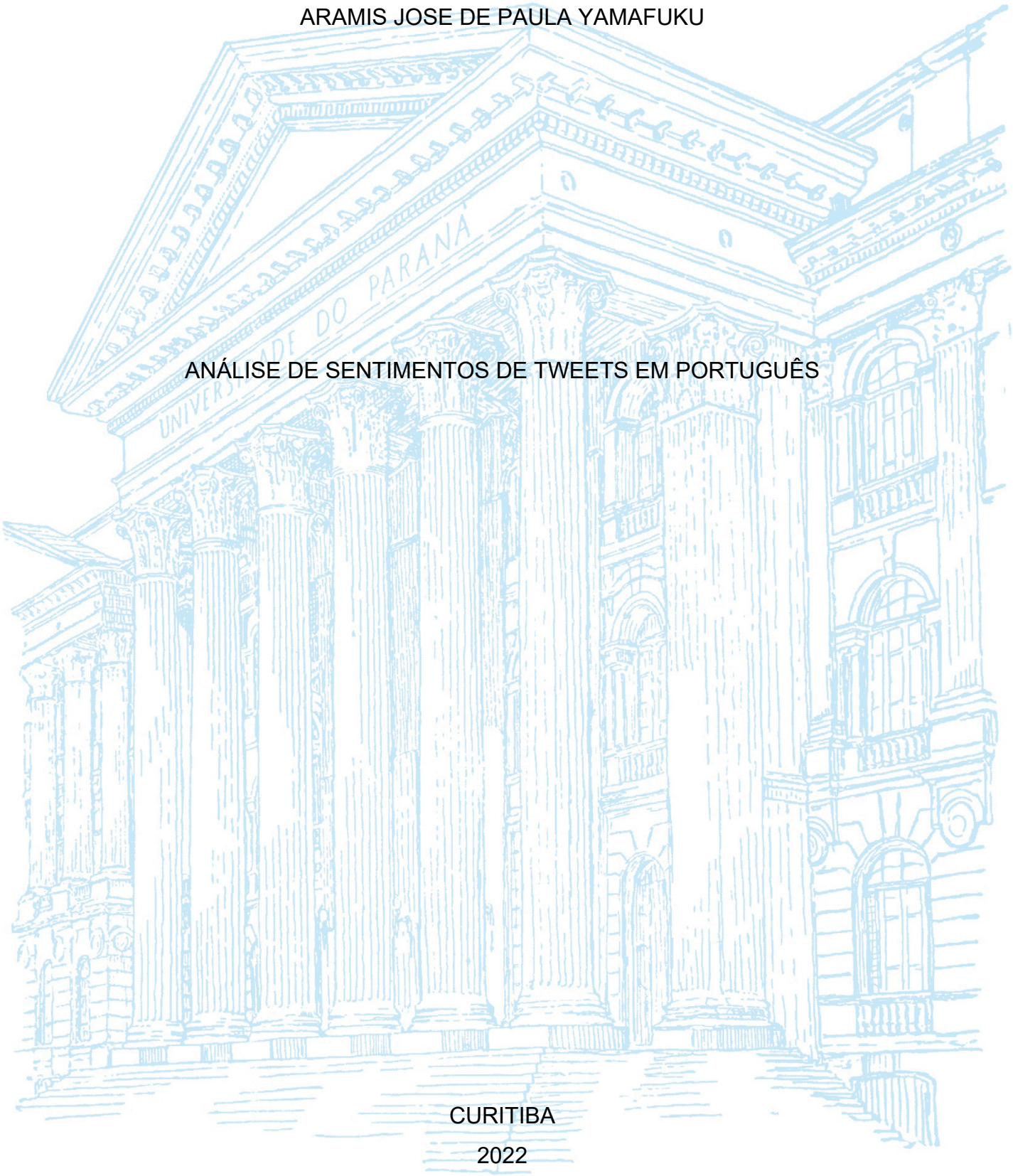
UNIVERSIDADE FEDERAL DO PARANÁ

ARAMIS JOSE DE PAULA YAMAFUKU



ANÁLISE DE SENTIMENTOS DE TWEETS EM PORTUGUÊS

CURITIBA

2022

ARAMIS JOSE DE PAULA YAMAFUKU

ANÁLISE DE SENTIMENTOS DE TWEETS EM PORTUGUÊS

Trabalho de Conclusão de Curso apresentado ao curso de Pós-Graduação em Inteligência Artificial Aplicada Setor de educação profissional e tecnológica, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Especialista em Inteligência Artificial.

Orientador: Prof. Dr. Alexander Kutzke

CIDADE

2022

MINISTÉRIO DA EDUCAÇÃO
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
CURSO DE PÓS-GRADUAÇÃO INTELIGÊNCIA ARTIFICIAL APLICADA - 40001016348E1

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INTELIGÊNCIA ARTIFICIAL APLICADA da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **ARAMIS JOSÉ DE PAULA YAMAFUKU** intitulada: **Análise de Sentimentos de Tweets em Português**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 27 de Outubro de 2022.

