Coordenadoria de Tecnologia da Informação
Tecnologia em Análise e Desenvolvimento de Sistemas

Aplicação para Análise de Sentimentos em Tweets

Alexis Cesar Ruiz de Almeida

Sorocaba
Junho – 2022

Coordenadoria de Tecnologia da Informação
Tecnologia em Análise e Desenvolvimento de Sistemas

Alexis Cesar Ruiz de Almeida

Aplicação Para Análise de Sentimentos em Tweets

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Sorocaba – FATEC, como parte dos pré-requisitos para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas

Orientadora: Profa. Dra. Maria das Graças J.M. Tomazela

Sorocaba
Junho – 2022

Agradecimento

Primeiramente à existência, que me possibilita experienciar as mais diversas e únicas situações, e que me permitiu conhecer as pessoas nomeadas abaixo.

Aos meus pais, Paulo Cesar Ferraz de Almeida e Elaine Costa Ruiz de Almeida, por todo amor e carinho, e pelo apoio a todo momento da minha vida, me possibilitando oportunidades maravilhosas que contribuíram muito para me tornar a pessoa que sou.

À minha orientadora, Maria das Graças Junqueira Machado Tomazela pelo grande apoio, correções, ensinamentos, risadas e pela amizade.

Ao Centro Paula Souza pelo ensino técnico e superior de qualidade, que expandiram meus horizontes e me capacitaram a enfrentar muitos obstáculos.

A todos aqueles que contribuíram para o acontecimento deste trabalho.

“Faça ou não faça. Tentativa não há.”
(Mestre Yoda)

RESUMO

As pesquisas na área de Análise de Sentimentos, embora recentes, vem evoluindo constantemente. Muitas são as técnicas para se realizar a análise de sentimento em qualquer que seja o tipo de dado, ao mesmo tempo em que diversos obstáculos surgem e são contornados por abordagens diferentes. A análise de sentimentos tem aplicações em diversas áreas, podendo ser encontrada na literatura em pesquisas sociais, extração de opinião de clientes e identificação de eventos. Com isso, o objetivo deste trabalho foi desenvolver um software para mineração de opiniões baseado em palavras-chave que pode ser utilizado para identificação de sentimento em relação às diversas organizações, temas, partidos etc. Para a realização deste trabalho foi utilizada a pesquisa experimental. Inicialmente foram levantados os conceitos-chave da pesquisa: Análise de Sentimentos, Mineração de Opinião e Algoritmos Classificadores, bem como os trabalhos relacionados ao tema proposto. Para a realização do experimento foi primeiramente solicitado o acesso a API do Twitter, seguido da fase de construção da base de dados, pré-processamento, processamento e avaliação do modelo gerado, com um comitê de classificadores composto pelos algoritmos: Naïve Bayes, *Support Vector Machine* e Regressão Logística. Como resultado, obteve-se um modelo de classificação eficaz para determinar os sentimentos do público-alvo em relação a um determinado conteúdo, com uma acurácia de 96,21%.

Palavras-chave: Análise de Sentimentos; Mineração de Opinião; Aprendizado de Máquina; Algoritmos Classificadores; Processamento de Linguagem Natural; Redes Sociais; Twitter.

ABSTRACT

Research in the Sentiment Analysis area, although recent, is in constant evolution. There are many techniques for performing sentiment analysis on any type of data, as different obstacles arise and are circumvented by different approaches. Sentiment analysis has applications in several areas and can be found in the literature on social research, customer opinion extraction and event identification. Thus, the objective of this work is to develop an opinion mining software based on keywords that can be used to identify feelings in relation to different organizations, themes, political parties, etc. To carry out this work, experimental research was used. Initially, the key concepts of the research were raised: Sentiment Analysis, Opinion Mining and Classification Algorithms, as well as the works related to the proposed theme. To carry out the experiment, access to the Twitter API was first requested, followed by the database construction phase, pre-processing, processing and evaluation of the generated model, with a classifier committee composed of the algorithms: Naïve Bayes, Support Vector Machine and Logistic Regression. As a result, a classification model effective in determining the feelings of the target audience in relation to certain content was obtained with an accuracy of 96.21%.

Keywords: Sentiment Analysis; Opinion Mining; Machine Learning; Classifying Algorithms; Natural Language Processing; Social Networks; Twitter.

Sumário

Lista de Figuras

Lista de Quadros e Tabelas

INTRODUÇÃO

A Análise de Sentimentos é um campo dentro da área de Processamento de Linguagem Natural que tem como objetivo identificar opiniões, sentimentos e até emoções em informações subjetivas (LIU, 2010). Há muitas aplicações para a técnica de análise de sentimento, principalmente no âmbito organizacional em que as empresas podem obter de forma mais ágil milhares de opiniões de seus clientes a respeito de seus produtos como pode ser visto em Sarlan, Nadam e Basri (2014).

A rede social Twitter é uma ótima escolha para o estudo da análise de sentimentos, pois, além de ser uma rede social popular e com foco em textos, as publicações contam com um limite de 280 caracteres, facilitando assim a análise realizada (FELL e LUKIANOVA, 2019).

Para que o computador determine a classe de um texto, como por exemplo, positivo e negativo, são utilizados algoritmos classificadores. Cada classificador tem seu próprio método de atribuir uma classe a um documento, desde cálculos probabilísticos simples até redes neurais. Normalmente, o cálculo realizado pelos algoritmos retorna um valor, após processar cada palavra presente no texto, representando a tendência do documento de pertencer a cada classe pré-determinada. A capacidade de um algoritmo predizer a que conjunto um dado elemento pertença com base em seus atributos se dá por conta de um treinamento a priori, sendo este denominado um aprendizado supervisionado; também há a possibilidade de um algoritmo identificar a classe de um elemento sem um treinamento prévio, sendo um aprendizado não-supervisionado (Feldman, 2013).

Um dos problemas mais comuns durante a análise é a má escrita dos textos, principalmente no Twitter (MARTÍNEZ-CÁMARA, E. et al., 2012). Os usuários costumar utilizar várias gírias, abreviar palavras, se referir ao mesmo elemento de formas diferentes durante o texto etc. Isso está ligado com a quantidade limitada de caracteres na publicação, o que faz com que os usuários adotem uma escrita mais rápida e informal. Para tratar essa questão, são utilizadas técnicas de processamento de linguagem natural, como o processo de redução de palavras,

remoção de palavras vazias, remoção de caracteres especiais, links, *hashtags* etc. Há diversas técnicas a se utilizar para tornar o texto mais fácil de se analisar para o computador, e diversas delas são implementadas, por exemplo, na biblioteca Natural Language Toolkit (NLTK) da linguagem Python.

Diante desse contexto o problema de pesquisa que norteou este trabalho foi: “Quais os algoritmos de mineração de opinião possuem melhor acurácia no julgamento de textos obtidos por meio do Twitter?”

Desta forma, este trabalho teve como objetivo explorar os algoritmos utilizados na mineração de opinião para alcançar uma maior acurácia no julgamento dos textos obtidos por meio do Twitter.

A hipótese é que a utilização de algoritmos classificadores em conjunto pode obter melhor desempenho na classificação de textos obtidos do Twitter.

Neste trabalho foi utilizada a pesquisa experimental, que segundo Gil (2007) consiste na determinação de um objeto de estudo, fazer a seleção de variáveis que sejam capazes de influenciá-lo, definir meios para controlar e observar os efeitos que esta variável manipulada possa produzir neste objeto. Assim foram usados alguns dos classificadores mais utilizados para este fim, com desenvolvimento em linguagem de programação Python3 com auxílio de algumas de suas bibliotecas desenvolvidas para este campo de pesquisa. A critério de estudo, o classificador Naïve Bayes foi desenvolvido pelo próprio autor.

O estudo está dividido da seguinte forma: no primeiro capítulo se encontra a revisão da literatura a respeito dos conceitos e técnicas abordados neste estudo: análise de sentimentos, processamento de linguagem natural, aprendizado de máquina e algoritmos classificadores. Além disso são apresentados os trabalhos relacionados a esta pesquisa. No segundo capítulo está a metodologia utilizada para a realização deste trabalho, no terceiro capítulo os resultados obtidos são apresentados bem como uma discussão sobre eles, por fim, no quarto e último capítulo está a conclusão e notas sobre trabalhos futuros e limitações.

1. Revisão da Literatura

1.1. Análise de Sentimentos

A análise de sentimentos, também conhecida como mineração de opinião, é uma área de estudo relativamente nova, que começou a ganhar atenção da comunidade a partir de 2001. É considerado como sendo mineração de opinião qualquer estudo computacional que envolva opinião (sentimento, avaliação, ponto de vista, emoção e subjetividade) de forma textual (LIU, 2010). A mineração de opinião se refere a área de processamento de linguagem natural, ou, linguística computacional; área que envolve o estudo computacional de sentimentos, opiniões e emoções expressados em texto (SARLAN, NADAM e BASRI. 2014).

Devido à quantidade de informação disponível na Web que cresce continuamente, mal é possível acessar as informações sem a ajuda de um mecanismo de pesquisa; é inviável para um ser humano coletar, categorizar e organizar diversos textos de opinião para usá-los como uma ferramenta de apoio a decisão em um período eficaz de tempo, dessa maneira sistemas de descoberta de opinião são necessários. (LIU, 2010; TSYTSARAU e PALPANAS, 2011).

Muitas são as aplicações da análise de sentimento, algumas das mais utilizadas são: pesquisa (OLENSCKI, et al. 2020), política (SOUZA, 2019), monitoramento de crises (AGUIAR, et al. 2018), detecção de textos ofensivos (COUTINHO e MALHEIROS, 2020) e organizacional (SARLAN, NADAM e BASRI. 2014).

Segundo Feldman (2013) a análise de sentimentos pode ser feita por meio das quatro seguintes abordagens diferentes: em nível de documento, em nível de sentença, em nível de aspecto e comparativa.:

- Análise em nível de documento: essa é a forma mais simples de análise e assume que o documento como um todo representa a opinião do autor sobre um objeto principal. Para essa análise existem duas abordagens principais: aprendizado supervisionado e não supervisionado;
- Análise em nível de sentença: esse método compreende que em um mesmo documento de texto, pode haver mais de uma opinião, mesmo se tratando

da mesma entidade. Aqui se assume que a entidade discutida na sentença é conhecida; também se assume que em cada sentença há uma opinião. Antes de iniciar a análise, é necessário determinar se as sentenças são objetivas ou subjetivas: apenas as subjetivas serão analisadas. A maioria dos métodos utiliza a abordagens supervisionadas para classificar as sentenças em duas classes.

