Utilizando Análise de Sentimentos e SVM na Classificação de Tweets Depressivos

Omar Andres Carmona Cortes
Wesley Eduardo de Oliveira Melo
omar@ifma.edu.br
wesley.eduardo@acad.ifma.edu.br
Institute Federal de Educação, Ciência e Tecnologia do Maranhão (IFMA)
Departamento de Computação (DComp) – Bacharelado em Sistemas de Informação
São Luis, MA, Brasil

ABSTRACT

The number of depression cases has grown worldwide. The World Health Organization estimates that 5.8% of the Brazilian population already present depression symptoms. In the world, 4.8% of the entire population has presented some symptoms. These data are alarming because they represent about 12 million people only in Brazil and 368 million worldwide. Therefore, it is essential to build applications that adequately identify the population's feelings about depression to drive public health policies. Appropriate policies can save money on public health and keep people active. Thus, this work investigates how to apply machine learning in classifying depression posts on Tweeter. The data were extracted from the social media network, reaching a total of 31.177 tweets classified as depressive and non-depressive. The application was implemented in Python with Pandas and SciKit Learning. Results have shown that SVM overcomes the Naive Bayes algorithm and can reach an accuracy of 94%, precision of 91%, a recall of 91%, and an F1 Score of 91%.

KEYWORDS

Aprendizagem de Máquina, SVM, Depressão, Análise de Sentimentos

1 INTRODUÇÃO

A ideia da análise de sentimentos é determinar o conteúdo emocional de um texto através do uso do computador [1], sendo que para realizar essa transformação de um texto para um sentimento é necessário a utilização de uma área conhecida como Processamento de Linguagem Natural (PLN). Na verdade, a análise de sentimentos foi inicialmente definida como uma sub-tarefa de PLN [2] e depois estendida para uma tarefa de aprendizagem de máquina.

Uma das formas mais comuns de se fazer essa análise é através da contagem de palavras e os sentimentos associados a elas. Assim, através de uma tabela de sentimentos, na qual cada palavra tem associada a ela um valor, é possível determinar a proporção de palavras negativas e positivas sobre um determinado assunto e fornecer uma pontuação geral sobre o sentimento.

O ponto fraco da contagem de palavras é que a pontuação associada a cada palavra deve ser definida por um especialista humano, ou seja, pode estar enviesada. Além disso, alguns desafios serão encontrados pela frente na análise de sentimentos como os apresentados a seguir:

- Identificar quais palavras tem um sentimento positivo e quais tem um sentimento negativo, sendo uma o oposto da outra. Além disso, palavras neutras também devem ser identificadas;
- Identificar sentenças com palavras que parecem indicar um sentimento, mas que não indicam sentimento algum. Como por exemplo, "É bom praticar mais de uma hora de exercícios por dia para combater a diabetes?". Embora a palavra "bom" tenha um cunho positivo, nessa sentença a palavra não deve fazer essa indicação já que é uma pergunta sobre o assunto;
- Identificar sentimentos em sentenças irônicas. É um caso parecido ao anterior, porém a palavra com sentido positivo quer dizer exatamente o contrário. Como por exemplo,"Que carro bom! Quebrou em dois dias.";
- Identificar sentimentos em metáforas. Por exemplo, "dormiu tanto que formou um vale no colchão.". É extremamente difícil identificar sentimentos em frases contendo sentenças desse tipo.

Algoritmos tradicionais de aprendizagem de máquina podem lidar com alguns desses aspectos, mas eles podem falhar em sua análise. Nesse contexto, algoritmos podem ser testados e analisados para identificar qual deles podem superar esses desafios de forma eficaz. Sendo assim, este trabalho investiga a utilização do algoritmo *Support Vector Machine* (SVM) para identificar postagem depressivas em redes sociais, mais especificamente no Tweeter. Também será utilizado o algoritmo chamado de *Naive Bayes* (NB) para formar uma base de comparação, sendo que a mesma será realizada usando métricas tradicionais da aprendizagem de máquina, tais como precisão, acurácia, sensibilidade e *F1 Score*.

A escolha pela depressão se dá pois o número de brasileiros diagnosticados com depressão alcançou níveis preocupantes. A OMS estima que mais de 12 milhões de pessoas apresentam sintomas da doença somente no Brasil. No mundo esse número também é alarmante, pois mais de 368 milhões de pessoas apresentam algum tipo de sintoma de depressão. Outro ponto preocupante é que a depressão não tem idade, atacando desde crianças até aposentados, prejudicando os mais diversos setores econômicos e familiares. Um dado interessante, mas também assustador, é que estima-se que em torno 5% das pessoas até 18 anos já tenham tido algum tipo de sintoma [3], que se tratado adequadamente não será propagado

pelo restante da vida adulta. Vale destacar que tratar-se cedo pode causar um impacto positivo na economia, pois não irá afetar as pessoas economicamente ativas.

Em relação a população mundial, como já mencionado, em torno de 368 milhões de pessoas tem ou já teve algum tipo sintoma, ou seja, é uma epidemia mundial [4] que não pode ser jogada para debaixo do tapete. Nesse âmbito, pode se dizer que quanto mais formas de lidar, identificar e tratar a doença, melhor seja para o Brasil ou para o mundo.

Em termos econômicos um estudo aponta que o crescimento da doença entre 1990 e 2013 é de 50% [5]. Em termos de tratamento, o relatório acusa, ainda, que a cada US$ 1 investido no tratamento da depressão ou ansiedade, o retorno é de US$ 4. Por outro lado, o custo do tratamento em 36 países (ricos, pobres e emergentes) entre o período de 2016 e 2030, considerando psicoterapia e medicamento alcançará a espetacular cifra de US$ 147 bilhões [6]. Além disso, segundo o mesmo estudo, qualquer investimento em tratamento compensa, pois com apenas 5% de retorno na força de trabalho pode representar até US$ 400 de volta no mercado.

Dessa forma, este artigo esta dividido da seguinte maneira: a Seção 2 apresenta os trabalhos correlatos; a Seção 3 mostra os conceitos básicos de análise de sentimentos e SVM; a Seção 4 apresenta como a base de dados foi obtida, a configuração dos experimentos e seus resultados; finalmente, a Seção 5 ilustra as conclusões do trabalho e os trabalhos futuros.