ALEXANDER ROBERT KUTZKE
Presidente da Banca Examinadora

RAZER ANTHOM NIZER ROJAS MONTAÑO
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Análise de sentimentos de tweets em português

Aramis José de Paula Yamafuku
Especialização em Inteligência Artificial Aplicada - Universidade Federal do Paraná
Curitiba, Brasil
aramis.yamafuku@ufpr.br

Alexander Kutzke
Especialização em Inteligência Artificial Aplicada - Universidade Federal do Paraná
Curitiba, Brasil
alexander@ufpr.br

Resumo — Com a evolução exponencial das interações digitais entre empresas e clientes e a velocidade com que a acessibilidade a internet cresceu, atualmente responder seus clientes não é suficiente para as empresas, elas precisam ser cada vez mais rápidas, responder no timing adequado e ser assertivas nessas respostas. Dentro deste contexto, este trabalho tem o objetivo de apresentar resultados de classificação de sentimentos com textos extraídos do Twitter para algumas marcas do mercado de CFT (cosméticos, fragrâncias e higiene pessoal) e alguns concorrentes diretos do ramo de presenteáveis e bomboniére utilizando alguns algoritmos de IA (inteligência artificial) como: Árvore de Decisão, Random Forest, Naive Bayes, SVM, Regressão Logística e LSTM. O classificador que apresentou melhor resultado foi o SVM com análises de unigrama e a abordagem TF-IDF (*Term Frequency - Inverse Document Frequency)* o algoritmo obteve uma acurácia de 75% para classificação.
Palavras-chaves: Interações, algoritmos, sentimentos, twitter

Abstract — With the exponential evolution of digital interactions between companies and customers and the speed with which internet accessibility has grown, currently responding to their customers is not enough for companies, they need to be faster and faster, respond in the proper timing and be assertive in those answers. Within this context, this work aims to present sentiment classification results with texts extracted from Twitter for some brands in the CFT market (Cosmetics, Fragrances and Toilletries Market) and some direct competitors in the gift and bonbonniere business using some algorithms of AI (artificial intelligence) such as: Decision Tree, Random Forest, Naive Bayes, SVM, Logistic Regression and LSTM. The classifier that presented the best result was the SVM with unigram analysis and the TF-IDF (Term Frequency - Inverse Document Frequency) approach, the algorithm obtained an accuracy of 75% for classification.
Keywords: Interactions, algorithms, feelings, twitter

I. INTRODUÇÃO

Não é de hoje que as interações digitais entre clientes e empresas acontecem, e estão ganhando muita força nos últimos anos, principalmente em 2020, início da pandemia da COVID-19. Neste período, muitas lojas físicas ficaram fechadas e a compra online era, em muitos casos, a única opção. Segundo MCC-ENET, primeiro indicador para acompanhamento da evolução dos preços do varejo online brasileiro, em 2020 o volume de vendas no e-commerce brasileiro aumentou 73,88% [1].

Além do aumento das vendas online, a acessibilidade a internet pela população brasileira também aumentou. O percentual de domicílios com acesso à internet chegou a 82,7% em 2019 segundo Governo [2]. Quando considera apenas regiões urbanas o número aumenta para 86,7%. Como consequência do aumento do volume de vendas online e uma maior acessibilidade a internet pela população, também aumentam a quantidade de interações online entre clientes e empresas, seja para tirar dúvidas, realizar compras, fazer elogios ou reclamações. Devido a esse aumento de interações, muitas empresas precisam se adaptar, não apenas em responder aos seus consumidores, mas responder de forma rápida e assim ajudar a formar a reputação digital [3] da empresa. Segundo a Review Trackers [4], 63,6% dos consumidores pesquisam no Google antes de comprar qualquer produto. Outro ponto identificado é que 53% dos consumidores esperam ser respondidos em menos de uma semana.

Em uma pesquisa da Forbes [5] informa que as duas principais características das empresas líderes em *Customer Experience* são: Comunicação e Escuta. Essas duas características reforçam ainda mais que o relacionamento com o cliente não é um monologo, segundo Daniela Cachich [6] “A publicidade passou por um momento de monólogo, virou um diálogo e agora a gente tem uma retroalimentação entre marcas e consumidores”.

Existe um grande desafio para muitas empresas que é utilizar de todos esses novos dados e interações a seu favor para conseguir melhorar toda a jornada e experiência do cliente. Será essencial dar as respostas no tempo correto e conseguir ter uma visão mais ampla e completa de sua empresa, conseguindo enxergar as dores de seus consumidores e pontos a melhorar em seus processos. Segundo Ivan Petri [7], a análise dos dados e a velocidade da informação são essenciais para entender o consumidor e entregar o que ele busca. Algumas empresas já se ajustaram esse novo cenário, porém ainda existem muitas que precisam se adequar. Em um estudo realizado pela Super Office [8],

62% das empresas não respondem aos e-mails de suporte ao cliente.

Muitas empresas nos períodos de maior movimento (Natal, *Black Friday*, entre outros), precisam alocar diversas pessoas para identificar de forma manual o sentimento das interações dos clientes originadas pelo site da empresa para depois direcionar para o departamento correto. Todo esse processo manual, deixa o fluxo muito lento e moroso o que acaba implicando em um tempo maior para responder aos clientes e um custo operacional muito alto.

Buscando uma solução para esse problema, este trabalho realiza análise de sentimentos em textos em português, utilizando processamento de Linguagem Natural (PLN) e técnicas de aprendizado de máquina para conseguir distinguir o sentimento de cada sentença com o objetivo de apoiar a empresa na questão de identificar sentimentos em 3 polaridades diferentes (positivos, negativos e neutros) de forma mais rápida.

Para a extração dos textos, o trabalho guiou-se em obter informações dos *tweets*, nome dado as postagens realizadas no Twitter com limitação de 140 caracteres por *post*. Atualmente, o Twitter é uma das redes sociais mais utilizadas pelos clientes para realizar interações com as empresas e conta com milhões de usuários e com mais de 500 milhões de postagens por dia [11].

Com estas informações tem-se o intuito de responder o seguinte questionamento: Será que existem padrões textuais nestas interações com a qual seja possível identificar de forma automática aspectos-chave do produto/atendimento?

É apresentado um comparativo entre performance de alguns algoritmos de classificação para uma base de dados capturada do Twitter utilizando como filtro o nome de algumas marcas do mercado de CFT (*cosmetics, fragrancies and toilletries)*, também de alguns concorrentes diretos de presenteáveis e *bomboniere*.

