

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM COMPUTACIONAL

**COMPARAÇÃO DE MODELOS DE ANÁLISE DE
SENTIMENTOS A PARTIR DE DADOS COLETADOS NA REDE
TWITTER: UM ESTUDO DE CASO A PARTIR DE
MENSAGENS A RESPEITO DE UM MODELO DE
SMARTPHONE**

por

Giovanna de Oliveira Moreira

Dissertação para obtenção do Título de
Mestre em Modelagem Computacional

Orientador: Prof. Dr. Leonardo Ramos Emmendorfer
Coorientador: Prof. Dr. Vinicius Menezes de Oliveira

Rio Grande, Agosto de 2022

Giovanna de Oliveira Moreira

“COMPARAÇÃO DE MODELOS DE ANÁLISE DE SENTIMENTOS A PARTIR DE DADOS COLETADOS NA REDE TWITTER: UM ESTUDO DE CASO A PARTIR DE MENSAGENS A RESPEITO DE UM MODELO DE SMARTPHONE”

Dissertação apresentada ao Programa de Pós-Graduação em Modelagem Computacional da Universidade Federal do Rio Grande - FURG, como requisito parcial para obtenção do Grau de Mestra. Área de concentração: Modelagem Computacional.

Aprovada em:

BANCA EXAMINADORA

Documento assinado digitalmente
 LEONARDO RAMOS EMMENDORFER
Data: 22/08/2022 15:40:10-0300
Verifique em <https://verificador.iti.br>

Prof. Dr. Leonardo Ramos Emmendorfer (Orientador – UFSM/PPGMC-FURG)

Documento assinado digitalmente
 EMANUEL DA SILVA DIAZ ESTRADA
Data: 23/08/2022 08:37:47-0300
Verifique em <https://verificador.iti.br>

Prof. Dr. Emanuel da Silva Diaz Estrada (FURG)

Documento assinado digitalmente
 GUSTAVO HENRIQUE DALPOSSO
Data: 22/08/2022 16:12:32-0300
Verifique em <https://verificador.iti.br>

Prof. Dr. Gustavo Henrique Dalposso (UTFPR)

Rio Grande - RS
2022

Ficha Catalográfica

M838c Moreira, Giovanna de Oliveira.

Comparação de modelos de análise de sentimentos a partir de dados coletados na rede *Twitter*: um estudo de caso a partir de mensagens a respeito de um modelo de *Smartphone* / Giovanna de Oliveira Moreira. – 2022.

77 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-Graduação em Modelagem Computacional, Rio Grande/RS, 2022.

Orientador: Dr. Leonardo Ramos Emmendorfer.

Coorientador: Dr. Vinicius Menezes de Oliveira.

1. Análise de sentimentos 2. Processamento de linguagem natural
3. Aprendizado de máquina 4. *Twitter* I. Emmendorfer, Leonardo
Ramos II. Oliveira, Vinicius Menezes de III. Título.

CDU 004.738.5

Catálogo na Fonte: Bibliotecário José Paulo dos Santos CRB 10/2344

Dedico meu trabalho ao meus pais Marcos Flávio Pinto Moreira
e Fernanda de Oliveira Moreira, ao meu querido irmão
Filipe de Oliveira Moreira, por toda ajuda, amor e carinho.
Em especial ao meu pai, nada disto seria possível sem o seu apoio e incentivo.

AGRADECIMENTOS

Ao meu orientador, Prof. Dr. Leonardo Ramos Emmendorfer, por toda a ajuda e ensinamentos durante essa etapa.

Ao meu co-orientador, Prof. Dr. Vinicius Menezes de Oliveira por teu aceitado embarcar nesta conosco.

Aos professores Dr. Emanuel da Silva Diaz Estrada e Dr. Gustavo Henrique Dalposso pelas sugestões e participações na minha qualificação e defesa.

Aos professores do PPGMC que direta ou indiretamente cruzaram meu caminho.

Ao Prof. Dr. Guilherme Dotto, pela co-autoria em um dos artigos.

Aos meus pais, ao meu irmão, ao meu namorado por todo amor e carinho.

Ao meu avô, à minha tia, por todo o auxílio em Rio Grande.

Ao meus sogros, ao meu cunhado, pelo carinho e auxílio.

Às minhas queridas amigas, Tatiana Schein e Leticia Santos, por toda a parceria, força e carinho.

Agradeço a CAPES pelo incentivo à minha formação através da concessão de bolsa de estudos na modalidade de mestrado.

RESUMO

A análise de sentimentos é um campo de estudo com potencial crescente de informação devido ao aumento na utilização das redes sociais, o que tem gerado uma quantidade enorme de dados disponíveis para análise em tempo real. Atualmente o Twitter (twitter.com) está entre as redes sociais mais utilizadas no mundo, sendo esta uma plataforma de grande interesse para pesquisadores e empresas no que diz respeito à análise de sentimentos. O Twitter permite ao usuário publicar mensagens em até 280 caracteres, links de vídeos e fotos, entre outros, além de permitir a obtenção destes dados. A análise de sentimentos proporciona *feedbacks* da opinião do público sobre vários tópicos, tais como a imagem de uma empresa e a aceitação de determinado produto, entre tantos outros. Na era computacional a análise de sentimentos é realizada com a utilização de algoritmos computacionais, existindo vários deles disponíveis na literatura para a aplicação da análise de sentimentos. Apesar disso, ainda há a necessidade da avaliação do emprego destes algoritmos quando se trata da análise de sentimentos de clientes. Todo o conjunto de dados presente no trabalho, passou pelas etapas de coleta, limpeza, pré-processamento e treinamento. Dessa forma, esta proposta de trabalho avaliou a utilização dos algoritmos Naive Bayes, *Support Vector Machine* e Regressão Logística para a análise de sentimentos sobre o *smartphone* iPhone a partir de mensagens coletadas na rede social Twitter. A partir das métricas de validação o Modelo Naive Bayes apresentou uma precisão de 75%, acurácia de 72% e F1-score de 75%. O modelo Regressão Logística apresentou uma precisão de 74%, acurácia de 71% e F1-score de 72%. Por fim, o modelo *Support Vector Machine* apresentou uma precisão de 74%, acurácia de 68% e F1-score de 71%. Mesmo utilizando a validação cruzada e um classificador de votação não houve aumento nas métricas dos modelos.

Palavras-chaves: Análise de sentimentos, Processamento de linguagem natural, Aprendizado de máquina, Twitter.

ABSTRACT

Sentiment analysis is a field of study with growing information potential due to the increase in the use of social networks, which has generated a huge amount of data available for analysis in real time. Currently Twitter (twitter.com) is among the most used social networks in the world, and this is a platform of great interest to researchers and companies when it comes to sentiment analysis. Twitter allows the user to publish messages in up to 280 characters, links to videos and photos, among others, and allows the collection of this data. Sentiment analysis provides feedbacks of the public's opinion on various topics, such as the image of a company and the acceptance of a certain product, among many others. In the computational age, sentiment analysis is performed using computer algorithms, and there are several of them available in the literature for the application of sentiment analysis. Despite this, there is still a need for evaluation of the use of these algorithms when it comes to customer sentiment analysis. All the data set present in this work went through the stages of collection, cleaning, pre-processing, and training. Thus, this work proposal evaluated the use of Naive Bayes, Support Vector Machine and Logistic Regression algorithms for sentiment analysis about the iPhone smartphone from messages collected on the social network Twitter. From the validation metrics the Naive Bayes Model showed a precision of 75%, accuracy of 72%, and F1-score of 75%. The Logistic Regression model showed a precision of 74%, accuracy of 71%, and F1-score of 72%. Finally, the Support Vector Machine model presented a precision of 74%, accuracy of 68%, and F1-score of 71%. Even using cross-validation and a voting classifier there was no increase in the metrics of the models.

Keywords: Sentiment Analysis, Natural Language Processing, Machine Learning, Twitter.

ÍNDICE

1	INTRODUÇÃO	14
1.1	Justificativa	15
1.2	Objetivo	15
1.3	Organização do documento	16
2	REFERENCIAL TEÓRICO	17
2.1	Processamento de linguagem natural	17
2.1.1	Técnicas baseadas em léxico	18
2.1.2	Técnicas baseadas em aprendizado de máquina	18
2.2	Modelos	20
2.2.1	Regressão Logística (LR)	20
2.2.2	Support Vector Machine (SVM)	23
2.2.3	Naive Bayes (NB)	26
2.3	Métricas de Avaliação	29
2.3.1	Precisão	30
2.3.2	Acurácia	31
2.3.3	Recall	31
2.3.4	F1-Score	32
2.3.5	Validação cruzada	32
2.3.6	Classificador de votação	33
3	TRABALHOS RELACIONADOS	35
4	MATERIAIS E MÉTODOS	39
4.1	Materiais	39
4.2	Métodos	42
4.3	Etapas do trabalho	43
5	RESULTADOS E DISCUSSÕES	44
5.1	Coleta de dados	44
5.2	Pré processamento	46
5.3	Treinamento e teste	55
5.4	Análise	58
5.5	Validação	59
6	CONCLUSÕES	64
7	SUGESTÃO PARA TRABALHOS FUTUROS	66
8	REFERÊNCIAS	67
9	APÊNDICE	73
9.1	APÊNDICE 1: SCRIPT ANÁLISE DA FREQUÊNCIA DE PALAVRAS	73
9.2	APÊNDICE 2: RESULTADOS VALIDAÇÃO CRUZADA	75

LISTA DE FIGURAS

Figura 2.1: Esquema do aprendizado supervisionado.	19
Figura 2.2: Curva logística.	21
Figura 2.3: Limite Linear.	23
Figura 2.4: Limite complexo.	24
Figura 2.5: Reorganização dos limites.	24
Figura 2.6: Exemplo machos e fêmeas.	25
Figura 2.7: SVM para machos e fêmeas.	26
Figura 2.8: Matriz de confusão.	29
Figura 4.1: Ilustração do método proposto para realização deste trabalho.	39
Figura 4.2: Logo Twitter.	40
Figura 4.3: Logo Apple.	41
Figura 5.1: Configuração dos dados.	49
Figura 5.2: Contagem dos rótulos.	50
Figura 5.3: Nuvem de palavras do conjunto de dados.	51
Figura 5.4: Distribuição de frequência de palavras em frases positivas.	54
Figura 5.5: Distribuição de frequência de palavras em frases negativas.	54
Figura 5.6: Conjunto de dados com frases positivas e negativas.	57
Figura 9.1: Naive Bayes validação cruzada.	75
Figura 9.2: Regressão Logística validação cruzada.	76
Figura 9.3: SVM validação cruzada.	77

LISTA DE TABELAS

Tabela 1:	$P(x, y)$	20
Tabela 2:	$P(y x)$	20
Tabela 3:	Coeficientes das variáveis do modelo de regressão	22
Tabela 4:	Classificação das casas	27
Tabela 5:	Artigos que empregaram métodos supervisionados na análise de sentimento.	38
Tabela 6:	Conjunto de dados sem a polaridade neutra.	50
Tabela 7:	Frases positivas.	52
Tabela 8:	Frases negativas.	52
Tabela 9:	Configuração dos dados.	57
Tabela 10:	Dados balanceados.	58
Tabela 11:	Resultado dos modelos.	60
Tabela 12:	Métricas das matrizes de confusão.	60
Tabela 13:	Média das métricas de validação cruzada dos modelos.	61

LISTA DE ABREVIATURAS

AB	<i>AdaBost</i>
API	<i>Application Programming Interface</i>
BNB	Bernoulli Naive Bayes
DP	<i>Deep Learning</i>
DT	<i>Decision Tree</i>
FN	Falso Negativo
FP	Falso Positivo
GNB	Gaussian Naive Bayes
IA	Inteligência artificial
JSON	<i>JavaScript Object Notation</i>
KN	<i>K-Nearest</i>
LR	<i>Regressão Logística</i>
LSVC	<i>Linear SVC</i>
ME	<i>Maximum Entropy</i>
ML	<i>Machine Learning</i>
MNB	Multi Naive Bayes
NB	Naive Bayes
PDCA	<i>Plan Do Check Act</i>
PLN	Processamento de linguagem natural
RF	<i>Random Forest</i>
RNC	Redes Neurais Covolucionais
SA	<i>Sentiment Analysis</i>
SPAM	<i>Sending and Posting Advertisement in Mass</i>
SVM	<i>Support Vector Machine</i>
TF-IDF	<i>Term Frequency – Inverse Document Frequency</i>
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo

LISTA DE CÓDIGO FONTE

1	Biblioteca coleta dados	44
2	Acesso aos dados do Twitter.	44
3	Especificações para a extração dos <i>tweets</i>	45
4	Exemplos de <i>tweets</i> extraídos.	45
5	Atributos presentes formato JSON.	46
6	Tokenização.	46
7	Resultado da tokenização.	46
8	Limpeza dos <i>tweets</i>	47
9	Exemplos de <i>tweets</i> extraídos após a limpeza.	47
10	Palavras mais frequentes de acordo com a polaridade.	51
11	Resultado da vetorização das frases de exemplo.	55
12	Mapeamento da primeira frase.	55
13	Mapeamento da segunda frase.	55
14	Vetorizando todos os <i>tweets</i> do conjunto de dados.	56
15	Ajustando a proporção de dados de treinamento e de teste.	56
16	Ajuste apenas dos dados de treinamento (X_{train}).	57
17	Balanceando os dados.	58
18	Sklearn Naive Bayes.	59
19	Sklearn Regressão Logística.	59
20	Sklearn SVM.	59
21	Vetorização.	60
22	Pontuação de precisão.	60
23	Classificador de votação.	62
24	Análise frequência de palavras a partir da polaridade	73
25	Naive Bayes validação cruzada.	75
26	Regressão Logística validação cruzada.	76
27	<i>Support Vector Machine</i> validação cruzada.	77

LISTA DE QUADRO

1	Configuração do método de validação cruzada.	33
2	Desempenho dos modelos.	61

1 INTRODUÇÃO

Atualmente é muito comum as pessoas expressarem suas opiniões na internet, especialmente nas redes sociais. Essas opiniões tornaram-se os dados fundamentais das empresas para a análise de sentimentos com o objetivo de verificar o quão positiva ou negativa é a sua relação/imagem com o consumidor através do produto/serviço/atendimento fornecidos. A partir dessa análise podem daí ser estabelecidas estratégias mais eficazes com o intuito de sobrevivência da empresa no mercado. Dessa forma, o uso da web para criação e compartilhamento tem gerado oportunidade para análise e mineração de dados (Pang & Lee, 2008) [46].

A Análise de Sentimentos, também conhecida como “Mineração de Opinião”, é um campo dentro do Processamento de Linguagem Natural (PLN) que constrói sistemas que identificam e extraem opiniões dentro de frases ou textos. Além de identificar a opinião, a análise ainda é capaz de extrair características como, por exemplo, sobre o assunto que está em questão, reconhecendo de quem veio essa opinião e se foi dito algo positivo ou negativo em relação ao assunto. As opiniões são essenciais para quase todas as atividades humanas e são as principais influenciadoras de nosso comportamento (Farhadloo & Rolland, 2016) [19].

Dessa forma, tornou-se relevante estudar estratégias para investigação desses conteúdos já que as informações públicas e privadas na Internet estão sempre crescendo. Com isso, existem cada vez mais textos que fornecem uma fonte rica de informações e evidências sobre diversos assuntos em sites de *reviews* sobre produtos, fóruns, blogs e em sua grande maioria mensagens compartilhadas em redes sociais.

A Análise de Sentimento é normalmente realizada a partir de um processamento de linguagem natural (PLN), ou seja, transforma informações não-estruturadas em dados estruturados. Este processo utiliza uma das duas abordagens básicas: classificadores baseados em regras, em que as regras derivadas do estudo linguístico são aplicadas à análise de sentimento, e classificadores de aprendizado de máquina em que algoritmos estatísticos são utilizados para aprender sinais de sentimentos automaticamente (Chong et al., 2014) [14].

Entretanto, a análise de sentimento não é uma tarefa fácil, pois geralmente é usada no processamento de um grande volume de textos opinativos que podem conter erros de ortografia, abreviações na escrita, conteúdo subjetivo, figuras de linguagem, “emojis”, figuras, entre outros. Por exemplo, imagine a seguinte situação, uma pessoa após um longo e exaustivo dia de trabalho chega em casa e se depara com a pia repleta de louça suja e diz: “Que maravilha! Eu não fiz nada o dia todo”. A um ser humano adulto seria completamente compreensível tal situação, ou seja, a fala do sujeito carrega uma figura de linguagem e quer dizer exatamente o oposto do que está dito. Mas e o algoritmo? Seria capaz de identificar o que realmente queria ser dito? É a partir de frases como essa, com um certo grau de complexidade e com uma exigência maior em relação aos detalhes que se inicia o processo cauteloso do processamento de linguagem natural.

1.1 Justificativa

As mídias sociais surgem nesse novo contexto como importantes ferramentas de comunicação e publicidade, podendo se traduzir em vantagem ou desvantagem competitiva, de acordo com a maneira como as empresas gerenciam sua utilização. É possível perceber ainda que com o desenvolvimento tecnológico e o crescimento da internet houve um aumento significativo no volume de informações disponíveis. A evolução destes meios de comunicação proporciona cada vez mais agilidade e facilidade na divulgação, bem como a compra e venda, aumentando a concorrência entre empresas que buscam a atualização constante de suas divulgações por informações (Dehghani, 2013) [17].

Com isso, as pessoas sentiram necessidade em estar conectadas em qualquer lugar e a qualquer hora do dia, o que gerou um grande avanço em tecnologias portáteis capazes de estarem presentes no dia a dia das pessoas. Os *smartphones* permitem uma conexão rápida e eficaz com o mundo todo, permitindo assim o consumo desenfreado por informação diariamente.

Essa necessidade de informação e conexão refletiu significativamente na procura por *smartphones*, tornando assim, a indústria cada vez mais agressiva. Com relação aos próximos cinco anos, a expectativa é de uma alta controlada nas vendas, porém otimista, segundo a consultoria de mercado: 1,5 bilhão de unidades em 2023 e 1,7 bilhão em 2025. As previsões se assemelham a um estudo publicado pela consultoria de mercado IDC, no qual ela prevê uma alta de cerca de 9% nas vendas em 2021 em relação à 2020 (IDC, 2020) [29].

Quando uma pessoa escreve um comentário ou uma avaliação, ela provavelmente comenta vários aspectos do produto (negativos e positivos) e isso é uma informação valiosa para as empresas, tanto para melhorar seus produtos e processos, como para melhorar sua imagem perante os consumidores.

No entanto, a análise de sentimentos não é um trabalho fácil. Existem vários idiomas no nosso planeta e o processamento da linguagem natural é recente no que diz respeito ao desenvolvimento de softwares e inteligência artificial. Portanto, há a necessidade de um estudo mais aprofundado na área de análise de sentimentos com relação aos algoritmos de identificação (Gaur & Sharma, 2017) [21].

1.2 Objetivo

O objetivo geral deste trabalho é comparar algoritmos baseados em aprendizado de máquina para realizar a classificação de sentimento ao serem aplicados em mensagens do Twitter (em português brasileiro) relacionados à compra do *smartphone* iPhone.

Para atingir o objetivo geral serão atingidos os seguintes objetivos específicos:

- Coletar mensagens relacionadas ao iPhone na rede social Twitter;
- Pré-processar os dados com o intuito de excluir caracteres, *emojis*, urls entre outros;
- Descrever fundamentos estatísticos que servirão de base para o código computacional;

- Adaptar algoritmos já existentes na literatura de acordo com a necessidade da pesquisa;
- Classificar dados afim de treinar os algoritmos;
- Testar os algoritmos a partir dos dados obtidos;
- Avaliar os algoritmos utilizando índices estatísticos;
- Comparar os resultados encontrados.