2 TRABALHOS CORRELATOS

Como já mencionado, a análise de sentimentos tem com objetivo identificar o conteúdo emocional de um texto. Sua aplicação começou a ficar evidente na última década e cresceu rapidamente desde 2004 [7]. No entanto, um dos principais desafios no momento é que grande parte dessas pesquisas foram desenvolvidas para língua inglesa [8], acredita-se que isso tenha sido influenciado pelo grau de maturidade que as ferramentas de PLN tem no referido idioma. No entanto, sua aplicação em outras línguas também tem sido identificado, como por exemplo, Coreano [9], Tailandês [10] e Árabe [11], dentre outros.

Em inglês, Zachini [12] investigou a análise de sentimentos no idioma coletando dados do Tweeter, utilizando como *string* de busca os termos *"anxiety, depression e mental health"*, visando encontrar relatos de depressão. A partir disto, o autor obteve 3574 tweets depressivos para treinamento. Em seguida, para completar a outra classe, os *tweets não-depressivos*, foram coletados de um dataset muito utilizado chamado *Sentiment140* [13], outros 3574 tweets, totalizando 7148 tweets divididos entre a classe depressiva e não depressiva. O autor também utilizou técnicas de processamento de linguagem natural (PNL) que ajudaram no processo de padronização dos textos, como por exemplo, a remoção de palavras vazias e a lematização (técnica otimizada para o idioma inglês). Para realizar o processo de classificação de tweets, utilizou-se o modelo de aprendizagem de máquina SVM, obtendo-se uma acurácia otimizada de 99.7%.

Corrêa [14] desenvolveu um trabalho de análise de sentimentos em relação aos filmes indicados ao Oscar 2017. A construção do projeto teve como fonte de dados a extração de *tweets* no idioma inglês. O autor tinha como um dos objetivos predizer qual filme seria o grande vencedor da premiação e quais estariam entre os menos prestigiados. A *string* de busca utiliza para extração de tweets nesse contexto teve como base os títulos dos filmes concorrentes a esse prêmio, são eles: "arrival", "fences", "hacksaw ridge", "hell or high water", "hidden figures", "la la land", "lion", "manchester by the sea" e "moonlight". Após o processo de extração inicial, o autor optou por um processo de rotulação manual de *tweets* divididos em três classes, positivos, negativos e neutros. Obteve-se como resultado uma base rotulada com 1.444 tweets para a classe positiva, 1.362 tweets para a classe negativa e 429 para a classe neutra, somando um total de 3.235. Com a realizações de técnicas de Processamento de Linguagem Natural e a utilização do classificador Naive Bayes, o autor obteve uma acurácia de 74.1%

Já em português, Nascimento [15] realizou um estudo de caso sobre a rede social do YouTube para a extração de comentários sobre vídeos relacionados à depressão no idioma português do Brasil. A extração dos dados foi realizada através da API do YouTube, utilizando como *string* de busca os termos *"depressao + suicidio"*, que resultou no total de 26.908 vídeos relacionados. Desse total foram coletados 312.055 comentários, porém a versão final do *dataset* utilizada para o experimento teve um total de 718 comentários divididos entre a classe com indícios de depressão e sem indício de depressão. Outro aspecto relevante que levou o autor a diminuir o tamanho do *dataset* foi a remoção de comentários usando critérios de balanceamento tal como a quantidade de caracteres abaixo de 100. Para a análise dos comentários utilizaram-se os algoritmos de classificação NB, MNB, LMT, J48 e SVM. Dos algoritmos utilizados, o modelo SVM teve o melhor desempenho chegando a um valor de 80.4% para a média *F1 Score*.

Pereira [16] buscou obter informações sobre popularidade dos candidatos à presidência da república do Brasil 2018 com base em publicações presentes na rede social Tweeter durante o período eleitoral. O trabalho desse autor tem uma importância significativa no que diz respeito a comparação de outros trabalhos realizados também no idioma português do Brasil. O processo de extração de *tweets* teve com string de buscas termos relacionados aos nomes dos candidatos, como Jair Bolsonaro, Lula, Guilherme Boulos e dentre outros. O processo de extração resultou em um *dataset* com 1.014.752 *tweets*, onde desse total foram utilizados na etapa de treinamento apenas 18.000 por questões de balanceamento e outros critérios do autor. Assim como em outros trabalhos no contexto de análise de sentimento, utilizou-se também técnicas de Processamento de Linguagem Natural. Os algoritmos de Naive Bayes e SVM foram os escolhidos como modelo de classificação de *tweets*. O modelo SVM foi superior em termos de acurácia, obtendo-se o valor de 86% no processo de análise da popularidade de um determinado candidato.

Nenhum dos trabalho citados utilizou uma quantidade considerável de tweets no processo de treinamento dos algoritmos de classificação. Isso pode gerar falhas em algumas classificações de tweets, ainda que os resultado das métricas cheguem próximos dos 100%. O *dataset* utilizado neste trabalho contém 31.177 tweets, divididos entre a classe depressiva e não depressiva. Um dos maiores objetivos do aumento do *dataset* é tentar uma melhoria nos resultados em termos de métricas como acurácia e *F1 Score* que superem

os 90% no idioma português, visto que geralmente nessa língua há dificuldades de classificações por conta de suas particularidades.

Além disso, não é o objetivo deste trabalho indicar um diagnóstico sobre depressão, pois isso só pode ser feito por um profissional habilitado. O objetivo é mostrar que através da análise de sentimentos usando aprendizagem de máquina é possível identificar tweets depressivos, desde que eles existam na base de dados. Nesse contexto, este trabalho pode servir como auxiliar no processo de diagnóstico para um profissional da área.

3 ANÁLISE DE SENTIMENTOS

A análise de sentimentos, também chamada de mineração de opiniões, tem como objetivo minerar ou até mesmo entender o conteúdo emocional de um texto ou trecho de texto. Assim, uma palavra pode ter conotação positiva ou negativa, ou até mesmo caracterizar outros tipos de sentimentos como, por exemplo, surpresa ou nojo. Segundo Feldman [17], a análise de sentimentos é definida como a tarefa de encontrar opiniões sobre entidade ou assuntos específicos.