- Análise em nível de aspecto: as abordagens anteriores funcionam bem quando, tanto o documento, quanto as sentenças individuais, tratam de um único objeto, porém, há muitos casos em que esses objetos têm atributos e o texto apresenta uma opinião diferente para cada atributo. Isso ocorre com frequência em avaliações de produtos. Por exemplo, em uma avaliação sobre um *smartphone* o autor pode escrever que gostou da câmera, mas detestou o rápido desgaste de bateria. A análise em nível de aspecto tem como foco o reconhecimento dos sentimentos de um documento e os aspectos a que eles se referem. Uma abordagem clássica utilizada para atingir esse objetivo é extrair frases nominais e manter somente aquelas que possuam frequência acima de um nível determinado experimentalmente. Após separar a lista de aspectos, é utilizado um algoritmo para classificar a polaridade de cada expressão. A polaridade de cada aspecto é determinada por uma média ponderada das polaridades das expressões de sentimentos inversamente ponderadas pela distância entre o aspecto e a expressão.
- Análise comparativa: esta abordagem tem como objetivo identificar sentenças que tenham opiniões comparativas, nas quais se compara a entidade A com a entidade B e se extrai quais são as entidades e as respectivas sobre elas. Isso é possível utilizando uma base de palavras que possa indicar comparação.

Há diversos obstáculos que dificultam o processo da análise de sentimentos. Muitos foram contornados por diferentes estratégias apresentadas na literatura. Alguns destes problemas são (SARLAN, NADAM e BASRI. 2014; MARTINEZ-CÁMARA, et al. 2012):

- Textos escritos informalmente, com muitas abreviações, idiomas e uso de jargão;

- Dados esparsos, ou seja, o autor do texto pode se referir à mesma entidade com abreviações e formas irregulares, principalmente quando os caracteres da publicação são limitados, como é o caso do Twitter;
- Falta de contexto;
- Uso de *emoticons*.

Muitos outros problemas podem ser encontrados na literatura como: negação, dupla negação, comparação, sarcasmo, entre outros. Por esse motivo, diversas técnicas são empregadas par maximizar a capacidade de análise, como o processamento de linguagem natural.

1.2. Processamento de Linguagem Natural

O processamento de linguagem natural (PLN) é uma área que explora como os computadores podem ser usados para compreender e manipular textos ou falas em línguas humanas naturais. (CHOWDHURY, 2005). O PLN é composto por diversas técnicas computacionais para analisar e representar textos de ocorrência natural em um ou mais níveis da análise linguística: fonológico, morfológico, léxico, sintático, semântico, discurso, e, pragmático (LIDDY, 1998).

A Análise de Sentimentos utiliza várias técnicas de PLN para facilitar o processo de identificação das palavras. Algumas dessas técnicas são:

- *Stemming*: é a técnica que busca encontrar palavras derivadas no texto e reduzi-las para suas respectivas raízes (LOVINS, 1968).
- *Tokenization*: é definido em Webster e Kit (1992) como sendo uma espécie de pré-processamento responsável por identificar unidades básicas para o futuro processamento; no caso de textos, essas unidades (*token*) são as palavras, cada token é delimitado por um espaço em branco.
- Remoção de *Stopwords*: consiste em remover palavras irrelevantes para a análise que não carregam valor sentimental. São palavras muito comuns como ‘de’ e ‘para’. (SILVA e RIBEIRO, 2003).

1.3. Aprendizado de Máquina

O aprendizado de máquina é um campo da Inteligência Artificial que tem como objetivo construir modelos matemáticos capazes de predizer ou classificar algo com base em uma base de dados de exemplo (ZHANG, 2020).

A análise de sentimentos utiliza o aprendizado de máquina para julgar um dado texto pertencente a uma classe ou não. Segundo Feldman (2013) e Zhang (2020), os dois principais meios de aprendizado são:

Aprendizado supervisionado: é assumido que existe um número finito de classes, por exemplo ‘positivo’ e ‘negativo’ em que o documento será classificado e os dados que serão disponibilizados para o treinamento do modelo são anotados com sua respectiva classe. Com os dados para treino, o algoritmo classificador aprende os padrões e poderá analisar e classificar outros documentos.

Aprendizado não supervisionado: nesta abordagem, os dados que serão disponibilizados para treino do modelo não são anotados com suas classes. É tarefa do próprio algoritmo analisar os padrões e utilizá-los para discriminar grupos a partir desses dados.

Aprendizado por reforço: é o aprendizado análogo ao aprendizado animal. Diferente do aprendizado supervisionado, não há um treino sobre o que é bom ou ruim; o agente precisa perceber quando algo bom ocorreu por meio de um *feedback*. Um *feedback* positivo é chamado de recompensa e o agente precisa estar programado para identificar essa recompensa e não a tratar apenas como qualquer outra entrada sensória. (RUSSEL e NORVIG, 2013).

1.4. Algoritmos Classificadores

1.4.1. Naïve Bayes

O algoritmo Naïve Bayes é utilizado para classificar instâncias associando-as a uma determinada classe (HAN, KAMBER e PEI, 2011), classificações realizadas através deste algoritmo são chamadas de classificações bayesianas. O algoritmo é baseado no Teorema de Bayes, que recebeu o nome em homenagem a seu criador, Thomas Bayes, um reverendo presbiteriano que viveu no século 18. A classificação tem como função associar um dado elemento a uma classe

baseando-se em um modelo preenchido previamente indicando a qual classe pertence cada registro, desta forma o algoritmo busca através da probabilidade calcular qual classe tem a maior chance de estar atribuída ao dado elemento visando alcançar o máximo de acurácia.

O Teorema de Bayes é utilizado para descrever a probabilidade de um evento ocorrer (ser verdadeiro) após o acontecimento de outro evento. Em seu trabalho, Han ,Kamber e PEI (2011) demonstram que, considera-se que B é um elemento composto por N atributos e A é uma hipótese de cada instância B pertencer a uma classe específica denotada por C, é preciso calcular P(A|B), isto é, a probabilidade da hipótese A ser verdadeira sabendo-se do valor do atributo B. P(A) é a probabilidade inicial de A ser verdadeiro para qualquer B e P(B|A) é então a probabilidade de B ocorrer em condição de A. O teorema é expresso por meio da equação matemática 1:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (1)$$

Onde P(B) é diferente de 0.

A Classificação Bayesiana ocorre da seguinte forma: é dividido um conjunto para treinamento do classificador, contendo as instâncias e suas respectivas classes. Cada instância, por sua vez pode conter N atributos, como no caso de textos que possuem N palavras. Podem existir M classes, e cada instância (X) estará associada a uma dessas classes. Desta forma, o classificador atribuirá um valor para cada atributo de cada instância representando sua probabilidade de pertencer a cada classe. Portanto, uma instância (X) será associada a uma classe (Ci) se a inequação 2 for verdade:

$$P(C_i|x) > P(C_j|x) \dots P(C_i|x) > P(C_m|x) \quad (2)$$

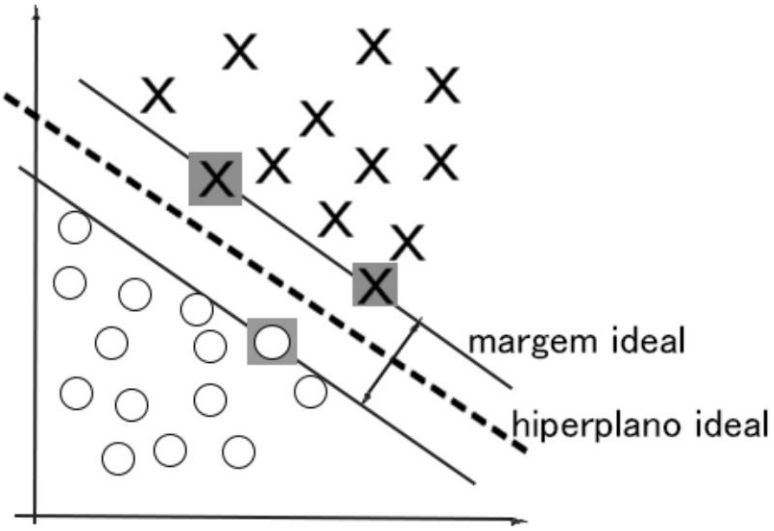
Onde 1 ≤ i, j ≤ m e i ≠ j, m.

Ainda segundo o teorema, cada classe possui um valor representando sua hipótese inicial representado por A.

1.4.2. Support Vector Machine (SVM)

Segundo Hearst (1998) as máquinas de vetores de suporte é uma técnica de análise que pode ser utilizada também para categorizar textos. O aprendizado por meio de vetores de suporte pode ser utilizado tanto para casos simples como complexos. Este modelo de classificação foi introduzido por Cortes e Vapnik (1995) e tem como objetivo encontrar uma divisão entre duas ou mais classes em um hiperplano. A figura 1 demonstra um exemplo de classes separáveis em um espaço bidimensional; uma linha é tracejada na metade da distância entre os dois elementos opostos mais próximos; esses elementos também formam os vetores de suporte, deixando no centro uma margem que pode pender tanto para uma classe quanto para outra. Um exemplo de um problema separável em um espaço bidimensional pode ser visto na figura 1, os vetores de suporte, marcados com quadrados cinzas, definem a margem da maior separação entre as duas classes.

Figura 1 – Problema possível de se separar em um plano bidimensional.



Fonte: Cortes e Vapnik (1995) – Adaptado

Alguns problemas podem surgir ao tentar dividir duas classes por meio de um classificador de vetores de suporte:

- Em um caso de classes A e B, o objeto B mais próximo do limite de A pode estar longe dos demais objetos pertencentes de B, o que levaria a classificar objetos de forma errada.