1.3 Organização do documento

Este documento está organizado da seguinte forma:

- Capítulo 2: Apresenta uma revisão sobre as abordagens mais comumente utilizadas para a Análise de Sentimento, além das métricas de avaliação empregadas no trabalho;
- Capítulo 3: Revisa os estudos encontrados na literatura;
- Capítulo 4: Apresenta os materiais e a métodos utilizados na pesquisa;
- Capítulo 5: Apresenta os resultados e discussões acerca do trabalho;
- Capítulo 6: Conclusões;
- Capítulo 7: Trabalhos futuros;
- Capítulo 8: Apresenta as referências utilizadas neste trabalho;
- Capítulo 9: Apêndices.

2 REFERENCIAL TEÓRICO

Nesta seção será apresentado todo o referencial teórico envolvido nesta proposta de trabalho.

A inteligência artificial (IA) tem se destacado com o objetivo prático de agrupar várias tecnologias, como redes neurais artificiais, algoritmos, sistemas de aprendizado, entre outros, que conseguem simular capacidades humanas ligadas à inteligência (Saleh, 2019) [57].

Além de a IA poder auxiliar na simplificação de processos de análise, especialmente quando se trata de tomada de decisão baseada em dados, ela é capaz de organizar dados e lhes conferir maior clareza na interpretação. O sistema ainda pode desenvolver processos que envolvam correlações, regressões e estruturação de análises dos dados gerados (Singh et al., 2013) [58].

Outra vantagem da utilização do sistema de inteligência artificial está na capacidade de avaliar uma grande quantidade de dados a uma velocidade relativamente elevada e retornar as informações mais relevantes à análise.

Todavia, apesar de todas as possibilidades a partir da IA, ainda é possível encontrar tecnologias que contribuem para que ela evolua e facilite a sua atuação, como é o caso do *Machine Learning* (ML), *Deep Learning* (DL) e o Processamento de Linguagem Natural (PLN) (Janiesch et al., 2021) [30].

2.1 Processamento de linguagem natural

Um dos desafios da área da inteligência artificial está na subárea denominada de Processamento de Linguagem Natural (PLN), a qual tem por objetivo reproduzir processos de desenvolvimento humano relacionados à linguagem. Entre as ferramentas empregadas neste estudo estão os algoritmos e a programação.

Por meio de processos, as máquinas podem compreender melhor os textos, o que envolve reconhecimento de contexto, extração de informações e desenvolvimento de resumos. O PLN pode ser usado em áreas como atendimento ao consumidor e na produção de relatórios corporativos a fim de auxiliar o entendimento da relação empresa e consumidor (Rindfleisch, 1996) [54].

A ação mais importante no Processamento de Linguagens Naturais é a conversão do texto em uma representação legível aos algoritmos computacionais, possibilitando assim a extração de informação contidas nos textos por meio de ferramentas tecnológicas.

Porém, a mineração de texto ainda é uma tecnologia em expansão e, por isso, ainda não há consenso sobre até onde vai a abrangência da recente área, mas todo o escopo de Processamento de Linguagem Natural se enquadra como mineração de texto (Witten et al., 2011) [70].

A classificação pode ser realizada por meio de: (a) técnicas de aprendizagem de máquina com o uso predominante de métodos supervisionados de aprendizagem como os de classificação ou de regressão (Medhat et al., 2014) [40], e (b) técnicas baseadas em léxico, ou seja, pontuação de palavras (Liu, 2010) [38] de acordo com o sentimento que as palavras contêm.

2.1.1 Técnicas baseadas em léxico

Métodos baseados em léxicos ou dicionários são métodos que utilizam listas e dicionários de palavras associadas a sentimentos específicos, palavras marcadas por polaridade ou pontuação de polaridade para determinar a pontuação geral de opinião de um determinado texto. Por exemplo, a orientação semântica ou polaridade é definida como uma medida da subjetividade e opinião de uma palavra, frase, sentença ou documento (Taboada et al., 2011) [63]. Assim, é possível verificar se uma frase (ou um documento) é positivo ou negativo.

Por exemplo, considere a frase “Eu gosto de abacaxi”. Nesse método, cada palavra recebe uma pontuação, a qual denominou-se de polaridade, podendo ter os valores -1, 0 ou +1, onde +1 é para palavras positivas, 0 para palavras neutras e -1 para palavras negativas. Nesse caso suponhamos que a pontuação da frase seja 0, +1, 0, +1. Neste contexto, há 2 palavras neutras, 2 positivas e nenhuma negativa. De maneira simples, a partir disso pode-se concluir que é uma frase positiva. Essa pontuação pode variar de acordo com o tipo de técnica que se trabalha. Alguns autores preferem ainda, construir o seu próprio dicionário de polaridades.

A principal vantagem desses métodos é que eles não precisam de dados de treinamento. Um léxico de sentimento útil conteria atribuições de orientação de polaridade (positiva e negativa) e também a força do sentimento ou, em alguns casos, o grau de centralidade para a categoria de sentimento.

Entretanto, eles são menos explorados em análise de sentimentos em comparação a outras abordagens. O léxico poderia ser muito bem aplicado a textos mais formais, como documentos ou reportagens, onde estão presentes frases formais, dentro dos padrões ortográficos da língua. No entanto, quando o interesse está nas redes sociais como o Twitter por exemplo, o léxico perde a capacidade de identificar o sentimento devido à singularidade da escrita na plataforma, já que contém um grande número de peculiaridades textuais (p.ex.: “n = não”, “vc= você”) e expressões coloquiais, além de possuir uma natureza dinâmica com novas expressões e *hashtags*. Nesses casos, fica claro a vantagem da criação do seu próprio dicionário de polaridades a partir do conjunto de dados.

2.1.2 Técnicas baseadas em aprendizado de máquina

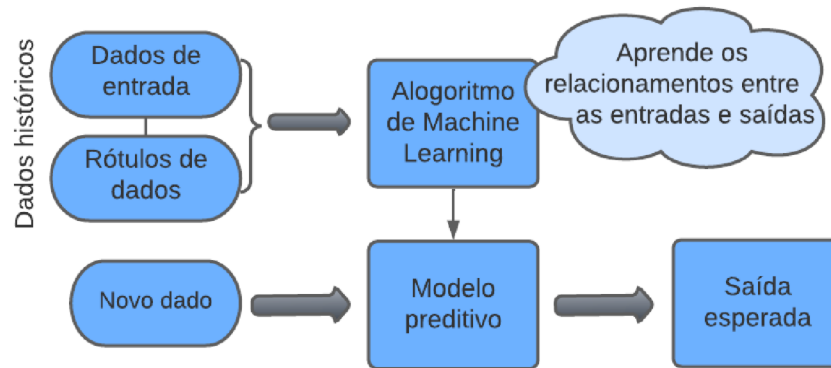
O Aprendizado de Máquina é considerado um subárea da Inteligência Artificial e tem por objetivo o desenvolvimento de técnicas e métodos que possibilitem o aprendizado do computador.

A análise de sentimento sob esta abordagem propõe que um conjunto de dados seja dividido em dois conjuntos, um de treinamento e outro para teste. O classificador aprende com os dados de treinamento e (às vezes) constrói um modelo que posteriormente é usado para classificar os documentos do conjunto de teste.

Alguns dos exemplos mais comuns de aprendizagem de máquinas são os algoritmos que *streamings* utilizam, por exemplo, a plataforma faz sugestões de filmes com base em filmes vistos no passado ou quando lojas recomendam outros produtos com base em produtos comprados

anteriormente. O objetivo é que a representação gerada seja capaz de produzir saídas corretas para novas entradas não apresentadas previamente. A figura 2.1 a seguir ilustra o funcionamento de um modelo de aprendizado supervisionado:

Figura 2.1: Esquema do aprendizado supervisionado.



Fonte: Autoria própria (2022).

Em geral, existem dois tipos diferentes de modelos de classificação: modelos generativos (Naive Bayes, Modelos Hidden Markov, etc.) e modelos discriminativos (Regressão Logística, *Support Vector Machine* (SVM), entre outros). A principal diferença é que um modelo generativo tenta modelar a distribuição de probabilidade conjunta e, em seguida, calcula a probabilidade condicional usando o teorema de Bayes, enquanto um modelo discriminativo modela diretamente a forma conjunta.

A abordagem de aprendizado de máquina supervisionado geralmente atinge uma precisão maior do que a abordagem não supervisionada para análise de sentimento. O aprendizado supervisionado ocorre quando o modelo aprende a partir de resultados pré-definidos, já no não supervisionado não existem resultados pré-definidos para o modelo utilizar como referência para aprender. Dessa forma, trabalhar com esse tipo de abordagem requer uma construção de um gigantesco número de dados. Também pode ser caro e demorado, além disso após construído em um domínio poderá implicar baixa precisão quando aplicado em domínio diferente (Akinsola, 2017) [1].

Modelos Generativos x Modelos Discriminativos

Modelos generativos e discriminativos são modelos de aprendizado de máquina muito utilizados. Por exemplo, Regressão Logística, SVM's são modelos discriminativos bastante conhecidos. Já como exemplo de modelos generativos comumente usado é possível citar Naive Bayes, Modelos Hidden Markov e entre outros (Osislogu, 2019) [44].

Classificadores generativos aprendem a distribuição de probabilidade conjunta $P(x, y)$. O principal objetivo é ser capaz de explicar como os dados são gerados, de modo que, a partir

da captura do processo que gerou os dados, o modelo possa realizar previsões sobre os novos exemplos. O diferencial do classificador generativo é ser capaz de aprender a distribuição de dados subjacentes. Para as previsões os classificadores generativos transformam a probabilidade conjunta ($P(x, y)$) em uma probabilidade condicional ($P(y|x)$) (lê-se: “probabilidade de y dados os valores de x ”), usando a regra de Bayes (Ulusoy & Bishop, 2006) [67].

Já de modo contrário, os classificadores discriminativos tentam encontrar limites que separam as classes. São testados e verificados todos os possíveis limites e o que possuir menor erro é selecionado. Esses limites podem ser rígidos ou flexíveis. Um limite flexível permite que alguns exemplos sejam classificados de maneira incorreta. Este tipo de limite é possível encontrar no modelo SVM.

Suponha que existam dados de entrada x e é necessário classificá-los em rótulos y . Como observado, o modelo generativo irá aprender a distribuição de probabilidade conjunta $P(x, y)$ e o modelo discriminativo irá aprender a distribuição de probabilidade condicional $P(y|x)$. Agora imagine que você tenha os seguintes dados no formulário (x, y) : $(1, 0)$, $(1, 0)$, $(2, 0)$, $(2, 1)$.

As Tabelas 1 e 2 representam o exemplo citado acima.

Tabela 1: $P(x, y)$		
	$y=0$	$y=1$
$x=1$	1	0
$x=2$	$1/2$	$1/2$

Fonte: Autoria própria (2022).

Tabela 2: $P(y x)$		
	$y=0$	$y=1$
$x=1$	1	0
$x=2$	$1/2$	$1/2$

Fonte: Autoria própria (2022).

A distribuição $P(y|x)$ é a distribuição natural para classificar um determinado exemplo x em uma classe y . Modelo de algoritmos generativos $P(x, y)$, que pode ser transformado $P(y|x)$ pela aplicação da regra de Bayes e depois usado para classificação.

2.2 Modelos

2.2.1 Regressão Logística (LR)

A regressão logística é uma abordagem que permite estimar a probabilidade associada à ocorrência de determinado evento diante de um conjunto de variáveis explanatórias, além de ser usada para determinar a saída ou resultado quando houver uma ou mais variáveis independentes.

A regressão logística pertence à família de classificadores conhecidos como classificadores exponenciais ou log-lineares. Basicamente, a regressão logística binomial se refere a um classificador que classifica uma observação em uma de duas classes, enquanto a regressão logística multinomial é usada ao classificar em mais de duas classes. Vale ressaltar que a regressão logística é um classificador discriminativo, sendo assim, a função de um classificador probabilístico é escolher qual rótulo de saída y atribuir a uma entrada x , escolhendo o y que maximiza essa probabilidade (Wankhade et al., 2017) [69].

Em um caso mais simples, da variável dependente Y assumir apenas dois possíveis estados (0 ou 1) e haver um conjunto de p variáveis independentes $X_1, X_2, \dots, X_p$. O modelo de regressão logística pode ser escrito da seguinte forma:

$$P(Y = 1) = \frac{1}{1 + e^{-g(x)}} \quad (2.1)$$

onde,

$$g(x) = B_0 + B_1X_1 + \dots B_pX_p \quad (2.2)$$

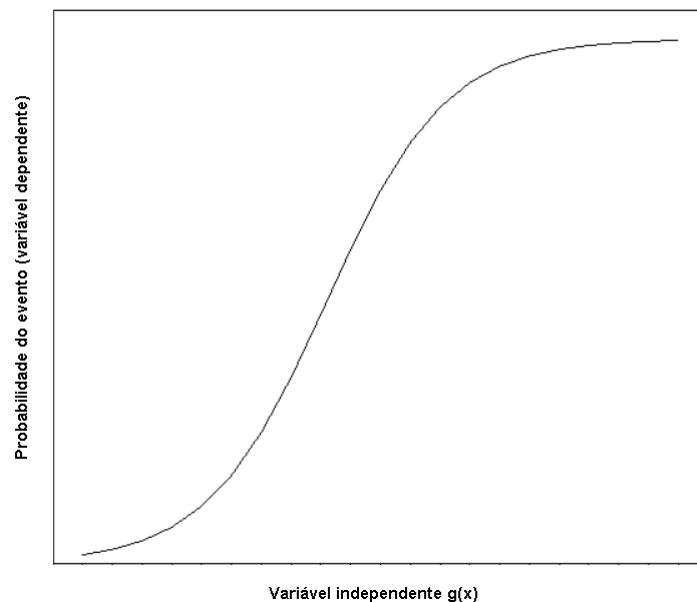
Os coeficientes $B_0, B_1, \dots, B_p$ são estimados a partir do conjunto dados pelo método da máxima verossimilhança. Considerando uma certa combinação de coeficientes $B_0, B_1, \dots, B_p$ e variando os valores de X (Hosmer & Lemeshow, 2000) [26].

Dessa forma:

- $g(x) \rightarrow +\infty$, então $P(Y = 1) \rightarrow 1$,
- $g(x) \rightarrow -\infty$, então $P(Y = 1) \rightarrow 0$.

Uma função logística ou curva logística tem um formato de S comum (curva sigmoide), como mostra a Figura 2.2.

Figura 2.2: Curva logística.



Fonte: Autoria própria (2022).

Algumas das interpretações a partir dos coeficientes são:

- Identifica-se que o impacto do coeficiente da variável independente sobre a razão de chances;

- Determina-se o efeito que os coeficientes exercem sobre a chance de um evento ocorrer;
- Ressalta-se que um coeficiente:
 - positivo aumenta a probabilidade;
 - negativo diminui a probabilidade.

Para utilizar o modelo de regressão logística para discriminação de dois grupos, a regra de classificação baseada em Hair et al., (1998)[24] é a seguinte:

- se $P(Y = 1) > 0,5$ então classifica-se $Y = 1$;
- se $P(Y = 1) < 0,5$ então classifica-se $Y = 0$.

Para obter-se uma boa estimativa da eficiência classificatória do modelo, recomenda-se separar a amostra em duas partes:

- uma parte para estimação do modelo;
- outra parte para testar a eficiência da classificação (*holdout sample*) (Hair et al., 1998) [24].

Utilizar a regressão logística possui algumas vantagens, por exemplo, sua facilidade em lidar com variáveis independentes categóricas, ela fornece resultados em termos de probabilidade, facilita a classificação em categorias, não necessita de um grande número de suposições além de possuir um alto grau de confiabilidade (Hosmer & Lemeshow, 2000) [26].

Para facilitar o entendimento, utilizou-se como exemplo a aplicação proposta no trabalho de Ferreira et al. (2012) [20].

A aplicação se baseia na análise de concessão de crédito de uma agência bancária localizada em Viçosa-MG. A base amostral foi composta por 74 operações de crédito realizadas no ano de 2007. A variável dependente foi a inadimplência, sendo atribuído ($Y = 1$) aos clientes inadimplentes e ($Y = 0$) aos clientes adimplentes. Já as variáveis independentes são a idade do cliente, o tempo de relacionamento com o banco, a renda, o limite do cheque especial, o estado civil e a escolaridade.

A Tabela 3 apresenta os coeficientes das variáveis do modelo de regressão.

Tabela 3: Coeficientes das variáveis do modelo de regressão

Variável	Coeficiente estimado
Idade Cliente (IC)	-1,68895
Tempo de relacionamento (TR)	26,67029
Renda (RD)	0,05831
Limite cheque especial (LC)	-0,09597
Estado Civil (EC)	29,65605
Escolaridade (ES)	-11,90192
Constante	-169,58336

Fonte: Adaptado de Ferreira et al. (2012).

Vale ressaltar que os sinais dos coeficientes são importantes para o julgamento dos resultados, um coeficiente positivo aumenta a probabilidade de ocorrência do evento, enquanto o negativo diminui. Nesse caso, o evento de interesse ($Y = 1$) é a inadimplência. A constante representa o β_0 no modelo de regressão. Dessa forma, o modelo de regressão é expresso pela equação 2.3:

$$\hat{p} = \frac{1}{1 + e^{-(-169,583 - 1,689 \times IC + 26,67 \times TR + 0,0583 \times RD - 0,096 \times LC + 29,656 \times EC - 11,902 \times ES)}} \quad (2.3)$$

Desta forma, a partir dos resultados obtidos, os autores fazem algumas observações, por exemplo, clientes de maior idade e maior grau de escolaridade tendem a ser adimplentes. Porém, ainda observou-se que clientes com maior tempo de relacionamento com o banco apresentam maiores problemas em relação à inadimplência.

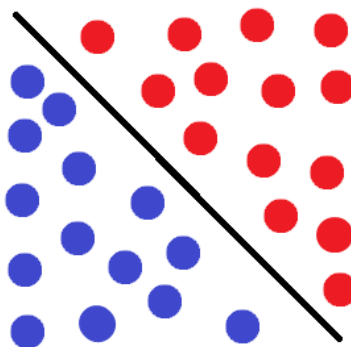
2.2.2 Support Vector Machine (SVM)

O Support Vector Machine (SVM) foi proposto pela primeira vez em 1992, apresentado por Boser, Guyon e Vapnik no COLT-92 (Boser et al., 1992) [9]. SVMs são um conjunto de métodos de aprendizado supervisionado usados para classificação e regressão cujo objetivo é categorizar determinado conjunto de pontos de dados.

Em outras palavras, o *Support Vector Machine* (SVM) é uma ferramenta de classificação e previsão de regressão que usa a teoria do aprendizado de máquina para ampliar a precisão, evitando assim, o ajuste excessivo aos dados. Os SVMs podem ser definidos como sistemas que usam as hipóteses de funções lineares em um espaço de características de alta dimensão, treinadas com um algoritmo de aprendizagem focado em otimização que implementa uma orientação de aprendizagem enfatizando a teoria de aprendizagem estatística (Awad & Khana, 2015) [7].

De uma forma mais dinâmica, ele separa um conjunto de objetos com diferentes classes utilizando o conceito de plano de decisão. Por exemplo, na Figura 2.3 é possível visualizar duas classes de objetos, azul ou vermelho. Essa linha define o limite em que se encontram os objetos.

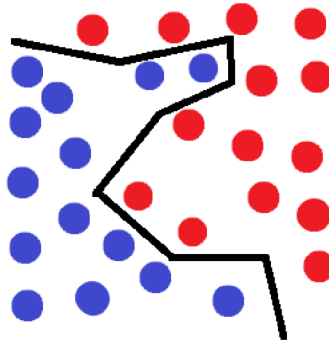
Figura 2.3: Limite Linear.



Fonte: Autoria própria (2022).

No entanto, os problemas costumam ficar mais elaborados quando é necessário realizar uma separação ótima por meio de estruturas mais complexas, como é possível observar na Figura 2.4. Ao entrarem novos objetos na análise, eles serão classificados como “azuis” ou “vermelhos”. Dessa forma, o SVM propõe a classificação de novos objetos (teste) com base nos dados disponíveis (treinamento).