Uma forma de se analisar o sentimento de um texto é considera-lo como uma combinação de palavras individuais e o sentimento contido no texto é a soma de todos os sentimentos expressados. Essa é uma forma comum de se tratar a análise de sentimentos, porém, não a mais eficiente. Nesse caso, pode-se recorrer a algoritmos de aprendizagem de máquina para esse fim.

A aprendizagem de máquina é uma subárea da ciência da computação que evoluiu do estudo de reconhecimento de padrões e da teoria do aprendizado computacional em inteligência artificial. Em 1959, Arthur Samuel conceitualizou a aprendizagem de máquina como o "campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados" [18]. Dentre esses algoritmos podem ser citados os já mencionados SVM [19] e o Naive Bayes [20].

3.1 SVM

SVM ou Máquinas de Vetor de Suporte é um conjunto de métodos de aprendizado supervisionado utilizados tanto para classificação quanto para regressão. Neste trabalho, o SVM será utilizado como método de classificação.

Um modelo SVM é uma representação de exemplos como pontos no espaço, mapeados de maneira que os exemplos de cada classe sejam divididos por um espaço claro que seja tão amplo quanto possível, como mostrado na Figura 1, na qual se observa a linha que divide as duas classes e as linhas pontilhadas denominadas de vetor de suporte [21]. Essa divisão é chamada de margem (m), sendo que o objetivo da SVM é maximizar m, ou seja, maximizar a distância entre os hiperplanos do vetor de suporte.

Assim, a ideia no SVM é identificar um hiperplano $h(x) = w.x+b$, no qual $w \,.\, x$ é o produto escalar entre os vetores w e x e $w \in X$ é o vetor normal ao hiperplano. No espaço euclidiano w representa o coeficiente angular da reta, sendo que a reta principal é dada por $w.x+b = 0$ e as marges por $w.x+b = -1$ e $w.x+b = 1$, para as linhas inferior e superior, respectivamente. O valor de b é computado a partir da média de todos os vetores de suporte possíveis. Detalhes sobre como obter b podem ser vistos no trabalho de [22]. Nesse contexto, um classificador pode ser construído usando a Equação 1.

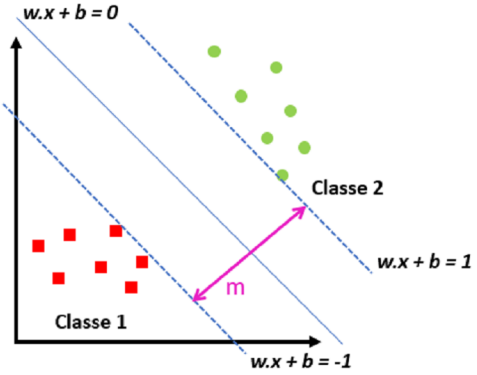

Figura 1: Classificador linear SVM

$$g(x) = sgn(h(x)) = \begin{cases} +1, \text{se } w \,.\, x + b > 0 \\ -1, \text{se } w \,.\, x + b < 0 \end{cases} \quad (1)$$

Usando manipulações algébricas, a margem m de separação entre objetos pode ser obtida pela minimização de $||w||$ [23]. Quando não é permitido que haja dados de treinamento entre as margens, essa SVM é dita com margens rígidas. Caso o hiperplano não seja suficiente para separar as classes, então pode-se usar técnicas de aumento de dimensionalidade, assim o que não pode ser separado em 2 dimensões por uma reta, pode dependendo dos dados ser separável em 3 dimensões, e assim por diante [21].

As SVMs são robustas diante de dados de grandes dimensões sobre os quais outras técnicas de aprendizado obtêm classificadores super ou sub ajustados [22]. Outra vantagem em sua utilização é que existe somente uma configuração ótima para a SVM em seu conjunto de treinamento. Além disso, o uso de funções Kernel na não-linearização das SVMs torna o algoritmo eficiente, pois permite a construção de simples hiperplanos em um espaço de alta dimensão de forma tratável do ponto de vista computacional [24]. Por outro lado, a principal limitação das SVMs está na sensibilidade a escolha de seus parâmetros, pois pode levar a modelos imprecisos.

3.2 Naive Bayes

Naive Bayes é um classificador popular nos anos 90 que foi bastante utilizado como filtro de *spam*. A ideia do algoritmo é utilizar o teorema de Bayes, apresentado na Equação 2, para determinar a qual classe pertence uma observação, tupla ou registro [21].

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (2)$$

como $P(x_1, x_2, ..., x_m|c_k) = P(x_1|c_k) \times P(x_2|c_k) \times ... \times P(x_n|c_k)$, pode-se generalizar que $P(c_k|x_1, x_2, ..., x_n|) \cong P(x_1|c_k) \times P(x_2|c_k) \times ... \times P(x_n|c_k) \times P(c_k)$, na qual x_i é um preditor (campo da base de dados) e c_k representa a classe sendo avaliada. A saída do classificar é dada pela Equação 3, também chamada de *Maximum a Posteriori*, que indica a qual classe a instância irá pertencer, ou seja, pertencerá àquela de maior probabilidade.

$$y = argmax_k \; P(c_k) \times \prod P(x_i|c_k) \quad (3)$$

De acordo com [25], o Naive Bayes apresenta as seguintes vantagens: (i) possui uma base matemática sólida; (ii) é rápido; e (iii) apesar de simples, apresenta de modo geral um bom desempenho em tarefas de classificação. Por esse motivo, este algoritmo é utilizado muitas vezes como base de comparação com outros algoritmos de classificação.

4 EXPERIMENTOS

4.1 Construção da Base de Dados

A base de dados é o principal recurso para a realização do treinamento de algoritmos de classificação. Vale ressaltar que no contexto de aprendizagem de máquina, quanto maior for o *dataset*, em conjunto com uma boa qualidade do conteúdo, melhores serão os resultados alcançados. Assim, a Figura 2 representa a sequencia de procedimentos utilizados para a construção do *dataset*.