- As classes podem estar misturadas não sendo possível traçar uma linha reta que as separe;

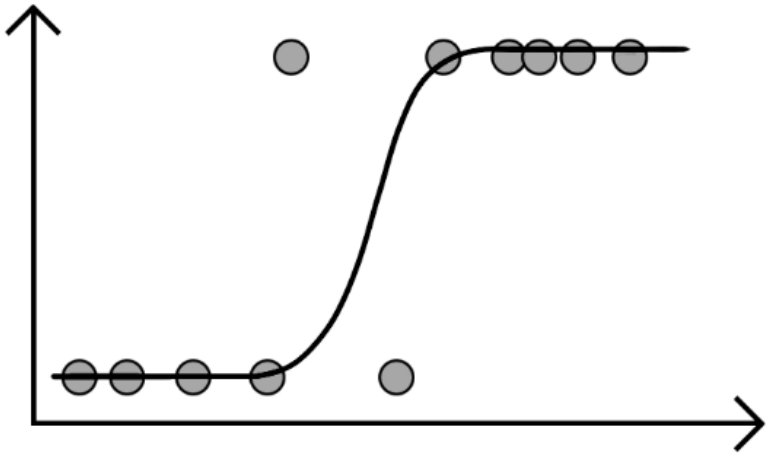
O primeiro caso pode ser tratado utilizando o método de validação cruzada, em que é testada a classificação para distâncias diferentes entre os elementos de cada classe, os dois pontos (um de cada classe) que resultarem na menor margem de erro, serão utilizados para formar os vetores de suporte e a margem. Para isso ser possível, tem-se que aceitar a possibilidade de classificações erradas em que um objeto B está muito próximo de A pois foi classificado erroneamente. O segundo problema não pode ser resolvido no mesmo hiperplano, pois, só é possível distanciar os elementos de tal forma a encontrar uma divisão elevando-os a um hiperplano de maior dimensão, para isso, utiliza-se o que é chamado de Função Kernel. Essas funções operam em hiperplanos superiores sem ter que computar as coordenadas de cada elemento neles, tornando assim a computação menos custosa. Por exemplo, em um plano unidimensional, pode-se elevar cada elemento ao quadrado para então formar uma curva possível de separar (HEARST, 1998).

1.4.3. Regressão Logística

A Regressão Logística é um modelo que tem como função classificar um grande conjunto de dados com variáveis categóricas e frequentemente binárias, por exemplo ‘sucesso’ e ‘fracasso’ (GONZALES, 2018).

A regressão logística é preferível ao cálculo de regressão linear para a classificação, pois o segundo retorna uma reta no plano de distribuição dos dados não se ajustando adequadamente a eles, além de sua variável dependente não ser categórica e sim uma variável aleatória contínua (FIGUEIRA. 2006), que pode trazer classificações erradas para o modelo. Para melhor classificar os elementos, é necessária uma função sigmoide, pois sua flexibilidade permite melhor distribuir a probabilidade de um elemento pertencer a certa classe no plano, como apresentado na figura 2.

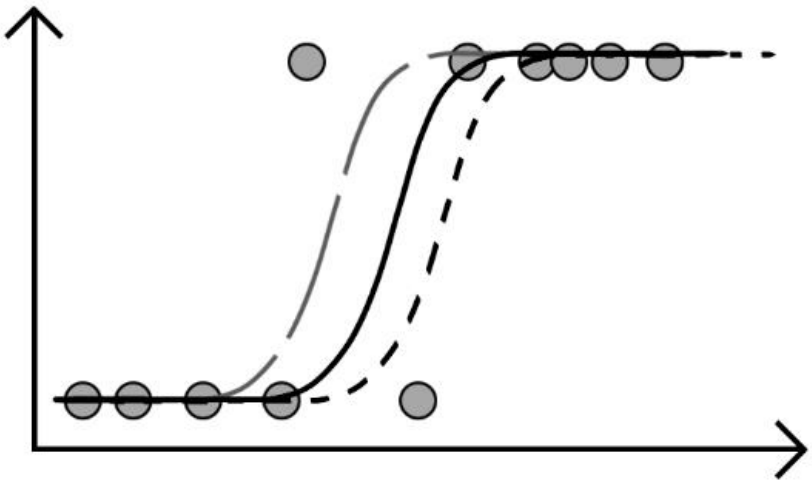
Figura 2 – Função sigmoide que se ajusta à distribuição dos elementos no plano



Fonte: Elaborada pelo autor

Diferentes curvas podem se ajustar bem aos elementos (figura 3). Regressão logística é o nome dado ao método utilizado para encontrar a curva que tenha uma melhor flexibilidade e acurácia, ou seja, obtenha a menor margem de erro na distribuição dos elementos. No plano cartesiano, como visto na figura 3, o eixo vertical varia de 0 a 1 representando as classes, e, o eixo horizontal representa o valor do dado elemento a ser classificado.

Figura 3 – Possíveis diferentes curvas que se ajustem à distribuição dos elementos no plano



Fonte: Elaborada pelo autor

1.5. Comitê de Classificadores

A abordagem de comitê consiste em utilizar um grupo de algoritmos classificadores para obter uma maior precisão no julgamento do sentimento presente no texto, em relação ao uso de apenas um algoritmo classificador (OPITZ e MACLIN, 1999). A análise é realizada individualmente pelos algoritmos previamente treinados e, em seguida, é escolhido um método para determinar a polaridade do texto com base nos resultados obtidos com os classificadores (AGUIAR, et al. 2018; KUMAR e SINGH. 2017). O método mais comumente utilizado é o de voto majoritário (*Voting*), no qual, se por exemplo, têm-se 5 classificadores, 2 resultam em 'negativo' para um texto enquanto os outros 3 resultam em 'positivo' o sentimento final será positivo. É comum a variante desse método em que se faz o uso de uma média ponderada levando em consideração a acurácia de cada classificador. Além desse método, outros dois mais utilizados são chamados de *Bagging* e *Boosting* (KUMAR e SINGH, 2017).

De acordo com Breiman (1996) o método *Bagging* consiste em dividir a base de dados de treino em subconjuntos dos dados originais, mesmo que estes dados se repitam nas '*bags*'; em seguida cada subconjunto é utilizado para treinar um

modelo em que a quantidade de modelos é igual à quantidade de subconjuntos da base de treino. Por fim, cada modelo é utilizado para classificar o conjunto e uma média é calculada (assim como em voto majoritário) para obter o resultado.

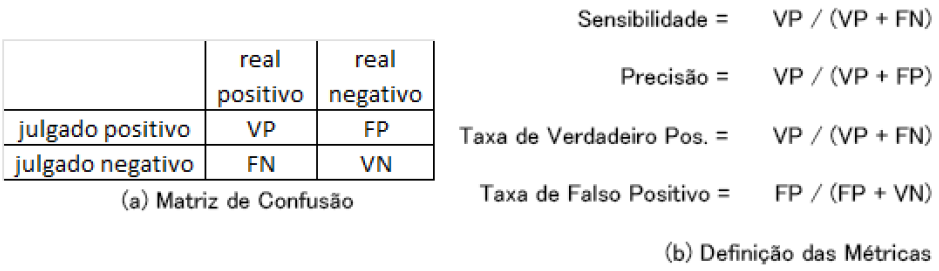
Já o método *Boosting* varia um pouco do método *Bagging*: é separado um subconjunto do conjunto original e utilizado para treinar um modelo. Em seguida, é gerado um novo subconjunto com base no desempenho do classificador anterior. No *Boosting*, exemplos julgados de forma incorreta pelo classificador anterior são escolhidos com mais frequência do que os que foram julgados corretamente. Desta forma, este método tenta desenvolver novos classificadores que são melhores em julgar os exemplos em que o comitê está falho (OPITZ e MACLIN, 1999; FREUND e SCHAPIRE, 1996).

Opitz e Maclin (1999) destacam que, enquanto o Bagging geralmente apresenta acurácia superior a um classificador individual, muitas vezes é inferior ao método *Boosting*, enquanto esse, pode gerar comitês com acurácia inferior à de um classificador individual.

1.6. Análise de Desempenho de Classificadores

Para se analisar o desempenho de cada classificador, se tratando de sua acurácia, é utilizada uma matriz de confusão, possibilitando assim que sejam calculados diversos indicadores de performance. Nessa matriz são identificados os valores que resultaram em: verdadeiro positivo (VP), falso positivo (FP), falso negativo (FN) e verdadeiro negativo (VN); com isso, é possível calcular a precisão, *f-score*, revocação entre outras medidas, como mostrado na figura 4, na qual VP é a quantidade de predições corretas para a classe ‘positivo’ assim como VN é a quantidade de predições corretas para a classe ‘negativo’, e, FP representa a quantidade de predições incorretas para ‘positivo’ enquanto FN representa as predições incorretas para ‘negativo’.

Figura 4 – Métricas comuns de avaliação de aprendizado de máquina



	real positivo	real negativo
julgado positivo	VP	FP
julgado negativo	FN	VN

(a) Matriz de Confusão

Sensibilidade =	VP / (VP + FN)
Precisão =	VP / (VP + FP)
Taxa de Verdadeiro Pos. =	VP / (VP + FN)
Taxa de Falso Positivo =	FP / (FP + VN)

(b) Definição das Métricas

Fonte: Davis e Goadrich. (2006) – Adaptado

- Além da matriz, é comum o uso do Coeficiente Kappa de Cohen introduzido em Cohen (1960) para se medir a concordância entre dois classificadores. Este coeficiente varia de 0 até 1, dessa forma, quanto maior o valor obtido, maior a concordância entre as partes.

Os valores utilizados na matriz de confusão são os resultados da fase de teste de um modelo. Há diversas maneiras de dividir o conjunto de dados em grupos de treinamento e teste dos modelos. Um método muito utilizado é chamado de *holdout* (HAN, KAMBER e PEI 2011). Nesse método o conjunto de dados é dividido em dois grupos distintos para o treinamento e teste. Uma proporção muito comum utilizada na divisão dos conjuntos é a de 2/3 dos dados para o grupo de treinamento e o 1/3 restante para o grupo de teste. Segundo Han, Kamber e Pei (2011), esse método produz uma estimativa pessimista, e consequentemente, mais realista, porque somente uma porção inicial dos dados é usada para derivar o modelo.

1.7. Trabalhos Relacionados

Este estudo foi possível graças a diversos outros trabalhos desenvolvidos sobre temas relacionados e que fazem uso da análise de sentimentos. A literatura utilizada nesta seção foi inteiramente consultada por meio do mecanismo de busca Google Acadêmico, considerando as pesquisas dos últimos quatro anos.