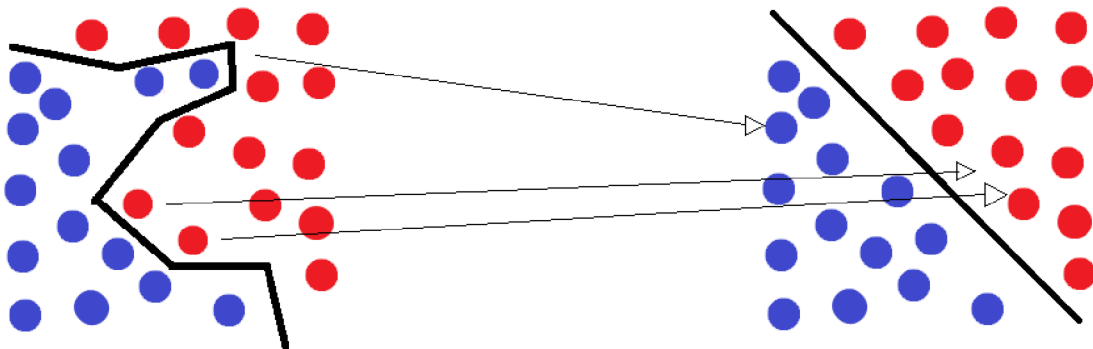
Figura 2.4: Limite complexo.



Fonte: Autoria própria (2022).

Na Figura 2.5 observa-se a complexidade em separar os objetos originais. A partir disso, utiliza-se um conjunto de funções matemáticas com o objetivo de reorganizar esses objetos, conhecido como *Kernels*.

Figura 2.5: Reorganização dos limites.



Fonte: Autoria própria (2022).

Dessa forma, em vez de construir uma curva complexa, como no esquema esquerdo, o SVM propõe, nesse caso, essa linha ótima capaz de separar os pontos azuis dos vermelhos.

Sendo assim, o SVM é usado para fins de classificação. Ele também pode ser usado em análises de regressão e detecção de *outliers* (dados que se diferenciam drasticamente de todos

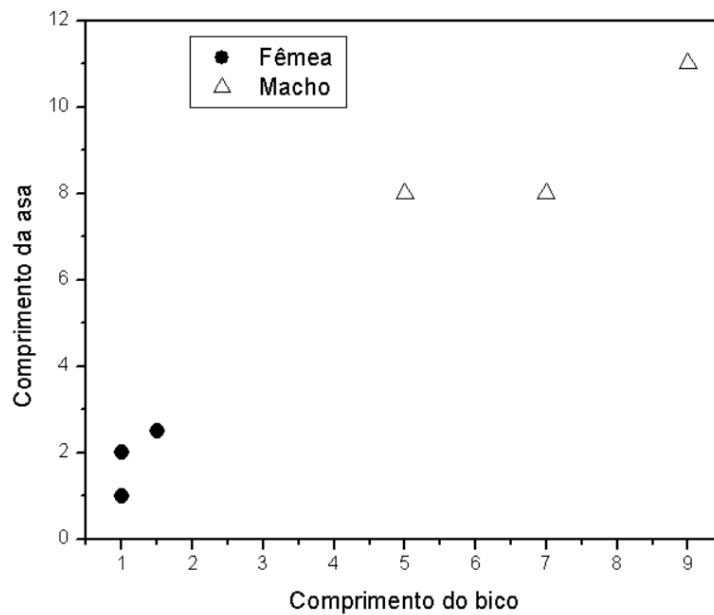
os outros). Além de evitar as armadilhas de representações dimensionais muito altas sendo assim, bastante utilizada quando se possui essas características (Lewis, 2004) [37].

Imagine a seguinte situação, existe uma população composta por 50% machos e 50% de fêmeas. A partir de uma amostra deseja-se criar um conjunto de regras para que seja possível classificar o gênero do restante da população.

Para simplificar, suponha que os dois fatores de diferenciação identificados sejam o comprimento da asa e o comprimento do bico.

Os círculos na Figura 2.6 representam as fêmeas e os triângulos representam os machos.

Figura 2.6: Exemplo machos e fêmeas.



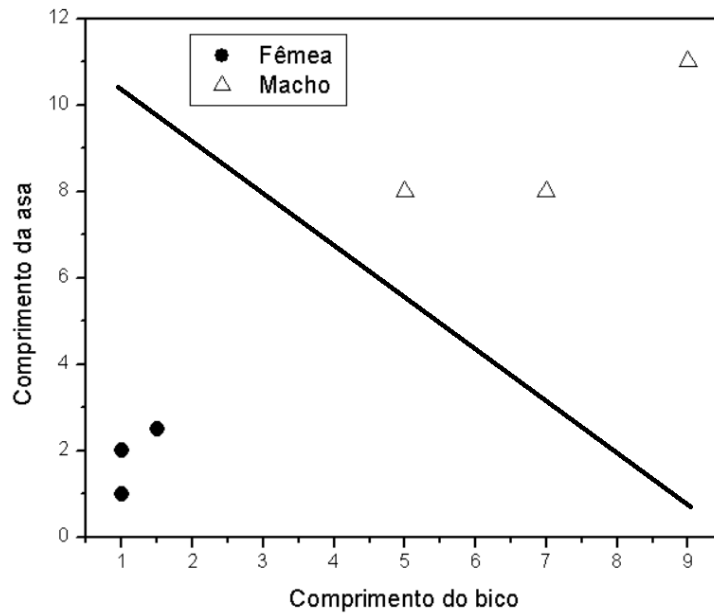
Fonte: Autoria própria (2022).

Alguns *insights* esperados do gráfico são:

1. Os machos da amostra têm um comprimento médio de bico e asa maior que as fêmeas;
2. As fêmeas da amostra tem um comprimento médio de bico e asa menor que o machos.

Observa-se que existe um indivíduo com 10cm de comprimento de asa e comprimento de bico de 8cm, o melhor palpite seria classificar esse indivíduo como um macho. Nesse caso, Vetores de Suporte são simplesmente as coordenadas da observação individual. Por exemplo, (4, 4) é um vetor de suporte que corresponde a uma fêmea, pois é uma fronteira que melhor segrega as classes, como mostra a Figura 2.7.

Figura 2.7: SVM para machos e fêmeas.



Fonte: Autoria própria (2022).

A maneira mais fácil de interpretar a função objetivo em um SVM é encontrar a distância mínima da fronteira do vetor de suporte mais próximo (isso pode pertencer a qualquer classe).

2.2.3 Naive Bayes (NB)

O algoritmo Naive Bayes é um classificador probabilístico baseado no “Teorema de Bayes” (representado pela equação (2.4)), o qual foi criado por Thomas Bayes (Rutten, 2010) [55].

$$prob(B|A) = \frac{prob(A|B)prob(B)}{prob(A)} \quad (2.4)$$

onde,

- $P(B|A)$ é a probabilidade posterior da classe (B, alvo) dada preditor (A, atributos).
- $P(B)$ é a probabilidade original da classe.
- $P(A|B)$ é a probabilidade que representa a probabilidade do preditor dada a classe.
- $P(A)$ é a probabilidade original do preditor.

De modo mais simples,

$$prob(B|A) = \frac{\text{Quantas vezes aparece } B}{\text{Quantidade total de } B \text{ no documento}} \quad (2.5)$$

As contagens de frequência das palavras são armazenadas em tabelas de hash (estrutura de dados especial que associa chaves de pesquisa a valores. Seu objetivo é fazer uma busca rápida e obter o valor desejado a partir de uma chave simples) durante a fase de treinamento. Também são removidas as palavras duplicadas do documento, pois elas não adicionam mais informação.

Atualmente o algoritmo ficou bastante conhecido na área de aprendizado de máquina, de modo a categorizar textos baseados na frequência que as palavras são usadas, dessa forma podendo ser usado para identificar se um *e-mail* é SPAM ou até mesmo sobre qual assunto se refere determinado texto (Tope, 2019) [64].

A característica predominante do algoritmo, a qual também o faz receber “*naive*” (ingênuo) no nome, é que ele desconsidera completamente a correlação entre as variáveis, ou seja, ele considera que a ocorrência de todas palavras são independentes umas das outras dado o contexto da categoria.

O algoritmo funciona de forma probabilística, ele faz uso direto do teorema de Bayes para probabilidades condicionais ao classificar os textos. Como é um método de aprendizado de máquina supervisionado, ele precisa de uma etapa de treino onde os estimadores ótimos para os parâmetros do modelo são calculados (Lewis, 1998; Goldberg, 1989) [36] [23].

Imagine a seguinte situação, um casal precisa analisar qual a melhor opção para a compra de sua primeira casa, para isso eles começam classificando todas as opções já visitadas como apresenta a Tabela 4.

Tabela 4: Classificação das casas

Material	Quintal	Tipo	Região	Boa escolha
Alvenaria	Grande	Térrea	Afastada	Sim
Alvenaria	Grande	Térrea	Central	Sim
Madeira	Grande	Sobrado	Residencial	Não
Alvenaria	Grande	Sobrado	Central	Sim
Alvenaria	Pequeno	Térrea	Residencial	Sim
Madeira	Pequeno	Térrea	Residencial	Sim
Madeira	Pequeno	Térrea	Afastada	Sim
Alvenaria	Pequeno	Sobrado	Central	Não
Madeira	Pequeno	Térrea	Central	Não
Alvenaria	Grande	Térrea	Residencial	Sim
Madeira	Grande	Sobrado	Afastada	Sim
Madeira	Pequeno	Sobrado	Afastada	Não
Alvenaria	Pequeno	Sobrado	Residencial	Sim
Madeira	Grande	Térrea	Residencial	Não

Fonte: Autoria própria (2022).

Com essa base já classificada, dada uma casa térrea feita de madeira que possui quintal pequeno na região residencial é uma boa opção? Essa casa pode ser representada por:

$$x = [madeira, pequeno, térrea, residencial]. \quad (2.6)$$

E a pergunta, em forma de probabilidade condicional:

$$P(y = sim|x). \quad (2.7)$$

Essa probabilidade deve ser lida como:

- Qual a probabilidade da casa ser uma boa opção dado que ela é de madeira, pequeno, térrea, residencial?

Pelo Teorema de Bayes, tem-se que:

$$P(y = \text{sim}|x) = \frac{P(x|y = \text{sim})P(y = \text{sim})}{P(x)}, \quad (2.8)$$

com $P(x|y = \text{sim})$ sendo denominado *likelihood*, $P(y = \text{seguro})$ é o *prior* e $P(x)$ o preditor.

Com essa fórmula, e a Tabela de exemplos é possível calcular a probabilidade de cada nova casa não classificada pertencer a uma classe ou outra. A partir disso obtém-se:

$$x = [\text{madeira}, \text{pequeno}, \text{térrea}, \text{residencial}]. \quad (2.9)$$

- $P(\text{madeira}) = 7/14$
- $P(\text{pequeno}) = 7/14$
- $P(\text{térreo}) = 8/14$
- $P(\text{residencial}) = 6/14$
- $P(x) = 2352/38416 = 0.06$

E, finalmente:

- | | |
|---|---|
| • $P(\text{madeira} \mid y = \text{sim}) = 3/9$ | • $P(\text{madeira} \mid y = \text{não}) = 4/5$ |
| • $P(\text{pequeno} \mid y = \text{sim}) = 4/9$ | • $P(\text{pequeno} \mid y = \text{não}) = 3/5$ |
| • $P(\text{térreo} \mid y = \text{sim}) = 6/9$ | • $P(\text{térreo} \mid y = \text{não}) = 2/5$ |
| • $P(\text{residencial} \mid y = \text{sim}) = 4/9$ | • $P(\text{residencial} \mid y = \text{não}) = 2/5$ |
| • $P(x y = \text{sim}) = 288/6561 = 0.04$ | • $P(x y = \text{não}) = 48/625 = 0.07$ |

Com isso pode-se calcular:

$$P(y = \text{sim} \mid x) = 0.04 \cdot 0.64/0.06 = 0.43$$

$$P(y = \text{não} \mid x) = 0.07 \cdot 0.36/0.06 = 0.42$$

Note que $P(y = \text{sim} \mid x) \neq 1 - P(y = \text{não} \mid x)$.

Resumindo, pode-se expressar da seguinte maneira:

$$\hat{y} = \underset{k}{\operatorname{argmax}}(C_k) \prod_{i=1}^n p(x_i|C_k) \quad (2.10)$$

onde,

- C_k : probabilidade da classe;

- $x_i|C_k$: probabilidades condicionais.

É possível observar, a partir do exemplo, o valor que corresponde a escolha ser sim, uma boa opção ou não, uma opção ruim. Apesar da proximidade dos resultados, acrescentar ou diminuir mais características, bem como outra opção de escolha pode gerar mudanças significativas nos resultados encontrados.

2.3 Métricas de Avaliação

Os modos mais utilizados para a avaliação dos resultados obtidos da classificação de texto são abrangência (*recall*), precisão (*precision*), acurácia (*accuracy*), F1-Score, validação cruzada e classificador de votação.

Para entender melhor cada métrica, primeiro é necessário entender alguns conceitos.

Matriz de confusão

Uma matriz de confusão é uma tabela que indica os erros e acertos do modelo, o que é feito comparando-se com o resultado esperado. A Figura 2.8 apresenta um exemplo de uma matriz de confusão (Markoulidakis et al., 2021) [39].

Figura 2.8: Matriz de confusão.

		Detectada	
		Sim	Não
Real	Sim	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Fonte: Autoria própria (2022).

sendo:

- VP : classificação correta da classe Positivo;
- FP : erro em que o modelo previu a classe Positivo quando o valor real era classe Negativo;
- VN : classificação correta da classe Negativo;
- FN : erro em que o modelo previu a classe Negativo quando o valor real era classe Positivo.

Para termos um exemplo, imagine que houvesse um modelo que, dada os sintomas de um paciente, dissesse se o mesmo estaria ou não com gripe. Porém, para esse contexto, utilizou-se uma lista com os dados reais, e uma lista de dados preditos pelo modelo, ambos dados criados. Utilizou-se 1 para sim e 0 para não. Dessa forma:

- valores-reais: [1, 0, 1, 0, 0, 0, 1, 0, 1, 0];
- valores-preditos: [1, 0, 0, 1, 0, 0, 1, 1, 1, 0].

Neste caso:

- Verdadeiro positivo (VP): ocorre quando no conjunto real, a classe que buscamos foi prevista corretamente. Por exemplo, quando o paciente está com gripe e o modelo previu corretamente que ele está com gripe;
- Falso positivo (FP): ocorre quando no conjunto real, a classe que buscamos prever foi prevista incorretamente. Exemplo: o paciente não está com gripe, mas o modelo disse que ela está;
- Verdadeiro negativo (VN): ocorre quando no conjunto real, a classe que não buscamos prever foi prevista corretamente. Exemplo: o paciente não está com gripe, e o modelo previu corretamente que ele não está;
- Falso negativo (FN): ocorre quando no conjunto real, a classe que não buscamos prever foi prevista incorretamente. Por exemplo, quando o paciente está com gripe e o modelo previu incorretamente que ele não está com gripe.

Assim, o modelo:

- Previu paciente doente 3 vezes corretamente (VP=3);
- Previu não doente 4 vezes corretamente (FP=4);
- Previu doente 1 vez incorretamente (VN=1);
- Previu não doente 2 vezes incorretamente (FN=2).

2.3.1 Precisão

Precisão se trata da eficiência na avaliação de uma classe (positivos ou negativos), sendo dada pela razão entre os verdadeiros da classe e a soma dos verdadeiros e falsos desta classe (Dalianis, 2018) [15]. Assim, para os positivos a precisão será:

$$Precisao = \frac{VP}{VP + FP} \quad (2.11)$$

sendo,

- VP : Verdadeiro Positivo;
- FP : Falso Positivo.

A precisão pode ser usada em uma situação em que os Falsos Positivos são considerados mais prejudiciais do que os Falsos Negativos.

Fazendo um breve cálculo do exemplo citado acima o modelo apresenta uma precisão de aproximadamente 43%.

2.3.2 Acurácia

A acurácia é a medida mais intuitiva, se trata de uma avaliação geral dos acertos, ou seja, é a razão entre o total de acertos e o total de avaliações, considerando as classes verdadeiro e falso (Vujovic, 2021) [68].

$$Acuracia = \frac{VP + VN}{VP + FP + FN + VN} \quad (2.12)$$

sendo,

- VP : Verdadeiro Positivo;
- FP : Falso;
- VN : Verdadeiro Negativo;
- FN : Falso Negativo.

A acurácia é uma boa indicação geral do desempenho do modelo.

Da mesma forma, realizando o cálculo com base no exemplo citado acima o modelo apresenta uma acurácia de aproximadamente 40%.

2.3.3 Recall

O *recall* se trata da medida que avalia a frequência com que o classificador classifica corretamente (Hossin, 2015) [27]. Esse método é definido como a razão entre verdadeiros positivos e a soma de verdadeiros positivos com negativos falsos:

$$Recall = \frac{VP}{VP + FN} \quad (2.13)$$

sendo,

- VP : Verdadeiro Positivo;
- FN : Falso Negativo.

O *recall* pode ser usada em uma situação em que os Falsos Negativos são considerados mais prejudiciais que os Falsos Positivos.

E por fim, realizando o cálculo com base no exemplo citado acima o modelo apresenta um recall de aproximadamente 60%.

2.3.4 F1-Score

O F1-Score, também chamado de F-score, é uma medida da precisão de um modelo em um conjunto de dados. É usado para avaliar sistemas de classificação binária, que classificam exemplos em “positivos” ou “negativos”.

O F-Score é uma forma de combinar a precisão e o *recall* do modelo, e é definido como a média harmônica da precisão e do *recall* do modelo. O F-Score é comumente usado para avaliar sistemas de recuperação de informações, como mecanismos de busca, e também para muitos tipos de modelos de aprendizado de máquina, em particular no processamento de linguagem natural. É possível ajustar o F-Score para dar mais importância à precisão sobre o *recall*, ou vice-versa. Os F-Score ajustados comuns são o F0.5-score ($\beta = 0.5$) e o F2-score ($\beta = 2$), bem como o F1-score ($\beta = 1$) padrão (Powers, 2011) [47].

A fórmula para o F1-score padrão é a média harmônica da precisão e do *recall*, como mostrado abaixo:

$$F_1 = \frac{2}{\frac{1}{recall} + \frac{1}{precisao}} = 2 \times \frac{precisao \times recall}{precisao + recall} \quad (2.14)$$

Dessa forma,

$$F_1 = \frac{VP}{VP + \frac{1}{2}(FP + FN)} \quad (2.15)$$

Um modelo perfeito tem um F-score de 1.

2.3.5 Validação cruzada

A validação cruzada é um método estatístico de avaliação e comparação de algoritmos de aprendizado, dividindo os dados em dois segmentos: um usado para aprender ou treinar um modelo e o outro usado para validar o modelo.

Na validação cruzada, os dados são primeiro particionados em k segmentos ou dobras de tamanho igual (ou quase igual). Subsequentemente, k iterações de treinamento e validação são realizadas de tal forma que dentro de cada iteração uma dobra diferente dos dados é mantida para validação enquanto as $k - 1$ dobras restantes são usadas para aprendizado (Refaeilzadeh et al., 2009) [53].

A acurácia da validação cruzada, por exemplo, é a média de todas as precisões alcançadas nos conjuntos de validação (Berrar, 2018) [8]. Esse processo é repetido várias vezes, gerando (aleatoriamente) novas partições de treinamento e teste a cada vez, como apresenta a Quadro 1.

Quadro 1: Configuração do método de validação cruzada.

iteração 1	iteração 2	iteração 3	iteração 4	iteração 5
teste	treinamento	treinamento	treinamento	treinamento
treinamento	teste	treinamento	treinamento	treinamento
treinamento	treinamento	teste	treinamento	treinamento
treinamento	treinamento	treinamento	teste	treinamento
treinamento	treinamento	treinamento	treinamento	teste

Fonte: Autoria própria (2022).