O processo de construção do *dataset* se inicia com o cadastro e solicitação de acesso à API do Tweeter. Além de dados de contatos, também são solicitados conteúdos textuais que informem detalhadamente as intenções que levaram a essa solicitação. O acesso através de API não é restrito, mas não pode ser usado para inferir dados discriminatórios tais como, gênero, etnia, orientação política ou religiosa, dentre outros. Ao final desta etapa e com a requisição aprovada, as chaves de acesso podem ser utilizadas para de fato usar os recursos da API.

Com o acesso liberado é necessário definir os termos das *strings* de busca, cujos objetivos são extrair dados tanto para a classe *depressiva* quanto para a classe *não depressiva*. A *string* de busca para a classe *depressiva* é formada pelos seguintes termos: *"depressão, ansiedade, angústia, saúde mental e suicídio"*. Para a classe *não depressiva* diversos termos foram utilizados durante o processo de extração, como por exemplo, *bem estar, cinema, viagem, futebol, filmes, família*, dentre outros diversos termos que não configuram um estado emocional depressivo. Em seguida foi implementado o código que realiza o processo de extração dos tweets em conformidade com as *strings* de busca.

A etapa seguinte é a definição de qual banco de dados será utilizado para o armazenamento dos tweets extraídos. Escolheu-se o MongoDB Atlas como ferramenta de banco de dados NoSql, que corresponde a estruturas não unicamente relacionais. Outro ponto a ser considerado é o fato de que o MongoDB tem uma estrutura orientada a documentos, ideal para armazenamento no formato de textos, além de ser também compatível com o formato JSON, o mesmo utilizado como retorno de dados extraídos utilizando a API do Tweeter.

O processo de extração e armazenamentos de tweets teve inicio no mês de maio de 2020 e se estendeu até mês de novembro de 2020. Nesse período foram coletados **15.747** tweets da classe *depressiva* e **15.430** tweets da classe *não depressiva*, somando um total de **31.177** tweets. A Figura 3 resume os dados obtidos de acordo com as *strings* de busca.

Além das informações textuais de cada postagem, também foram recuperados outros dados como o *id* do tweet, data de postagem, quantidade de caracteres, localização e informações sobre a autoria das postagens, de acordo com as normas de privacidade da rede social. Essas informações adicionais podem ser utilizadas em outros experimentos de várias formas, como por exemplo a extração de tweets tendo como parâmetro um usuário, o que abre espaço para uma análise de sentimentos com base em perfis específicos.

Finalmente, implementou-se o código para recuperar os dados armazenados no *dataset* e transferi-lo para um arquivo CSV contendo apenas o texto do tweet e sua respectiva rotulação que indicasse a qual classe pertence, depressiva ou não depressiva. Dessa forma, é possível utilizar alguns recursos da biblioteca Pandas, que é utilizada também para o armazenamento dos dados em uma estrutura de dados conveniente chamada de *Data Frame* [26], que é composto por duas ou mais series, que equivalem a colunas ou atributos. Além disso, a utilização e importação de um arquivo CSV representa um ganho maior na velocidade de acesso aos dados por parte dos algoritmos de classificação, já que não é necessário acessar a nuvem e a biblioteca Pandas possui diversas funções para manipulação de arquivos.

4.2 Pré-Processamento

O ponto de partida para utilizar os algoritmos de aprendizagem de máquina SVM e Naive Bayes é a criação do *dataset* que foi apresentado na seção anterior. No entanto, outro ponto que precisa ser levado em conta é tentar obter um balanceamento entre as classes disponíveis na base, ou seja, é interessante que não haja um diferença muito grande em termos de quantidade entre classes, pois isso pode levar à necessidade de se utilizar técnicas de balanceamento. O que neste trabalho foi evitado.

No contexto de qualidade da base de dados, o ideal seria que houvesse uma revisão por parte de especialistas na área, como psiquiatras, psicólogos e/ou profissionais relacionados. Essas revisões seriam importante para evitar equívocos na rotulação dos tweets, pois rotulações erradas podem atrapalhar principalmente no processo de análise de um tweet jamais visto pelo modelo de classificação.

O primeiro passo do pré-processamento é realizar a leitura do arquivo CSV que foi criado. Em seguida são criado dois vetores chamados *tweets* e *classes*, onde o vetor *tweets* recebe somente as informações dos textos e o vetor *classes* apenas as classificações correspondentes a cada tweet. A ligação do tweet e sua classificação mesmo em vetores separados ocorre através do índice que é mantido. Essa separação servirá para a realização da etapa de treinamento dos algoritmos de classificação, pois a função que é utilizada requer essas informações de forma separada. A Figura 4 mostra a sequencia de passos utilizado nessa etapa.

A maioria dos passos executados são provenientes da área de Processamento de Linguagem Natural (PLN) e foram utilizados através da biblioteca NLTK [27], cujo o objetivo é padronizar textos modificando ou removendo conteúdos irrelevantes para o modelo de classificação. Outra vantagem do pré-processamento é a diminuição do tamanho do texto e consequentemente do tempo de processamento destes dados.

Nesse contexto, as primeiras operações de pré-processamento são a remoção de pontuações, *hashtags*, menções de usuários, endereços URL, palavras que não começam com letras de A até Z, palavras vazias (*stop words*) e palavras adicionais sem relevância. As palavra vazias não adicionam sentido ao texto, pois são usadas para dar coesão e contexto, mas não fazem sentido específico

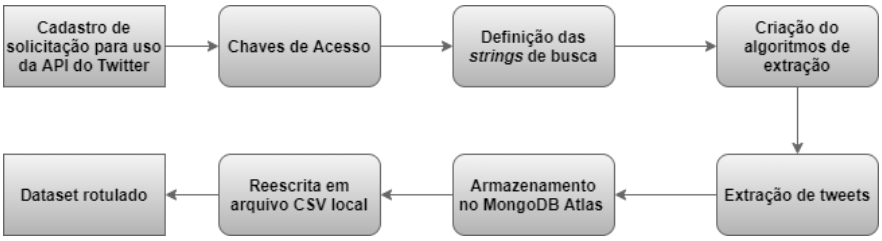

Figura 2: Procedimentos para a criação do *Dataset*

Idioma		Português do Brasil		
String de busca	**Tweets depressivos**	"depressão, ansiedade, angústia, saúde mental e suicídio"		
	Tweets não depressivos	"bem estar, cinema, viagem, futebol, filmes, família, outros"		
Dataset		Tweets Depressivos	Tweets não Depressivos	Total
		15.747	15.430	31.177

Figura 3: *Dataset*

quando olhadas individualmente, como por exemplo, as preposições, artigos e pronomes, dentre outras.