Trupthi, Pabboju e Narasimha (2017) objetivaram em seu estudo desenvolver um sistema interativo que informa os sentimentos de tweets postados na rede social usando a ferramenta Hadoop para processar o grande volume de dados; A

pontuação dessas palavras extraídas pelo MapReducer foram armazenados no banco de dados MongoDB. O foco principal do trabalho foi realizar a análise de sentimentos de tweets em tempo real. A análise dos tweets foi baseada no algoritmo Naïve Bayes. Para o projeto também foi necessária a utilização do conjunto de bibliotecas Natural Language Toolkit (NLTK), que realizou o processamento de linguagem natural, deixando o texto estruturado e converteu *emoticons* em palavras que os representam. Segundo os autores, o sistema desenvolvido foi capaz de buscar um tópico e informar a quantidade presente de emoções nos tweets coletados, além de exibir um gráfico de *donut* para comparação das emoções. Esse sistema também tem sua eficiência melhorada a cada consulta devido ao seu módulo de treinamento. Os tweets analisados são recentes e em tempo real, portanto os dados coletados podem ser considerados relevantes. A limitação do sistema se deu pelo algoritmo de classificação ter sido baseado em uni-gramas, isto é, palavra por palavra é analisada e não a sentença como um todo, devido a isso, a semântica deixa de ser considerada. Outra limitação é que o sistema foi designado para funcionar apenas com textos na língua inglesa.

Sailunaz e Alhajj (2019) tiveram como objetivo em seu estudo desenvolver um sistema de recomendação baseado na análise de sentimento em posts do Twitter para destacar usuários e assuntos que propagam um determinado sentimento. Para isso, foi estruturada uma base de dados dos tweets (coletados de forma randômica, localizados em um certo assunto por meio da API do Twitter e também uma ferramenta de Web Scraping) com uma rede de emoções baseada nos textos dos usuários; nesse conjunto de dados também foram coletadas as respostas desses tweets e informações dos autores dessas respostas. Para seus experimentos, foram coletados tweets sobre eventos e assuntos recentes, com uma variedade de emoções, sendo removidos coisas desnecessárias como menções, emoticons, caracteres não alfanuméricos e links. Após isso, os textos foram processados com o Processamento de Linguagem Natural (PLN). A classificação dos tweets foi realizada com o algoritmo Naïve Bayes. Os tweets foram classificados com sentimentos positivos, negativos ou neutros e as seis emoções humanas básicas segundo o modelo de emoções de Ekman que são: Raiva, Desgosto, Medo,

Alegria, Tristeza e Surpresa. As respostas dos tweets também foram classificadas com ‘concordância’ e ‘oposição’. Todas as seis emoções que os autores analisaram puderam ser caracterizadas em mais de uma dimensão, por exemplo: surpresa é uma emoção que pode estar relacionada tanto a um sentimento positivo quanto a um negativo. Também foi possível analisar, que sentimentos opostos a um tweet não indicam necessariamente discordância do tweet. O estudo mostrou ainda que, na maioria dos casos, os usuários que respondem a tweets normalmente compartilham de emoções ou sentimentos similares e concordam com o conteúdo da mensagem, embora haja casos em que é clara a discordância e oposição de sentimentos.

O estudo de Manguri, Ramadhan e Mohammed Amin (2020) teve como objetivo analisar os sentimentos presentes no Twitter relacionados ao surto do coronavírus. Os dados coletados estavam relacionados a apenas duas palavras-chaves: “COVID-19” e “coronavírus”. A extração dos dados foi realizada entre 09/04/2020 e 15/04/2020. Os dados foram extraídos do Twitter utilizando a API de pesquisa do Twitter e a linguagem de programação Python com a biblioteca Tweepy; em seguida, a análise de sentimentos foi realizada com a biblioteca TextBlob. Aproximadamente 500.000 tweets foram coletados para a análise. Ao fim do estudo, as informações foram representadas em gráficos de barras. As informações mostraram que tweets positivos (36%) superaram os tweets com sentimentos negativos (14%), enquanto a neutralidade está presente em 50% dos tweets. Outra informação apresentada, foi a de que a maior parte dos tweets foram objetivos (aproximadamente 64%) enquanto a outra parte consistia em tweets subjetivos (22%) ou sem característica clara (14%). Por fim, foi observado que a emoção predominante nos usuários estava entre ‘Calmo e contente’ e ‘Aliviado’.

Aguiar et al. (2018) propuseram um método para estimar sentimentos em redes sociais (com foco no Twitter) para a língua portuguesa. Foi utilizada a abordagem de Comitê, portanto foram utilizados no trabalho algoritmos de aprendizado de máquina para a classificação e esses foram avaliados por meio de testes estatísticos de precisão e desempenho. Os algoritmos utilizados no trabalho foram: Naive Bayes, SVM, Árvore de Decisão, Random Forest e Regressão

Logística. Os algoritmos foram treinados com uma base já rotulada de tweets, disponibilizada pelo grupo de pesquisa MiningBR, com sentimentos identificados e classificados como positivos, negativos e neutros. O pré-processamento teve duas etapas: Normalização e Transformação do Texto, na primeira foi empregada a biblioteca do Python, Natural Language Toolkit, para limpeza do texto, extração de *tokens* e remoção de *stopwords*; já na segunda o texto foi transformado em uma tabela chamada de o Bag f Words, em que cada palavra foi armazenada com seu número de ocorrências, para isso foi utilizada a biblioteca Scikit-Learn do Python. Com essa biblioteca também foi possível utilizar uma função com suporte ao Grid Search, utilizado para otimização de parâmetros que foi utilizada para selecionar bons parâmetros para o algoritmo de aprendizado de máquina realizar suas buscas. O Comitê de classificadores é utilizado de modo que a predição é feita por meio de 'votos' (considerando como peso, a acurácia dos algoritmos), prevalecendo a classificação mais presente nos algoritmos. Nos resultados do trabalho, o Comitê mostrou uma maior acurácia se comparado aos algoritmos individuais; esse conjunto de algoritmos também superou as ferramentas disponíveis para análise de texto, como o Watson da IBM e o Microsoft Text Analytics.

Souza (2019), em sua pesquisa, teve como objetivo analisar a eficiência de um comitê de classificadores para mineração de opinião focado em eleitores brasileiros. Nesse estudo as etapas de desenvolvimento da proposta foram: coleta, pré-processamento, processamento e avaliação; após isso os resultados alcançados e uma discussão sobre eles. O autor, no pré-processamento, utilizou de: tokenização, letras minúsculas, remoção de URL, *hashtags*, menções e *stopwords*; utilizando o modelo de n-gramas de tamanho 2 (essas tarefas foram realizadas com auxílio da biblioteca em Python, NLTK). Também foi calculado o peso de cada termo conforme o esquema Termo Frequência-Frequência de Documento Inverso. Os algoritmos utilizados no comitê foram: Support Vector Machine, Multinomial Naive Bayes, Passive Agressive e Regressão Logística. O corpus utilizado foram tweets no domínio de 'Política' e alcançou 90,26% de acurácia para a língua portuguesa e 77,94% para a língua inglesa. A análise de sentimentos foi realizada em nível de documento, pois os tweets, embora curtos,

podem conter mais de uma sentença. Foram selecionados alguns usuários de candidatos e *hashtags* para a extração dos tweets, excluindo da coleta os retweets. Os 2670 tweets foram anotados manualmente como sendo positivos, negativos ou neutros. Ao fim da análise, foi demonstrado a eficiência superior do uso de um comitê em comparação ao uso de apenas um classificador; também pôde ser observado que a melhor configuração para a análise foi utilizar todas as técnicas citadas anteriormente de processamento de linguagem natural, exceto a filtragem (remoção de menções, *hashtags* e links). A abordagem do comitê *Boosting* também se mostrou superior as outras abordagens experimentadas: *Voting* e *Bagging*.

O estudo de Nemes e Kiss (2020) teve como objetivo principal desenvolver e treinar um modelo de predição de sentimentos baseado em Rede Neural Recorrente: um tipo de rede neural artificial que reconhece padrões em sequências de dados, como texto. Esse modelo foi escolhido pelos autores para evitar muitos problemas da abordagem tradicional léxica. Não foi utilizado nenhum *dataset* existente, pois, segundo os autores, esses poderiam estar desatualizados. A base de textos minerada estava relacionada ao tema COVID-19 e não foi rotulada apenas em positivo ou negativo, mas foram etiquetadas também as emoções, com o objetivo de obter mais precisão e detalhes na análise realizada. Para a coleta, foi utilizada a biblioteca Tweepy. O modelo foi construído e treinado usando a biblioteca Tensorflow (Python). Uma comparação entre o modelo desenvolvido e a biblioteca TextBlob foi realizado e mostrou que os autores conseguiram atingir um de seus objetivos: reduzir ao máximo o número de sentimentos neutros (0%) enquanto o classificador padrão da biblioteca teve a maior parte de seus resultados neutros (~30%); mas ambos os classificadores mostraram resultados parecidos, se ignorando a neutralidade: a positividade foi maior do que a negatividade mesmo em tempos de pandemia.

Em seu estudo, Olenscki et al (2020) realizaram a mineração de opinião sobre o medicamento cloroquina no Twitter. A motivação para esse estudo foi devido ao medicamento ter gerado grande repercussão após declarações favoráveis feitas pelos presidentes do Brasil e dos Estados Unidos de que a cloroquina seria eficiente no tratamento da COVID19. Para esse estudo, foi utilizado

o aprendizado de máquina, e foram analisados os sentimentos de cerca de 70 mil frases. Primeiramente foram definidas palavras-chave para a coleta; em seguida os tweets foram coletados, pré-processados e alguns (2000) selecionados para o treinamento do modelo, os quais foram julgados manualmente. Embora o pré-processamento tenha como objetivo aprimorar a acurácia da análise, análises preliminares com o classificador Support Vector Classifier mostrou que não aplicar nenhum método de pré-processamento gerou uma acurácia maior (56,25%). Os dados de treino do modelo foram classificados em 3 grupos: a favor do uso da cloroquina para o tratamento do COVID19 (1), neutros ou sem opinião (0) e contra o uso do medicamento (-1). Foram utilizados diversos algoritmos classificadores no processamento e a acurácia de cada um foi calculada com o objetivo de comparação. Os resultados mostraram que, tweets a favor do uso do medicamento foram os mais presentes.