Para o desenvolvimento deste projeto foi necessário compreender conceitos e teorias e estes estão dispostos na Seção II onde constam detalhamento de atendimento a clientes na atualidade, EDA, PLN, trabalhos relacionados, os algoritmos LSTM, SVM, Naive Bayes, Decision Tree, Random Forest e Regressão Logística. A Seção III trata de Materiais e Métodos aplicados na pesquisa, e apresenta os temas: Coleta e estruturação dos dados, EDA, Pré-processamento dos dados, Separação dos dados e Treinamento do modelo. A Seção IV, detalham os Resultados dos modelos e por último a seção V apresenta as considerações finais.

II. FUNDAMENTAÇÃO TEÓRICA

Este trabalho tem como objetivo testar diversos algoritmos para identificar sentimentos em 3 polaridades diferentes (positivo, negativo e neutro) para textos capturados do Twitter utilizando Processamento de Linguagem Natural (PLN) e técnicas de aprendizado de máquina. Este capítulo irá explorar os assuntos necessários para realização deste trabalho.

Esta seção está subdividida em 9 seções onde a seção A apresenta sobre o atendimento a clientes na atualidade, a seção B informa sobre a Análise de Dados Exploratórios (EDA), a seção C descreve sobre PLN ou Processamento de Linguagem Natural, a D mostra alguns trabalhos relacionados ao tema. A partir da seção E são descritos alguns algoritmos sendo *Long Short-Term Memory* (LSTM) o primeiro, a F o *Support Vector Machine* (SVM), a seção G apresenta o Naives Bayes, a H o Random Forest, a I detalha o algoritmo Decision Tree (Arvore de decisão), a seção J demonstra o algoritmo de Logistic Regression (Regressão Logística), na seção K é apresentado as métricas de performance utilizadas no artigo e por último, a seção L apresenta sobre *Bag-of-words* e *TF-IDF*, técnicas para *feature extraction.*

A. ATENDIMENTO A CLIENTES NA ATUALIDADE

Não é de hoje que as empresas sabem da importância da satisfação e de uma boa jornada / experiência do cliente. Existe concorrência em praticamente todas as áreas e sai na frente a empresa que está melhor preparada e foca no pós venda, no estreitamento do relacionamento com o cliente e não se preocupa apenas em realizar uma venda, mas sim em encantar seu cliente ao longo da jornada. Como diz uma frase de Philip Kotler, conhecido como pai do marketing, "a melhor propaganda é feita por clientes satisfeitos." [9].

Philip Kotler [10] também dizia que: "Conquistar novos clientes custa entre 5 e 7 vezes mais do que manter os já existentes". Manter o cliente é essencial, mas muito difícil de atingir pois para isso demanda de um bem muito precioso na relação empresa-consumidor, a confiança.

Para conquistar a confiança é necessário estar muito próximo ao cliente, focar no pós venda, investir em tecnologia, pessoas e atendimento. Isso envolve todos os setores da empresa e nos dias de hoje, com alto volume de interações digitais, com a informação trafegando cada vez mais rápido e chegando a todos os cantos do mundo em um instante, se torna um desafio maior ainda.

A empresa ReviewTrackers [4], realizou uma pesquisa em 2021 para obter informações dos consumidores e identificaram que:

- 63% dos consumidores dizem que fazem pesquisas no Google antes de visitar uma loja;
- Consumidores esperam que as marcas respondam suas interações e eles estão desapontados. 53% dos consumidores esperam que sejam respondidos em menos de 1 semana. 63% informaram que as lojas nunca os responderam;
- Avaliações negativas afastam os clientes, 94% dos consumidores informaram que avaliações negativas os convencem de não ir a uma determinada loja;
- Clientes não confiam em lojas com menos de 4 estrelas de avaliação. 80% dos consumidores informaram que eles realmente confiam em empresas com avaliações entre 4 e 5 estrelas.

Essa pesquisa mostra um pouco da importância da reputação digital [3] e os desafios que as empresas tem pela

frente. Com os clientes cada vez mais digitais, pesquisando na internet antes de comprar em uma loja e sempre à espera de um retorno rápido por parte das empresas, é de suma importância investir em tecnologia para conseguir dar tração e ganhar velocidade nas interações com os clientes e assim, conseguir melhorar sua reputação e conquistar a confiança dos clientes.

B. EDA - ANÁLISE DE DADOS EXPLORATÓRIO

O processo de análise exploratória (*EDA - Exploratory Data Analysis*) foi originalmente desenvolvido por John Tukey, matemático norte-americano. Também, conforme USP [12], o processo e exploração é essencial para o entendimento dos dados e das relações existentes entre as variáveis e resumir suas principais características, geralmente usando métodos de visualização de dados.

Segundo Chatfield [13], o processo de EDA tem como um de seus objetivos a descrição dos dados e nesse processo inclui validar a qualidade dos dados, realizar cálculo estatísticos sumarizados, plotar os gráficos apropriados e talvez utilizar técnicas de *data-analytics* mais avanças como Análise dos Componentes Principais (PCA - *Principal Component Analysis*).

C. PLN – PROCESSAMENTO DE LINGUAGEM NATURAL

Segundo Liddy [14], processamento de Linguagem Natural (PLN ou NLP – *Natural Language Processing*) é o conjunto de técnicas teórico-computacionais que analisam e representam dados textuais, com o objetivo de processar linguagem humana para vários tipos de tarefas e aplicações [14].

PNL é uma das ramificações da Inteligência Artificial que ajuda a computadores a entender e interpretar a linguagem humana / linguagem natural.

Pode-se utilizar a PNL em diversas disciplinas, como ciências da computação e informação, linguística, matemática, engenharia elétrica e eletrônica, inteligência artificial e robótica, psicologia etc. As aplicações da PNL incluem vários campos de estudos, como tradução automática, processamento de texto em linguagem natural, interfaces de usuário, recuperação multilíngue e *cross language information* (CLIR), reconhecimento de fala, inteligência artificial, sistemas especialistas, entre outros.