Para modificação do conteúdo textual é aplicado uma técnica chamada de *stemização*, que consiste na redução de palavras flexionadas ou derivadas à sua base, transformando palavras próximas em uma só [28]. Um exemplo da aplicação é a redução das palavras "estudar, estudou, estudo e estudando"ao termo "estud", que representa a base de todas as variações mencionadas.

Aplicou-se em seguida um método importante chamado de vetorização, visto que dependendo do algoritmo, os algoritmos de aprendizagem de máquina podem não estão receber entradas de texto. Essa técnica permite que os textos dos tweets sejam convertidos em valores numéricos, tornando possível que os dados se organizassem em um modelo matemático que irá generalizar os problemas a serem classificados [12]. O *CountVectorizer* é a função que implementa essa tarefa e fornece uma maneira simples de *tokenizar* uma coleção de documentos de texto e criar um vocabulário de palavras conhecidas.

O último passo antes da utilização dos algoritmos de aprendizagem de máquina é realizar uma etapa de separação dos dados para que o modelo possa ser testado e validado. Neste trabalho utilizou-se a técnica de re-amostragem conhecida como validação cruzada através da função *cross_val_predict*, presente na biblioteca *sklearn*.

4.3 Configuração do Experimento

O computador utilizado em condições exclusivas para a realização dos experimentos possuí as seguintes especificações:

- **Modelo do Computador:** Dell Inspiron I15-5566-a30p
- **Processador:** Intel Core i5 7200u 2.5 GHz
- **Memória RAM:** 8 GB DDR4 2400 MHz
- **Armazenamento:** SSD Kington A400 240GB
- **Sistema Operacional:** Windows 10 Pro x64 versão 1909
- **Ambiente de Desenvolvimento:** PyCharm Professional
- **Versão do Python:** 3.8

A Figura 5 representa o fluxo utilizado no experimento, desde a fase de entrada dos dados até a etapa de geração das métricas. Os resultados referentes à matriz de confusão e as métricas serão apresentados Seção 4.5.

O experimento é iniciado através da leitura do arquivo CSV local com o conteúdo do *dataset*. Utilizou-se também uma arquitetura baseados em modelos com Pipelines [29], cujo objetivo é ganhar simplicidade e flexibilidade, reduzindo códigos e automatizando fluxos. Em seguida, com os dados separados em treinamento e teste, executaram-se os algoritmos de aprendizagem de máquina Naive Bayes e SVM nos dados de treinamento.

Em seguida, utilizou-se a técnica de validação cruzada mencionada previamente. Essa validação consiste em separar os dados de treino e teste em várias partes distintas, cuja representação pode ser descrita na imagem 6, na qual pode-se observar que em cada *fold* um conjunto diferente de dados (sem intersecção entre eles) é testado. Assim, no final a base inteira acaba por ser testada.

A função *cross_val_predict* recebeu como parâmetro o modelo de classificação, os dados de treinamento, teste e um parâmetro k que se refere ao número de grupos em que um *dataset* irá ser repartido, sendo utilizados os valores $k = 5$ e $k = 10$. Para que seja possível quantificar os resultados da validação cruzada, algumas métricas foram utilizadas, como a acurácia, precisão, sensibilidade e F1 Score.

4.4 Métricas

Antes de especificar o cálculo das métricas é importante apresentar o conceito de matriz de confusão, que nada mais é do que uma tabela que mostra as frequências de classificação para cada classe do modelo [30]. A Figura 7 ilustra um exemplo de uma matriz de confusão na forma genérica, na qual faz-se a relação entre o que esta na base de dados e o que foi classificado pelo algoritmo.

A relação entre o real e a predição é baseado em quatro possíveis resultados elencados a seguir:

- **Verdadeiro positivo (VP):** Ocorre quando a classe que se está buscando foi prevista corretamente.
- **Falso positivo (FP):** Ocorre quando a classe que se está buscando prever foi prevista como verdadeira incorretamente.
- **Verdadeiro Negativo (VN):** Ocorre quando a classe que se está buscando foi prevista como negativa de forma correta.
- **Falso Negativo (FN):** Ocorre quando a classe que se esta buscando foi prevista como negativa incorretamente.

A seguir apresenta-se o cálculo das métricas acurácia, precisão, sensibilidade e F1 Score, nas Equações 4, 5, 6 e 7, respectivamente.

