2. PROCESSAMENTO DE LINGUAGEM NATURAL PARA ANÁLISE DE SENTIMENTO UTILIZANDO PYTHON

Guilherme Noronha

2.1 INTRODUÇÃO

Um dos desafios da pesquisa social aplicada é o processamento e gerenciamento de abundância de dados. A era do *Big Data* exige que o(a) pesquisador(a) esteja capacitado(a) para manusear tecnologias que capturam, limpam e extraiam a informação de dados disponibilizados nas mais variadas fontes. Somente em 2022, usuários, na *internet,* movimentaram 97 *zettabytes* de dados, o equivalente a 10^{21} *bytes*. A cada minuto, usuários no *Twitter* compartilharam mais de 347 mil *tweets*, 231 milhões de e-mails foram enviados e 16 milhões de mensagens de texto foram trocadas (Domo, 2022).

Como lidar com esses dados, principalmente quando eles estão na forma desestruturada de textos? A área responsável por lidar com esse problema é o Processamento de Linguagem Natural, ou PLN. Ela é responsável por estudar técnicas que convertam a linguagem natural em formas capazes de serem processadas por computadores, permitindo a análise automatizada de dados. O PLN conta com ajuda de aprendizado de máquina que, segundo Burkov (2019), são métodos de inteligência artificial capazes de fazer tarefas sem serem explicitamente programados para tal.

A área moderna do PLN data os anos do pós-Segunda Guerra Mundial com iniciativas de Noam Chomsky (1972) por meio da chamada gramática gerativa e a tradução por máquinas criada pela Universidade de Georgetown e IBM (Hutchins, 2004), a citar algumas. Trata-se de um campo extenso de pesquisa com aplicações distintas, por exemplo, tradução de idiomas, sintetizador de voz, sumarizador de textos, reconhecimento de entidades, extração de informação e análise de sentimentos, tema deste capítulo.

Análise de sentimentos é uma técnica que permite extrair informação subjetiva de textos. Nela, o(a) pesquisador(a) é, a partir de um conjunto de documentos, capaz de identificar, extrair e quantificar *metadados* importantes para análise (os sentimentos presentes). Embora a análise seja comumente associada aos sentimentos negativo, positivo e neutro, essa técnica pode se aplicar a diferentes classificações que o pesquisador deseja medir. Só na língua portuguesa, por exemplo, existem mais de setecentos sentimentos que podem ser usados para classificação (Ramos; Freitas, 2019).

Este capítulo pretende auxiliar o(a) pesquisador(a) da área de ciência social aplicada a dominar a metodologia básica do processamento de linguagem natural para fazer análise de sentimentos. Por meio de um caso de uso utilizando o *Python*, explica-se o passo a passo desde o tratamento de dados até a construção de um analisador de sentimentos que seja capaz de medir com alto grau de acurácia sentimentos que podem ser aplicados a diferentes tipos de pesquisa.

Este capítulo foi dividido da seguinte maneira: a seção 2 explica as tecnologias usadas para o desenvolvimento do caso de uso; a seção 3 apresenta um caso de uso detalhando cada passo desde a análise exploratória, a transformação, modelagem e análise de resultados. A seção 4 traz algumas pesquisas feitas nos últimos cinco anos que exemplificam a análise de sentimentos como método. Por fim, a seção 5 traz as considerações finais.

2.2 TECNOLOGIAS USADAS

Para o exemplo mostrado neste capítulo foram usados o *Python* e algumas bibliotecas para PLN e aprendizado de máquina como *spaCy*[1], o *scikit-learn*[2] e o *Pandas*[3]. O código-fonte utilizado pode ser encontrado no *Github*[4].

1 Disponível em: https://github.com/explosion/spaCy .

2 Disponível em: https://scikit-learn.org/.

3 Disponível em: https://pandas.pydata.org/.

4 Disponível em: https://github.com/guilhermenoronha/sentiment_analysis_chapter.

O *spaCy* é um *software* de código aberto usado para processamento avançado de linguagem natural. Seu código-fonte é mantido e desenvolvido pela comunidade. Na versão 3.0, o *spaCy* oferece suporte para mais de 70 idiomas, incluindo o português, técnicas que acompanham a evolução do estado da arte em PLN, componentes de análise prontos para uso, suporte para aplicação de aprendizado de máquina, entre outras funcionalidades. Neste capítulo o *spaCy* será usado como ferramenta para limpeza e tratamento de texto.

O *scikit-learn* é uma biblioteca que oferece soluções de aprendizado de máquina para *Python*. Ela fornece diferentes pacotes para a resolução de diferentes problemas de aprendizado, como classificação, regressão e *clusterização* de dados. A análise de sentimentos é, em sua natureza, um problema de classificação e o *scikit-learn* possui meios para facilitar essa implementação. Assim como o *spaCy*, ele também é de código aberto, além de ser amplamente adotado tanto academicamente quanto industrialmente. Neste capítulo o *scikit-learn* foi usado para treinar o modelo de aprendizado de máquina que fará a análise de sentimentos automática.

Por fim, a biblioteca *Pandas* é a responsável pela análise e manipulação de dados. Ela é a principal biblioteca do *Python* para manipulação de dados no formato tabular. Além de armazenar dados no formato de linhas e colunas, ela possui uma série de funções embutidas úteis na análise de dados. *Pandas* foi usado neste capítulo para armazenar a base e fazer as análises iniciais.

Instruções de como instalar e utilizar o código-fonte podem ser encontradas no *Github* do próprio capítulo.

2.3 ANÁLISE DE SENTIMENTO NA PRÁTICA

É preciso fazer uma ressalva antes de passar para a parte prática. Como dito anteriormente, a análise de sentimentos é um problema de classificação de dados. Ou seja, dado um texto, quer-se saber qual classe ele possui. No exemplo deste capítulo, veremos se o sentimento é positivo ou negativo.