Segundo Covington [15], a PNL se subdivide em 5 aspectos da linguagem natural agrupados em 3 subgrupos:

- Som: fonologia;
- Estrutura: morfologia e sintaxe;
- Significado: semântica e pragmática.

A fonologia está relacionada ao reconhecimento dos sons que compõe as palavras. A morfologia é a análise da formação das palavras (e.g. um fato morfológico na língua portuguesa é que para formar a maior parte dos plurais dos substantivos, basta adicionar "s" ao final da palavra). Na sintaxe, analisa a sentença da estrutura. A semântica analisa o significado da palavra e a pragmática lida com a relação de contexto da linguagem.

Neste estudo utiliza 4 algoritmos que já foram utilizados em trabalhos relacionados com Processamento de Linguagem Natural (PLN), os algoritmos são:

- *Long short-term memory - LSTM*;
- *Support Vector Machine - SVM;*
- *Naives Bayes;*
- *Random Forest.*

O principal objetivo deste trabalho foi classificar sentimentos positivos, negativos e neutros em textos retirados de tweets relacionados com o mercado de CFT.

D. PLN – TRABALHOS RELACIONADOS

No artigo de Kansaon [16] de 2018, foi apresentado um estudo de PLN para classificação de sentimentos em mensagens do Twitter no idioma português brasileiro. Foi realizada a coleta de dados via API oficial do Twitter, com 12.814 *tweets* coletados, também foi realizado o tratamento dos dados e armazenados em um repositório no Azure, por último foram utilizados modelos para realizar a classificação de sentimentos (tristeza, chateação, felicidade, amor, raiva, inveja e ironia).

FIGURA 1 - detalhamento do fluxo de coleta dos dados. Extraído de Kasaon[16]

Para melhor acurácia dos modelos, além dos textos, foram utilizados as *hashtag*, *emojis* e *emoticons* dos *tweets* para melhor classificação e acurácia.

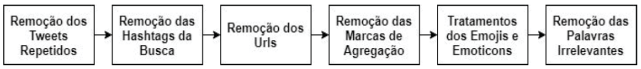


FIGURA 2 - Etapas do processamento de cada tweet. Extraído de Kasaon[16]

Após as etapas de processamento de *tweets*, utilizaram os seguintes algoritmos:

- *Naive Bayes;*
- *Naive Bayes Multinominal;*
- *Naive Bayes Multinominal Updateable;*
- *Sparge Generative Model;*
- *DMNB Text;*
- *Complement Naive Bayes;*
- *Bayesian Logistc Regression;*

•*IBK;*

•*Forest;*

•*Random Committee.*

*F*oram utilizados 33,3% dos dados para teste e 66,6% para treino e para evitar problemas de representatividade no conjunto de testes, realizaram a validação cruzada (de 10 partições). O método *Naive Bayes* Multinominal Updateable obteve melhor resultado dos algoritmos aplicados, chegando a 85% de acerto na classificação. O resultado foi considerado satisfatório de acordo com outras pesquisas realizadas na área.

TABELA I - Taxa de acerto dos melhores algoritmos para detectar sentimentos positivos x negativos. Extraído de Kansaon[16]

Algoritmo	Amor x Triste	Feliz x Triste	Feliz x Chateado	Taxa de Acerto Média
Naive Bayes Multinominal Updateable	85,54%	81,35%	79,60%	82,16 %
Naive Bayes Multinominal	85,41%	81,02%	79,60%	82,01 %
Complement Naive Bayes	85,64%	80,34%	79,54%	81,84 %

Além de avaliar a taxa de acerto, também utilizaram as métricas F-Measures e ROC² para avaliar a performance dos modelos. Nessa visão o *Complement Naïve Bayes* apresentou melhor performance frente aos outros modelos.

Em outro artigo, Junqueira e Fernandes[17], também abordaram o tema de análise de sentimento na língua Português brasileiro comparando diversos algoritmos (*Naïve Bayes, SVM,* Máxima Entropia, *Random Forest* e Árvore de Decisão) e a abordagem léxica (indicar sentimento positivo ou negativo) durante as Olímpiadas de 2016.

Para realizar a análise foram coletados diariamente os *tweets* com a hashtag #rio2016 por um período de 80 dias, sendo 40 dias antes e 40 após o início dos jogos Olímpicos. O total de *tweets* coletados foi de 988.512 mensagens, número já desconsiderando as duplicidades (*retweets*). Após a coleta foi realizada uma amostragem aleatória com 7.000 mensagens com classificação de sentimento manual posterior com os seguintes atributos: neutra, positiva, negativa ou spam.

Para tratamento, utilizaram conversão para minúsculo, remoção de URLs, remoção de letras duplicadas, remoção de acentos, *stopwords*, usuários e hashtags.

Ao final o algoritmo SVM apresentou maior assertividade na classificação, atingindo 89,5% de acurácia em uma proporção de 90% treino e 10% para teste.

E. LONG SHORT-TERM MEMORY (LSTM)

O algoritmo *Long Short-Term Memory* (LSTM) é um tipo de Rede Neural Recorrente (RNN - *Recurrent Neural Networks*), que pode ser utilizada em diversos cenários de Processamento de Linguagem Natural.

A motivação para criação do LSTM foi conseguir contornar o problema das redes neurais profundas de instabilidade e que tendem a desaparecer as camadas anteriores, conhecido como desaparecimento de gradientes *[18][19]*.