Por exemplo, suponha que você escolheu usar 10% dos seus 100 dados como dados de teste. Então, seu conjunto de teste na primeira repetição pode ser os pontos 64, **90**, 63, **42**, 65, 49, 10, 68, 96, 48. Na próxima execução, seu conjunto de teste pode ser **90**, 60, 23, 67, 16, 78, **42**, 17, 73, 26. Como as partições são feitas independentemente para cada execução, o mesmo ponto pode aparecer no conjunto de teste várias vezes.

A vantagem desse método é que a proporção da divisão de treinamento/teste não depende do número de iterações, o que é útil para grandes conjuntos de dados. Por outro lado, a desvantagem desse método é que você não está executando iterações suficientes, portanto, alguns exemplos de conjuntos de teste podem nunca ser selecionados ou outros podem ser selecionados mais de uma vez (Kohavi, 1995) [33].

2.3.6 Classificador de votação

O aprendizado de conjunto, ou método *Ensemble*, é uma meta-abordagem geral para aprendizado de máquina que busca um melhor desempenho preditivo combinando as previsões de vários modelos. Esse método combina de alguma forma suas decisões, ou seus algoritmos de aprendizado, ou diferentes visões de dados, ou outras características específicas para obter previsões mais confiáveis e precisas (Re & Valentini, 2012) [52].

Os tipos de técnicas *Ensemble* incluem:

- *Bagging or Bootstrap Aggregation*;
- *Boosting*;
- *Stacking Classifier*;
- *Voting Classifier*.

Um classificador de votação (*Voting Classifier*) é um modelo de aprendizado de máquina que treina em um conjunto de vários modelos e prevê uma saída (classe) com base em sua maior probabilidade de classe escolhida como saída (Anagnostopoulos & Skourlas, 2014) [4].

Os critérios de votação podem ser de dois tipos:

- **Votação Difícil:** A classe de saída prevista é uma classe com a maior maioria de votos, ou seja, a classe que teve a maior probabilidade de ser prevista por cada um dos classificadores.

Exemplo: Suponha que três classificadores previram a classe de saída (A, A, B) , então aqui a maioria previu A como saída. Portanto , A será a previsão final.

- Votação Suave: A classe de saída é a previsão baseada na média de probabilidade dada a essa classe.

Exemplo: Suponha que dada alguma entrada para três modelos, a probabilidade de previsão para classe A = (0.27, 0.42, 0.31) e B = (0.35, 0.48, 0.33). Portanto, a média para a classe A é 0.33 e B é 0.386 , o vencedor é claramente a classe B porque teve a maior probabilidade média de cada classificador.

3 TRABALHOS RELACIONADOS

Os classificadores mais utilizados para endereçar o problema de análise de sentimentos em Twitter são os classificadores *Support Vector Machine* e Naive Bayes.

Gautam e Yadav (2014) [22] para analisar os sentimentos de clientes a partir de *tweets* propuseram um conjunto de técnicas de aprendizado de máquina com análise semântica para classificação de frases e avaliações de produtos com base em dados do twitter. A técnica Naive Bayes apresentou um resultado melhor que SVM, que está sendo submetido a um modelo uni-grama, que dá um resultado melhor do que usá-lo sozinho. Além disso, a precisão é melhorada quando a análise semântica do WordNet é utilizada sozinha, levando-o de 88,2% para 89,9%. Entretanto, os autores afirmam que se o conjunto de dados de treinamento for aumentado existe uma melhora no processo de identificação de frases relacionadas ao vetor de recursos.

Neethue e Rajasree (2013) [43] analisaram postagens do Twitter sobre produtos eletrônicos como celulares, *laptops*, etc., usando a abordagem de Aprendizado de Máquina. Em seguida, apresentaram um novo vetor de recursos para classificar os *tweets* como positivos, negativos e extrair a opinião das pessoas sobre os produtos. Na primeira etapa, os recursos específicos do Twitter são extraídos e adicionados ao vetor de recursos. Depois disso, esses recursos são removidos dos *tweets* e novamente a extração de recursos é feita como se fosse feita em um texto normal. A precisão da classificação do vetor de características é testada usando diferentes classificadores como Naive Bayes, SVM, entre outros classificadores. Todos esses classificadores apresentaram uma precisão quase semelhante para o novo vetor de recursos, sendo assim, esse vetor de recursos tem um bom desempenho para o domínio de produtos eletrônicos.

Hasan et al., (2018) [25] apostam na adoção de uma abordagem híbrida que envolve um analisador de sentimento que incluiu o aprendizado de máquina. Além disso, esta pesquisa se utiliza de vários analisadores de sentimento com algoritmos de aprendizado de máquina para determinar a abordagem com a maior taxa de precisão para aprender sobre sentimentos eleitorais. A pesquisa focou em comparar os léxicos de sentimento (W-WSD, SentiWordNet, TextBlob), além de testar seus resultados com dois classificadores diferentes NB e SVM. Embora os resultados do TextBlob tenham sido relativamente melhores, obtiveram o melhor resultado ao analisar *tweets* com W-WSD, como fica claro nos resultados.

Nas pesquisas de Clark e Wicentowski (2013) [13], o sentimento de frases individuais em um *tweet* é rotulado usando uma combinação de classificadores treinados em uma variedade de recursos lexicais. Os classificadores são combinados estimando a precisão dos classificadores em cada *tweet*. O desempenho é medido ao usar apenas os dados de treinamento fornecidos e separadamente ao incluir dados externos. Os autores perceberam que, ao estimar a confiança do classificador por frase, o sistema pode priorizar classificadores confiáveis e ignorar os menos confiáveis antes da combinação. Dessa forma, essa diferença no desempenho pode ser uma indicação de que o conjunto de dados suplementares era mais ruidoso do que esperavam ou que era mais aplicável ao domínio do que antecipavam. É perceptível também uma diferença entre a precisão da classificação positiva e negativa. Eles justificam que essa diferença é provavelmente

devido a um viés positivo no conjunto de treinamento utilizado (os dados de treinamento fornecidos são 64% positivos, 36% negativos) ou uma seleção de recursos que favoreceram o sentimento positivo.

Nas pesquisas feitas por Ramadhan et al., (2017) [50] os autores na primeira etapa coletaram dados usando a API (Application Programming Interface), um conjunto de normas que possibilita a comunicação entre plataformas através de uma série de padrões e protocolos do Twitter com os nomes de cada candidato na Eleição do Governador de Jacarta. Os dados coletados então se tornaram entrada para a etapa de pré-processamento. Na próxima etapa objetivou-se extrair o recurso de cada *tweet* a ser listado. A lista de recursos foi transformada em vetor de recursos na forma binária e transformada novamente usando o método TF-IDF (*Term Frequency - Inverse Document Frequency*), onde TF calcula a frequência em que o termo aparece no documento, ou seja, quanto maior for a frequência no documento, maior será a importância do termo. Já o cálculo do IDF considera que termos que se repetem frequentemente nos textos.

O conjunto de dados consistiu em dois tipos de dados, treinamento e teste. O treinamento foi rotulado manualmente. Já a validação cruzada K-Fold, que consiste em dividir o conjunto total de dados em k subconjuntos mutuamente exclusivos do mesmo tamanho e, a partir daí, um subconjunto é utilizado para teste e os $k-1$ restantes são utilizados para estimação dos parâmetros, fazendo-se o cálculo da acurácia do modelos testando assim o desempenho do algoritmo. Dessa forma, com base no resultado do teste, a precisão obtida atingiu 74% em média com a proporção dos dados de treinamento e dados de teste em 90:10. Os autores ainda concluíram que de todos os experimentos, o parâmetro mais influente na obtenção de bons resultados é a composição dos dados de treinamento e teste. Quanto mais dados de treinamento em comparação com a quantidade de dados de teste, maior será a precisão obtida.

No trabalho realizado por Prabhat e Khullar (2017) [48], os autores utilizaram Naïve Bayes e Regressão Logística para a classificação de comentários no Twitter. O desempenho dos algoritmos foi avaliado com base em diferentes parâmetros, como acurácia e precisão. A análise com regressão logística fornece resultados 10,1% mais precisos e 4,34% mais precisos com quase um quinto do tempo de implementação para o mesmo tamanho de conjunto de dados. Como neste artigo apenas *tweets* de texto foram usados, os autores afirmam que outros experimentos podem ser organizados em *tweets* de imagem.

Da Silva et al., (2014) [16] apresentaram uma abordagem que classifica automaticamente o sentimento de *tweets* usando conjuntos de classificadores em léxicos. Os *tweets* foram classificados como positivos ou negativos em relação a um termo de consulta. O conjunto de classificadores foi formado por Multinomial Naive Bayes, SVM, *Random Forest* e regressão logística. O trabalho baseou-se em demonstrar que conjuntos de classificadores formados por componentes diversificados - especialmente se vierem de diferentes fontes de informação, como dados textuais, *emojicons* e léxicos - podem fornecer resultados de última geração para esse domínio específico. Também foram comparadas estratégias promissoras para a representação de *tweets* (ou seja, *bag-of-words* e *hashing* de recursos) e mostrado suas vantagens e desvantagens.

O hashing (algoritmo que mapeia dados de comprimento variável para dados de comprimento fixo) de recursos demonstrou ser uma boa escolha no cenário de análise de sentimento de *tweet*, em que o esforço computacional é de suma importância. No entanto, quando o foco está na precisão, a melhor escolha é um pacote de palavras.

O trabalho realizado por Symeonidis et al., (2018) [62] comparou experimentalmente 16 técnicas de pré-processamento comumente usadas em dois conjuntos de dados do Twitter para Análise de Sentimentos, empregando quatro algoritmos de aprendizado de máquina populares: Linear SVC, Bernoulli Naïve Bayes, Regressão Logística e Redes Neurais Convolucionais. Avaliaram-se então, as técnicas de pré-processamento com base na precisão da classificação resultante e no número de recursos que produzem. Os experimentos de análise de sentimento no Twitter mostrou que algumas técnicas fornecem melhores resultados na classificação para ambos os conjuntos de dados usados, enquanto outras diminuem a precisão. As técnicas recomendadas são lematização (técnica de busca de palavras), substituição de repetições de pontuação, substituição de contrações e remoção de números. As técnicas não recomendadas pelos autores são: a remoção de pontuação, marcação de palavras em maiúsculas e tratamento de palavras em maiúsculas, substituição de gírias, substituição de negações por antônimos e correção ortográfica. Dependendo do classificador, os resultados variam e, se combinar essas técnicas, pode-se obter resultados diferentes. Os autores ainda concluem que a melhor maneira é substituir URLs e menções do usuário, substituir contrações, remover números, substituir repetições de pontuação e lematizar.

Neste artigo Rane e Kumar (2018) [51], trabalharam em um conjunto de dados composto de *tweets* para 6 das principais companhias aéreas da US *Airline Service* e realizaram uma análise de sentimento multiclasse. Essa abordagem começa com técnicas de pré-processamento usadas para limpar os *tweets* e, em seguida, representa esses *tweets* como vetores para fazer uma análise em nível de frase. A análise foi realizada usando 7 estratégias de classificação diferentes: árvore de decisão, *Random Forest*, SVM, *K-Nearest Neighbors*, regressão logística, Gaussian Naïve Bayes e AdaBoost. Os classificadores foram treinados com 80% dos dados e testados com os 20% restantes. O resultado do conjunto de teste é o sentimento do *tweet* que será: positivo, negativo ou neutro. As precisões foram calculadas para fazer uma comparação entre cada abordagem de classificação e a contagem geral de sentimento foi visualizada combinando todas as seis companhias aéreas. Por fim, os autores concluem que o classificador AdaBoost fornece uma precisão de 84,5%, a maior obtida, e afirmam que as precisões obtidas pelos classificadores são altas o suficiente para serem usadas pelo setor de investigação da satisfação do cliente.

Jianqiang e Xiaolin (2017) [31] estudaram que seis métodos diferentes de pré-processamento e como afetam a classificação da polaridade de sentimento no Twitter. Foi conduzido uma série de experimentos usando quatro classificadores para verificar a eficácia de vários métodos de pré-processamento em cinco conjuntos de dados do Twitter. Os resultados experimentais indicam que a remoção de URLs, a remoção de palavras de interrupção e a remoção de números afetam minimamente o desempenho dos classificadores. Além disso, substituir a negação e expandir os acrônimos (palavras formas pela inicial de outras palavras, ex: Anatel, Agência Nacional de

Telecomunicações) melhora a precisão da classificação.

Portanto, remover palavras de parada, números e URLs é apropriado para reduzir o erro, mas não afeta o desempenho. Já substituir a negação é eficaz para a análise de sentimento. Os classificadores Naive Bayes e *Random Forest* são mais sensíveis do que os classificadores de Regressão Logística e Máquina de Vetores de Suporte quando vários métodos de pré-processamento foram aplicados.

A Tabela 5 sintetiza os artigos citados neste capítulo a respeito de suas abordagens no estudo da análise de sentimentos.

Tabela 5: Artigos que empregaram métodos supervisionados na análise de sentimento.

Ref	Autores	Ano	Abordagem	Métodos
17	Gautam, G., Yadav, D.	2014	híbrida	NB, ME, SVM
18	Neethu, M. S., Rajasree, R.	2013	aprendizado	NB, ME, SVM
19	Hasan et al.	2018	aprendizado	NB, SVM
20	S. Clark, R. Wicentwoski.	2013	aprendizado	NB
21	Ramadhan et al.	2017	aprendizado	LR
22	Prabhat, A., Khullar, V.	2017	aprendizado	NB, LR
23	Da Silva et al.	2014	aprendizado	MNB, LR, SVM, RF
24	Symeonidis et al.	2018	híbrida	LSVC, BNB, LR, RNC
25	Rane, A., Kumar, A.	2018	aprendizado	RF, DT, SVM, KN, GNB, AB, LR
26	Jianqiang, Z., Xiaolin, G.	2017	aprendizado	NB, LR, SVM, RF

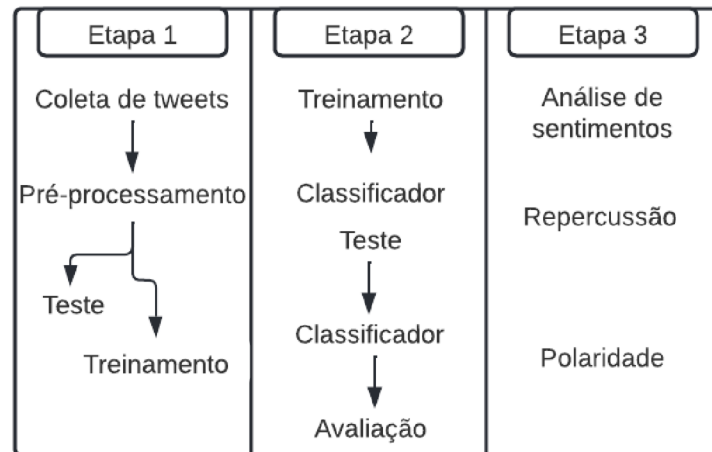
Fonte: Autoria própria (2022).

A revisão bibliográfica realizada mostrou que existem diversos grupos trabalhando com análise de sentimento e mineração de texto em redes sociais, para diferentes áreas de conhecimento. Entretanto, o foco desta revisão foi direcionado a trabalhos que utilizaram a técnica de AS e o Twitter como fonte de dados. Cada trabalho citado utilizou métodos de classificação diferentes e resultados satisfatórios são conseguidos de acordo com os autores. Com isso, foi possível notar que não existe um método ótimo para qualquer aplicação de classificação de sentimento em textos, sendo assim cada aplicação, conjunto de dados e método escolhido apresentam suas particularidades. Neste estudo serão utilizados três métodos diferentes para um mesmo conjunto de dados para a realização de AS de mensagens em português brasileiro compartilhadas na rede social Twitter, sendo também utilizados algoritmos de aprendizagem de máquina como na maioria dos trabalhos expostos acima.

4 MATERIAIS E MÉTODOS

Esta pesquisa baseou-se na execução de três etapas principais para cumprir o objetivo proposto: aprimoramento dos classificadores de sentimento, treinamento do classificadores e experimentação da análise de sentimento, repercussão e popularidade com mensagens do Twitter sobre *smartphones* iPhone. A Figura 4.1 apresenta uma visão geral do método e ilustra resumidamente esse fluxo de desenvolvimento da pesquisa.

Figura 4.1: Ilustração do método proposto para realização deste trabalho.



Fonte: Autoria própria (2022).

A seguir estão descritos os materiais, assim como os métodos utilizados para a realização desta pesquisa.

4.1 Materiais

A seguinte infraestrutura de hardware foi utilizada para realizar a pesquisa e desenvolvimento do trabalho:

- Laptop Acer Aspire 5, processador Intel Core™ i5-7200U 2.5GHz with Turbo Boost up to 3.1 GHz, NVIDIA GeForce MX150 with 2GB VRAM, 8 GB DDR4 Memory, 1000 GB HDD, para pesquisa bibliográfica, coletar e armazenar os dados, construção e testes do classificador e elaboração do experimento, análise dos resultados e escrita da dissertação.

Adicionalmente, os seguintes softwares e tecnologias utilizados para dar suporte ao desenvolvimento da pesquisa:

- Linguagem de programação Python (<https://www.python.org/>), de uso livre sob licença GPL, para todo o desenvolvimento do trabalho.
- IDE PyCharm (<https://www.jetbrains.com/pt-br/pycharm/download/>) sob licença Apache, para construção do classificador de sentimento.

- Distribuição da linguagem de programação Python:

Anaconda (<https://www.anaconda.com/>), sob licença BSD, para construção do classificador de sentimento.

- Microsoft Excel (<https://www.microsoft.com>), para a classificação de sentimento.

Rede Twitter

A rede social Twitter foi fundada em março de 2006 por Jack Dorsey, Evan Williams e Biz Stone, nos Estados Unidos. A plataforma oferece aos usuários um espaço para conversação e compartilhamento de conteúdos escritos, fotografias, vídeos e áudios. Além disso, uma das principais ferramentas do Twitter, os *Trending Topics*, disponibiliza aos seus usuários os assuntos mais falados do mundo no momento. Atualmente a rede social conta com mais de 316 milhões de usuários utilizando cerca de 35 idiomas (Twitter, 2022) [65]. O Twitter pode ser facilmente acessado pelo seu site ou em dispositivos móveis como aplicativo e encontrado através da sua marca mundialmente conhecida como “passarinho”, apresentado na Figura 4.2.

Figura 4.2: Logo Twitter.



Fonte: Twitter.com

O Twitter (twitter.com) está entre as redes sociais mais utilizadas no mundo. Plataformas assim permitem que o usuário filtre os interesses por profissão, afinidade, amizades ou interesses em comum, além de possibilitar compartilhamento de conhecimento e opiniões sobre diversos temas. No Twitter é possível publicar mensagens em até 280 caracteres, links de vídeo e fotos para que os seguidores confirmem o que está sendo feito, onde se está ou até mesmo a opinião sobre um determinado assunto.

Além disso, o Twitter dispõe de ferramentas que oferecem aos pesquisadores o acesso aos seus dados para que possam realizar análises e estudos (Albuquerque et al., 2013) [2]. Estes dados oferecem uma imensidade de informações e opiniões sobre diversos assuntos. Os dados podem ser coletados da API Twitter (<https://developer.twitter.com/en>). Essa plataforma permite amplo acesso aos dados públicos do Twitter para que desenvolvedores criem, analisem e auxiliem empresas privadas ou a própria plataforma. Os *tweets* estão gratuitamente disponíveis para desenvolvedores de software através de APIs públicas. Essas APIs podem ser classificadas em dois tipos (Kumar et al., 2013) [35]:

REST APIs: usam a estratégia *pull* para recuperação de dados, ou seja, para coletar informações o usuário deve explicitamente fazer uma solicitação.

Streaming APIs: usam a estratégia *push* para recuperação de dados, ou seja, depois que uma solicitação de informações é feita, a API fornece um fluxo contínuo de atualizações sem necessitar nenhuma outra solicitação.