Para que isso seja possível, é necessário treinar um *algoritmo* a fim de que ele seja capaz de identificar, automaticamente, as classes de sentimentos. Esse treinamento requer a presença de uma base de dados com textos pré-selecionados e classificados. Nos casos em que o(a) pesquisador(a) não disponha dessa base, ele deverá montá-la, preferencialmente usando textos semelhantes ao que deseja classificar automaticamente. O(A) pesquisador(a) deverá fazer o trabalho de coleta de dados e posterior classificação. A coleta pode ser feita por meio de bibliotecas para raspagem de dados como o *scrapy*[5] ou por meio de *APIs* (*Application Programming Interface*) especializadas. As *APIs* são aplicações que fornecem uma *interface* de comunicação a um determinado serviço, como *Twitter*, *Instagram*, Governo Federal etc. Elas podem ser encontradas nos sites dos fornecedores como o *tweepy*[6] para o *Twitter*, *Graph API*[7] para o *Instagram* e as *APIs* Governamentais[8] para o Governo Federal.

Após a coleta, o(a) pesquisador(a) deve anotar manualmente os textos com o sentimento que ele deseja classificar. A anotação de textos pode ser feita por meio de *softwares* especializados como o *Prodigy*[9], *Label Studio*[10] etc. Essa coleta inicial é feita somente com o intuito de treinar um *algoritmo* para que ele consiga classificar novos textos manualmente. Quanto maior a quantidade de textos coletados, melhor "treinado" será o *algoritmo*. Embora a coleta de dados seja uma etapa importante, ela não é contemplada neste capítulo cujo foco está apenas no processamento.

Neste capítulo usou-se uma base de dados de comentários de filmes feitos no *IMDB* (Internet Movie Database)[11], site especializado em crítica de obras audiovisuais. Essa base contém duas colunas de interesse: a primeira, com a crítica em língua portuguesa de um usuário sobre um determinado filme

5 Disponível em: https://scrapy.org/.

6 Disponível em: https://www.tweepy.org/.

7 Disponível em: https://developers.facebook.com/docs/instagram-api/.

8 Disponível em: https://www.gov.br/conecta/catalogo/apis/api-de-servicos.

9 Disponível em: https://spacy.io/universe/project/prodigy.

10 Disponível em: https://labelstud.io/.

11 Disponível em: https://www.imdb.com/.

e, a segunda, com a classificação do sentimento dessa crítica, podendo ter os valores "negativo" e "positivo". Uma primeira impressão da base pode ser vista na Figura 1.

Figura 1 - Exemplo de sentimentos para análise de sentimentos de críticas de filmes.

	id	text_pt	sentiment
0	1	Mais uma vez, o Sr. Costner arrumou um filme por muito mais tempo do que o necessário. Além das terríveis seqüências de resgate no mar, das quais há muito poucas, eu simplesmente não me importei com nenhum dos personagens. A maioria de nós tem fantasmas no armário, e o personagem Costers é realizado logo no início, e depois esquecido até muito mais tarde, quando eu não me importava. O personagem com o qual deveríamos nos importar é muito arrogante e superconfiante, Ashton Kutcher. O problema é que ele sai como um garoto que pensa que é melhor do que qualquer outra pessoa ao seu redor e não mostra sinais de um armário desordenado. Seu único obstáculo parece estar vencendo Costner. Finalmente, quando estamos bem além do meio do caminho, Costner nos conta sobre os fantasmas dos Kutchers. Somos informados de por que Kutcher é levado a ser o melhor sem pressentimentos ou presságios anteriores. Nenhuma mágica aqui, era tudo que eu podia fazer para não desligar uma hora.	negativo
12389	12391	Eu fui e vi este filme ontem à noite depois de ser persuadido por alguns amigos meus. Eu admitiria que estava relutante em vê-lo porque, pelo que eu sabia de Ashton Kutcher, ele só conseguia fazer comédia. Eu estava errado. Kutcher interpretou o personagem de Jake Fischer muito bem, e Kevin Costner interpretou Ben Randall com tal profissionalismo. O sinal de um bom filme é que ele pode brincar com nossas emoções. Este fez exatamente isso. Todo o teatro que foi vendido foi superado pelo riso durante a primeira metade do filme, e foi levado às lágrimas durante o segundo semestre. Ao sair do teatro, eu não só vi muitas mulheres em lágrimas, mas também muitos homens adultos, tentando desesperadamente não deixar ninguém vê-los chorando. Este filme foi ótimo, e eu sugiro que você vá vê-lo antes de julgar.	positivo

Fonte: Captura de tela (2023).

2.3.1 ANÁLISE EXPLORATÓRIA DE DADOS

A primeira etapa consiste em fazer uma análise exploratória de dados para entender melhor o universo que o(a) pesquisador(a) trabalhará. Essa análise busca compreender o aspecto da base de dados, identificar as colunas úteis, os possíveis tratamentos de dados a serem feitos, balanceamento de dados etc. Numa primeira análise é importante responder às seguintes perguntas: (1) Qual o tamanho da base?; (2) Quais colunas compõem essa base e qual a relevância de cada uma delas para o projeto? e; (3) Quais classes de sentimento essa base possui e como ela está distribuída?

A Figura 2 ilustra o comando inicial para carregar a base de dados para análise exploratória. Em seguida, usa-se o comando *df.shape* para obter a resposta da primeira pergunta: a base possui 49459 linhas e 3 colunas.

Figura 2 - Comandos para carregar a base de dados usando o *Pandas*.

```
df = pd.read_csv(
    'https://github.com/guilhermenoronha/sentiment_analysis_chapter/raw/main/dataset/sentiment_analysis.zip',
    sep=',',
    index_col=[0]
)
pd.set_option('display.max_colwidth', None)
```

Fonte: Elaborado pelo autor (2023).

Com o comando *df.head()* obtém-se uma amostra do conteúdo da base para responder à segunda pergunta da análise. O resultado é similar ao mostrado na Figura 1. As colunas da base são: *id*, *text_pt* e *sentiment*. Uma análise inicial nos diz que apenas as colunas *text_pt* e *sentiment* são de interesse para a classificação de sentimento. Para remover a coluna "*id*" (*identity*) usa-se o comando *df.drop(columns=['id'], inplace=True*).

Já, para responder à terceira pergunta, pode-se usar o comando *df['sentiment'].drop_duplicates()* a fim de identificar quantas classes de sentimento essa base possui e, o comando *df['sentiment'].value_counts()*, para realizar a contagem de registros de cada sentimento. Existem 24765 avaliações positivas e 24694 avaliações negativas. É uma proporção de 50,07% para sentimentos positivos e 49,93% para sentimentos negativos.