O objetivo de Coutinho e Malheiros (2020) foi utilizar a análise de sentimentos para combater a propagação de mensagens homofóbicas no Twitter escritas na língua portuguesa. A motivação deste trabalho foi o fato de que embora as redes sociais empreguem moderadores e recebam denúncias de conteúdo ofensivo, é inviável filtrar esses conteúdos manualmente, e isso acaba deixando passar muitos discursos ofensivos. Os autores coletaram mensagens que continham termos popularmente usados para referenciar a homossexuais de forma ofensiva; entretanto, a presença de uma palavra não determina o caráter homofóbico da mensagem, pois ela depende do contexto; pensando nisso, os autores optaram por utilizar um modelo de Regressão Logística para considerar apenas mensagens com sentimentos negativos e que continham termos potencialmente homofóbicos. Para comparar a análise da máquina (treinada com auxílio de uma biblioteca) com a percepção humana, questionários foram distribuídos para pessoas classificarem as mensagens como sendo ou não homofóbicas. O estudo resultou em uma acurácia de 61,48%, precisão de 66,67%, sensibilidade de 62,16% e *f-measure* de 64,33%. Os entrevistados tiveram uma concordância razoável (24,22%) segundo o coeficiente Kappa de Fleiss.

O objetivo do estudo de Souza, Souza e Meinerz (2021) foi analisar os sentimentos em tempo real de tweets publicados por veículos de notícias especializados no mercado de ações brasileiro. A motivação desse estudo foi o aumento exponencial de interessados no mercado de ações nos últimos tempos. Um classificador Naïve Bayes foi treinado com tweets previamente classificados (armazenados no Hadoop) utilizando a técnica Term Frequency – Inverse Document Frequency (TF-IDF) e o método Multinomial Naïve Bayes; foi utilizada pelos autores a biblioteca Machine Learning Library da plataforma Apache Spark. Após essas fases iniciais, novos tweets foram coletados em tempo real, específicos dos usuários escolhidos (por meio de seus Twitter IDs) para a análise. Os resultados foram exibidos por meio do Jupyter Notebook; por fim, o classificador obteve uma acurácia de 76,8%.

Com o levantamento dos trabalhos relacionados foi possível identificar ferramentas em comum entre eles que puderam contribuir ou até mesmo foram cruciais para o desenvolvimento desta pesquisa. Destaca-se ainda que cada autor fez uso de um ou mais algoritmos de classificação além de avaliar o desempenho do modelo. Alguns tópicos também foram identificados que puderam auxiliar na decisão das técnicas utilizadas para se elaborar o modelo de classificação usado neste estudo. As principais características de cada trabalho estão distribuídas no Quadro 1.

Quadro 1 – Trabalhos Relacionados

Autor (Ano)	Objetivo	Classificadores Utilizados	Destaques	Ferramentas Utilizadas
Sailunaz, e Alhajj (2019)	Desenvolver um sistema de recomendação de usuários e assuntos propagadores de sentimentos	Naïve Bayes	A emoção surpresa pode pertencer a sentimento positivo e negativo. Tweets e suas respostas podem ter sentimentos opostos mesmo em concordância.	Twitter API, Web Scraping
Trupthi, Pabboju e Narasimha. (2017)	Desenvolver um sistema interativo para identificação de sentimentos no Twitter com base em uma palavra-chave	Naïve Bayes	Melhoria do sistema a cada consulta nova devido ao aprendizado de máquina. Limitação por ser baseado em uni-gramas.	Twitter API, Hadoop, MongoDB, Natural Language Toolkit
Manguri, Ramadhan, e Mohammed Amin (2020)	Realizar a análise de sentimentos sobre a pandemia do coronavírus	Naïve Bayes	Tweets divididos em objetivos e subjetivos. Sentimento neutro predominante. Emoção mais presente entre 'Contentamento' e 'Aliviado'.	Twitter API, Tweepy, TextBlob
Aguiar, *et al*. (2018)	Propor um método de análise de sentimentos de tweets para a língua portuguesa com abordagem de comitê	Naïve Bayes, *Support Vector Machine*, Árvore de Decisão, *Random Forest*, Regressão Logística	Acurácia da abordagem de comitê superior a qualquer classificador utilizado individualmente e comparado as ferramentas IBM Watson e Microsoft Text Analytics.	Twitter API, base de tweets do grupo MiningBR, Natural Language Toolkit, Scikit-Learn
Souza. (2019)	Analisar o desempenho da abordagem de comitê para a mineração de opinião	*Multinomial Naïve Bayes*, *Support Vector Machine*, *Passive Aggressive*, Regressão Logística	Abordagem do comitê *Boosting* superior as abordagens *Voting* e *Bagging.* Melhor configuração não inclui remoção de menções, *hashtags* e links.	Twitter API, Natural Language Toolkit

Nemes, e Kiss. (2020)	Desenvolver, treinar e analisar o desempenho de um modelo de análise de sentimentos baseado em rede neural recorrente	Rede Neural Recorrente	Rotulação de emoções (não apenas positivo ou negativo). Redução ao máximo de resultados neutros.	Twitter API, Tweepy, Tensorflow
Pessanha, *et al*. (2020)	Analisar os sentimentos presentes no Twitter relacionados as *hashtags* covid19, fiqueemcasa e relacionadas	Abordagem Léxica Não Supervisionada	O sentimento mais presente foi negativo.	Twitter API, LexiconPT, SentiLex-PT02
Olenscki, *et al*. (2020)	Analisar a opinião dos usuários do Twitter a respeito do uso da cloroquina no combate ao coronavírus	*Support Vector Machine*, Regressão Logística, *Stochastic Gradient Descent*, *Maximum Entropy*, *Multinomial Naïve Bayes*	A maior parte dos tweets foram a favor do uso do medicamento.	Twitter API
Coutinho e Malheiros (2020)	Utilizar a analisar o desempenho da análise de sentimentos para detectar mensagens potencialmente homofóbicas em português no Twitter	Regressão Logística	Acurácia de 61,48% para identificar mensagens homofóbicas. Baixa concordância entre os entrevistados para pré-processamento.	Twitter API, Scikit-learn
Souza, Souza e Meinerz (2021)	Realizar a análise de sentimentos em tempo real de tweets publicados por veículos de notícias especializados no mercado de ações brasileiro	*Multinomial Naïve Bayes*	Análise de sentimento realizada em tempo real.	Twitter API, Hadoop, Apache Spark Machine Learning Library, Jupyter Notebook

2. Metodologia

2.1. Natureza da Pesquisa

Nesta pesquisa, optou-se pela abordagem experimental, que consiste em determinar um objeto de estudo, selecionar as variáveis que seriam capazes de exercer influência sobre o objeto, definir as formas de controle e de observação dos efeitos produzidos no objeto por essas variáveis (GIL, 2007).

2.2. Variáveis de Controle

As variáveis de controle consideradas nesta pesquisa foram:

- Base de dados utilizada;
- Seleção dos classificadores para o comitê;
- Técnicas utilizadas no pré-processamento;
- Método de divisão da base de dados para treino e teste;

2.3. Ferramentas Utilizadas

As ferramentas utilizadas neste experimento para as etapas de coleta, pré-processamento, construção do modelo de classificação e exibição são as descritas a seguir:

Twitter API[1] – API disponibilizada pela própria rede social para recuperar e analisar dados, bem como interagir em conversas no Twitter. A partir dela é possível obter dados de: tweets, usuários, mensagens diretas, listas, trends, mídias e localizações.

Python 3[2] – Python é uma linguagem de programação de fácil escrita e entendimento, e foi escolhida para este trabalho devido à grande variedade de bibliotecas feitas pela comunidade.

Tweepy[3] – Uma biblioteca para a linguagem Python que torna fácil a conexão com a API do Twitter para recuperar dados.

1 Disponível em <https://developer.twitter.com/en/docs/twitter-api/getting-started/about-twitter-api>, acesso em 7 de novembro de 2021.
2 Disponível em <https://www.python.org/>, acesso em 7 de novembro de 2021.
3 Disponível em <https://www.tweepy.org/>, acesso em 10 de novembro de 2021.

Scikit Learn[4] – Biblioteca Python com várias funcionalidades voltadas para aprendizado de máquina, como algoritmos classificadores.

PySimpleGUI[5] – Biblioteca para a linguagem Python para desenvolvimento de interfaces gráficas para dispositivos desktop de forma simples.

MongoDB[6] – Banco de dados orientado a objetos, que não precisa de uma estrutura construída previamente, dando a possibilidade de armazenar objetos no padrão JSON.

Pickle[7] – Biblioteca que permite salvar objetos instanciados na memória em um arquivo para posterior reutilização.

2.4. Coleta dos tweets

Os tweets (postagens feitas pelos usuários da rede social Twitter) são disponibilizados para coleta por meio de uma API (Application Programming Interface) oferecida pela própria plataforma, sendo necessário registrar-se e solicitar uma autorização para utilizá-la. Como a linguagem de programação utilizada foi Python3, foi usada a biblioteca da linguagem chamada Tweepy, que se conecta a essa API para receber os dados (que serão armazenados em uma tabela).

Para a coleta foi desenvolvida uma pequena aplicação (alocada no repositório do projeto) para busca de tweets com base em palavras-chave inseridas pelo usuário. Alguns termos como ‘economia’, ‘atendimento’, ‘reclamação’ e ‘qualidade’ foram utilizados para encontrar tweets relacionados ao propósito do trabalho. Tweets mais genéricos ou fora de tópico também foram coletados para obter uma maior variedade de palavras na base de dados. No total, foram coletados 400 tweets.

2.5. Anotação Manual

Os tweets coletados foram analisados pelo autor após a coleta, um por um, sendo julgados como contendo um sentimento positivo ou negativo, e, dessa forma cada tweet recebeu um identificador (em uma coluna da tabela de tweets) de sua

4 Disponível em <https://scikit-learn.org/stable/>, acesso em 10 de novembro de 2021.
5 Disponível em <https://pysimplegui.readthedocs.io/en/latest/>, acesso em 10 de novembro de 2021.
6 Disponível em <https://www.mongodb.com/pt-br>, acesso em 10 de novembro de 2021.
7 Disponível em <https://docs.python.org/3/library/pickle.html>, acesso em 19 de maio de 2022.

respectiva classe. Posteriormente, esses dados já classificados foram utilizados para a construção do modelo de classificação.