As APIs do Twitter incluem uma ampla variedade de dados que podem ser divididos em cinco grupos:

1. Contas e usuários;
2. Mensagens diretas;
3. Anúncios;
4. Ferramentas de *publisher* e SDKs;
5. *Tweets* e respostas (Twitter, 2021) [66].

Nesta pesquisa serão utilizados *tweets* e respostas por estarem disponíveis para os desenvolvedores, sendo permitido que acessem os *tweets* pesquisando por palavras-chave específicas ou solicitando conteúdo de contas específicas (Twitter, 2021) [66].

iPhone

No dia primeiro de abril de 1976, em Cupertino, no Vale do Silício que existia na Califórnia, em uma garagem que Steve Jobs, Steve Wozniak e Ronald Wayne começaram a produzir computadores em miniatura e assim surgiu a Apple. Mas só no ano de 2007 que foi apresentado por Steve Jobs o primeiro celular produzido pela empresa, o iPhone. E a partir disso já foram produzidos 34 modelos de iPhone diferentes, e desde então todos os anos nasce um novo modelo de iPhone.

A empresa conhecida mundialmente nem sempre foi como conhecemos hoje, o primeiro logotipo da Apple mostrava Isaac Newton embaixo de uma macieira. Porém, foi usado por pouco tempo, pois seria mais complicado reproduzi-lo. Alguns anos depois tornou-se o como é conhecida hoje em dia, a maçã mordida, onde faz associação da maçã com o conhecimento e a famosa mordida, em inglês, bite, que obviamente lembra byte, apresentada na Figura 4.3 (Alves, 2022) [3].

Figura 4.3: Logo Apple.



Fonte: Apple.com

Dessa forma, tomou-se como objeto de estudo a palavra iPhone para a avaliação dos modelos para análise de sentimento. A escolha do *smartphone* iPhone deu-se pelo fato da empresa Apple possuir uma grande diversidade de opiniões a partir de seus produtos, especialmente os celulares. Além de existir diversos concorrentes diretos que assim como a Apple, possuem uma frequência de lançamentos similar contribuindo para a disputada do melhor produto.

4.2 Métodos

Coleta dos dados

As mensagens utilizadas para a classificação foram coletadas da rede social Twitter. A escolha do Twitter se deu pelo fato de ser a rede social em maior ascensão atualmente e por oferecer uma ampla coleção de bibliotecas de acesso (<https://dev.twitter.com>). Além disso, os *tweets* postados são públicos por padrão, ou seja, qualquer usuário mesmo sem acesso ao Twitter pode ler os *tweets* sem necessidade de solicitar permissão específica.

Pré-processamento

Na etapa de pré processamento foi realizada toda a limpeza e iniciou-se a adaptação do texto para uma futura representação mais estruturada. O objetivo dessa etapa é detectar e excluir os *tweets* que são spam dentre os coletados na etapa anterior de forma a manter apenas os *tweets* relevantes.

Esse primeiro passo diz respeito à tokenização, ou seja, transformação de cada fala (strings contíguas) em uma sequência de palavras. Logo após, foram removidos as *stopwords* (termos não representativos para um documento como artigos, preposições, conjunções e pronomes), nomes próprios e nomes de cidades e estados brasileiros, bem como *emoticons* (representação gráfica de uma emoção feita através dos caracteres) e diversas outras palavras que se tornem insignificante na interpretação e no conteúdo de cada tweet, assim como pontuações.

Treinamento e teste

O algoritmo de aprendizagem visa encontrar regularidades estatísticas de forma que seja possível segmentar a base em subconjuntos similares entre si e dissimilares a objetos pertencentes a outros grupos (Hotho et al., 2005) [28]. Dessa forma, optou-se por dividir o conjunto de dados numa proporção 80:20, como proposto por Cavalcante e Barbosa (2017) [11].

Análise

A análise é o núcleo do processo da mineração de textos. Nesta etapa são aplicados algoritmos específicos capazes de extrair conhecimento não trivial e útil para algum processo de tomada de decisão. Nessa etapa se realiza a análise de sentimento dos *tweets* selecionados. Para isso, foram utilizados três modelos já existentes na literatura (SVM, NB, LR) bem como a adaptação de algoritmos também já existentes.

Validação

Medidas quantitativas e análises qualitativas foram aplicadas para verificar se o conhecimento extraído é realmente válido e relevante ao assunto estudado. Para validar o desempenho dos modelos de classificação, uma parte dos dados chamada de conjunto teste foi selecionada aleatoriamente e separada para não fazer parte do conjunto de treinamento, dessa forma esse conjunto foi testado e comparado com os verdadeiros rótulos. Assim foi possível verificar o desempenho dos classificadores.

Para problemas de duas classes (nesse caso positivo e negativo) quatro medidas são obtidas a partir da aplicação do classificador, são elas VP, VN, FP e FN, que podem ser vistos na matriz de confusão. A partir dessas medidas foi possível calcular taxas, e obter a acurácia, a precisão e o *recall*, bem como a realização do teste *F-score* e a utilização de métodos como a validação cruzada e um classificador de votação.

4.3 Etapas do trabalho

As etapas deste trabalho foram a “Coleta dos dados”, o “Pré-processamento”, o “Treinamento e teste”, a “Análise” e a “Validação”.

Perceba que as etapas do trabalho se confundem com os próprios métodos empregados nesta pesquisa. Dessa forma, no capítulo seguinte, “Resultados e Discussões”, estão apresentados os resultados obtidos para cada etapa deste trabalho, organizados de forma independente.

5 RESULTADOS E DISCUSSÕES

5.1 Coleta de dados

As mensagens utilizadas para a classificação foram coletadas da rede social Twitter. A escolha do Twitter se deu pelo fato de ser a rede social em maior ascensão atualmente e por oferecer uma ampla coleção de bibliotecas de acesso (<https://dev.twitter.com>). Além disso, os *tweets* postados são públicos por padrão, ou seja, qualquer usuário mesmo sem acesso ao Twitter pode ler os tweets sem necessidade de solicitar permissão específica. Para coletar os *tweets* foi implementado um mecanismo automático denominado de coletor, utilizado nas etapas mencionadas na Figura 4.1. O coletor por sua vez, foi escrito em linguagem Python utilizando biblioteca de código aberto, apresentado na Listagem 1.

```
1 import tweepy as tw
```

Listagem 1: Biblioteca coleta dados

O objetivo dessa biblioteca é realizar uma conexão persistente com a rede social e retornar continuamente os resultados com os *tweets*. Assim, essa estratégia favorece a coleta massiva e contínua dos *tweets*. Para isso, é necessário um fluxo de comunicação entre a aplicação solicitante e a aplicação servidora (no caso o Twitter).

Para isso é necessário entrar com as *keys* que permitem o acesso aos dados de desenvolvedores do Twitter. Essas *keys* podem ser obtidas facilmente respondendo um formulário de pedido na própria plataforma de desenvolvedores do Twitter, a plataforma leva em média três dias para análise do pedido e concessão das *keys*. A partir disto, é necessário a autenticação das chaves para que seja possível a pesquisa e extração dos dados, apresentado na Listagem 2.

```
1 consumer_key = 'XXXXXXXXXXXXXXXXXXXXX'
   consumer_secret = 'XXXXXXXXXXXXXXXXXXXXX'
3 access_token = 'XXXXXXXXXXXXXXXXXXXXX'
   access_token_secret = 'XXXXXXXXXXXXXXXXXXXXX'
```

Listagem 2: Acesso aos dados do Twitter.

A partir das *keys* é possível acesso ilimitado a todos os dados da plataforma e não somente ao objeto de pesquisa. É por isso que algumas observações devem ser feitas no momento da coleta de dados, por exemplo, o termo ou palavra que deseja em questão, nesse caso a palavra iPhone, bem como selecionar apenas *tweets* de maneira que os *retweets* não apareçam, aumentando o número de *spams* e prejudicando a polaridade do sentimento. É necessário solicitar que sejam escolhidos apenas *tweets* em português e de maneira estendida (280 caracteres), para que todos os caracteres do *tweet* estejam presentes na coleta, e, por fim o número de *tweets* que se deseja extrair, como apresenta a Listagem 3.

```

search_words = "Iphone"+"-filter:retweets" #para aparecer apenas uma vez
2 cursor_tweets = tw.Cursor(api.search, #entidade tweepy
    tweet_mode='extended', #280 caracteres
4    lang="pt", #apenas em portugues
    q=search_words).items(5) #busca em tempo real e quantidade
    desejada

```

Listagem 3: Especificações para a extração dos *tweets*.

Para que de fato os resultados se apresentassem de forma imparcial, coletaram-se de 500 a 1000 *tweets* por dia durante um período de 15 dias (15 a 30 de outubro de 2021).

Todos os *tweets* foram coletados de segunda a sexta sempre no mesmo horário, às 14 horas no horário de Brasília. Esta estratégia tinha por objetivo obter os dados mais atuais, para que não houvesse diferença devido ao período, ao público e aos eventos ocorridos. Apesar dessas especificidades, ainda sim, é difícil manter o controle do fluxo de dados, por exemplo, em relação aos acontecimentos diários, as *fake news*, lançamento de novos produtos, etc.

A análise de sentimentos com dados de fluxo contínuo torna difícil a obtenção de um resultado claro, por exemplo, imagine que hoje 70% dos dados estejam falando bem de um produto x, no próximo dia entraram mais *tweets* que totalizam 40% de comentários bons, ou seja, quando se trabalha com um conjunto de dados que recebe dados todos os dias, em um curto intervalo, a conclusão que se obtém é referente a este determinado período, diferente do que acontece com *feedbacks*, comentários, avaliações de sites, que de modo geral variam de forma mais lenta, estendendo, ou até mesmo ampliando a “validade” do sentimento em questão.

Após as especificações de pesquisa dos *tweets*, eles são impressos conforme apresenta a Listagem 4, gerada de forma aleatória:

```

1 2022-04-11 18:41:43
   no iphone esse numero me liga todo dia
3 2022-04-11 18:41:42
   @_bmats nao, xiaomi. Achei caro demais iphone kkkk ainda assim, xiaomi me
   decepcionou tirando meu amado p2.
5 2022-04-11 18:40:57
   iPhone X eh a prova d agua ?
7 2022-04-11 18:40:36
   oi apple tudo bem? @mundoapplebra poderia fazer uma nova atualizacao para poder
   BAIXAR apk no iphone?????? obrigado
9 2022-04-11 18:40:35
   @gijns Minha irma ta vendendo o iPhone 8 em otimo estado porque comprou o novo,
   amiga

```

Listagem 4: Exemplos de *tweets* extraídos.

É possível perceber que as mensagens apresentadas na Listagem 4 necessitam de pré processamento, visto que muitas informações contidas são irrelevantes para a análise, tais como,

nome do usuário, pontuações, datas e horários.

5.2 Pré processamento

É a partir da etapa de pré processamento que é realizada toda a limpeza dos dados e a adaptação do texto para uma futura representação mais estruturada. O principal objetivo desta etapa é detectar e excluir os *tweets* que são *spam* dentre os coletados na etapa anterior (coleta de dados) e manter apenas os *tweets* relevantes.

A partir dos *tweets* extraídos o próximo passo é colocar os dados no formato JSON que tem como propósito separar por atributo cada *tweet*, pois cada *tweet* possui diversas informações, tais como o ID do usuário, a localização, a data, o número de seguidores, as curtidas, os *retweets* entre outros atributos. Alguns exemplos de atributos existentes nos *tweets* são apresentados na Listagem 5.

```
dict_keys(['created_at', 'id', 'id_str', 'full_text', 'truncated', 'display_text_range', 'entities', 'metadata', 'source', 'in_reply_to_status_id', 'in_reply_to_status_id_str', 'in_reply_to_user_id', 'in_reply_to_user_id_str', 'in_reply_to_screen_name', 'user', 'geo', 'coordinates', 'place', 'contributors', 'is_quote_status', 'retweet_count', 'favorite_count', 'favorited', 'retweeted', 'lang'])
```

Listagem 5: Atributos presentes formato JSON.

Ainda assim, com todos os dados no formato JSON, é preciso retirar apenas o conteúdo necessário para dar prosseguimento a pesquisa. Dessa forma, criou-se um diretório vazio com o objetivo de colocar todo o atributo *full_text*, responsável por conter todos os *tweets*.

Apesar de separar apenas o conteúdo do *tweet* de todos os outros dados, ainda é necessário fazer algumas modificações. Esse processo diz respeito à tokenização, ou seja, transformação de cada fala (*strings* contínuas) em uma sequência de palavras, como apresenta a Listagem 6 e 7.

```
1 from nltk.tokenize import word_tokenize, RegexpTokenizer
   example = "A qualidade da camera do iphone esta melhor a cada atualizacao"
3 print(word_tokenize(example, language='portuguese'))
```

Listagem 6: Tokenização.

```
1 ['A', 'qualidade', 'da', 'camera', 'do', 'iphone', 'esta', 'melhor', 'a', 'cada',
   , 'atualizacao']
```

Listagem 7: Resultado da tokenização.

Optou-se pela remoção de todas as *stopwords* (termos não representativos para um documento como artigos, preposições, conjunções e pronomes), nomes próprios e nomes de cidades e estados brasileiros, bem como *emojicons* (representação gráfica de uma emoção feita através dos caracteres) e diversas outras palavras que se tornem insignificante à interpretação e ao conteúdo de cada *tweet*, assim como pontuações, como apresenta a Listagem 8.

```

1 import preprocessor as p
    input_file_name = "XXX" #formato .JSON ou .txt
3 p.clean_file(input_file_name) #limpeza basica
    p.tokenize(input_file_name) #tokeniza
5 p.OPT.URL #retira URLs
    p.OPT.MENTION #retira marcacoes
7 p.OPT.HASHTAG #retira hashtag
    p.OPT.EMOJI #retira emojis
9 p.OPT.SMILEY #retira emojis (no formato :), (; )
    p.set_options(p.OPT.URL, p.OPT.EMOJI) #mais de uma opcao ao mesmo tempo

```

Listagem 8: Limpeza dos *tweets*.

Em muitos casos é possível encontrar arquivos que já possuem as principais *stopwords* de cada língua, o que auxilia o trabalho visto a quantidade de palavras, principalmente na língua portuguesa.

Após todos esses processos, os *tweets* agora aparecem conforme a Listagem 9.

```

1 iphone esse numero liga todo dia
3 nao xiaomi Achei caro demais iphone ainda assim xiaomi decepcionou tirando meu
    amado
5 iPhone prova dagua
7 oi apple tudo bem poderia fazer uma nova atualizacao para poder baixar iphone
    obrigado
9 minha irma vendendo iPhone otimo estado porque comprou novo amiga

```

Listagem 9: Exemplos de *tweets* extraídos após a limpeza.

Esse tipo de limpeza e o tempo necessário pode variar de acordo com o seu conjunto de dados, o Twitter é uma rede que possui diversas conversas informais, sendo assim possui diversas gírias e abreviações. É possível ainda, que após a limpeza grosseira as gírias sejam substituídas pelas palavras corretas (ex: vc = você), sem perda de sentido da frase, ou ainda, a remoção delas. Geralmente, essa etapa é muito particular para cada conjunto de dados. Considerando, por exemplo, o texto de um editorial de um jornal qualquer, provavelmente a limpeza seria realizada com mais facilidade, pois é de se esperar que esse tipo de texto contenha uma escrita mais formal sem abreviações nem gírias.

Após essa fase, os dados já pré processados foram enviados no formato .csv para o software Excel® para que sejam classificados manualmente.

O processo de classificação manual permite que seja possível expressar o real sentimento da frase, sendo um processo realizado por um humano e não uma máquina. De fato, esse processo é lento e muito delicado, mas possibilita que sejam identificados figuras de linguagens, como metáforas, pleonasmos, ironias, etc, o que dificultaria o treinamento da máquina podendo levar a uma classificação errônea. Ainda assim, a partir da classificação manual é possível permanecer com a maior proximidade do *tweet* original, pois é muito mais prático para um humano compreender gírias, abreviações e figuras de linguagem, características estas bem presentes na rede social Twitter. Os rótulos possíveis foram:

- Neutro: mensagens que não expressam nenhuma opinião, sentimento ou uma dúvida;
- Positivo: mensagens que expressam um sentimento positivo;
- Negativo: mensagens que expressam um sentimento negativo, mesmo que escritas com a estrutura de textos positivos.

Como exemplo, as seguintes frases foram classificadas como neutras neste trabalho:

- Iphone foi produzido por uma empresa privada;
- Sem falar que tem até iPhone no jogo sendo que aparentemente se passa milhares de centenas de anos no passado;
- No iPhone aparece assim.

As frases seguintes foram classificadas como positivas neste trabalho:

- Chegou o ponto da minha vida no qual eu quero ter um iphone;
- Iphone no brasil mais barato que nos eua;
- Vou guardar dinheiro pra comprar o iphone.

E por fim, as frases abaixo foram classificadas como negativas neste trabalho:

- As notificações do iPhone com o IOS ficaram super estranhas;
- Venho diretamente do meu iPhone pra dizer que o Samsung Galaxy S21 Ultra o melhor celular do mundo;
- Alguém me deu um carregador de iphone.

Como tem-se por objetivo avaliar os algoritmos, é de suma importância que cada frase seja avaliada corretamente e que cada atribuição seja a mais real possível, visto que essa parte diz respeito a todo o treinamento do algoritmo, podendo assim, possivelmente, influenciar na

qualidade do algoritmo e de seus resultados. Além disso, classificar os dados manualmente, apesar da quantidade, possibilita que eles sejam utilizados como dados de treinamento para a criação de um futuro algoritmo de classificação de *tweets*.

Ao total foram coletadas 10.000 frases e separadas em duas colunas, a primeira contendo todas as frases do Twitter, enquanto a segunda contendo a pontuação referente ao sentimento, sejam eles +1, 0 ou -1. A Figura 5.1 apresenta a forma como os dados são apresentados nos programas, ressalte-se que frases acima de 45 caracteres não são possíveis a visualização completa.

Figura 5.1: Configuração dos dados.

	headline	label
0	queria um iphone novo	1.0
1	se nao fosse a camera desse iphone nao sei o q...	1.0
2	o que adianta comprar um iphone pra ter uma ca...	0.0
3	tenho carregadores aqui em casa e todos estrag...	-1.0
4	quero um iphone	1.0

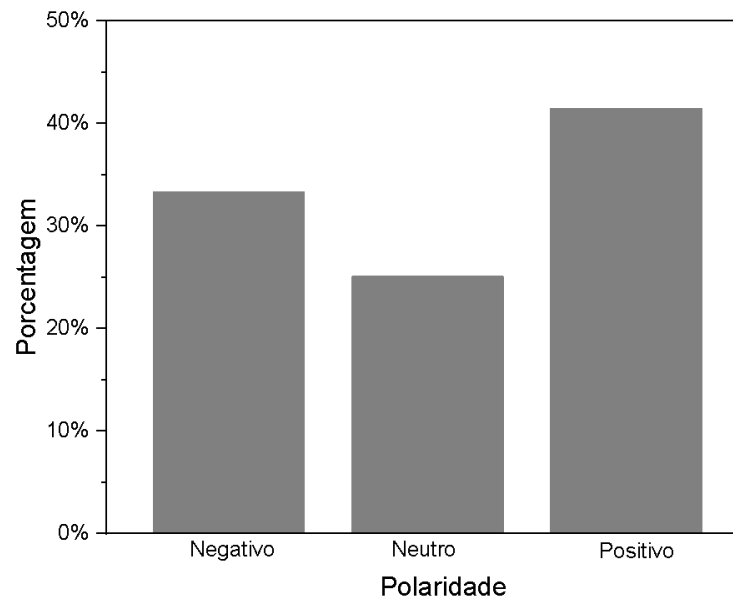
Fonte: Autoria própria (2022).