É fundamental entender como as classes de sentimento estão distribuídas dentro da base de estudos. Se o(a) pesquisador(a) deseja que o *algoritmo* aprenda a classificar igualmente todas as classes, é ideal que a proporção de registros de cada uma seja a mais próxima possível. Se o objetivo for, por exemplo, priorizar apenas a classificação correta de sentimentos negativos, é ideal que a base de dados tenha mais dados da classe de interesse. Na impossibilidade de ter essa proporção de dados, também é possível aplicar pesos diferentes na hora de classificá-los. Ou seja, configura-se o *algoritmo* de aprendizado para dar mais importância às classificações de interesse, ainda que estejam em menor número.

A proporção encontrada anteriormente já seria satisfatória, mas, para exemplo teórico, é possível balancear a base usando os seguintes comandos mostrados na Figura 3. Os comandos balanceiam a base removendo registros das categorias de maior quantidade até que elas tenham a mesma

quantidade da categoria com menor número de registros. O resultado é 24694 registros para ambos os sentimentos.

Figura 3 - Balanceando a base de dados.

```
min_rows = df.groupby('sentiment').apply(lambda x: len(x)).min()
df = df.groupby('sentiment').apply(lambda x: x.sample(min_rows)).reset_index(drop=True)
```

Fonte: Elaborado pelo autor (2023).

Após uma análise exploratória de dados, a próxima etapa é a transformação.

2.3.2 TRANSFORMAÇÃO DE DADOS

O processo de transformação trata-se de preparar os dados para serem consumidos por um usuário e/ou serviço. No exemplo deste capítulo, os dados serão consumidos pelo *pipeline* de aprendizado de máquina que será responsável por treinar e classificar opiniões de filmes. Saber quem (ou o quê) vai consumir os dados é importante para o(a) pesquisador(a), a fim de que ele(a) os transforme da maneira correta.

Existe um ditado cunhado por George Fuechsel, muito importante na análise de dados, que diz: "entra lixo, sai lixo." (*apud* Stenson, 2016, *online*). Fueschel queria dizer que alimentar um sistema com dados ruins traz resultados ruins, ou seja, classificações erradas, tomadas de decisões ruins, dados controversos, políticas imprecisas etc. Nesse sentido, a etapa de transformação mostra-se crucial para garantir a qualidade de entrada dos dados.

A análise de texto possui uma série de metodologias utilizadas para a padronização da linguagem natural. Elas são importantes para que diferentes textos possam ser processados por computadores. A metodologia para análise de sentimentos consiste em identificar quais são os conjuntos de palavras cujo significado é determinante na hora de classificar um texto segundo as classes de sentimentos. Por exemplo, as palavras "adorei" e "horrível" possuem alta significância na classificação de sentimento como positivo ou negativo.

A limpeza de dados é a técnica responsável por eliminar do texto partes cujo significado agrega pouco ou nada na tarefa de classificação. Para os

exemplos deste capítulo, aplicaram-se as seguintes transformações de dados: capitalização; lematização; remoção de *stopwords*, representações numéricas, *URLs*, *emoticons* e espaços em branco. As técnicas são brevemente descritas a seguir.

2.3.2.1 CAPITALIZAÇÃO DE TEXTO

Capitalizar um texto é transformá-lo inteiramente em caixa baixa ou caixa alta. Essa técnica é fundamental para padronizar o texto num mesmo conjunto de caracteres. Os computadores armazenam letras maiúsculas e minúsculas de formas diferentes. A letra 'A' e 'a', por exemplo, são armazenadas usando os códigos ASCII 65 e 97 (*American Standard Code for Information Interchange*) respectivamente. Ou seja, para um *algoritmo*, as palavras "Casa" e "casa" terão códigos ASCII distintos. De acordo com George *et al.* (2016) a capitalização de texto reduz o vocabulário e aumenta o poder estatístico e a validade dos resultados, trazendo benefícios para o aprendizado de máquina. Não há diferença entre capitalização em caixa baixa ou alta, embora haja uma adoção maior para capitalização em caixa baixa.

2.3.2.2 LEMATIZAÇÃO

A *lematização* é uma técnica com o propósito de normalizar a linguagem natural unificando palavras com o mesmo lema, mas flexionadas de formas diferentes. Por exemplo, as palavras "faria", "faz", "faço" pertencem ao mesmo lema: "fazer". Num processo de *lematização* todas as palavras são substituídas pelo seu lema equivalente. O propósito dessa transformação é similar ao proposto pela capitalização de texto: reduzir o vocabulário e aumentar o poder estatístico.

A *lematização* possui uma técnica similar chamada de *stemming*. Essa técnica pretende extrair os radicais das palavras. No exemplo do parágrafo anterior, o *stemming* geraria os seguintes resultados: "far", "faz" e "faç". O resultado seria pior, pois o tamanho do vocabulário ao final do processamento seria maior. No entanto, essa técnica destaca-se por preocupar-se em remover os *afixos*. As palavras "desfazendo" e "fazer" teriam o mesmo radical: "faz". A escolha entre um e outro depende do objetivo da

análise. Quando o contexto importa, usa-se o *lematizador*, caso contrário, o *stemming* (Balakrishnan; Lloyd-Yemoh, 2014).

2.3.2.3 STOP WORDS

Stop words são consideradas palavras extremamente comuns que agregam pouco ou quase nada para a análise de um documento. Geralmente são *stop words* as palavras funcionais que se encaixam nas classes de preposição, artigo, interjeição, pronome e conjunção. A lista de *stop words* pode variar de aplicação para aplicação e pode ser personalizada pelo(a) pesquisador(a), caso necessário. Segundo Hickman *et al.* (2022) as *stop words* aumentam o poder estatístico, mas reduz a capacidade de capturar o estilo de escrita do texto. Para análise de sentimentos, o estilo de escrita não é relevante.