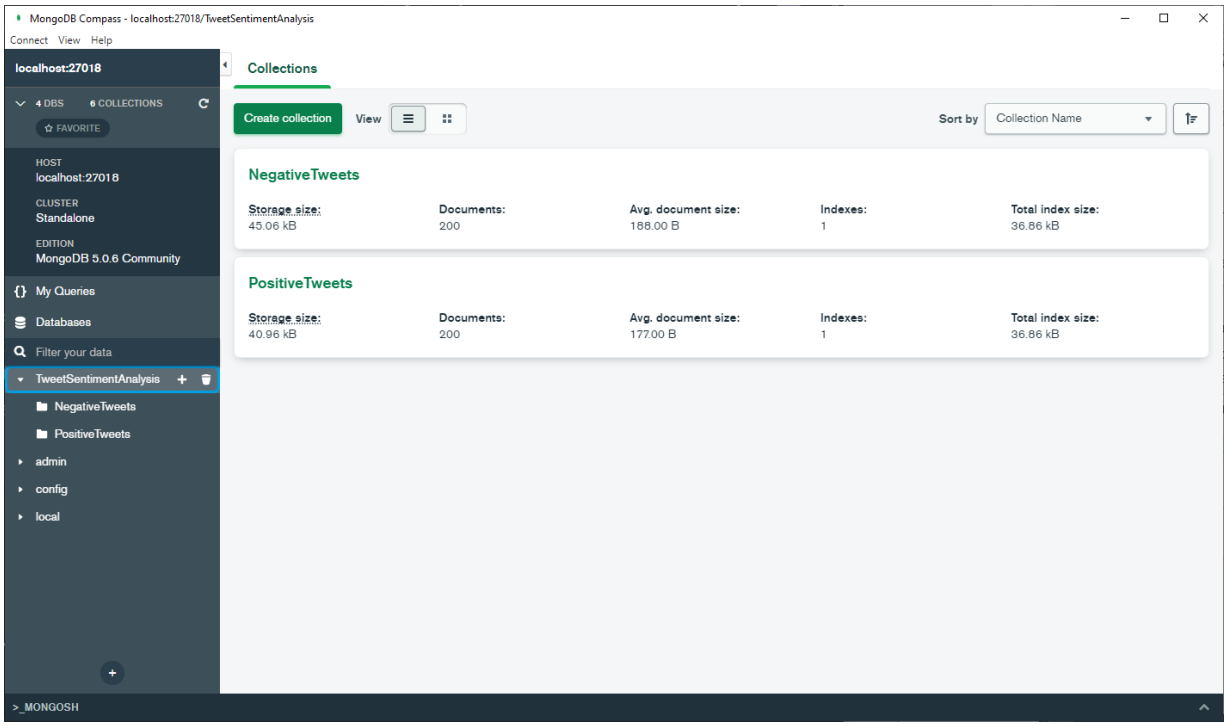
Exemplo de tweets classificados manualmente:

“Essa série é maravilhosa, eu estou impressionada!” – julgado pelo autor como positivo.

“Acho que esse foi o pior filme que eu já vi…” – julgado pelo autor como negativo.

Os tweets anotados foram armazenados em suas respectivas coleções (agrupamento de objetos em bancos não relacionais) no banco de dados como mostra a figura 5; 200 tweets foram classificados como positivos e 200 foram classificados como negativo, formando assim uma base de dados balanceada.

Figura 5 – Coleções de tweets no banco de dados MongoDB.



Fonte: Gerada pela ferramenta MongoDB Compass

2.6. Pré-Processamento

O pré-processamento é o conjunto de técnicas utilizadas antes do treinamento dos algoritmos, com o objetivo de deixar a base de dados mais limpa e eficaz para classificação, como por exemplo: removendo hiperlinks, caracteres especiais, resolvendo abreviações etc. Para esta etapa, foram utilizadas algumas

funções disponibilizadas pela biblioteca Natural Language Toolkit (NLTK) além de técnicas desenvolvidas manualmente para eliminar do texto elementos não benéficos para a análise; essas técnicas são apresentadas nas subseções a seguir.

2.6.1. Tokenização

A Tokenização é a técnica de "quebrar" um texto em pedaços pequenos, geralmente, as palavras que ele contém. Por exemplo, em um texto como "Eu gosto de ir ao museu!", a tokenização gera uma lista de palavras como a seguinte:

["Eu", "gosto", "de", "ir", "ao", "museu!"]

Dessa forma é possível calcular a probabilidade particular de cada palavra pertencer a uma das classes analisadas. Para esta etapa foi utilizada a função word_tokenize disponível na biblioteca NLTK. Contudo ainda foram necessários alguns passos para se preparar esses *tokens* para o processamento conforme apresentados nas seções de 2.6.2 a 2.6.4.

2.6.2. Filtragem e Padronização

Após a tokenização, foram descartados todos os *tokens* dispensáveis para a análise, isto é, *hashtags* (*tokens* que começam com o caractere #), menções (*tokens* que começam com o caractere @) e hiperlinks (*tokens* iniciados por 'www', 'http', 'https' etc.). Como os seguintes exemplos:

Hashtag:

Original: ["Eu", "estou", "empolgada", "#GameAwards", "#2021"]

Filtrado: ["Eu", "estou", "empolgada"]

Menção:

Original: ["Você", "é", "incrível", "@Angelo00,", "adorei", "a", "matéria!"]

Filtrado: ["Você", "é", "incrível", "adorei", "a", "matéria!"]

Hiperlinks:

Original: ["Que", "reportagem", "ridícula", "https://www.reportagem.abc"]

Filtrado: ["Que", "reportagem", "ridícula"]

Em seguida, com a lista de palavras contendo apenas os tokens relevantes, todas as letras foram convertidas para maiúsculas para maior praticidade no posterior processamento:

Original: [“Que”, “reportagem”, “ridícula”]

Padronizado: [“QUE”, “REPORTAGEM”, “RIDÍCULA”]

2.6.3. Remoção de Palavras Irrelevantes (*Stop Words*)

Para tornar mais otimizada e diminuir a tendência à neutralidade da extração do sentimento, foi utilizada uma técnica conhecida como remoção de *stopwords*, o que significa remover do texto palavras comuns e sem valor sentimental, como por exemplo: “de”, “para”, “e”, “é” etc.

Para isso foi necessária uma lista contendo palavras que se encaixam nesta definição específica da língua portuguesa e que é disponibilizada pela biblioteca NLTK removendo assim, todas as palavras dos tweets que estivessem presentes nesta lista.

2.6.4. Processo de *Stemming*

Para que os algoritmos considerem derivações de palavras possuindo o mesmo significado foi utilizada a técnica de *Stemming*, que consiste em reduzir uma palavra derivada para a sua raiz. Por exemplo, duas frases com palavras diferentes, mas derivadas da mesma raiz terão o mesmo valor:

Original: “Eu gosto de corridas”

Após a redução: “Eu gost de corr”

Original: “Eu gostei de correr”

Após a redução: “Eu gost de corr”

2.6.5. Modelo Bigrama

O modelo bigrama considera durante a análise um conjunto de duas palavras em vez de uma, por exemplo, na seguinte frase classificada com sentimento negativo: “Eu não gosto de você” o algoritmo analisaria cada token da seguinte forma:

[“Eu não”, “não gosto”, “gosto de”, “de você”]

O benefício da utilização desta técnica é que o algoritmo não identificará a palavra “gosto” como negativa, mas sim a combinação “não gosto” o que pode ser útil na identificação de negações.

2.7. Processamento

Na etapa de processamento se iniciou a construção e aprendizado dos algoritmos para futuras classificações. Os algoritmos utilizados neste experimento foram: Naïve Bayes, Support Vector Machine, e, Regressão Logística.

O algoritmo Naïve Bayes foi desenvolvido pelo próprio autor a título de aprendizado, enquanto os demais foram importados como funções da biblioteca Scikit-Learn.

2.8. Comitê de Classificadores

Com os algoritmos escolhidos anteriormente e já desenvolvidos ou importados de bibliotecas, foi criado um programa na linguagem Python3 para servir como um ponto de encontro dos classificadores, o próprio comitê, permitindo a distribuição da base de treinamento, classificação e decisão da polaridade final do texto.

O método implementado no programa para decidir a polaridade final do texto foi o método Voting, apresentado na Revisão da Literatura, capítulo 1, no qual é predominante o julgamento que mais resultou das classificações.

2.9. Treinamento dos Classificadores

Para o treinamento dos classificadores foi utilizada uma porção de 66,66% (neste caso 267) dos tweets obtidos na fase de coleta enquanto o restante (133 tweets, ou 33,33% da base de dados) foi reservado para teste, como especificado pelo método *holdout*, apresentado na seção 1.9. Para os classificadores SVM e Regressão Logística, foi utilizada a medida estatística TF-IDF (Termo Frequência-Frequência de Documento Inverso) por meio de uma funcionalidade disponibilizada pela biblioteca Scikit-Learn. Após treinado, cada modelo é salvo em um arquivo na pasta raiz da aplicação para posterior uso sem necessidade de repetir esta etapa, poupando assim tempo significativo em caso de grande volume de dados na base de treino e teste.