Apenas para satisfazer a curiosidade do leitor as frases 1, 2 e 3 na Figura 5.1 são:

1. “se nao fosse a camera desse iphone nao sei o que seria de mim”
2. “o que adianta comprar um iphone pra ter uma casa rebocada”
3. “tenho carregadores aqui em casa e todos estragaram as vezes eu odeio ter iphone na moral”

A partir da organização do conjunto de dados foi necessário avaliar a necessidade de se utilizar os dados que possuíam polaridade neutra, e sua relação com o todo. O resumo da classificação está apresentado na Figura 5.2.

Figura 5.2: Contagem dos rótulos.



Fonte: Autoria própria (2022).

É possível observar que os dados neutros estão em menor quantidade, e a partir disso optou-se por remover todos os dados cujo o sentimento era neutro para que fossem avaliadas apenas as classificações positivas e negativas, trabalhando com um conjunto de dados binomial.

As frases classificadas como neutras apresentam frases que não possuem compreensão com uma determinada clareza. Por exemplo, frases como: “do nada a loja do iPhone curtiu meu tweet” ou “alerta de oferta do iphone no Magazine Luiza” claramente não possuem uma opinião ou até mesmo, como a segunda, podendo ser considerada como spam.

A classificação manual possibilitou que menos dados fossem retirados do conjunto. Quando utiliza-se de uma classificação não manual o conjunto costuma conter uma enorme quantidade de dados neutros simplesmente pela dificuldade do algoritmo em interpretar o sentimento de cada frase, classificando-o precipitadamente como um comentário neutro, ou seja, a classificação manual propicia uma retirada menor de dados.

Por isso, o *dataframe* possui uma nova configuração, conforme está apresentado na Tabela 6.

Tabela 6: Conjunto de dados sem a polaridade neutra.

Polaridade	Quantidade
1	4017
-1	3227

Fonte: Autoria própria (2022).

A Tabela de dados agora contém apenas exemplos positivos e negativos.

Com as frases de polaridades positivas e negativas do *dataframe* foi possível gerar um recurso gráfico para descrever os termos mais frequentes de todo o conjunto de dados, conhecido como nuvem de palavras, apresentada na Figura 5.3.

Tabela 7: Frases positivas.

n	Palavra	F	%	F. A. %
1	iphone	4066	42,2	42,2
2	pra	617	6,4	48,6
3	comprar	391	4,1	52,6
4	so	346	3,6	56,2
5	pro	342	3,5	59,7
6	que	330	3,4	63,2
7	ta	293	3,0	66,2
8	vou	208	2,2	68,4
9	max	186	1,9	70,3
10	voce	170	1,8	72,1
11	ja	164	1,7	73,8
12	queria	163	1,7	75,4
13	novo	161	1,7	77,1
14	quero	160	1,7	78,8
15	ter	159	1,6	80,4
16	algum	158	1,6	82,1
17	vai	156	1,6	83,7
18	porque	144	1,5	85,2
19	melhor	139	1,4	86,6
20	nao	138	1,4	88,0
21	celular	134	1,4	89,4
22	to	132	1,4	90,8
23	ser	126	1,3	92,1
24	gb	120	1,2	93,4
25	apple	119	1,2	94,6
26	agora	113	1,2	95,8
27	para	106	1,1	96,9
28	dar	103	1,1	97,9
29	camera	100	1,0	99,0
30	da	100	1,0	100,0

Fonte: Autoria própria (2022)

Tabela 8: Frases negativas.

n	Palavra	F	%	F. A. %
1	iphone	3259	39,4	39,4
2	pra	511	6,2	45,5
3	que	382	4,6	50,2
4	bateria	325	3,9	54,1
5	ta	311	3,8	57,8
6	so	294	3,6	61,4
7	nao	248	3,0	64,4
8	celular	216	2,6	67,0
9	comprar	176	2,1	69,1
10	carregador	176	2,1	71,3
11	porque	168	2,0	73,3
12	ter	167	2,0	75,3
13	ja	158	1,9	77,2
14	pro	151	1,8	79,0
15	ser	141	1,7	80,7
16	vai	141	1,7	82,4
17	to	127	1,5	84,0
18	voce	125	1,5	85,5
19	android	115	1,4	86,9
20	ainda	112	1,4	88,2
21	vou	111	1,3	89,6
22	apple	110	1,3	90,9
23	algum	108	1,3	92,2
24	cara	103	1,2	93,5
25	to	98	1,2	94,6
26	agora	96	1,2	95,8
27	gente	91	1,1	96,9
28	round	90	1,1	98,0
29	nunca	84	1,0	99,0
30	ate	83	1,0	100,0

Fonte: Autoria própria (2022)

* F = Frequência, F.A = Frequência Absoluta.

De modo prático, a palavra mais positiva é “iphone”, mas como é apresentado na Tabela 8 a palavra “iphone” também é a mais negativa, o que já era de se esperar, visto que era o foco principal do estudo. No entanto, outras palavras também se repetiram nas Tabela 7 e 8, como “camera”, “apple”, “celular”, etc.

Vale a pena ressaltar que uma determinada palavra na análise positiva não possui a mesma ordem necessariamente na análise negativa. Por exemplo, a palavra “comprar” na análise positiva apresenta ordem $n = 3$ conforme a Tabela 7, já na análise negativa a palavra “comprar” apresenta ordem $n = 9$ conforme a Tabela 8.

As cinco primeiras palavras nas Tabelas 7 e 8 correspondem a uma frequência acumu-

lada percentual de 59,7% e 57,8% , respectivamente, ou seja, 16,7% das palavras (=5/30) são responsáveis por cerca de 60% das ocorrências. Neste momento é possível resgatar automaticamente a analogia com a regra 20-80 do Gráfico de Pareto, onde 20% das causas são responsáveis por 80% dos efeitos (Souza, 2010) [61].

Dessa forma, caso a empresa Apple estivesse em uma investigação sobre o modelo iPhone seria muito interessante avaliar as palavras presentes no primeiro 1/5 do ranking, pois ali informações importantes seriam encontradas. Como exemplo, na polaridade negativa as cinco primeiras palavras foram “iPhone”, “pra”, “que”, “bateria” e “ta”. Dentre estas cinco, a palavra “bateria” torna-se um possível foco para uma investigação mais aprofundada. Recorrendo ao banco de dados é possível compreender melhor a opinião negativa dos clientes iPhone sobre o item “bateria”.

Quanto às ocorrências de polaridade positiva, encontram-se como as cinco primeiras as palavras “iPhone”, “pra”, “comprar”, “so” e “pro”. Dentre as palavras mais comentadas, “comprar” aparece na terceira posição, permitindo também uma investigação a fim de entender melhor o porquê dos clientes iPhone desejarem comprar o iPhone. E a partir disso, da identificação do problema na análise negativa, e da identificação da oportunidade na análise positiva, a Apple teria uma oportunidade de melhorar seus produtos aplicando por exemplo um ciclo PDCA (Campos, 2004) [10] para melhoria, o que tornaria sua marca ainda mais desejada e lembrada como sinônimo de qualidade.

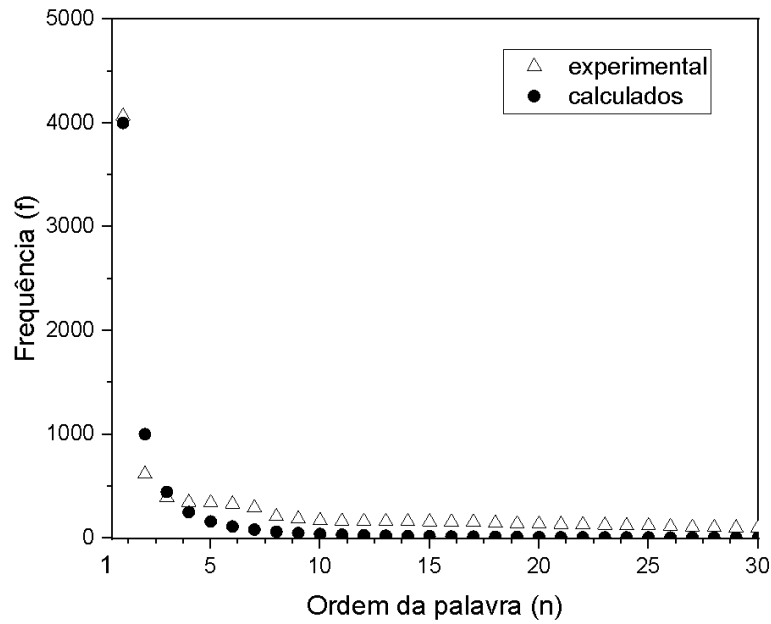
De forma mais minuciosa, as distribuições de frequência apresentadas nas Tabelas 7 e 8 podem ser descritas através da seguinte equação:

$$f = \frac{c}{r^n} \quad (5.1)$$

onde, n representa a ordem da palavra, e c uma contante.

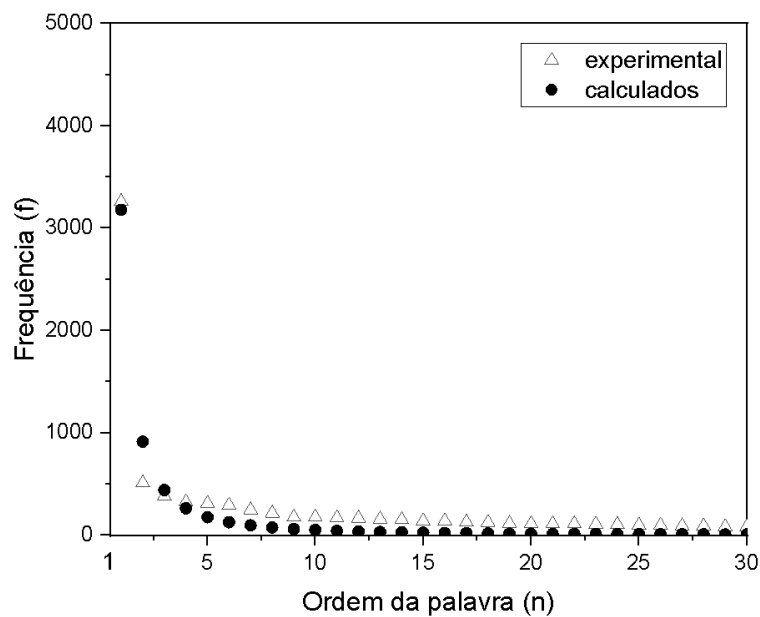
As Figuras 5.4 e 5.5 apresentam os ajustes obtidos dos dados calculados aos experimentais utilizando os códigos computacionais disponíveis no Apêndice 9.1.

Figura 5.4: Distribuição de frequência de palavras em frases positivas.



Fonte: Autoria própria (2022).

Figura 5.5: Distribuição de frequência de palavras em frases negativas.



Fonte: Autoria própria (2022).

Pode-se verificar de forma qualitativa em ambas as figuras a aproximação dos dados calculados ao experimentais, o que foi comprovado pelo coeficiente de relação R^2 com valor de 0,98 para as frases positivas e de 0,97 para as negativas.

A equação 5.1 para as frases positivas passa a ser dada por

$$f = \frac{1,99}{4000^n} \quad (5.2)$$

E para as frases negativas por:

$$f = \frac{1,79}{3178^n} \quad (5.3)$$

5.3 Treinamento e teste

Para treinar os modelos, precisa-se transformar todas as palavras em números, pois os algoritmos só trabalham com os números. Para realizar essa transformação, utilizou-se o Count-Vectorizer do sklearn. Para exemplificar, o resultado da vetorização das frases a seguir, é apresentado na Listagem 11.

- “Apesar do problema na ultima atualizacao Apple acredita no aumento das vendas”;
- “O aumento do preco dos produtos da Apple na ultima semana causou problema”.

```
array([[1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1],
      [0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0]],
      dtype=int64)
```

Listagem 11: Resultado da vetorização das frases de exemplo.

O vetorizador retira as palavras presentes no documento excluindo os artigos e as coloca em ordem alfabética de modo que a cada nova frase seja representada como a Listagem 11, onde 1 representa que a palavra está presente nesta frase e 0 que a palavra em questão não está.

Para ver isso mais claramente, verificou-se na Listagem 12 e 13 os nomes dos recursos mapeados para a primeira e segunda frase:

```
list(zip(X.toarray()[0], vect.get_feature_names()))
[(1, 'acredita'), (1, 'apesar'), (1, 'apple'), (1, 'atualizacao'), (1, 'aumento'),
 (0, 'causou'), (0, 'da'), (1, 'das'), (1, 'do'), (0, 'dos'), (1, 'na'),
 (1, 'no'), (0, 'preco'), (1, 'problema'), (0, 'produtos'), (0, 'semana'), (1,
 'ultima'), (1, 'vendas')]
```

Listagem 12: Mapeamento da primeira frase.

Da mesma forma para a segunda frase:

```
list(zip(X.toarray()[1], vect.get_feature_names()))
[(0, 'acredita'), (0, 'apesar'), (1, 'apple'), (0, 'atualizacao'), (1, 'aumento'),
 (1, 'causou'), (1, 'da'), (0, 'das'), (1, 'do'), (1, 'dos'), (1, 'na'),
 (0, 'no'), (1, 'preco'), (1, 'problema'), (1, 'produtos'), (1, 'semana'), (1,
 'ultima'), (0, 'vendas')]
```

Listagem 13: Mapeamento da segunda frase.

Por fim, dispõe-se de uma matriz $n \times m$, onde n são as frases, ou seja, cada *tweet*, e m representa cada palavra de todo o documento. Dessa forma, o exemplo acima possui duas frases (linhas) e 18 palavras no documento (colunas). Aplicando o vetorizador aos *tweets*, a impressão é apresentada na Listagem 14.

```

1 array([[0, 0, 0, ..., 0, 0, 0],
        [0, 0, 0, ..., 0, 0, 0],
3       [0, 0, 0, ..., 0, 0, 0],
        ...,
5       [0, 0, 0, ..., 0, 0, 0],
        [0, 0, 0, ..., 0, 0, 0],
7       [0, 0, 0, ..., 0, 0, 0]], dtype=int64)

```

Listagem 14: Vetorizando todos os *tweets* do conjunto de dados.

Note que o vetorizador por padrão armazena tudo em um *array* esparsa (arranjo de elementos no qual vários itens são idênticos), e usando `X.toarray()` apresenta a versão densa. Uma matriz esparsa é muito mais eficiente, pois a maioria dos valores em cada linha é 0.

Dessa forma, os algoritmos trabalharão com duas matrizes, a primeira contendo todas as palavras do documento e a segunda configurada no formato de coluna com todas as polaridades de cada frase.

Antes de começar o treinamento dos modelos é necessário especificar a proporção de dados que será utilizada em cada conjunto. É importante realizar esse processo antes de se trabalhar de qualquer maneira com os dados. Neste trabalho a proporção utilizada referente ao treinamento e ao teste dos modelos foi de 80 : 20. O tamanho de teste é 0,2 ou 20%. Isso significa que X_{test} e Y_{test} contêm 20% dos dados que reservados para testes, como apresentado na Listagem 15.

```

1 from sklearn.model_selection import train_test_split
3 X = df.headline
  Y = df.label
5
  X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2)

```

Listagem 15: Ajustando a proporção de dados de treinamento e de teste.

Apenas para reiterar, é importante evitar ajustar o vetorizador em todos os dados, pois deseja-se um conjunto de teste limpo para avaliar o desempenho. Colocar o vetorizador em tudo resultaria em vazamento de dados, causando resultados não confiáveis, pois o vetorizador não deveria saber sobre dados futuros.

A Listagem 16 apresenta como pode-se ajustar o vetorizador e transformar X_{train} em uma etapa.

```

1 from sklearn.feature_extraction.text import CountVectorizer
2
3 vect = CountVectorizer(binary=True)
4
5 X_train_vect = vect.fit_transform(X_train)

```

Listagem 16: Ajuste apenas dos dados de treinamento (X_{train}).

Um dos problemas mais comuns no aprendizado de máquina é trabalhar com um conjunto de dados desequilibrado. Os dados são chamados de desbalanceados ou desequilibrados quando ocorre uma pequena incidência de uma categoria em comparação com as demais categorias (Kotsiantis et al., 2005) [34].

Uma das consequências desse desequilíbrio é que o modelo terá uma tendência a dar muitos “alarmes falsos”, ou seja, na prática ele irá responder muito bem entradas para as classes majoritárias (positivas), mas terá um desempenho inferior para as minoritárias (negativas).

Como apresentando na Figura 5.2 os dados estavam ligeiramente desequilibrados, e é possível verificar a partir da Tabela 9.

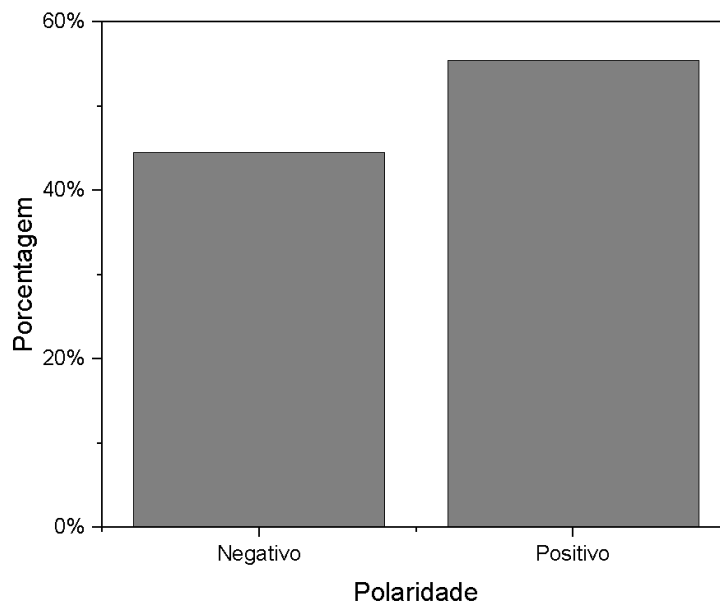
Tabela 9: Configuração dos dados.

Polaridade	Quantidade
1	4017
-1	3227

Fonte: Autoria própria (2022).

Na Figura 5.6 verifica-se esse pequeno desbalanceamento entre as frases de polaridade positiva e negativa.

Figura 5.6: Conjunto de dados com frases positivas e negativas.



Fonte: Autoria própria (2022).

Uma maneira de resolver esse problema é superamostrar os exemplos na classe minoritária. Isso pode ser alcançado simplesmente duplicando exemplos da classe minoritária no conjunto de dados de treinamento antes de ajustar um modelo. Isso pode equilibrar a distribuição de classes, mas não fornece nenhuma informação adicional ao modelo.

Dessa forma, é possível equilibrar os dados usando uma forma de sobre-amostragem chamada SMOTE (*Synthetic Minority Oversampling TEchnique*). O SMOTE funciona selecionando exemplos próximos no espaço de recursos, desenhando uma linha entre os exemplos no espaço de recursos e desenhando uma nova amostra em um ponto ao longo dessa linha. Este procedimento pode ser usado para criar tantos exemplos sintéticos para a classe minoritária quantos forem necessários Chawla et al., 2002) [12].

As Listagem 17 e apresenta a realização dessa etapa de balanceamento com os dados.

```

1 from imblearn.over_sampling import SMOTE
  sm = SMOTE()
3
  X_train_res, y_train_res = sm.fit_resample(X_train_vect, y_train)
5
  unique, counts = np.unique(y_train_res, return_counts=True)
7 print(list(zip(unique, counts)))

```

Listagem 17: Balanceando os dados.

Neste trabalho, as frases são as observações e o sentimento positivo/negativo são as categorias. Este é um problema de classificação binomial, que tem como objetivo prever se um *tweet* é positivo ou negativo.

Dessa forma, a Tabela 10 apresenta a nova quantidade de frases de cada polaridade.

Tabela 10: Dados balanceados.

Polaridade	Quantidade
1	3218
-1	3218

Fonte: Autoria própria (2022).

As classes agora estão balanceadas para o conjunto de treinamento.