2.3.2.4 REMOÇÃO DE OUTRAS CATEGORIAS DE PALAVRAS

Nesta subseção estão incluídos os tratamentos para remoção de *emoticons*, representações numéricas, pontuações e *URLs*. O objetivo de todos é o mesmo: aumentar o poder estatístico de aprendizado. Os *emoticons* podem identificar sentimentos, mas exige que uma transformação à parte seja aplicada para gerenciar diferentes representações e isso foge do escopo deste capítulo. Além disso, a remoção de *emoticons* aumenta a validade do aprendizado. As *URLs* são como *stop words* e carregam pouco ou nenhum significado para a análise. Ele é válido para representações numéricas e pontuações. Embora alguns autores como Goldbeck *et al.* (2012) defendam o uso da exclamação '!' como identificador de personalidade, isso não se aplica ao contexto de análise de sentimentos.

2.3.2.5 OUTRAS TÉCNICAS

As técnicas apresentadas acima não são as únicas presentes dentro do contexto de PLN para transformação de dados. Pode-se também corrigir erros ortográficos, expandir acrônimos e abreviações e/ou fazer controle do uso de negação nos textos. Cada técnica pode produzir efeitos diferentes no processo de aprendizado de máquina e seu uso deve ser considerado pelo(a)

pesquisador(a). É recomendado que o(a) pesquisador(a) faça testes com diferentes técnicas para chegar ao resultado ótimo. Para mais detalhes sobre técnicas de processamento de linguagem natural, ver (Hickman *et al.*, 2022).

2.3.2.6 TRANSFORMANDO OS DADOS

Para transformar os dados usando as técnicas citadas nas subseções anteriores, usaram-se as bibliotecas *emoji*, *spaCy* e *Pandas*. O *spaCy* foi responsável pela capitalização em caixa baixa, remover pontuações, *stopwords*, representações numéricas e *URLs*. A biblioteca *emoji* removeu os *emoticons*. Todas essas transformações foram adicionadas numa função em *Python* executada pelo *Pandas* em cada um dos textos da base de dados. A Figura 4 mostra o código-fonte usado para a transformação.

Figura 4 - Código-fonte da transformação de dados.

```
def clean_text(sentence):
    doc = nlp(sentence)
    tokens = [token.lemma_.lower() for token in doc
              if  not token.is_punct and # Filter punctuation
                  not token.is_stop  and # Filter stopwords
                  not token.like_num and # Filter numeric representations
                  not token.like_url     # Filter urls
             ]
    cleaned_text = emoji.replace_emoji(' '.join(tokens), replace='') # Remove emoticons
    return cleaned_text.replace("  ", " ") # Remove extra whitespaces
df['processed_text'] = df['text_pt'].apply(clean_text)
```

Fonte: Elaborado pelo autor (2023).

O resultado da transformação é armazenado na coluna *processed_text* apenas para efeito de comparação[12]. A coluna *text_pt* pode ser removida, pois não é usada no processo de aprendizado de máquina. A Figura 5 mostra um exemplo de uma opinião antes e depois da transformação.

12 O processo de transformação é custoso e demorado. É esperado um tempo de execução entre 15 e 30 minutos em um computador com *16GB* de memória *RAM* e um processador *i7* da 11ª geração.

Figura 5 - Resultado da transformação de texto.

text_pt	processed_text
Robin Williams é excelente neste filme e é uma pena que o material não seja páreo para ele. Isso pode funcionar se você comprar o "U-S-A! Número Um!" mentalidade, mas história sábia nada acontece. É uma pena, já que o filme está realmente tentando dizer alguma coisa, e diz sinceramente. Apenas não causa um impacto emocional suficiente.	robin williams excelente filme pena material ser páreo funcionar comprar u-s-a mentalidade história sábio acontecer pena filme realmente tentar algum sinceramente causar impacto emocional suficiente

Fonte: Captura de tela (2023).

2.3.3 CRIAÇÃO DO MODELO DE APRENDIZAGEM

A análise de sentimentos objetiva consumir os dados analisados e transformados previamente por um *algoritmo* de aprendizado de máquina. Esse *algoritmo* usará a base de dados para aprender os padrões que identificam um sentimento na hora de avaliar um filme no *IMDB*. Uma vez treinado, esse *algoritmo* pode ser usado para identificar opiniões novas.

A biblioteca responsável por usar e treinar *algoritmos* de aprendizado de máquina é o *scikit-learn*. Ela possui, por padrão, uma série de *algoritmos* diferentes que podem ser aplicados tanto para análise de sentimentos quanto para outros tipos de PLN. Para criar um modelo de aprendizado, o(a) pesquisador(a) deve escolhê-lo previamente e, depois, aplicar uma última transformação para adequar os dados ao modelo de consumo do *algoritmo*.

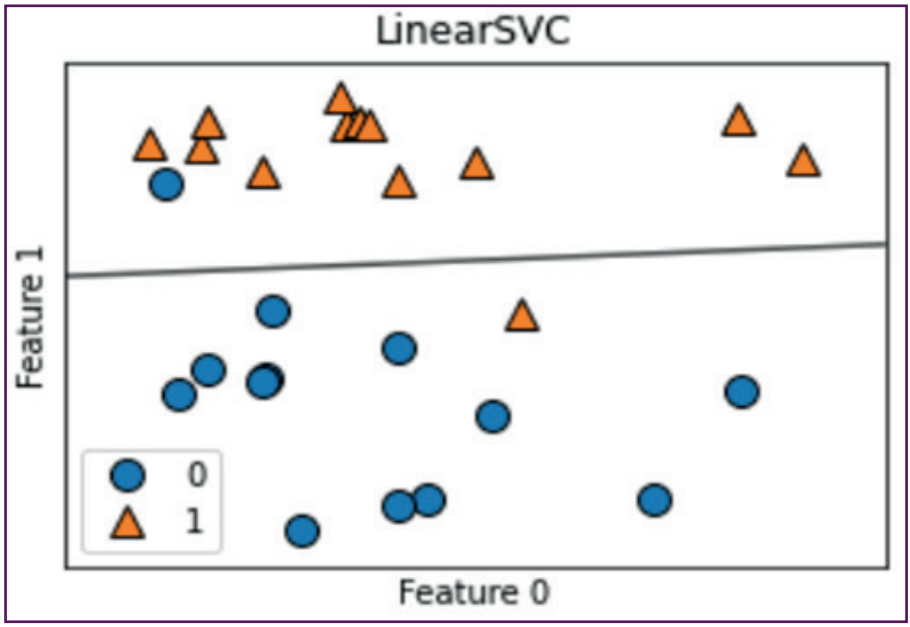
2.3.3.1 ESCOLHENDO O MÉTODO DE APRENDIZADO DE MÁQUINA

O *algoritmo* escolhido para a execução deste capítulo foi *LinearSVC*, mas ele não é o único e não necessariamente o melhor. Cabe ao(à) pesquisador(a) testar e aplicar diferentes métodos para entender qual é aquele que se adequa melhor à base de dados. Aprendizado de máquina não é o escopo deste capítulo. Mais informações sobre o assunto podem ser encontradas em (Burkov, 2019).