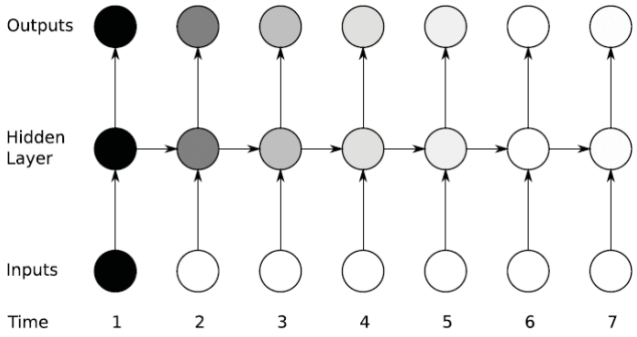


FIGURA 3 - The vanishing gradient problem for RNNs. Extraído de Sak [20]

O LSTM utiliza de um mecanismo específico em suas camadas ocultas, denominado *"célula de memória"*. Isto permite que ela consiga "*lembrar*" das informações que armazenou mesmo depois de várias interações. Para isso a LSTM utiliza-se de 3 portões para controlar o estado de cada célula sendo eles:

- *Forget Gate*: tem a função de definir qual informações serão descartadas ou mantidas. Esta decisão é realizada por uma camada sigmoide;
- *Input Gate*: Tem a função de decidir quais novas informações devem ser armazenadas no estado da célula. A decisão é realizada por uma sigmóide, em seguida uma camada *tanh* cria um vetor com os novos valores a serem adicionados;
- *Output Gate*: Tem o objetivo de identificar e extrair informações úteis do estado da célula atual para ser apresentadas como uma saída é feita pelo *output gate*.

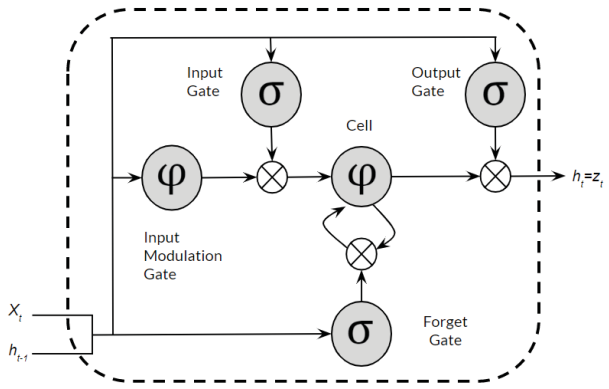


FIGURA 4 - Long Short-term Memory Cell. Fonte: Elaborado pelo autor.

Segundo Lindemann e Vietza[21], as células LSTM possuem *gates, input*, *forget* e *output*, que, no estado da célula, permitem inserir informações úteis, descartar informações inúteis e extrair informações úteis do estado da célula para gerar uma saída, respectivamente. Este fluxo de informação controlado dentro da célula permite que a rede memorize múltiplas dependências de tempo com características diferentes.

F. SUPPORT VECTOR MACHINE (SVM)

O Support Vector Machine (SVM), desenvolvido por Vapnik [22] em 1995, são um grupo de métodos de aprendizado de máquina que utiliza a arquitetura de uma rede neural recorrente (RNN), são utilizados tanto para classificação quanto para regressão. Utilizado para classificar, processar e prever séries temporais com intervalos de tempo de duração desconhecida entre outras aplicações.

Seu conceito foi fundamentado nos princípios da Minimização do Risco Estrutural (*Structural Risk Minimization* - SRM), construindo um hiperplano que maximiza a margem de separação entre diferentes classes de dados.

Segundo Vapnik, Stitson, Weston e Gammerman [34], o SVM realiza a minimização de riscos estruturais, cria um classificador com dimensão minimizada. Se a dimensão é pequena, as expectativas de erros são baixas o que resulta em uma boa generalização.

O SVM, originalmente desenvolvido para classificações binárias, busca a construção de um hiperplano como superfície para decisão. A construção dos hiperplanos pode ser feita de forma linear utilizando o *Kernel Linear*, conforme demonstrado na FIGURA 6, com retas conseguem realizar as classificações.

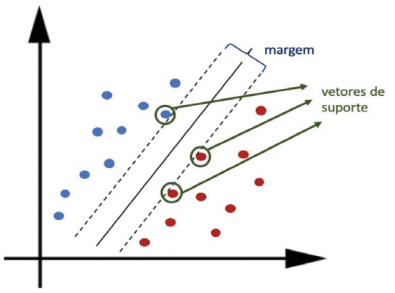


FIGURA 5 - Classificação Linear por SVM. Extraído de Escovedo e Koshiyama[23]

Para problemas não lineares, o algoritmo utiliza de funções de *kernels* para mapear o espaço de entrada podendo ser dos tipos Polinomial, Gaussiano (Radial) ou Sigmóide.

TABELA II - Funções de Kernel mais comuns

Tipo de Kernel	Função $\kappa(Xi, Xj)$	Parâmetros
Polinomial	$(\delta(Xi \,.\, Xj) + \kappa)^d$	δ, κ e d
Gaussiano	$\exp(-\sigma \lvert\lvert Xi - Xj \rvert\rvert^2)$	σ
Sigmoidal	$\tanh(\delta(Xi \,.\, Xj) + \kappa)$	δ e κ

Existem dois problemas comuns no SVM: *outliers* e exemplos rotulados erroneamente como ruídos. Para minimizar esses problemas o SVM permite trabalhar com o margens suaves, que permite que alguns dos pontos fiquem entre os hiperplanos de separação dos dados. Para otimizar esse classificador e controlar quanto os pontos poderão invadir à margem utiliza-se a seguinte função:

$$\begin{aligned} &\underset{\beta_0, \beta_1, \ldots, \beta_p, \epsilon 1, \ldots, \epsilon_n, M}{maximize} \quad M \\ &subject\ to \sum_{j=1}^{p} \beta_j^2 = 1, \\ &y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i), \\ &\epsilon_i \geq 0, \ \sum_{i=1}^{n} \epsilon_i \leq C, \end{aligned} \tag{1}$$

Onde:

- M - É a variável que representa a margem que procura se maximizar.
- ε - Variáveis de folga para permitir que alguns dados violem as restrições da maximização da margem.
- C - Constante chamada Custo, parâmetro não-negativo que representa o limite de erros que o classificador pode cometer.