5.4 Análise

Após todo o trabalho com os dados prosseguiu-se para os algoritmos, mas antes disso, foi necessário a utilização da biblioteca *sklearn* e ajustá-la aos dados de treinamento, como apresentado nas Listagens 18, 19 e 20.

```

1 from sklearn.naive_bayes import MultinomialNB
3 nb = MultinomialNB()
5 nb.fit(X_train_res, y_train_res)
7 nb.score(X_train_res, y_train_res)

```

Listagem 18: Sklearn Naive Bayes.

0.7420758234928527

```

1 from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
3 lr.fit(X_train_res, y_train_res)
5 lr.score(X_train_res, y_train_res)

```

Listagem 19: Sklearn Regressão Logística.

0.7823602100710535

```

from sklearn.svm import LinearSVC
2 svc = LinearSVC()
4 svc.fit(X_train_res, y_train_res)
6 svc.score(X_train_res, y_train_res)

```

Listagem 20: Sklearn SVM.

0.7935765512940443

Naive Bayes, Regressão Logística e *Support Vector Machine* ajustaram os dados de treinamento e estão prontos para realizar as previsões. Verificou-se uma precisão de ajuste de 74%, 78%, e 79% para o NB, RL, SVM, respectivamente.

5.5 Validação

Vetorizou-se o conjunto de teste e utilizou-o para prever se cada título de teste é positivo ou negativo. Como é necessário evitar qualquer vazamento de dados, é preciso apenas transformar e não reajustar, como mostra a Listagem 21.

```

X_test_vect = vect.transform(X_test)
2
y_pred = **.predict(X_test_vect) #nb, lr, svc
4
y_pred

```

Listagem 21: Vetorização.

```
array([ 1., 1., 1., ..., -1., -1., 1.])
```

Com isso, y_{pred} contém uma previsão para cada linha do conjunto de teste, ou seja, a primeira frase contém a polaridade positiva, a segunda positiva e assim por diante. Com esse resultado de previsão é necessário passá-lo para uma métrica *sklearn* com os rótulos verdadeiros (classificados manualmente) para obter uma pontuação de precisão, pontuação *F1* e gerar uma matriz de confusão, como apresenta a Listagem 22.

```

1 from sklearn.metrics import accuracy_score, f1_score, confusion_matrix
3 print("Acuracia: {:.2f}%".format(accuracy_score(y_test, y_pred) * 100))
print("\nF1 Score: {:.2f}%".format(f1_score(y_test, y_pred) * 100))
5 print("\nMatriz de Confusao:\n", confusion_matrix(y_test, y_pred))

```

Listagem 22: Pontuação de precisão.

As Tabelas 11 e 12 apresentam as métricas encontradas de cada modelo.

Tabela 11: Resultado dos modelos.

	Precisão	Acurácia	Recall	F1-Score
Naive Bayes	74,75%	72,19%	74,71%	74,80%
Regressão Logística	74,38%	70,74%	69,61%	71,92%
Support Vector Machine	73,68%	68,32%	67,40%	70,71%

Fonte: Autoria própria (2022)

Tabela 12: Métricas das matrizes de confusão.

	VP	VN	FP	FN
Naive Bayes	598	448	202	201
Regressão Logística	537	448	233	231
Support Vector Machine	546	444	195	264

Fonte: Autoria própria (2022)

A maioria das frases classificadas erradas são de polaridade positiva, ou seja, os modelos classificaram como negativas frases que são positivas. Uma possível resposta para esse acontecimento é o fato de se utilizar uma classificação manual. Esse tipo de classificação permite que as frases confusas, bem como, as ironias, sarcasmos e outras figuras de linguagem consigam ser interpretadas corretamente, dificultando com que o algoritmo entenda o “padrão” das

mensagens, pois há um maior número de diversidade e categorias dentro de uma só polaridade, contribuindo para o baixo valor de ajuste dos modelos. Essa técnica explica o motivo do conjunto de dados possuir menos comentários com polaridade neutra, o que acaba sendo facilmente escolhido quando o algoritmo enfrenta dificuldade na interpretação (Kirilenko et al., 2017) [32].

O Quadro 2 apresenta o desempenho dos modelos em cada uma das métricas, onde o símbolo de “check” verde é dado ao modelo que obteve o melhor índice.

Quadro 2: Desempenho dos modelos.

	Precisão	Acurácia	Recall	F1-Score	VP	VN	FP	FN
Naive Bayes	✓	✓	✓	✓	✓	✓		✓
Regressão Logística						✓		
Support Vector Machine							✓	

Fonte: Autoria própria (2022)

Analisando o Quadro 2, é possível verificar que o modelo Naive Bayes apresentou melhores métricas do que os demais, exceto para os Verdadeiros Positivos onde existe um empate com o modelo Regressão Logística e os Falsos Positivos onde o melhor modelo foi o *Support Vector Machine*.

Outros trabalhos também se utilizaram do aprendizado de máquina com dados do Twitter. Os resultados de acurácia de Nascimento et. al. (2012) [42] no estudo sobre notícias variaram de 70% a 80% para o modelo Naive Bayes. Kini e Patil (2021) [18] obtiveram acurácia de cerca de 88% para o *Support Vector Machine* no estudo sobre qual algoritmo de aprendizado de máquina oferece melhor precisão. O modelo de Regressão Logística obteve 67,2% de acurácia no trabalho proposto por Saad e Yang (2019) [56] analisando sentimentos de *tweets* com base em regressão ordinal usando técnicas de aprendizado de máquina.

A fim de avaliar de outra maneira os modelos, utilizou-se o método de validação cruzada onde são gerados dez vezes dados de treinamento e teste em posições diferentes.

A Tabela 13 apresenta as médias dos resultados obtidos em cada iteração do método de validação cruzada. As matrizes de confusão e as demais métricas são apresentadas nas Listagens 30, 31 e 32 e nas Figuras 9.1, 9.2 e 9.3 no Apêndice 9.2.

Tabela 13: Média das métricas de validação cruzada dos modelos.

	Acurácia	F1-Score
Naive Bayes	70,46%	72,76%
Regressão Logística	69,67%	72,02%
Support Vector Machine	67,76%	69,78%

Fonte: Autoria própria (2022)

Verifica-se na Tabela 13 que pelo método da validação cruzada os três modelos apresentaram uma diminuição nos valores de acurácia. Esse tipo de desempenho pode ocorrer pelo fato da escolha aleatória em cada iteração. Se a amostra escolhida contiver uma grande quantidade de dados que os modelos classificam de forma errada, então a acurácia será menor. Como no

fim é realizado uma média de cada iteração esse fato pode contribuir para que a acurácia da validação cruzada seja inferior às dos próprios modelos.

Diversos trabalhos na literatura optam por trabalhar com a validação cruzada, como é o caso de Ramadhan et. al. (2017) [50] que obtiveram a partir da validação cruzada uma acurácia média de 74% para a Regressão Logística. Soumya e Pramod (2020) [60] para o modelo Naive Bayes encontraram uma acurácia média de 71,4%. Já o modelo *Support Vector Machine* para Rajput e Solanki (2016) [49] alcançou uma acurácia média de 81,9%.

Após a avaliação de cada classificador individualmente, verificou-se se o método *ensembling*, classificador de votação, obterá métricas superiores às já obtidas.

Utilizou-se o *sklearn*, *VotingClassifier*, apresentado na Listagem 23, que tem como padrão uma votação por regra da maioria, ou seja, votação difícil.

```
1 from sklearn.ensemble import VotingClassifier
```

Listagem 23: Classificador de votação.

O classificador de votação obteve acurácia de 69,41% e F1-Score de 72,10%. A matriz de confusão obtida foi:

$$Confusão = \begin{bmatrix} 432,3 & 204,4 \\ 238,9 & 573,4 \end{bmatrix} \quad (5.4)$$

Mesmo utilizando um metaclassificador que equilibra as fraquezas dos classificadores individuais em um conjunto de dados específicos, os resultados mostraram valores próximos à média das acurácias dos modelos (70,41%).

Esse tipo de métrica é uma boa opção para quando os modelos possuem valores próximos de modo que não mascarem modelos que obtiveram valores baixos para uma determinada classe.

Os resultados obtidos neste trabalho se assemelham aos encontrados por Atallah e Al-mousa (2019) [6], onde o classificador de votação obteve uma acurácia 3% maior que os modelos individualmente. Assiri et. al. (2020) [5] a partir do classificador de votação alcançaram uma acurácia cerca de 3% maior que a média encontrada pelos modelos. Mung e Phyu (2020) [41] obtiveram uma acurácia cerca de 4% maior que a média encontrada a partir dos modelos individualmente.

Uma hipótese que pode ter influenciado o resultado dos modelos é a maneira de especificar a polaridade, por exemplo, o trabalho desenvolvido por Pak e Paroubek (2010) [45] permite que haja uma pontuação que varie de 0 a 1, ou seja, valores decimais. Desse modo, o modelo não precisa especificar positivo ou negativo, simplesmente possibilitando mais do que apenas duas opções de escolha para uma frase, e sim, um intervalo dentro de cada polaridade.

Da mesma forma que comentado anteriormente, o valor encontrado para a precisão do ajuste dos modelos com os dados de treinamento pode ter um influência significativa para o desempenho encontrado, quando um modelo não consegue capturar o relacionamento entre os exemplos de entrada (geralmente denominado X) e os valores de destino (geralmente denominado Y).

Apesar disso, os valores obtidos nesse trabalho não destoam dos presentes na literatura.

Cabe ressaltar que outros produtos, serviços ou eventos possam ser utilizados como o objeto de pesquisa, ou seja, a partir do modelo escolhido espera-se também que o modelo possa analisar outras palavras além de “iPhone”.

Apesar de não ser o objetivo deste trabalho alguns pontos sobre a análise de sentimentos devem ser levados em conta, visto que opiniões mudam frequentemente a partir de eventos que ocorrem de maneira localizada. Por exemplo, empresas costumam utilizar a plataforma como uma maneira de *marketing*, criando contas para elogiar os próprios produtos ou até mesmo criticá-los de modo que o assunto venha à tona de maneira orgânica, permitindo a visualização da opinião de seus consumidores de maneira rápida, fácil e principalmente, natural (Soboleva, 2018) [59].

Portanto, entender a plataforma e os recursos que ela oferece ao usuário é essencial quando se trata de análise de sentimentos no Twitter. A frequência de uso e o conhecimento da plataforma permite ao pesquisador melhor compreensão dos assuntos e da maneira com que eles surgem, característica necessária quando se fala de classificação manual.

6 CONCLUSÕES

Este trabalho teve como objetivo explorar conceitos da Análise de Sentimentos em redes sociais. Nesse contexto, o Twitter foi escolhido para compor a base de dados a ser classificada por suas vantagens e facilidades. A pesquisa teve seu foco em comparar os modelos Naive Bayes, Regressão Logística e *Support Vector Machine* a partir de métricas como, acurácia, precisão, F1-score e *recall*. O conjunto de dados foi formado a partir de *tweets* em português sobre o *smartphone* iPhone.

Com base nos resultados obtidos têm-se as seguintes conclusões quanto à coleta de dados, o pré-processamento, o treinamento e teste, à análise e à validação:

- *Coleta*: Foi identificado que poucas pesquisas foram desenvolvidas na área de análise de sentimento para documentos em português, sobretudo em redes sociais. Essa etapa é fundamental, pois proporciona a coleta de um grande conjunto de *tweets* relacionados ao domínio sem a necessidade de o usuário precisar pesquisar no Twitter para encontrar todas “discussões” sobre um determinado assunto. A Rede Twitter contribuiu para o andamento da pesquisa, a permissão ao acesso dos dados ocorreu em três dias e permite que sejam escolhidos dados de qualquer período, quantidade e foco de estudo.

- *Pré-processamento*: A opção de classificação manual possibilitou a evolução do trabalho semântico permitindo, por exemplo, *tweets* com sarcasmo ou ironia, distinção entre *tweets* opinativos e informativos, *tweets* com várias opiniões descritivas sobre um ou vários assuntos, distinção entre *tweets* formais e informais, etc. De fato, essa classificação contribuiu para os resultados apresentados, deixando a dúvida do quanto os resultados se distorceriam caso a classificação fosse realizada por meios computacionais. Isso torna essa etapa da ferramenta muito importante para o seu resultado final. No entanto o pré processamento pode ter sido impactado pela maneira a qual se classificaram os dados, seja por uma polaridade específica (-1, 0, +1) ou por um intervalo que represente uma polaridade (ex: positivos = 1 a 5).

- *Treinamento e teste*: A etapa de treinamento e teste é totalmente necessária, pois completa o objetivo da ferramenta, que é prover o sentimento desses *tweets*, além de servir como teste para validar o desempenho dos modelos. É nesse momento que ocorre o contato dos modelos com os dados. E a partir disso, Naive Bayes, Regressão Logística e *Support Vector Machine* obtiveram uma pontuação de ajuste de, respectivamente, 74%, 78% e 79%. Pontuações relativamente baixas para um ajuste, é possível que haja um aumento se houver diminuição do volume de regularização (treinamento).

- *Validação*: De uma maneira geral, algoritmos Naive Bayes, Regressão Logística e *Support Vector Machine* apresentaram resultados no modelo binomial de até 74% de acerto, utilizando as classes: “positiva” e “negativa”. O algoritmo Naive Bayes apresentou melhores parâmetros em todas as categorias, exceto na classificação das frases negativas (houveram mais falsos positivos) onde ficou atrás do algoritmo *Support Vector Machine*. As matrizes de confusão apresentaram similaridade na quantidade de manchetes mal classificadas, tanto positivas quanto negativas.

De modo geral, observa-se que os modelos conseguem prever corretamente cerca de 56% de *tweets* positivos e 44% negativos. A validação cruzada apresentou uma precisão média e a pontuação da F1 semelhantes ao que foi visto em uma única iteração, ou seja, neste trabalho não houve resultado significativo em comparação aos demais resultados. Embora o classificador de votação tenha um bom desempenho na média, ele não apresentou diferenças nos resultados para Naive Bayes, Regressão Logística e *Support Vector Machine*.

De uma forma geral, a flexibilidade que a análise de sentimentos possui é muito grande e, portanto, abrange um universo muito amplo de pesquisa e alcance de informações que podem ser exploradas em pesquisas futuras.

Sugestões de trabalhos futuros serão apresentados no próximo capítulo.

7 SUGESTÃO PARA TRABALHOS FUTUROS

Com base no estudo realizado sobre a comparação de algoritmos para a análise de sentimento sugere-se para o prosseguimento das pesquisas:

- Avaliar o desempenho de outros modelos, por exemplo, *Random Forest*, Redes Neurais Convolucionais, Ada Boost, Árvore de Decisão, *K-Nearest Neighbors*, etc.
- Avaliar a influência da classificação por escala;
- Comparar com outras abordagens emergentes na área de análise de sentimento como o *Deep Learning*;
- Utilizar outras redes sociais;
- Utilizar a geolocalização para identificar regiões que mais comentam sobre tais assuntos identificando a polaridade das mensagens;
- Trabalhar com uma proporção diferente para dados de treinamento e teste;
- Avaliar a influência do tamanho do conjunto de dados no desempenho dos modelos;
- Comparar entre as redes sociais, estudando o comportamento dos usuários a fim de verificar características de cada uma delas;
- Analisar tentativas de influência identificando o surgimento de assuntos emergentes e como eles se relacionam com a mudança de sentimento;
- Identificar criadores de assuntos emergentes por meio de uma classificação de confiabilidade dos assuntos surgem;
- Utilizar conjuntos de dados de outros idiomas para treinar o classificador;
- Avaliar o desempenho dos modelos a partir de uma especificação decimal da polaridade (valores entre 0 e 1);
- Analisar outro tipo de produto ou serviço;
- Criar um algoritmo para a classificação.

8 REFERÊNCIAS

- [1] Akinsola, J E T. (2017). **Supervised Machine Learning Algorithms: Classification and Comparison**. International Journal of Computer Trends and Technology (IJCTT). 48. 128 - 138. 10.14445/22312803/IJCTT-V48P126.
- [2] Albuquerque, J. Mulatinho, N. Nunes, A. M. Queiroz, F. S. Lima, S.R. et al. **Difusão da informação no Twitter: Fluxo da informação na audiência invisível**. ENANCIB 2013 [Internet].
- [3] Alves, K. **História da Apple: a marca da maçã mais famosa do mundo**. 2021. Disponível em: <https://www.consolidesuamarca.com.br/blog/historia-da-apple>.
- [4] Anagnostopoulos, T., Skourlas, C. (2014). **Ensemble Majority Voting Classifier for Speech Emotion Recognition and Prediction**. Journal of Systems and Information Technology. 16. 10.1108/JSIT-01-2014-0009.
- [5] Assiri, A. S., Nazir, S., Velastin, S. A. **Breast Tumor Classification Using an Ensemble Machine Learning Method**. Journal of Imaging, 6(6), 39. (2020). doi:10.3390/jimaging6060039
- [6] Atallah, R., Al-Mousa, A. **Heart Disease Detection Using Machine Learning Majority Voting Ensemble Method**. 2019 2nd International Conference on New Trends in Computing Sciences (ICTCS) (2019). doi:10.1109/ictcs.2019.8923053
- [7] Awad, M., Khanna, R. (2015). **Support Vector Machines for Classification**. 10.1007/978-1-4302-5990-9_3.
- [8] Berrar, D. (2018). **Cross-Validation**. 10.1016/B978-0-12-809633-8.20349-X.
- [9] Boser, B. E.; Guyon, I. M.; Vapnik, V. N. **A Training Algorithm for Optimal Margin Classifiers**. In: Annual Workshop on Computational Learning, 5, 1992, Pittsburgh. ACM Press. Pittsburgh: Haussler D, jul 1992. p.144-152 .
- [10] Campos, V. F. **Gerenciamento da rotina do trabalho do dia dia**.9.ed.Nova Lima: INDG Tecnologia e Serviços Ltda, 2004
- [11] Cavalcante, P. E. C. and Malheiros, Y. d. A. (2017). **Um dataset para análise de sentimentos na língua portuguesa**. Trabalho de conclusão de curso, Bacharel em Sistemas de Informação, universidade Federal da Paraíba.
- [12] Chawla, Nitesh., Bowyer, Kevin., Hall, Lawrence., Kegelmeyer, W.. (2002). **SMOTE: Synthetic Minority Over-sampling Technique**. J. Artif. Intell. Res. (JAIR). 16. 321-357. 10.1613/jair.953.

- [13] Clark, S. Wicentwoski, R. : **SwatCS: combining simple classifiers with estimated accuracy**, Second Joint Conference on Lexical and Computational Semantics (*SEM), Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), vol. 2, Association for Computational Linguistics, Atlanta, Georgia, USA, 2013, pp. 425–429
- [14] Chong, W. Y., Selvaretnam, B., Soon, L.-K. (2014). **Natural Language Processing for Sentiment Analysis: An Exploratory Analysis on Tweets**. 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology. doi:10.1109/icaiet.2014.43
- [15] Dalianis, H. (2018). **Evaluation Metrics and Evaluation**. *Clinical Text Mining*, 45–53. doi:10.1007/978-3-319-78503-5_6
- [16] Da Silva, N. F. F., Hruschka, E. R., Hruschka, E. R. (2014). **Tweet sentiment analysis with classifier ensembles**. *Decision Support Systems*, 66, 170–179.
- [17] Dehghani, M., (2013). **The Role of Social Media on Advertising: A Research on Effectiveness of Facebook Advertising on Enhancing Brand Image**. *Studies in Computational Intelligence*, 59p. Thesis (Master of Arts) - Eastern Mediterranean University, Gazimağusa, North Cyprus, 2013.
- [18] Disha C. K., Harshali P. **Sentiment Analysis of Tweets using Machine Learning**. *International Journal of Advance Research, Ideas and Innovations in Technology* 7.3 (2021). www.IJARIIT.com.
- [19] Farhadloo, M., Rolland, E. (2016). **Fundamentals of Sentiment Analysis and Its Applications**. *Studies in Computational Intelligence*, 1–24.
- [20] Ferreira, M. A. M.; Celso, A. S dos S.; Neto, J. E. B. **Aplicação do modelo logit binomial na análise do risco de crédito em uma instituição bancária**. *Revista de Negócios, Blumenau*, v. 17, n. 1, p. 41-59, 2012.
- [21] Gaur, N., Sharma, N. (2017). **Sentiment Analysis in Natural Language Processing**. *International Journal of Engineering and Techniques*. 3, no. 3, pp. 144-148.
- [22] Gautam, G., Yadav, D. (2014). **Sentiment analysis of twitter data using machine learning approaches and semantic analysis**. 2014 Seventh International Conference on Contemporary Computing (IC3).
- [23] Goldberg, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. Reading, MA, USA: Addison-Wesley, 1989.
- [24] Hair, J. F., Anderson, R. E., Tatham, R. L., and Black, W. C. (1998). **Multivariate Data Analysis**, Fifth Edition, Prentice-Hall International, Inc.