O *LinearSVC* é um *algoritmo* da categoria dos classificadores. O *acrônimo SVC* vem do termo *Support Vector Classification*, ou seja, ele faz classificações baseadas em vetores de suporte. Para isso, cada opinião da base de dados, que está em formato de texto, deve ser transformada num vetor para que o *LinearSVC* possa ser aplicado. Em linhas gerais, o *LinearSVC* classifica as classes da base de dados traçando uma *reta ótima* que divide

melhor os vetores em um plano. O *LinearSVC* pressupõe que vetores que estejam próximos um dos outros possuam semelhanças entre si. Logo, espera-se que vetores de classificações positivas e negativas estejam agrupadas em lados opostos do plano (Fan *et al.*, 2008). A Figura 6 ilustra um exemplo de classificação do *LinearSVC*.

Figura 6 - Exemplo de classificação usando o *LinearSVC* num plano de duas dimensões.



Fonte: Captura de tela (2023).

2.3.3.2 VETORIZAÇÃO DOS DADOS

A *vetorização* de textos possui uma técnica simples e funcional. Primeiro, calcula-se o tamanho do vetor sendo igual ao tamanho do vocabulário da base de dados. Cada eixo do vetor corresponde a uma palavra do vocabulário. Se um texto possui uma palavra, o valor do eixo correspondente é preenchido com um valor maior que zero e, zero, caso contrário.

Uma metodologia amplamente usada em PLN para calcular o melhor valor para cada palavra do documento é o *TF-IDF*, do inglês *Term*

Frequency–Inverse Document Frequency. O *TF-IDF* calcula a importância que cada palavra tem em um documento, dando um valor entre 0 e 1. Esse valor é calculado pela frequência que essa palavra aparece no documento e, depois, é multiplicado pela frequência invertida em que essa palavra aparece em diferentes documentos da base de dados. Em outras palavras, quanto mais uma palavra aparece em um documento específico e mais rara ela é entre os demais documentos, mais valor ela possui (Qaiser; Ali, 2018). A *vetorização* da análise de sentimentos aplica os valores calculados de *TF-IDF* para cada palavra em cada documento.

2.3.3.3 CRIANDO E TREINANDO O MODELO DE APRENDIZADO

O código mostrado na Figura 7 descreve uma série de passos para criar e treinar o modelo de aprendizado. As variáveis *tfidf* e *svm* correspondem ao processo de *vetorização* e aprendizado citados nas subseções anteriores. Esses processos são sequenciados dentro da variável *pipe*, que é um *pipeline* de processamento. As variáveis *X* e *y* correspondem aos dados que serão usados e as classes de sentimentos.

Figura 7 - Código-fonte da modelagem e treinamento do algoritmo de aprendizado de máquina.

```
tfidf = TfidfVectorizer()
svm = LinearSVC()
steps = [('tfidf',tfidf),('svm',svm)]
pipe = Pipeline(steps)
X = df['processed_text']
y = df['sentiment']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)
pipe.fit(X_train,y_train)
y_pred = pipe.predict(X_test)
```

Fonte: Elaborado pelo autor (2023).

Esses dados são divididos em variáveis de treinamento (*X_train, y_train*) e variáveis de teste (*X_test, y_test*). É recomendado, por padrão, que o pesquisador separe 80% dos dados para treinamento e 20% dos dados para teste. Se a base de dados for massiva (*big data*), recomenda-se aumentar o percentual de dados de treinamento.

Por fim, temos a função *fit* que executa o treinamento do modelo baseado nos dados de treinamento fornecidos. Os resultados do treinamento são medidos usando as variáveis de testes e são discutidos na próxima seção.

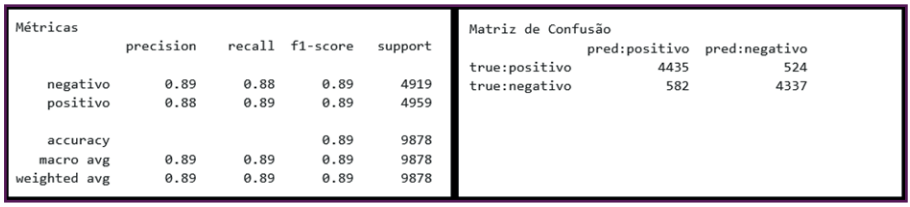
2.3.4 ANALISANDO OS RESULTADOS

Os resultados do treinamento de um *algoritmo* de aprendizado de máquina são obtidos por meio das métricas de precisão, revocação e *F-SCORE*. A precisão mede a proporção de acertos que o modelo teve em relação ao total de predições. Já a revocação mede a proporção de documentos relevantes classificados em relação ao total possível de classificações. Precisão e revocação são excludentes, ou seja, para ter uma boa precisão é preciso renunciar a uma boa revocação e vice-versa. Quando ambas as métricas são importantes, o *F-SCORE* é a melhor escolha, pois trata-se de uma média harmônica entre precisão e revocação (Burkov, 2019).

A escolha do uso entre precisão, revocação e *F-SCORE* depende do objetivo que o(a) pesquisador(a) almeja com a tarefa de análise de sentimentos. Se a prioridade é classificar corretamente os sentimentos, a precisão é mais importante. Caso a prioridade seja recuperar corretamente as classes de interesse, escolhe-se, então, a revocação. O *F-SCORE* é geralmente usado como uma *métrica padrão* de avaliação para comparação de estudos.