Segundo Liu e Jiang [24], apesar de o SVM obter sucesso numa vasta área de aplicação, um problema notório aparece na sua aplicação prática. Esse problema inclui principalmente o parâmetro de penalidade C (*Cost*), o parâmetro da função de perda (*Loss*), e os parâmetros na função de *kernel* (por exemplo, parâmetros de largura da função de *kernel* RBF - *Radial Basis Function*), e seu efeito para mais ou para menos na performance do SVM.

A função de perda (*Loss*) é definida na equação (2).

$$L(f(x), y) = \max\{|f(x) - y| - 3, 0\} \tag{2}$$

Algumas das principais características das SVMs são [25]:

- Boa capacidade de generalização - os classificadores gerados por uma SVM em geral alcançam bons resultados em termo de generalização. Essa capacidade é medida por sua eficiência na classificação de dados que não pertençam ao conjunto utilizado em seu treinamento, portanto, é evitado o overfitting (memoriza os padrões de treinamento, gravando suas peculiaridades e ruídos, ao invés de extrair as características gerais que permitirão a generalização ou reconhecimento de padrões não vistos durante o treinamento);
- Robustez em grandes dimensões – as SVMs são robustas diante de objetos de grandes dimensões, como por exemplo, imagens. Comumente há a ocorrência de overfitting nos classificadores gerados por outros métodos inteligentes sobre esses tipos de dados;

- Teoria bem definida – as SVMs possuem uma base teórica bem estabelecida dentro da Matemática e Estatística.

G. NAIVE BAYES

O algoritmo Naive Bayes é um classificador probabilístico baseado no Teorema de Bayes, criado por Thomas Bayes (1701-1761).

Segundo Mukherjee e Sharma [32], o modelo Naive Bayes é um modelo bem simplificado, o classificador opera em uma forte suposição de independência, onde a probabilidade de um atributo não afeta a probabilidade de outro.

Pandey [33] informa que o *Naive Bayes* é um modelo muito utilizado para discriminar diferentes objetos baseados em certas características. Muito populares com spam de e-mails, filtragem colaborativa para mecanismos de recomendação e análise de sentimentos.

Sua equação é ilustrada na equação 3:

$$P(A/B) = \frac{P(B/A)\ x\ P(A)}{P(B)} \quad (3)$$

onde:

- P(A|B) é a probabilidade de que a classe A aconteça dado que o atributo B aconteceu — probabilidade a posteriori;
- P(B|A) é a probabilidade do atributo B ser observado dado que a classe A ocorreu — verossimilhança;
- P(A) é a probabilidade da classe A acontecer — probabilidade a priori;
- P(B) é a probabilidade de ocorrência do atributo B — probabilidade a priori.

Devido a seu equacionamento matemático simples, seu custo de processamento é baixo e por consequência é um algoritmo rápido, além disso requer uma quantidade pequena de dados para treino.

É um classificador muito versátil e dado a sua velocidade de processamento, pode ser utilizado para predições em tempo real, além disso, também é utilizado para predições multiclasses, classificação de textos, filtro de *spam*, análise de sentimentos e sistemas de recomendação.

H. DECISION TREE

Segundo Quinlan [35], a indução de árvores de decisão é uma maneira eficiente de aprendizado por exemplos. As árvores de decisão são uma das mais populares escolhas para o aprendizado e raciocínio de sistemas que trabalham com aprendizado supervisionado.

A árvore de decisão estabelece nós (*decision nodes*), como em um fluxograma (ilustrado na FIGURA 9), que se relacionam entre si por uma hierarquia. Existe o nó-raiz, que é um dos atributos da base de dados e nós-folhas (*leaf nodes*) que são as classes ou valores que será gerado como resposta.

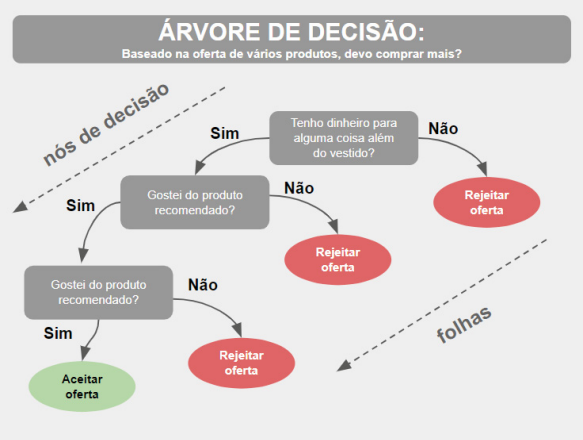


FIGURA 6 - *Decision Tree*. Fonte: Elaborado pelo autor.

Com essas estruturas de *Root node* e *leaf nodes*, a árvore de decisão permite comparar possíveis ações com base em seus custos, probabilidades e benefícios.

Segundo Quinlan [26], uma árvore de decisão utiliza uma estratégia de dividir para conquistar, onde um problema complexo é decomposto em subproblemas mais simples e recursivamente a mesma estratégia é aplicada a cada subproblema.

É um algoritmo recursivo, ele repete o mesmo padrão sempre que vai entrando em novos níveis de profundidade. Geralmente começa com um único nó, que se divide em possíveis resultados, cada um desses nós leva a nós adicionais que se ramificam num formato de árvore.

Seu objetivo é identificar quais nós deverão ser encaixados em cada posição, identificar o nó raiz e todos os nós de decisão abaixo com a melhor divisão que dê o máximo ganho de informação.

É comum utilizar abordagens de ganho de informação e a entropia para calcular os nós a direita e a esquerda, identificando quais conjunto de dados atendem melhor as condições que levam à um lado ou para outro.