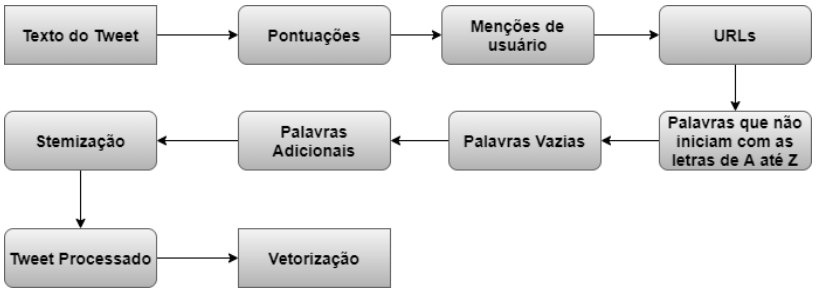

Figura 4: Pré-Processamento de Tweets

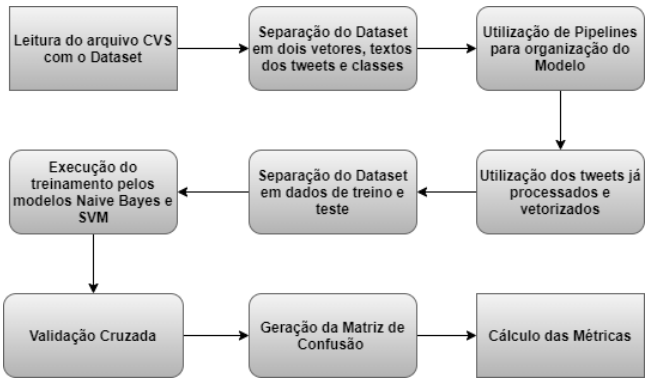

Figura 5: Fluxo do Experimento

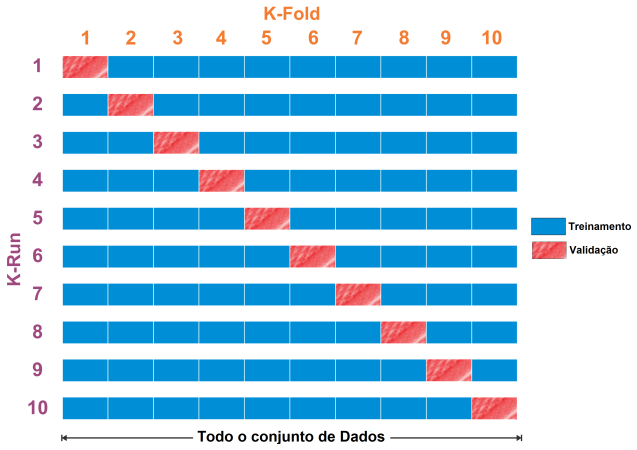

Figura 6: Método de validação cruzada com k = 10
Fonte: Google Imagens

$$acuracia = \frac{VP + VN}{VP + FN + VN + FP} \quad (4)$$

$$precisao = \frac{VP}{VP + FP} \quad (5)$$

$$sensibilidade = \frac{VP}{VP + FN} \quad (6)$$

		Predito	
		Classe Positiva	Classe Negativa
Real	Classe Positiva	**VP** Verdadeiro Positivo	**FN** Falso Negativo
	Classe Negativa	**FP** Falso Positivo	**VN** Verdadeiro Negativo

Figura 7: Matriz de Confusão Genérica

$$f1\_score = \frac{2\ x\ precisao\ x\ sensibilidade}{precisao + sensibilidade} \quad (7)$$

4.5 Resultados

Os resultados foram gerados com a execução dos já mencionados classificadores Naive Bayes e SVM. Executou-se o primeiro experimento com um sub-conjunto de 5 mil tweets e o segundo com a base completa contendo 31.177. Outro parâmetro utilizado nos experimentos esta relacionado com o modelo de validação cruzada, usando o valor$k = 5$ e $k = 10$ em cada experimento, respectivamente.

Das Figuras 8 a 15 são exibidos os resultados obtidos para cada um dos modelos, contendo a matriz de confusão e em seguida as métricas. É mostrado também informações sobre o parâmetro k utilizado na validação cruzada e os totais de tweets em cada experimento. Os resultados referentes às métricas estão de acordo com a distribuição apresentada em cada matriz de confusão.

K = 5	Classe Depressiva	Classe não Depressiva	Totais
Classe Depressiva	2.100	400	2500
Classe não Depressiva	967	1.533	2500
Totais	**3.067**	**1.933**	**5000**

Acurácia	Precisão	Recall	F1_Score
0.73	**0.74**	**0.73**	**0.72**

Figura 8: Experimento 1 | 5.000 tweets | k = 5
Modelo: Naive Bayes

Para o modelo SVM, além das métricas já utilizadas, utilizou-se também um método de otimização chamado Grade de Busca *(Grid-searching)*. Esse método irá construir um modelo para cada combinação possível de parâmetros que lhe foi dado e será feito uma validação cruzada para cada modelo. Utilizou-se os parâmetros C,

K = 10	Classe Depressiva	Classe não Depressiva	Totais
Classe Depressiva	2205	295	2500
Classe não Depressiva	911	1.589	2500
Totais	3.116	1.884	5000

Acurácia	Precisão	Recall	F1_Score
0.76	0.78	0.76	0.76

Figura 9: Experimento 1 | 5.000 tweets | k = 10
Modelo: Naive Bayes

K = 5	Classe Depressiva	Classe não Depressiva	Totais
Classe Depressiva	14.824	923	15.747
Classe não Depressiva	4.564	10.866	15.430
Totais	19.388	11.789	31.177

Acurácia	Precisão	Recall	F1_Score
0.82	0.84	0.82	0.82

Figura 10: Experimento 1 | 31.177 tweets | k = 5
Modelo: Naive Bayes

K = 10	Classe Depressiva	Classe não Depressiva	Totais
Classe Depressiva	14.801	946	15.747
Classe não Depressiva	4.061	10.369	15.430
Totais	18.862	12.315	31.177

Acurácia	Precisão	Recall	F1_Score
0.84	0.85	0.84	0.84

Figura 11: Experimento 1 | 31.177 tweets | k = 10
Modelo: Naive Bayes

kernel e *gamma*. A constante *C* representa uma regulação, o kernel pode ser definido como linear, poly e rbf. O parâmetro *gamma* é o coeficiente para o parâmetro kernel. Além dos valores definidos pelo *kernel*, utilizou-se no parâmetro *C* os valores 1.5, 10, 100, 1000 e para *gamma* os valores 1e-7, 1e-4, 1e-2 e 0.1.

A pesquisa de grade é uma técnica de ajuste que tenta calcular os valores ideais de hiperparâmetros. Cada valor presente nesses parâmetros testa todas as combinações entre eles. A função da Grade de Busca consegue avaliar a performance de acerto para cada modelo resultante de cada combinação e no final determina qual foi a melhor combinação dos parâmetros [12].