As métricas podem ser extraídas do modelo de aprendizado de máquina por meio de uma *matriz de confusão*. Essa matriz resume o quão eficiente foi o teste do modelo. Em um eixo encontram-se as classes de predição e noutro encontra-se a predição feita em relação a essa classe. O *scikit-learn* já calcula as métricas para o(a) pesquisador(a) e a matriz de precisão deve ser usada apenas como um complemento da análise. A Figura 8 mostra o resultado do teste assim como a *matriz de confusão*:

Figura 8 - Métricas e matriz de confusão do modelo.



Métricas

	precision	recall	f1-score	support
negativo	0.89	0.88	0.89	4919
positivo	0.88	0.89	0.89	4959
accuracy			0.89	9878
macro avg	0.89	0.89	0.89	9878
weighted avg	0.89	0.89	0.89	9878

Matriz de Confusão

	pred:positivo	pred:negativo
true:positivo	4435	524
true:negativo	582	4337

Fonte: Captura de tela (2023).

Pode-se tirar algumas conclusões ao analisar as métricas da Figura 8. A precisão para a classe positiva, por exemplo, é de 0,88. Ou seja, espera-se que a predição do *algoritmo* classifique corretamente 88% das vezes. Já a revocação para a mesma classe é de 0,89 que representa que o *algoritmo* consegue recuperar 89% das vezes um sentimento dessa classe. O *F-SCORE* para ambos os sentimentos possui valor de 0,89, refletindo a média harmônica entre precisão e revocação.

As métricas *macro avg* e *weighted avg* são formas diferentes de calcular as métricas. A primeira usa uma média simples entre todas as classes, enquanto a segunda usa uma média ponderada com valores de suporte escolhidos pelo(a) pesquisador(a). O *weighted avg* é importante quando o(a) pesquisador(a) deseja valorizar mais uma classe em detrimento da outra. Como esse valor não foi informado para o *algoritmo*, o *weighted avg* é igual ao *macro avg* (valores de suporte iguais a 1).

Por fim, a acurácia mede a quantidade de sentimentos classificados corretamente pela razão do total de sentimentos classificados. A *matriz de confusão* da Figura 8 é apenas uma forma de validar as métricas acima mencionadas. Ela é interpretada da seguinte maneira: a primeira linha representa a classe de sentimento positivo, sendo que 4435 análises foram classificadas corretamente e 524 incorretamente. Depois que o *algoritmo* foi treinado, ele é capaz de fazer predições conforme mostra a Figura 9.

Figura 9 - Analisando novos sentimentos.

```
sentence_1 = 'Achei esse filme muito bom'
sentence_2 = 'Perdi duas horas da minha vida'
print(f'Sentence: {sentence_1}\nSentiment:{pipe.predict([sentence_1])[0]}')
print(f'Sentence: {sentence_2}\nSentiment:{pipe.predict([sentence_2])[0]}')
```

```
Sentence: Achei esse filme muito bom
Sentiment:positivo
Sentence: Perdi duas horas da minha vida
Sentiment:negativo
```

Fonte: Elaborado pelo autor (2023).

2.4 APLICAÇÕES DE ANÁLISE DE SENTIMENTO EM PESQUISA

Este capítulo trouxe algumas pesquisas que usaram análise de sentimentos nos últimos cinco anos com o objetivo de ilustrar ao leitor as possibilidades dessa técnica dentro de diferentes assuntos das ciências sociais aplicadas. Alsaeedi e Khan (2019) testaram diversas técnicas de aprendizado de máquina para fazer análise de sentimentos no *Twitter*. Os autores encontraram que o *SVM*, da mesma família que o *LinearSVC*, foi a técnica que entregou os melhores resultados. Oliveira *et al*. (2019) analisaram os sentimentos no *Twitter* em relação aos programas públicos do governo Dilma Rousseff. Segundo os autores, as políticas públicas deveriam ser pautadas conforme os discursos da população civil. Os resultados indicaram os programas "Mais Médicos" e "Bolsa Família" como de maior rejeição, enquanto "Pronatec" e "Minha Casa, Minha Vida" tiveram a maior aceitação.

Shaukat *et al*. (2020) se propuseram a usar redes neurais e um dicionário para fazer análise de sentimentos da base do *IMDB* em língua inglesa. Os resultados encontrados por eles foram um F-SCORE de 0,91, pouco melhor que o exemplo usado neste capítulo. Coutinho e Malheiros (2020) usaram o Twitter para fazer a classificação de mensagens homofóbicas. Os autores chegaram a um *F-SCORE* de 0,64. A justificativa para o resultado é a dificuldade de entendimento consensual sobre o que é ou não uma mensagem homofóbica. Os autores utilizaram entrevistas para coleta de dados e obter classificações de cada mensagem. Como não houve consenso quanto à classificação, o modelo não pôde ser treinado corretamente.

Chauhan, Sharma e Sikka (2021) usaram análise de sentimentos para tentar prever os resultados de eleições passadas baseados em dados do *Twitter* e *Facebook*. Embora os resultados tenham sido mistos, os autores concluem que a análise de sentimentos pode ser sim um termômetro para medir as intenções de votos dos eleitores nas redes sociais. Souza, Souza e Meinerz (2021) analisaram sentimentos no mercado de ações em tempo real em uma tentativa de prever a oscilação de preços. A aplicação dos autores atingiu um *F-SCORE* de 0,76. A justificativa para o valor é a baixa captura de *tweets* no período de análise.

Mahyoob *et al*. (2022) usaram análise de sentimentos em *tweets* para medir a percepção da população em relação à emergência causada pela variante *Ômicron* da *COVID-19*. Os pesquisadores dividiram os sentimentos em subclasses que mediam sua força, variando entre o fraco até muito forte. Os resultados indicaram que a população estava preocupada e que esses indicadores poderiam ser usados para a adoção de medidas públicas para acalmá-la. Ainda usando o *Twitter* como fonte de dados, Paes *et al*. (2022) mediram o sentimento de usuários brasileiros em relação ao desmatamento da floresta amazônica, identificando picos de rejeição de até 60%. Os autores identificaram uma correlação entre os picos de rejeição com notícias divulgadas, destacando a importância do papel da mídia para fiscalizar e denunciar atividades nocivas à população brasileira.

