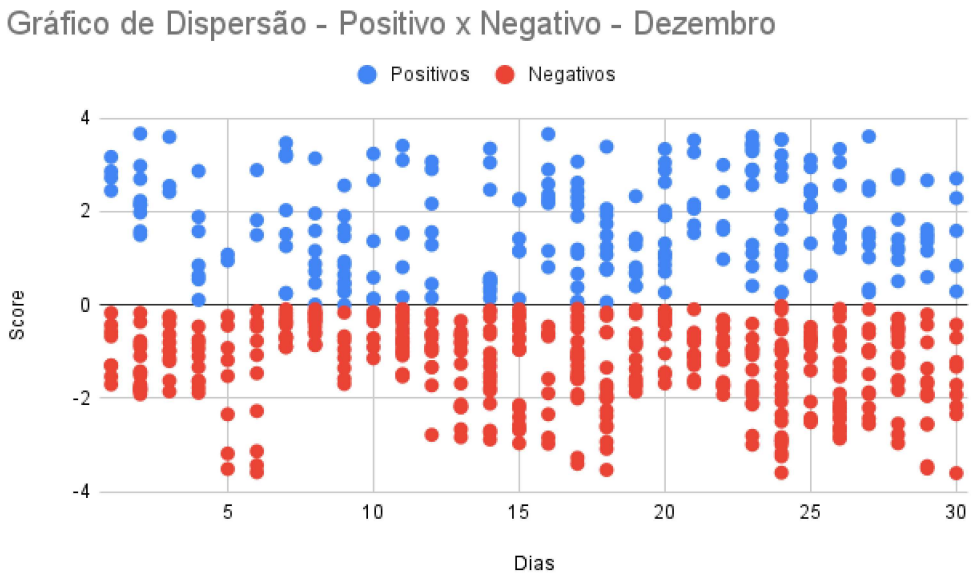

Figura 13. *Scores* do pior mês coletado, encontrando em maior quantia *scores* negativos perto das datas festivas comumente realizadas no final do ano.

5. Conclusão

Utilizando o *Naive Bayes* para análise de sentimentos foi possivel descobrir como uma parcela da população brasileira que faz parte da rede social *Twitter* se sentiu com relação a pandemia do Coronavírus no país. Era esperado pelo autor que a reação geral fosse negativa e a análise de sentimentos demonstrou ser negativa.

Uma das dificuldades foi encontrar uma forma de analisar os textos em português, pois como a maioria da comunidade é internacional e a comunicação é feita em Inglês, a maioria das bases pré-rotuladas para o treinamento também é em Inglês, o que dificulta analisar pois há muitas formas de se escrever em português, e com significados extremamente diferentes dependendo das gírias empregadas.

A parte mais interessante do trabalho foi poder acompanhar como era a reação dos usuários do *Twitter* com relação a pandemia do Coronavírus no país. As diferentes ondas da pandemia mostraram picos de negatividade no Twitter, e é algo que pode ser continuado a ser observado em um futuro.

Referências

Amarouche, K., Benbrahim, H., e Kassou, I. (2015). Product opinion mining for competitive intelligence. *Procedia Computer Science*, 73.

Antunes, M. N., Silva, C., Guimarães, M. C. S., e Rabaço, M. (2014). Social media monitoring: the dengue e-monitor. Núcleo de Editoração SBI.

Bussab, W. O. e Morettin, P. A. (2010). *Estatística Básica*. Saraiva, 6th edition.

D'Andrea, A., Ferri, F., Grifoni, P., e Guzzo, T. (2015). Approaches, tools and applications for sentiment analysis implementation. *International Journal of Computer Applications*, 125:26–33.

Duarte, E. S. (2013). Sentiment analysis on twitter for the portuguese language. Faculdade de Ciências e Tecnologia.

Junior, G. M. d. O. (2010). Máquina de vetores suporte: estudo e análise de parâmetros para otimização de resultado.

Liu, B. (2012). Sentiment analysis and opinion mining. In *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool Publishers.

Marques, L. K. S. e Vidigal, F. (2018). In *Prosumers e redes sociais como fontes de informação mercadológica: uma análise sob a perspectiva da inteligência competitiva em empresas brasileiras*. Núcleo de Editoração SBI.

Medhat, W., Hassan, A., e Korashy, H. (2014). Sentiment analysis algorithms and applications. *Shams Engineering Journal*, 5(4):1093–1113.

Nasukawa, T. e Yi, J. (2003). Sentiment analysis: capturing favorability using natural language processing. *K-CAP '03: Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77.

Padilha, T. P. P. e Evangelista, T. (2014). Monitoramento de posts sobre empresas de e-commerce em redes sociais utilizando análise de sentimentos. III Brazilian Workshop on Social Network Analysis and Mining.

Shukri, S., Yaghi, R., Aljarah, I., e Alsawalqah, H. (2015). Twitter sentiment analysis: A case study in the automotive industry. *2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, pages 1–5.

Silva, N. F. F. (2016). Análise de sentimentos em textos curtos provenientes de redes sociais. *Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2016.*