K = 5	Classe Depressiva	Classe não Depressiva	Totais
Classe Depressiva	2.022	478	2.500
Classe não Depressiva	402	2.098	2.500
Totais	2.424	2.576	5.000

Acurácia	Precisão	Recall	F1_Score
0.82	0.82	0.82	0.82

Acurácia Otimizada	Melhor Combinação de Parâmetros
0.86	{'C': 1.5, 'gamma': 0.01, 'kernel': 'rbf'}

Figura 12: Experimento 2 | 5.000 tweets | k = 5
Modelo: SVM

K = 10	Classe Depressiva	Classe não Depressiva	Totais
Classe Depressiva	2.054	446	2500
Classe não Depressiva	355	2.145	2500
Totais	2.409	2.591	5000

Acurácia	Precisão	Recall	F1_Score
0.84	0.84	0.84	0.84

Acurácia Otimizada	Melhor Combinação de Parâmetros
0.87	{'C': 1.5, 'gamma': 0.01, 'kernel': 'rbf'}

Figura 13: Experimento 2 | 5.000 tweets | k = 10
Modelo: SVM

K = 5	Classe Depressiva	Classe não Depressiva	Totais
Classe Depressiva	14.190	1.557	15.747
Classe não Depressiva	1.554	13.876	15.430
Totais	15.744	15.473	31.177

Acurácia	Precisão	Recall	F1_Score
0.90	0.90	0.90	0.90

Acurácia Otimizada	Melhor Combinação de Parâmetros
0.93	{'C': 10, 'gamma': 0.01, 'kernel': 'rbf'}

Figura 14: Experimento 2 | 31.177 tweets | k = 5
Modelo: SVM

K = 10	Classe Depressiva	Classe não Depressiva	Totais
Classe Depressiva	14.456	1.291	15.747
Classe não Depressiva	1.432	13.998	15.430
Totais	15.888	15.289	31.177

Acurácia	Precisão	Recall	F1_Score
0.91	0.91	0.91	0.91

Acurácia Otimizada	Melhor Combinação de Parâmetros
0.94	{'C': 10, 'gamma': 0.01, 'kernel': 'rbf'}

Figura 15: Experimento 2 | 31.177 tweets | k = 10
Modelo: SVM

A figura 16 representa um resumo geral das métricas obtidas em cada modelo de classificação.

Modelo	k	Quant. Tweets	Acurácia	Acurácia Otimizada	Precisão	Sensibilidade	F1_Score
Naive Bayes	5	5.000	0.73	-	0.74	0.73	0.72
Naive Baye	10	5.000	0.76	-	0.78	0.76	0.72
Naive Baye	5	31.177	0.82	-	0.84	0.82	0.82
Naive Baye	10	31.177	0.84	-	0.85	0.84	0.84
SVM	5	5.000	0.82	**0.86**	0.82	0.82	0.82
SVM	10	5.000	0.84	**0.87**	0.84	0.84	0.84
SVM	5	31.177	0.90	**0.93**	0.90	0.90	0.90
SVM	10	31.177	0.91	**0.94**	0.91	0.91	0.91

Figura 16: Resumo das métricas obtidas

De acordo com os resultados obtidos, observa-se que o modelo SVM foi superior em relação ao modelo Naive Bayes, assim como apresentado em diversos outros trabalhos nesse contexto. Vê-se

também que o aumento do valor de k usado no método de validação cruzada e aumento na quantidade de tweets gerou melhores resultados. Entretanto, esses variações de aumento de quantidade não necessariamente implicam em bons resultados, outras variáveis devem ser consideradas, como o modelo de implementação, qualidade do *dataset* e rotulação consistente de tweets na etapa de separação das classes.

5 CONCLUSÕES

Este trabalho apresentou como a análise de sentimentos baseada em aprendizagem de máquina pode ser utilizada para auxiliar na classificação de tweets depressivos e não depressivos em redes sociais. Foram utilizados os algoritmos Naive Bayes e SVM de aprendizagem de máquina, sendo que o modelo SVM se mostrou mais eficaz que o modelo de Naive Bayes, trazendo como resultado uma acurácia de 94% no experimento realizado com a quantidade máxima de tweets do *dataset*.

Além disso, este trabalho apresenta uma contribuição relevante em como construir uma base de dados em português para realizar o processo de análise de sentimento, já que grande parte dos trabalhos na área utiliza como base de dados a língua inglesa, que já possui inúmeras ferramentas para sua utilização.

Com relação a eficácia dos algoritmos utilizados, a presença de erros nas classificações que impedem o modelo de alcançar métricas próximas dos 100% podem ter ocorrido pelo fato de que muitas vezes um tweet extraído para a classe depressiva pode de fato não pertencer a essa classe. Por isso o ideal seria que os tweets utilizados no modelo tivessem sido revisados por equipes técnicas da área de psicologia e/ou psiquiatria de modo a garantir o máximo possível de qualidade nas rotulações.

Finalmente, pode-se observar também que a rede social do Tweeter é um excelente ambiente para a extração de informações textuais e aplicação em experimentos dessa natureza. Diante disso é possível de fato disponibilizar através deste projeto informações que possam de algum modo contribuir com o tratamento da depressão, pois a referida doença tem tirado muitas vidas no século corrente, tendo seus índices aumentados principalmente em épocas de pandemia e isolamento social.

Para trabalhos futuros, pretende-se obter o apoio de especialistas (psicólogos e psiquiatras) que possam ajudar no processo de rotulação de tweets, aumentando a qualidade do *dataset*. Um outro objetivo é classificar tweets com base em um perfil específico de um usuário, através das informações adicionais presentes no MongoDB Atlas, obtidas na fase inicial do processo de extração. Em outras palavras, o objetivo é classificar não apenas um tweet, mas sim um perfil de um usuário como sendo da classe depressiva ou não, mas isso só é possível com a ajuda de profissionais da área.