2.5 CONSIDERAÇÕES FINAIS

Este capítulo teve como objetivo mostrar ao(à) pesquisador(a) de ciências sociais aplicadas o poder de PLN, mais especificamente a análise de sentimentos e suas múltiplas aplicações. A área de PLN é extensa quanto à variedade de técnicas e metodologias a serem aplicadas em pesquisa. Embora este capítulo traga as principais técnicas usadas na área, ele não é extensivo quanto às possibilidades de aplicação. Um estudo mais detalhado sobre a área pode ser lido em Wankhade, Rao e Kulkarni (2022).

As seções 2 e 3 trouxeram ao leitor um caso de análise de sentimentos na prática. Tanto a base de dados quanto o código-fonte em *Python* foram fornecidos ao(à) leitor(a) para que ele(a) consiga replicá-lo e, se necessário, readaptá-lo ao próprio contexto de pesquisa. As tecnologias usadas são as mais atuais

dentro do contexto da área, fazendo uso de bibliotecas tradicionais tanto de aprendizado de máquina quanto do processamento de textos.

A seção 4 mostrou diferentes aplicações que podem ser desenvolvidas usando a análise de sentimentos nos últimos cinco anos. Embora não seja uma pesquisa exaustiva, é suficiente para apresentar ao leitor ideias de aplicações, bem como possíveis locais para extração e captura de dados (Banks *et al.*, 2018). A análise de sentimentos se beneficia muito das redes sociais e da explosão de dados da *internet*.

Como contribuição, espera-se que este capítulo ajude o(a) leitor(a) a conduzir sua própria pesquisa na área de análise de sentimentos. As pesquisas em PLN dependem muito do desenvolvimento em diferentes idiomas, sendo ainda mais importante o desenvolvimento de pesquisas para o português do Brasil. Segundo a *National Science Board* (2018), o Brasil é apenas o 11º país em termos de pesquisa científica. Ao produzir mais pesquisas para o português, espera-se que esse *ranking* melhore em médio prazo.

REFERÊNCIAS

ALSAEEDI, A.; KHAN, M. Z. A study on sentiment analysis techniques of Twitter data. **International Journal of Advanced Computer Science and Applications - IJACSA**, Cleckheaton, v. 10, n. 2, p. 361-374, 2019.

BALAKRISHNAN, Vimala; LLOYD-YEMOH, Ethel. Stemming and Lemmatization: A Comparison of Retrieval Performances. **Lecture Notes on Software Engineering - LNSE**, [*s. l.*], v. 2, n. 3, p. 262-267, Aug. 2014. DOI 10.7763/LNSE.2014.V2.134. Disponível em: http://www.lnse.org/show-34-165-1.html. Acesso em: 19 set. 2023.

BANKS, G. C.; WOZNYJ, H. M.; WESSLEN, R. S.; ROSS, R. L. A review of best practice recommendations for text analysis in R (and a user-friendly app). **Journal of Business and Psychology**, Berlin, v. 33, n. 4, p. 445-459, Jan. 2018. Disponível em: https://link.springer.com/article/10.1007/s10869-017-9528-3. Acesso em: 19 set. 2023.

BURKOV, A. **The hundred-page machine learning book**. Quebec City: Andriy Burkov, 2019. v. 1, p. 32. ISBN 978-1999579500. Disponível em: http://ema.cri-info.cm/wp-content/uploads/2019/07/2019BurkovTheHundred-pageMachineLearning.pdf. Acesso em: 19 set. 2023.

CHAUHAN, Priyavrat; SHARMA, Nonita; SIKKA, Geeta. The emergence of social media data and sentiment analysis in election prediction. **Journal of Ambient Intelligence and Humanized Computing**, Berlin, v. 12, n. 2, p. 2601-2627, Feb. 2021. DOI 10.1007/s12652-020-02423-y. Disponível em: https://link.springer.com/10.1007/s12652-020-02423-y. Acesso em: 19 set. 2023.

CHOMSKY, N. **Studies on semantics in generative grammar**. Berlin: De Gruyter Mouton, 1972. (Series Janua Linguarum, n. 107). Disponível em: https://www.degruyter.com/document/doi/10.1515/9783110867589/html. Acesso em: 19 set. 2023.

COUTINHO, V. M. M. S.; MALHEIROS, Y. Detecção de mensagens homofóbicas em português no Twitter usando análise de sentimentos. *In*: BRAZILIAN WORKSHOP ON SOCIAL NETWORK ANALYSIS AND MINING (BraSNAM), 9., Cuiabá, 2020. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2020. p. 1-12. Disponível em: https://sol.sbc.org.br/index.php/brasnam/article/view/11158/11029. Acesso em: 19 set. 2023.

DOMO. Data Never Sleeps 10.0. 2022. Disponível em: https://www.domo.com/data-never-sleeps. Acesso em: 19 set. 2023.

FAN, R.-E.; CHANG, K.-W.; HSIEH, C.-J.; WANG, X.-R.; LIN, C.-J. LIBLINEAR: A library for large linear classification. **Journal of Machine Learning Research**, New York, v. 9, n. 9, p. 1871-1874, Aug. 2008.

GEORGE, Gerard; OSINGA, Ernst C.; LAVIE, Dovev; SCOTT, Brent A. Big Data and Data Science Methods for Management Research. **Academy of Management Journal**, New York, v. 59, n. 5, p. 1493-1507, out. 2016. DOI 10.5465/amj.2016.4005. Disponível em: http://journals.aom.org/doi/10.5465/amj.2016.4005. Acesso em: 19 set. 2023.

GOLBECK, J.; ROBLES, C. G.; EDMONDSON, M.; TURNER, K. Predicting personality from twitter. *In:* IEEE THIRD INTERNATIONAL CONFERENCE ON PRIVACY, SECURITY, RISK AND TRUST AND 2011 IEEE

THIRD INTERNATIONAL CONFERENCE ON SOCIAL COMPUTING, Boston, MA, 2011. **Proceedings** […]. New York: IEEE, 2012. p. 149-156. DOI 10.1109/PASSAT/SocialCom.2011.33. Disponível em: https://ieeexplore.ieee.org/document/6113107. Acesso em: 19 set. 2023.