2.10. Avaliação

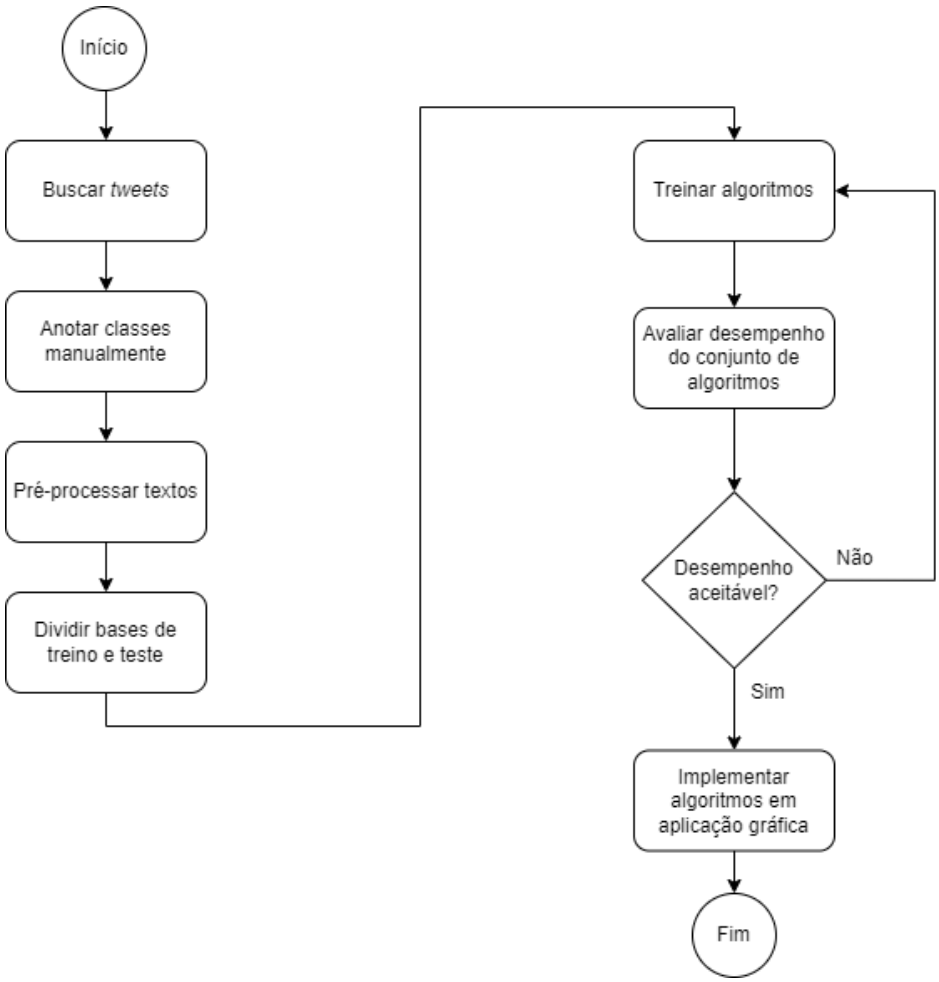
A avaliação foi realizada calculando a acurácia por meio dos acertos, erros e quantidade total de avaliações, com base em classificações feitas para a porção da base de tweets reservada para teste, que corresponde a 33,33% dos tweets coletados (neste caso 133). Cada algoritmo foi avaliado individualmente, além disso, o próprio comitê (conjunto de classificadores) teve sua avaliação. Os modelos foram avaliados 10 vezes para alcançar uma acurácia média visto que a acurácia pode apresentar variações dependendo da seleção do conjunto de treino e teste.

O fluxo que sumariza as tarefas necessárias para o desenvolvimento deste experimento está representado na próxima seção.

2.11. Fluxo de Trabalho

O fluxo de trabalho, apresentado na Figura 6, descreve os passos do desenvolvimento do comitê tendo como sua fase inicial a busca dos textos para compor a base de dados para treino e teste ao mesmo tempo em que a segunda etapa já está em ação, na qual os sentimentos dos textos são anotados. Em seguida os textos sofrem um tratamento inicial para que então a base de dados seja dividida e utilizada para treinar e testar os algoritmos. As etapas finais descrevem um processo iterativo de treino até que o desempenho seja aceitável antes do modelo ser implementado de fato em uma aplicação gráfica.

Figura 6 – Fluxo de trabalho do estudo



Fonte: Elaborada pelo autor

3. Resultados e Discussão

Neste experimento foi utilizada uma base de dados de tweets balanceada, visto que enquanto a base permanecia desbalanceada, as classificações pendiam para a polaridade com maior quantidade de dados para treino. A base contém 400 tweets, sendo 200 classificados como positivos e 200 classificados como negativos, ambos anotados manualmente pelo autor.

3.1. Resultados Obtidos

Durante o desenvolvimento deste experimento, foram testadas combinações diferentes de técnicas para o pré-processamento a fim de obter o maior nível de acurácia. A versão final do comitê tem seu pré-processamento composto pelas técnicas de: remoção de *stopwords*, *stemming*, remoção de caracteres não alfanuméricos; remoção de hiperlinks, *hashtags* e menções.

A acurácia para cada classificador e o comitê (modelo composto pelos 3 classificadores), para uma base de dados contendo 400 tweets classificados como positivos ou negativos foi a seguinte:

- Support Vector Machine: 82,19%
- Regressão Logística: 81,28%
- Naïve Bayes: 79,16%
- Comitê: 96,21%

A tabela 1 apresenta os resultados obtidos de cálculos realizados a partir das matrizes de confusão geradas para cada modelo.

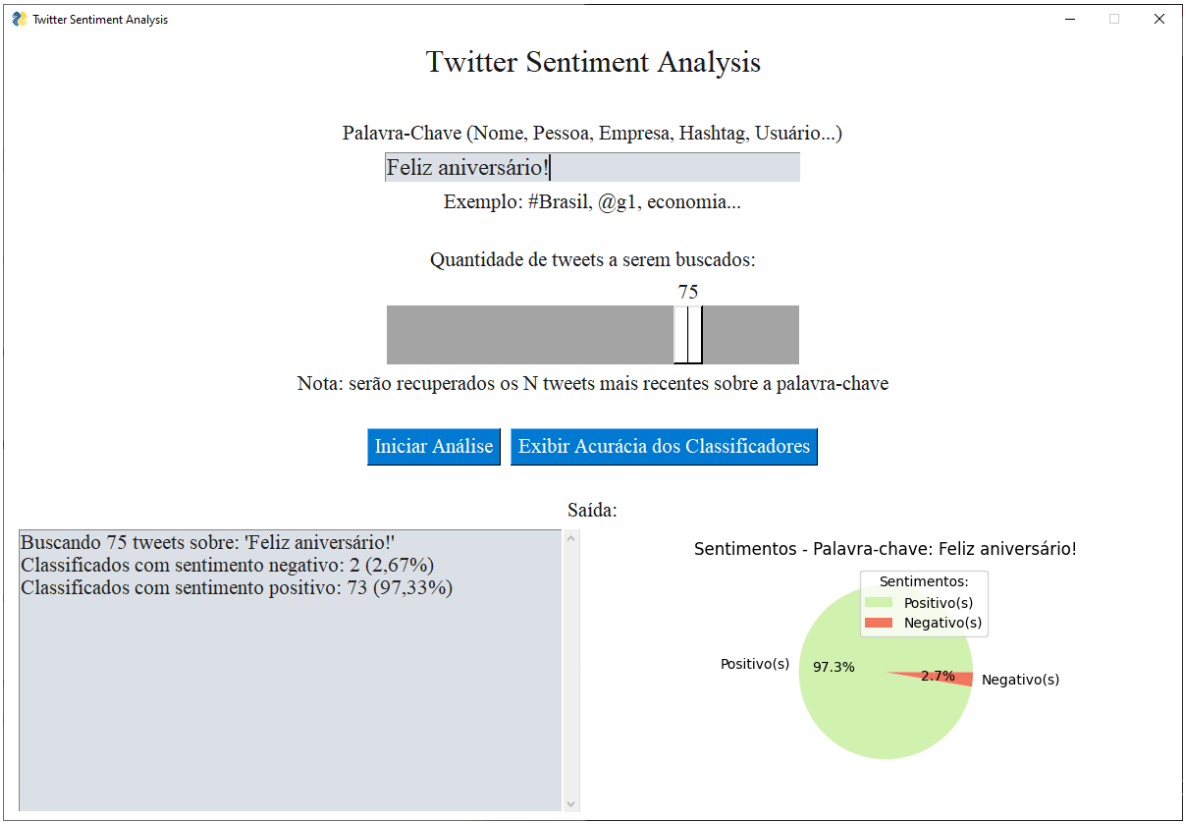
Tabela 1 – Resultados gerados pelos modelos de classificação

Classificador	Acurácia	Precisão (Positivo)	Precisão (Negativo)	Sensibilidade (Positivo)	Sensibilidade (Negativo)
SVM	82,19%	88,40%	84,12%	85,91%	86,88%
Regressão Logística	81,28%	87,09%	65,71%	69,23%	85,18%
Naïve Bayes	79,16%	79,36%	78,26%	76,92%	80,59%
Comitê	96,21%	96,96%	96,96%	96,96%	96,96%

Fonte: Elaborada pelo autor

Os classificadores foram integrados a uma aplicação gráfica que permite buscar palavras-chaves no Twitter, possibilitando ao usuário escolher um intervalo entre 1 e 100 tweets, ao final exibindo os resultados (junto a um gráfico) referentes a quantidade dos tweets classificados como positivos e a quantidade dos tweets classificados como negativos. A aplicação final é apresentada na figura 7.

Figura 7 – Captura da aplicação final



Fonte: Elaborada pelo autor

O repositório deste trabalho se encontra armazenado no GitHub[8] acompanhado de uma descrição de como utilizar a aplicação. Devido a políticas do Twitter, não é permitido que tweets sejam armazenados e expostos publicamente, desta forma, é necessário que o utilizador da aplicação construa ou importe sua própria base de dados para treino e teste dos classificadores.

3.2. Discussão

Algumas particularidades puderam ser notadas no desenvolvimento do comitê. Durante a coleta, alguns tweets inseridos na base de dados apresentavam múltipla polaridade da mesma forma como foi mencionado na seção 1.1. no tópico de Análise a Nível de Aspecto; esses tweets geralmente tratavam de resenhas de

8 Repositório disponível em <https://github.com/AlexisCesar/tweet-sentiment-analysis>.

produtos ou serviços; alguns tweets desse gênero foram removidos da base de dados para não desequilibrar a polaridade de cada coleção no banco de dados.

Outro ponto foi que, mesmo durante a coleta manual, pôde-se notar que a maioria dos tweets, em diversos assuntos, têm a polaridade negativa predominante. Este fato somado a base de dados desbalanceada proporcionava resultados que praticamente excluíam a polaridade positiva. Porém depois do balanceamento da base de dados, as classificações ficaram mais equilibradas.

Foi possível perceber também o grande impacto da fase de pré-processamento e suas técnicas, uma vez que os classificadores apresentavam uma acurácia em torno de 60% antes da aplicação das medidas de preparação do texto.

Em Souza (2019) a abordagem *Voting*, apresentou uma acurácia de 59,10% para uma base de dados de tweets em português, balanceada, ao passo que neste trabalho, a mesma abordagem resultou em uma acurácia superior de 96,21%.

Em Aguiar (2018), o Comitê de Classificadores foi composto por cinco modelos de classificação, com uma acurácia de 86,5%. A sua base de dados para treino e teste foi muito maior que a deste trabalho (2516 tweets) porém apresentou acurácia inferior se comparada ao modelo gerado neste experimento, diferença que pode estar ligada ao uso de uma base de dados desbalanceada ou com ruídos, no caso de Aguiar (2018).