A entropia é o grau de pureza do conjunto, definindo a medida de "falta de informação" do conjunto de dados. Sua equação é representada pela fórmula abaixo onde em um determinado conjunto de dados **S**, com instâncias pertencentes à classe *i,* com probabilidade *Pi.*

$$Entropia\ (S) = \sum p_i log_2 p_i \quad (4)$$

O ganho de informação é definido pela redução na entropia. Ganho(S,A) significa a redução esperada na entropia de S, ordenando pelo atributo A. O ganho é dado pela seguinte equação:

$$Ganho(S,A) = Entropia(S) - \sum_{v\ \epsilon\ Valores(A)} \frac{|S_v|}{|S|}.Entropia(S_v) \quad (5)$$

As árvores de decisão são algoritmos importantes para o desenvolvimento de modelos inteligentes, esses algoritmos tem ótima explicabilidade, leitura e compreensão, são

modelos robustos e muito eficientes, principalmente quando associados a um número considerável de árvores de decisão (*random forest*).

I. RANDOM FOREST

O *Random Forest* (RF) é um algoritmo de aprendizagem supervisionada utilizados para Classificação ou Regressão, sua primeira proposta foi realizada por Tin Kan Ho de Bells Labs em 1995.

Segundo Breiman [36], *Random Forest* é uma combinação de preditores de árvores de modo que cada árvore depende dos valores de um vetor aleatório amostrado independentemente e com a mesma distribuição para todas as árvores da floresta. O uso de uma seleção aleatória de recursos para dividir cada nó, produz taxas de erros que comparam favoravelmente ao algoritmo *Adaboost*.

A ideia principal por trás do *Random Forest* é tentar mitigar problemas como:

- Injetar aleatoriedade no treinamento das árvores
- Combinar a saída de várias árvores aleatórias em um classificador único

Segundo Yin, Criminisi, Winn e Essa [27], os classificadores de RF demonstraram produzir menos erros de testes do que as árvores de decisões convencionais.

Seu funcionamento consiste em criar uma floresta, ou seja, um conjunto de árvores de decisão aleatórias onde cada árvore é treinada utilizando um subconjunto de amostras selecionadas aleatoriamente e com repetição. A quantidade total de amostras utilizadas no subconjunto é o total de amostras do conjunto original de treinamento.

Random Forest utiliza-se de diversos classificadores de *Decision Tree* como base de aprendizado para classificação.

A Floresta é um conjunto de diversas árvores de decisão, conforme ilustrado na FIGURA 7.

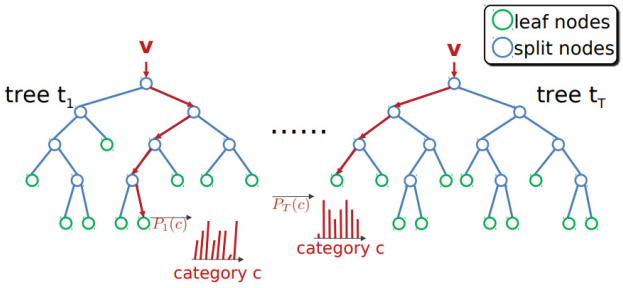


FIGURA 7 - Floresta é um conjunto de várias árvores de decisões. Extraído de Safaripour[28]

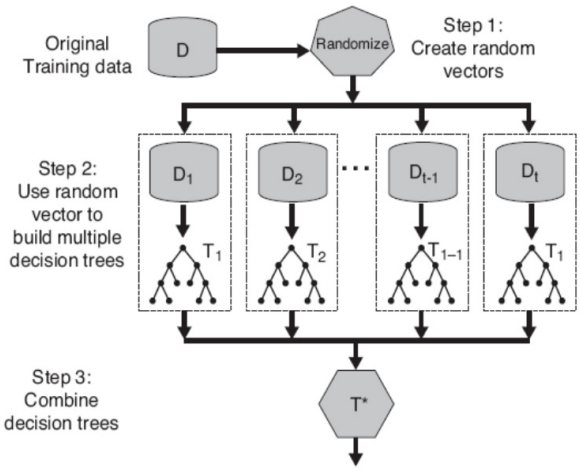


FIGURA 8 - *Random Forest - Flow Diagram*. Extraído de Safaripour [32]

O fluxo da *Random Forest* é separado em 3 passos, ilustrados na FIGURA 8 onde:

- 1 - Criação de vetores de subconjuntos de dados aleatórios (*bootstrap aggregation*);
- 2 - Utilizar os vetores randomicamente gerados para construir e multiplicar as árvores de decisão;
- 3 - A instância de teste deve percorrer cada árvore da floresta, combinar os resultados e a classe definida será a mais votada.

Segundo Safaripour [28], a *Random Forest* é fácil para construção e para predição, também, devido ao fator aleatoriedade, tem resistência com relação ao over-fitting, é possível utilizar sem o pré-processamento ou redimensionamento da base e é resistente a outliers e valores nulos.

J. REGRESSÃO LOGÍSTICA

Segundo Hosmer e Lemeshow [29], os métodos de regressão tem se tornado componente importante para qualquer problema de análise de dados que envolvem uma relação entre uma variável resposta (dependente) e uma ou mais variáveis explicativa (independentes).

O objetivo dos modelos de regressões são encontrar o melhor ajuste e mais interpretável para descrever a relação entre uma variável de resultado (dependente ou resposta) e um conjunto de variáveis independentes (preditores ou explicativas). O que distingue o modelo de regressão logística do modelo de regressão linear é que o a variável de saída da regressão logística é binária ou dicotômica, com isso a curva logística tem um comportamento probabilístico no formato da letra S enquanto a regressão linear é uma reta.