HICKMAN, L.; THAPA, S.; TAY, L.; CAO, M.; SRINIVASAN, P. Text preprocessing for text mining in organizational research: Review and recommendations. **Organizational Research Methods**, Thousand Oaks, v. 25, n. 1, p. 114-146, 2022. Disponível em: https://doi.org/10.1177/1094428120971683. Acesso em: 19 set. 2023.

HUTCHINS, W. J. The Georgetown-IBM experiment demonstrated in January 1954. *In:* CONFERENCE OF THE ASSOCIATION FOR MACHINE TRANSLATION IN THE AMERICAS, 6th, Washington, DC, 2004. **Proceedings** [...]. Berlin: Springer, 2004. p. 102-114. Disponível em: https://link.springer.com/chapter/10.1007/978-3-540-30194-3_12. Acesso em: 19 set. 2023.

MAHYOOB, M.; AL-GARAADY, J.; ALBLWI, A.; ALRAHAILI, M. Sentiment analysis of public tweets towards the emergence of SARS-CoV-2 Omicron variant: a social media analytics framework. **Engineering, Technology & Applied Science Research**, Pátras, v. 12, n. 3, p. 8525-8531, June 2022.

NATIONAL SCIENCE BOARD. Science and Engineering Indicators 2018. Broad-based, objective information on the U.S. and international S7E enterprise. Alexandria, VA: National Science Foundation, 2018. Disponível em: https://www.nsf.gov/statistics/2018/nsb20181/. Acesso em: 19 set. 2023.

OLIVEIRA, D. J. S.; BERMEJO, P. H. S.; PEREIRA, J. R.; BARBOSA, D. A. A aplicação da técnica de análise de sentimentos em mídias sociais como instrumento para as práticas da gestão social em nível governamental. **Revista de Administração Pública**, Rio de Janeiro, v. 53, n. 1, p. 235-251, jan./fev. 2019. DOI 10.1590/0034-7612174204.

PAES, V. J.; ARAÚJO, D.; BRITO, K.; ANDRADE, E. Análise de sentimento em tweets relacionados ao desmatamento da floresta amazônica. *In*: BRAZILIAN WORKSHOP ON SOCIAL NETWORK ANALYSIS AND MINING, 11., Niterói, 2022. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2022. p. 61-72. ISSN 2595-6094. Disponível em: https://sol.sbc.org.br/index.php/brasnam/article/view/20517. Acesso em: 19 set. 2023.

QAISER, S.; ALI, R. Text mining: use of TF-IDF to examine the relevance of words to documents. **International Journal of Computer Applications**, New York, v. 181, n. 1, p. 25-29, July 2018.

RAMOS, B.; FREITAS, C. "Sentimento de quê?": uma lista de sentimentos para a Análise de Sentimentos. *In*: SYMPOSIUM IN INFORMATION AND HUMAN LANGUAGE TECHNOLOGY (STIL), Oct. 15-18, 2019. **Anais** [...]. Salvador, BA, 2019. Disponível em: https://www.linguateca.pt/Repositorio/RamosFreitasSTIL2019.pdf. Acesso em: 19 set. 2023.

SHAUKAT, Z.; ZULFIQAR, A. A.; XIAO, C.; AZEEM, M.; MAHMOOD, T. Sentiment analysis on IMDB using lexicon and neural networks. **SN Applied Science,** Switzerland, v. 2, n. 148, p. 1-10, Jan. 2020. Disponível em: https://doi.org/10.1007/s42452-019-1926-x. Acesso em: 19 set. 2023.

SOUZA, Vinicius Augusto de; SOUZA, Érica Ferreira de; MEINERZ, Giovani Volnei. Análise de sentimento em tempo real de notícias do mercado de ações. **Brazilian Journal of Development**, Curitiba, v. 7, n. 1, p. 11084-11091, 2021. DOI 10.34117/bjdv7n1-758. Disponível em: https://www.brazilianjournals.com/index.php/BRJD/article/view/23959/19224. Acesso em: 19 set. 2023.

STENSON, Rob. Is This the First Time Anyone Printed, 'Garbage In, Garbage Out'?. **Atlas Obscura**. 14 Mar. 2016. Disponível em: http://www.atlasobscura.com/articles/is-this-the-first-time-anyone-printed-garbage-in-garbage-out. Acesso em: 19 set. 2023.

WANKHADE, Mayur; RAO, Annavarapu Chandra Sekhara; KULKARNI, Chaitanya. A survey on sentiment analysis methods, applications, and challenges. **Artificial Intelligence Review**, Berlin, v. 55, n. 7, p. 5731–5780, Oct. 2022. DOI 10.1007/s10462-022-10144-1. Disponível em: https://link.springer.com/10.1007/s10462-022-10144-1. Acesso em: 19 set. 2023.

DADOS DO AUTOR:

Guilherme Noronha

Guilherme Noronha é Graduado em Ciência da Computação pela PUC Minas com mestrado e doutorado em Gestão e Organização do Conhecimento pela UFMG. É entusiasta na área de dados com mais de dez anos de experiência e acumula trabalhos em diferentes áreas como proteção à privacidade, processamento de linguagem natural, aprendizado de máquina, modelagem, análise e engenharia de dados. Atualmente atuo como engenheiro de dados.

https://www.linkedin.com/in/noronha2001/
https://orcid.org/0000-0002-1422-2179
guilhermenoronha@2001@gmail.com

Como referenciar o capítulo 2:

NORONHA, Guilherme. Processamento de Linguagem Natural para análise de sentimento utilizando Python. *In*: SHINTAKU, Milton; MACÊDO, Diego José; MARIN, Luciano Heitor Gallegos (org.). **Tecnologias utilizadas em pesquisas acadêmicas em Ciências Sociais Aplicadas.** Brasília, DF: Ibict, 2023. cap. 2. p. 39-60. ISBN 978-65-89167-94-5. DOI: http://doi.org/10.22477/9786589167938cap2