Em Olenscki (2020), a acurácia para o classificador SVM foi de 56,25% e para o classificador Regressão Logística de 55,26%, o que é uma diferença considerável se comparado a acurácia destes mesmos modelos no presente trabalho: 82,19% para SVM e 81,28% para Regressão Logística.

Em Souza, Souza e Meinerz (2021), o classificador utilizado foi Naïve Bayes e obteve uma acurácia de 76,8%, ficando bem próximo a deste trabalho, 79,16% para o algoritmo Naïve Bayes. As técnicas de pré-processamento utilizadas em ambos trabalhos podem explicar a acurácia próxima.

4. Conclusão

Este trabalho que teve como problema de pesquisa “Quais os algoritmos de mineração de opinião possuem melhor acurácia no julgamento de textos obtidos por meio do Twitter?”, chega ao seu final com as seguintes considerações:

Foi desenvolvido um modelo de classificação composto por três classificadores, sendo implementado em uma aplicação gráfica. O comitê de classificadores utilizou da abordagem de votação (*Voting*) para decidir a classificação final de um texto. Para o treino e teste, alguns tweets foram coletados e anotados manualmente como sendo positivos ou negativos, sendo gravados posteriormente em um banco de dados as palavras e suas respectivas probabilidades (calculadas pelo algoritmo) de pertencer a cada classe.

Foi possível perceber a grande influência que a fase de pré-processamento exerce na classificação. Os resultados, foram bastante satisfatórios, mostrando que o comitê com uma acurácia de 96,21% superou os todos os três classificadores analisados individualmente.

Algumas limitações e dificuldades encontradas no decorrer do desenvolvimento deste experimento foram:

- Quantidade elevada de tweets neutros: ao realizar a coleta dos tweets, a maior parte dos tweets encontrados não possuíam sentimento, sendo esses tweets apenas mensagens informativas como notícias ou propagandas.
- *Emojis*: muitos usuários do Twitter adotam o uso de *emojis* para expressar o sentimento do tweet, e isto é algo que pode dificultar a análise quando se trata de ironia. Por exemplo, uma tendência atual nas mensagens das redes sociais entre os jovens é inserir um emoji de palhaço que pode indicar ironia quando se trata de um texto escrito positivamente ou um enfatizador quando se trata de um texto negativo.
- Enorme variação de gírias e abreviações: durante a coleta, foi possível perceber que em vários tweets os autores utilizaram diversas gírias e muitas abreviações de palavras, além do forte estrangeirismo. O linguajar utilizado em redes sociais, que sofre constante mudanças, pode degenerar um modelo de

classificação se esse não dispor de uma grande quantidade de dados para treino, e ainda, se esses dados não estiverem atualizados.

Apesar das limitações encontradas, os resultados dos experimentos tornam possível concluir que a hipótese inicial pôde ser confirmada e que os objetivos deste trabalho foram alcançados.

4.1. Trabalhos Futuros

Visto que grande parte dos trabalhos relacionados à mineração de opinião envolvem a rede social Twitter, especialmente as análises de comitês de classificadores, experimentos do uso de comitês para outras redes sociais bem como outros meios de tráfego de informação em forma de texto agregariam grande valor aos estudos sobre análise de sentimentos.

Estudos de métodos mais eficazes e menos exaustivos para a fase de coleta dos tweets devem ser introduzidos visando a facilitar a construção de grandes bases de dados para modelos de classificações mais abrangentes e precisos. Outro possível trabalho futuro é a adição de um módulo de autotreinamento do classificador, de forma que seja possível aumentar automaticamente sua base de dados.

A identificação do sentimento de um texto considerando *emoticons* e *emojis* bem como sua influência no aspecto de contradição ou ênfase do sentimento classificado é um tópico que deve ser trabalhado para aumentar a assertividade dos classificadores em redes sociais, principalmente no Twitter.

Referências

AGUIAR, E. J. *et al*. **Análise de Sentimento em Redes Sociais para a Língua Portuguesa Utilizando Algoritmos de Classificação**. Centro de Ciências Tecnológicas, Universidade Estadual do Norte do Paraná, Bandeirantes, 2018.

BREIMAN, L. **Bagging Predictors**. Departamento de Estatística, Universidade da Califórnia. Berkele, 1996.

CHOWDHURY, G. G. **Natural language processing**. Annual Review of Information Science and Technology, 2005.

COHEN, J. **A Coefficient of Agreement for Nominal Scales**. Educational and Psychological Measurement, 1960.

CORTES, C; Vapnik, V. **Support-vector networks**. Machine Learning, 1995.

COUTINHO, V; MALHEIROS, Y. **Detecção de Mensagens Homofóbicas em Português no Twitter usando Análise de Sentimentos**. Brazilian Workshop On Social Network Analysis and Mining (BRASNAM). Cuiabá, 2020.

DAVIS, J; GOADRICH, M. **The relationship between Precision-Recall and ROC curves**. Proceedings of the 23rd International Conference on Machine Learning, 2006.

FELDMAN, R. **Techniques and applications for sentiment analysis**. Communications of the ACM, 2013.

FELL, E; LUKIANOVA, N. **Twitter (Digital Media and Society)**. European Journal of Communcation, 2019.

FIGUEIRA, C. V. **Modelos de Regressão Logística**. Instituto de Matemática. Universidade Federal do Rio Grande do Sul. Porto Alegre, 2006.

FREUND, Y; SCHAPIRE, R. E. **Experiments with a New Boosting Algorithm**. Machine Learning: Proceedings of the Thirteenth International Conference. Murray Hill, 1996.

GIL, A. C. **Métodos e Técnicas de Pesquisa Social**. 6. ed. São Paulo: Atlas, 2008.

GONZALES, L. A. **Regressão Logística e suas Aplicações**. Monografia – Curso de Ciência da Computação. Universidade Federal do Maranhão. São Luís, 2018.

HAN J; KAMBER, M; PEI, J. **Data Mining:** Concepts and Techniques. 3. ed. San Francisco: Elsevier, 2011.

HEARST, M. **Support Vector Machines**. Intelligent Systems and their Applications, IEEE 13. Berkeley, 1998.

KUMAR, S. SINGH, R. **Comparative analysis of ensemble classifiers for sentiment analysis and opinion mining**. 2017 3rd International Conference on Advances in Computing, Communication and Automation. 2017.

LIDDY, E**. Enhanced text retrieval using natural language processing**. Bulletin of the American Society for Information Science. v. 24. 1998.

LIU, B. **Sentiment Analysis and Subjectivity**. Department of Computer Science. University of Illinois. 2. ed. Chicago, 2010.

LOVINS, J. B. **Development of a Stemming Algorithm**. Electronic Systems Laboratory. Massachusetts Institute of Technology. v. 11. Cambridge, 1968.

MANGURI, K. H; RAMADHAN, R. N; AMIN, P. R. M. **Twitter Sentiment Analysis on Worldwide COVID-19 Outbreaks**. Kurdistan Journal of Applied Research. v. 5. 2020.

MARTÍNEZ-CÁMARA, E. *et al*. **Sentiment analysis in Twitter**. Natural Language Engineering. Cambridge University. 2012.

NEMES, L. KISS, A. **Social media sentiment analysis based on COVID-19**. Journal of Information and Telecommunication. 2020.

OLENSCKI, J. *et al*. **Aplicação de análise de sentimentos no Twitter para avaliação da percepção pública quanto a cloroquina**. Simpósio Brasileiro de Computação Aplicada à Saúde. Porto Alegre: Sociedade Brasileira de Computação, 2020.

OPITZ, D; MACLIN, R. **Popular Ensemble Methods: An Empirical Study**. Journal of Artificial Intelligence Research. v. 11. 1999.

PANG, B; LEE, L. **Opinion Mining and Sentiment Analysis**. Foundations and Trends in Information Retrieval. v. 2. 2008.

PESSANHA, G. *et al*. **#FIQUEEMCASA: Análise de sentimento dos usuários do Twitter em relação ao COVID19**. HOLOS, v. 5. 2020.

RUSSEL, N; NORVIG, P. **Inteligência Artificial**. 3. ed. Editora Campus, 2013.

SAILUNAZ, K; ALHAJJ, R. **Emotion and sentiment analysis from Twitter text**. Journal of Computational Science, v. 36. 2019.

SARLAN, A; NADAM, C; BASRI, S. **Twitter Sentiment Analysis**. Computer Information Science. Universiti Teknologi PETRONAS. Perak, 2014.

SILVA, C; RIBEIRO, B. **The Importance of Stop Word Removal on Recall Values in Text Categorization**. Computer Science. Proceedings of the International Joint Conference on Neural Networks. Coimbra, 2003.

SOUZA, G. **Comitê de Classificador para Mineração de Opinião de Eleitores Brasileiros**. Bacharelado em Sistemas de Informação. Universidade Federal Rural de Pernambuco. Serra Talhada, 2019.

SOUZA, V. A; SOUZA, E. F; MEINERZ, G. V. **Análise de Sentimento em Tempo Real de Notícias do Mercado de Ações**. Brazilian Journal of Development. Curitiba 2021.

STEHMAN, S. V. **Selecting and Interpreting Measures of Thematic Classification Accuracy**. Remote Sensing of Environment. Editora Elsevier. Syracuse, 1997.

TEIXEIRA, L. A. **Métodos de Regressão para Aprendizado por Reforço**. Monografia (Especialização) – Curso de Ciência da Computação. Universidade Federal de Juiz de Fora. Juiz de Fora, 2016.

TRUPTHI, M; PABBOJU, S; NARASIMHA, G. **Sentiment Analysis On Twitter Using Streaming API**. 2017 IEEE 7th International Advance Computing Conference. Telangana, 2017.

TSYTSARAU, M; PALPANAS, T. **Survey on mining subjective data on the web**. Data Mining and Knowledge Discovery. Trento, 2011.

WEBSTER, J; KIT, C. **Tokenization as the initial phase in NLP**. Proceedings of the 14th conference on Computational linguistics. v. 4. City Polytechnic of Hong Kong. Kowloon, 1992.

ZHANG, X.-D. **A Matrix Algebra Approach to Artificial Intelligence**. 1. ed. Springer Singapore, 2020. p. 223.