- [25] Hasan, A., Moin, S., Karim, A., Shamshirband, S. (2018). **Machine Learning-Based Sentiment Analysis for Twitter Accounts**. Mathematical and Computational Applications, 23(1), 11.
- [26] Hosmer, D. W., Lemeshow, S. (2000). **Applied logistic regression (Wiley Series in probability and statistics)**. Wiley-Interscience Publication. ISBN: 0471356328
- [27] Hossin, M., Sulaiman, M. N. (2015). **A Review on Evaluation Metrics for Data Classification Evaluations**. International Journal of Data Mining Knowledge Management Process. 5. 01-11. 10.5121/ijdkp.2015.5201.
- [28] Hotho, Andreas., Nürnberger, Andreas., Paass, Gerhard. (2005). **A Brief Survey of Text Mining**. LDV Forum - GLDV Journal for Computational Linguistics and Language Technology. 20. 19-62.
- [29] IDC. (2020) **Global Smartphone Shipments Expected to Drop Nearly 10% in 2020, But a Strong 5G Push Is Expected to Bring the Market Back to Growth in 2021, According to IDC**. Disponível em: <https://www.idc.com/getdoc.jsp?containerId=prUS46802520>
- [30] Janiesch, C., Zschech, P., Heinrich, K. (2021). **Machine learning and deep learning**. Electronic Markets. doi:10.1007/s12525-021-00475-2
- [31] Jianqiang, Z., Xiaolin, G. (2017). **Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis**. IEEE Access, 5, 2870–2879.
- [32] Kirilenko, A. P., Stepchenkova, S. O., Kim, H., Li, X. (Robert). **Automated Sentiment Analysis in Tourism: Comparison of Approaches**. Journal of Travel Research, (2017). doi:10.1177/0047287517729757
- [33] Kohavi, R. (1995). **A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection**. IJCAI.
- [34] Kotsiantis, S., Kanellopoulos, D., Pintelas, P. (2005). **Handling imbalanced datasets: A review**. GESTS International Transactions on Computer Science and Engineering. 30. 25-36.
- [35] Kumar, S., Morstatter, F., and Liu, H. (2013). **Twitter Data Analytics**. Springer Publishing Company, Incorporated.
- [36] Lewis, D. D. (1998). **Naive (Bayes) at forty: The independence assumption in information retrieval**. Lecture Notes in Computer Science, 4–15. doi:10.1007/bfb0026666
- [37] Lewis, J. P, **Tutorial on SVM**, CGIT Lab, USC, 2004.

- [38] Liu B. **Sentiment analysis and subjectivity**. Handbook of Natural Language Processing, Second Edition Taylor and Francis Group, Boca. 2010.
- [39] Markoulidakis, I.; Rallis, I.; Georgoulas, I.; Kopsiaftis, G.; Doulamis, A.; Doulamis, N. **Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem**. Technologies 2021, 9, 81. <https://doi.org/10.3390/technologies9040081>
- [40] Medhat, W., Hassan, A., Korashy, H. (2014). **Sentiment analysis algorithms and applications: A survey**. Ain Shams Engineering Journal, 5(4), 1093–1113. doi:10.1016/j.asej.2014.04.011
- [41] Mung, P. S., Phyu, S. **Effective Analytics on Healthcare Big Data Using Ensemble Learning**. 2020 IEEE Conference on Computer Applications(ICCA) (2020). doi:10.1109/icca49400.2020.9022853
- [42] Nascimento, P., Aguas, R., Lima, D.D., Kong, X., Osiek, B.: **Análise de sentimento de tweets com foco em notícias**. In I Brazilian Workshop on Social Network Analysis and Mining, 2012
- [43] Neethu, M. S., Rajasree, R. (2013). **Sentiment analysis in twitter using machine learning techniques**. 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT).
- [44] Osiogogu, U. (2019). **Semantic Sentiment Analysis Based on Probabilistic Graphical Models and Recurrent Neural Network**. 102p.Thesis (Master of Science) - African University of Science and Technology, Abuja, Nigeria, 2019.
- [45] Pak, A., Paroubek, P. **Twitter as a corpus for sentiment analysis and opinion mining**. Proceedings of LREC (2010).
- [46] Pang B, Lee L. **Opinion mining and sentiment analysis**. Found Trends Inf Retr.janeiro de 2008;2(1-2):1–135.
- [47] Powers, D. M. W. (2011). **Evaluation: From precision, recall and F-measure to ROC, informedness, markedness correlation**. J. Mach. Learn. Technol. 2. 2229-3981. 10.9735/2229-3981.
- [48] Prabhat, A., Khullar, V. (2017). **Sentiment classification on big data using Naïve bayes and logistic regression**. 2017 International Conference on Computer Communication and Informatics (ICCCI).
- [49] Rajput, R Solanki, Arun. **Real time sentiment analysis of tweets using machine learning and semantic analysis** (2016). 687-692. 10.1201/9781315364094-123.

- [50] Ramadhan, W. P., Novianty, S. T. M. T. A., Setianingsih, S. T. M. T. C. (2017). **Sentiment analysis using multinomial logistic regression**. 2017 International Conference on Control, Electronics, Renewable Energy and Communications (ICCREC).
- [51] Rane, A., Kumar, A. (2018). **Sentiment Classification System of Twitter Data for US Airline Service Analysis**. 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)
- [52] Re, M., Valentini, G.: **Ensemble Methods: A Review In Advances in Machine Learning and Data Mining for Astronomy**, pp. 563-594. Chapman Hall, Boca Raton (2012).
- [53] Refaeilzadeh, P., Tang, L. Liu, H. (2009). **Cross-Validation**. Encyclopedia of Database Systems. 532–538. 532-538. 10.1007/978-0-387-39940-9_565.
- [54] Rindflesch, Thomas. (1996). **Natural Language Processing. Annual Review of Applied Linguistics**. 16. 70 - 85. 10.1017/S0267190500001446.
- [55] Rutten, Lex. (2010). **Bayes' theorem: scientific assessment of experience**. International Journal of High Dilution Research. 9. 10.51910/ijhdr.v9i32.370.
- [56] Saad, S. E., Yang, J. (2019). **Twitter Sentiment Analysis Based on Ordinal Regression**. IEEE Access, 7, 163677–163685. doi:10.1109/access.2019.2952127
- [57] Saleh, Z. (2019). **Artificial Intelligence Definition, Ethics and Standards**.
- [58] Singh, G., Vedrtam, A., Sagar, D. (2013). **An Overview of Artificial Intelligence**. 10.13140/RG.2.2.20660.19840.
- [59] Soboleva, A. **Marketing with Twitter: Investigating Factors That Impact on the Effectiveness of Tweets**. Western Sydney University, Apr. 2018, hdl.handle.net/1959.7/uws:47360.
- [60] Soumya, S., Pramod, K.V. (2020). **Sentiment analysis of malayalam tweets using machine learning techniques**. ICT Express. doi:10.1016/j.icte.2020.04.003
- [61] Souza, F. P. **Engenhara da Qualidade**. Indaial: Uniasselvi, 2010. 198 p.
- [62] Symeonidis, S., Effrosynidis, D., Arampatzis, A. (2018). **A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis**. Expert Systems with Applications, 110, 298–310.
- [63] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M., “**Lexiconbased methods for sentiment analysis**,” Computational linguistics 37, no. 2, pp. 267-307, 2011.
- [64] Tope, M. **Email Spam Detection Using Naive Bayes Classifie**, International Journal of Science & Engineering Development Research, Vol.4, Issue 6, page no.1 - 7, June-2019.

- [65] Twitter. **Sobre a empresa.** 2022. Disponível em: <https://about.twitter.com/who-we-are/our-company> .
- [66] Twitter (2021). **Sobre as apis do twitter.** Recuperado de: <https://help.twitter.com/pt/rules-and-policies/twitter-api>.
- [67] Ulusoy, I., Bishop, C. M. (2006). **Comparison of Generative and Discriminative Techniques for Object Detection and Classification.** Lecture Notes in Computer Science, 173–195. doi:10.1007/11957959_9
- [68] Vujovic, Z. (2021). **Classification Model Evaluation Metrics.** International Journal of Advanced Computer Science and Applications. Volume 12. 599-606. 10.14569/I-JACSA.2021.0120670.
- [69] Wankhade, M., Chandra, A., Rao, S., Dara, S., Kaushik, B. (2017). **A Sentiment Analysis of Food Review using Logistic Regression.** 2456-3307.
- [70] Witten, I. H., Ian H., Frank, E., Hall, M. A., Mark A. e Pal, C. J. **Data mining: practical machine learning tools and techniques.** 3a ed. Burlington: Elsevier/Morgan Kaufmann, 2011.

9 APÊNDICE

9.1 APÊNDICE 1: SCRIPT ANÁLISE DA FREQUÊNCIA DE PALAVRAS

```

1  clc
   clear
3  // -----
   // ----- Definicao de global -----
5  // -----

7  global x_exp y_exp

9  // -----
   // ----- Dados de entrada -----
11 // -----

13 x_exp=[1:30];

15 y_exp=[4066 617 391 346 342 330 293 208 186 170 ...
164 163 161 160 159 158 156 144 139 138 ...
17 134 132 126 120 119 113 106 103 100 100]; //positivos

19 y_exp=[3259 511 382 325 311 294 248 216 176 176 ...
168 167 158 151 141 141 127 125 115 112 ...
21 111 110 108 103 98 6 91 90 84 83]; //negativos

23 x_exp=x_exp';
   y_exp=y_exp';

25
   // -----
27 // ----- MODELO -----
   // -----

29
   function y_teorico=y_cal(x,par)

31
       y_teorico=par(1)*x^(-par(2)); // par(1)=k par(2)=tau

33
   endfunction

35
   // -----
37 // ----- FUNCAO DE OTIMIZACAO -----
   // -----

39
   function e=myfun(par, x_exp, y_exp)

41

```

```

    global x_exp y_exp
43     e=y_cal(x_exp,par)-y_exp

45 endfunction

47 // -----
48 // ----- Definindo os valores iniciais de busca para os parametros -----
49 // -----

51 par_inicial=[1 1];

53 // -----
54 // ----- Chamando a rotina que vai encontrar os parametros -----
55 // -----

57 [fopt,par_opt, gopt] = leastsq(list(myfun,x_exp, y_exp),par_inicial)

59 // -----
60 // ----- Imprimindo os valores -----
61 // -----

63 printf('Valor da funcao de otimizacao = %6.4e\n',fopt);
64 printf('k = %6.4e\n',par_opt(1));
65 printf('tau = %6.4e\n',par_opt(2));

67 // -----
68 // ----- Calculando e Imprimindo R^2 -----
69 // -----

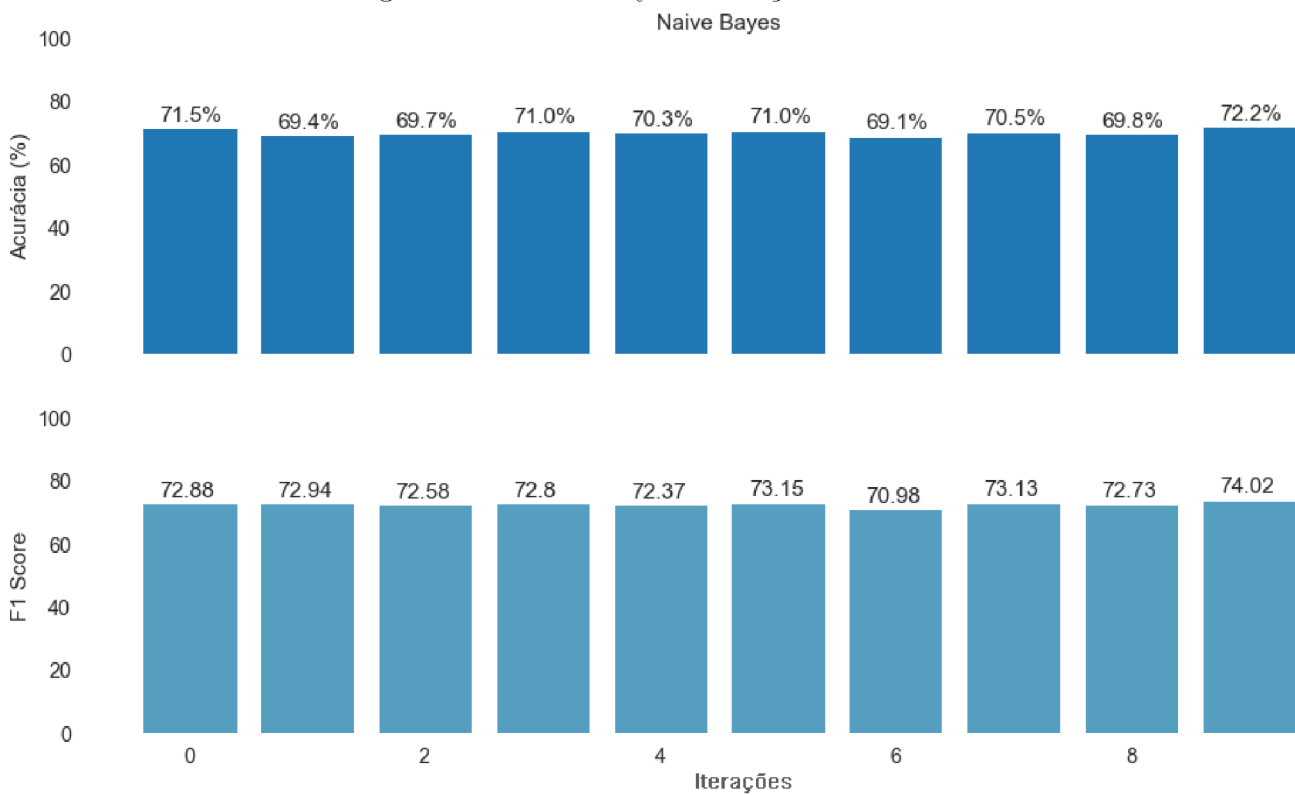
71 correlacao=correl(y_cal(x_exp,par_opt),y_exp);
72 printf('R^2 = %6.4e\n',correlacao^2);

73
74 previsao=y_cal(x_exp,par_opt)
75
76 // -----
77 // ----- Graficando y_experimental e y_calculado em funcao de x -----
78 // -----
79
80 plot(x_exp,previsao,'ro',x_exp,y_exp,'k+')
81 legend(['Calculado', 'Experimental'],1);
82 xtitle(' ','Ordem da palavra (n)', 'Frequencia (r)')
```

Listagem 24: Análise frequência de palavras a partir da polaridade

9.2 APÊNDICE 2: RESULTADOS VALIDAÇÃO CRUZADA

Figura 9.1: Naive Bayes validação cruzada.

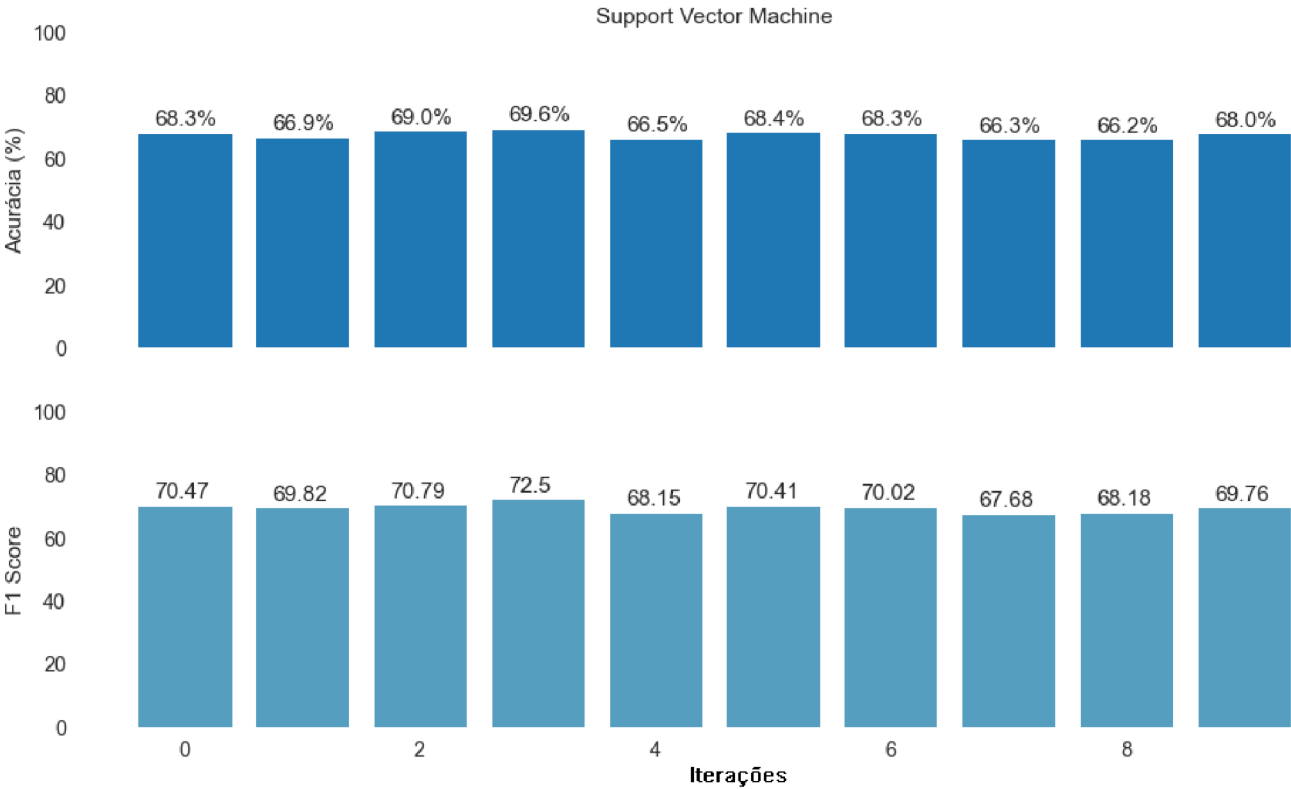


Fonte: Autoria própria (2022).

Media Acuracia : 70.46%					
Media F1 Score : 72.76%					
Media Matriz de Confusao :					
[449.2 201.9]					
[226.2 571.7]					
		precisao	recall	f1-score	dados
neg	0.70	0.71	0.70	669	
pos	0.74	0.74	0.74	780	
acuracia			0.72	1449	
media macro	0.72	0.72	0.72	1449	
media pesada	0.72	0.72	0.72	1449	

Listagem 25: Naive Bayes validação cruzada.

Figura 9.3: SVM validação cruzada.



Fonte: Autoria própria (2022).

Media Acuracia:	67.76%				
Media F1 Score:	69.78%				
Media Matriz de Confusao:					
	[442.2 208.1]				
	[259. 539.7]				
		precisao	recall	f1-score	dados
	neg	0.64	0.69	0.66	659
	pos	0.72	0.68	0.70	790
	acuracia			0.68	1449
	media macro	0.68	0.68	0.68	1449
	media pesada	0.68	0.68	0.68	1449

Listagem 27: *Support Vector Machine* validação cruzada.