REFERÊNCIAS

[1] J. Silge and D. Robinson. *Text Mining in R: A Tidy Approach.* O'Reilly, 2017.
[2] C. H. Miranda and J.e GuzmÃ¡n. A Review of Sentiment Analysis in Spanish. *Tecciencia*, 12:35–48, 06 2017. ISSN 1909-3667.
[3] Depressão não tem idade, 2019. URL https://www.em.com.br/app/noticia/tecnologia/2012/06/11/interna_tecnologia,299333/depressao-e-mal-que-nao-tem-idade.shtml. Accessado: 31/01/2019.
[4] Hospital Estadual de Urgencia da Região Sudeste. OMS considera depressão uma epidemia global, 2019. URL http://hursosantahelena.org.br/noticias/oms-considera-depressao-epidemia-global/#:~:text=Segundo%20a%20Organiza%C3%A7%C3%A3o%20Mundial%20de,mundo%3A%209%2C3%25. Accessado: 23/11/2020.
[5] GZHVida. Saiba por que os brasileiros são os mais ansiosos do mundo, 2017. URL https://gauchazh.clicrbs.com.br/saude/vida/noticia/2017/03/saiba-por-que-os-brasileiros-sao-os-mais-ansiosos-do-mundo-9750651.html#:~:text=O%20mesmo%20documento%20revela%20que,global%20de%20doen%C3%A7as%20n%C3%A3o%20fatais. Accessado: 23/11/2020.
[6] GreenMea. Estudo revela o custo da depressão e da ansiedade para a economia global, 2016. URL https://www.greenme.com.br/viver/saude-e-bem-estar/3194-estudo-custo-depressao-ansiedade. Accessado: 31/01/2019.
[7] M. V. Mäntylä, D. Graziotin, and M. Kuutila. The evolution of sentiment analysis—a review of research topics, venues, and top cited papers. *Computer Science Review*, 27:16 –32, 2018. ISSN 1574-0137.
[8] Z. Drus and H. Khalid. Sentiment analysis in social media and its application: Systematic literature review. *Procedia Computer Science*, 161:707–714, 2019. ISSN 1877-0509. The Fifth Information Systems International Conference, 23-24 July 2019, Surabaya, Indonesia.
[9] M. Song, H. Park, and K-S. Shin. Attention-based long short-term memory network using sentiment lexicon embedding for aspect-level sentiment analysis in korean. *Information Processing and Management*, 56(3):637–653, 2019.
[10] P. Sanguansat. Paragraph2vec-based sentiment analysis on social media for business in thailand. In *8th International Conference on Knowledge and Smart Technology (KST)*, pages 175–178, 2016.
[11] Itani. M., C. Roast, and S. Al-Khayatt. Developing resources for sentiment analysis of informal arabic text in social media. *Procedia Computer Science*, 117:129 – 136, 2017. Arabic Computational Linguistics.
[12] ZANCHINI. Vinícius Augusto Alves. Criação de um modelo de classificação de tweets depressivos utilizando máquina de vetores de suporte. 2019. 38 f. Brasil, Minas Gerais, 2019.
[13] Sentiment140 dataset with 1.6 million tweets. URL https://www.kaggle.com/kazanova/sentiment140.
[14] CORRÊA. Igor Tannús. Análise dos sentimentos expressos na rede social twitter em relação aos filmes indicados ao oscar 2017. 2017. 72 f. trabalho de conclusão de curso (graduação em sistemas de informação) – universidade federal de uberlândia, uberlândia, 2017. Brasil, Minas Gerais, 2017.
[15] Rodolpho Nascimento, Flavio Carvalho, and Gustavo Guedes. Identificando sintomas depressivos: um estudo de caso no youtube. In *Anais do VIII Brazilian Workshop on Social Network Analysis and Mining*, pages 119–130, Porto Alegre, RS, Brasil, 2019. SBC. doi: 10.5753/brasnam.2019.6554. URL https://sol.sbc.org.br/index.php/brasnam/article/view/6554.
[16] PEREIRA. Janailton Galvão. Análise de sentimentos da população brasileira em relação a eleição presidencial de 2018 através da rede social twitter. caicó, rn: 2019. 71f. trabalho de conclusão de curso (bacharelado) - universidade federal do rio grande do norte. centro de ensino superior do seridó. departamento de computação e tecnologia. 2019.
[17] R. Feldman. Techniques and applications for sentiment analysis. *Communications of The ACM*, 53(4):82–89, April 2013.
[18] P. Simon, editor. *Too Big to Ignore: The Business Case for Big Data.* Wiley, 2008.
[19] I. Steinwart and A. Christmann, editors. *Support vector machine.* Spriger, 2008.
[20] H. Saif, Y. He, and H. Alani. Semantic sentiment analysis of twitter. In P. Cudré-Mauroux, J. Heflin, E. Sirin, T. Tudorache, J. Euzenat, M. Hauswirth, J. X. Parreira, J. Hendler, G. Schreiber, A. Bernstein, and E. Blomqvist, editors, *The Semantic Web – ISWC 2012*, pages 508–524, Berlin, Heidelberg, 2012. Springer.
[21] T. Gonçalves, J. C. da Silva, and O. A. C. Cortes. Técnicas de mineração de dados: um estudo de caso da evasão no ensino superior do instituto federal do maranhão. *Revista Brasileira de Computação Aplicada*, 10(3):11–20, set. 2018. doi: 10.5335/rbca.v10i3.8427. URL http://seer.upf.br/index.php/rbca/article/view/8427.
[22] A. C. Lorena and A. C. P. L. F. de Carvalho. Uma introdução às support vector machines. *RITA*, XIV(2), 2007.
[23] C. Campbell. An introduction to kernel methods. In R. J. Howlett and L. C. Jain, editors, *Radial Basis Function Networks: Design and Applications.* Springer Verlag, Berlin, 2000.
[24] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, 2(2):1–43, 1998.
[25] L. Li and C. Li. Research and improvement of a spam filter based on naive bayes. In *2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, volume 2, pages 361–364, Aug 2015.
[26] Dataframe. URL https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html.
[27] Natural language toolkit. URL https://www.nltk.org/.
[28] Introdução ao processamento de linguagem natural (nlp). URL https://cienciaenegocios.com/processamento-de-linguagem-natural-nlp/.
[29] What is a pipeline in machine learning? how to create one? URL https://medium.com/analytics-vidhya/what-is-a-pipeline-in-machine-learning-how-to-create-one-bda91d0ceaca.

[30] Entendendo o que é matriz de confusão com python. URL https://medium.com/data-hackers/entendendo-o-que-%C3%A9-matriz-de-confus%C3%A3o-com-python-114e683ec509.