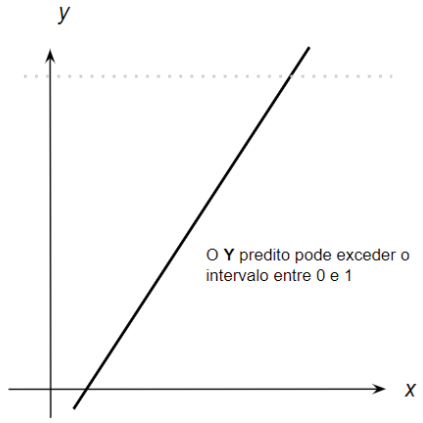


FIGURA 9 - Curva Regressão Linear. Adaptado de Hosmer e Lemeshow [29]

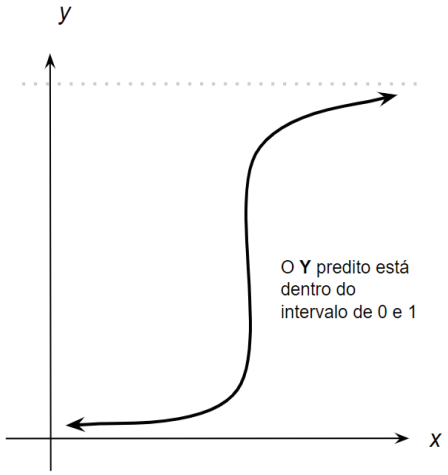


FIGURA 10 - Curva Regressão Logística. Adaptado de Hosmer e Lemeshow [29]

A regressão logística é uma técnica estatística com objetivo de classificação. A partir de um conjunto de observações irá gerar a probabilidade de um evento acontecer utilizando geralmente variáveis binárias (0 ou 1). No caso da variável Y assumir apenas dois possíveis estados e haver um conjunto de p variáveis independentes X_1, X_2, ..., X_p, a equação da regressão logística pode ser escrito da seguinte forma:

$$P(Y = 1) = \frac{1}{1+e^{-g(x)}} \quad (6)$$

Onde: $g(x) = B_0 + B_1X_1 + ... + B_pX_p$

Para utilizar o modelo de regressão logística para discriminação de dois grupos, a regra de classificação 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

Além da regressão logística binominal, também existem extensões do modelo logístico que permitem modelar a variação de variáveis ordinais são elas:

- Regressão Logística Ordinal: tem o objetivo de classificar categorias ordenadas (ex. Classificar um restaurante com nota de 1 a 10);
- Regressão Logística Multinomial: nessa vertente da RL, os classificadores podem incluir três ou mais categorias que não possuem ordem entre si. (Exemplo: Identificar qual tipo de automóvel preferido de um cliente).

Para este trabalho, é utilizada a Regressão Logística Multinomial, definido por:

$$g(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_{k'} \quad (7)$$

Onde:

β: são os parâmetros estimados
X: são as variáveis explicativas
k: quantidade de categorias para classificação

No modelo multinomial o número de equações é igual ao número de categorias das variáveis dependentes menos 1, ou seja, K-1, para este estudo a variável resposta possui três categorias para classificação de sentimentos sendo: “positivos”, “negativos” e “neutros”.

As vantagens da regressão logística é que é um algoritmo muito eficiente e não necessita de grandes quantidades de recursos computacionais, também tem fácil interpretação, além disso, tem uma fácil implementação e treinamento simples, o que torna esse algoritmo uma ótima base para medir o desempenho de outros algoritmos complexos.

K. MÉTRICAS DE PERFORMANCE

A avaliação da performance se faz necessária após construir um classificador, é necessário mensurar quão bom é o modelo para predição, essa fase é crucial a utilização de métricas apropriadas para cada tipo de problema. Os valores delas vão refletir a qualidade do modelo e caso sejam mal escolhidas, não será possível identificar se o modelo está performando de forma adequada.

A matriz de confusão é um dos métodos mais comuns e mais simples de visualizar a performance de um modelo de classificação.

A matriz indica quantos resultados ficaram em cada uma das quatro possíveis classificações conforme FIGURA 14:

- Verdadeiro positivo (TP - *true positive*): quando o método diz que a classe é positiva e, ao verificar a resposta, vê-se que a classe era realmente positiva;
- Verdadeiro negativo (TN - *true negative*): quando o método diz que a classe é negativa e, ao verificar a resposta, vê-se que a classe era realmente negativa;
- Falso positivo (FP - *false positive*): quando o método diz que a classe é positiva, mas ao verificar a resposta, vê-se que a classe era negativa;
- Falso negativo (FN - *false negative*): quando o método diz que a classe é negativa, mas ao verificar a resposta, vê-se que a classe era positiva.

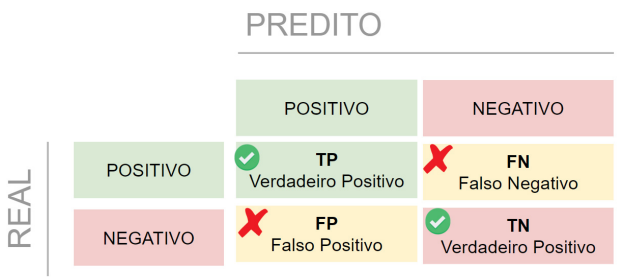


FIGURA 11 - Matriz de Confusão Fonte: Elaborado pelo autor.

O método mais simples e um dos mais importantes para avaliação de modelos de classificação é a acurácia, (*accuracy* ou ACC), ela avalia o percentual de acertos, também existe a sensibilidade, também chamada de revocação ou *recall,* esta avalia a capacidade do modelo detectar resultados classificados como positivo, a especificidade mensura a capacidade do método de detectar resultados negativos, a Precisão avalia a quantidade de verdadeiros positivos sobre a soma de todos os valores positivos e por último o *F-score* ou *F-measure* que é a média harmônica calculada com base na precisão e revocação.

Método	Fórmula
Acurácia	$\frac{VP+VN}{VP+VN+FP+FN}$
Revocação (Sensibilidade)	$\frac{VP}{VP+FN}$
Especificidade	$\frac{VN}{FP+VN}$
Precisão	$\frac{VP}{VP+FP}$
F1-score	$2 * \frac{Precisão * Revocação}{Precisão + Revocação}$

FIGURA 12 - Métricas de avaliação de classificação. Fonte: Elaborado pelo autor.

M. BAG OF WORDS e TF-IDF

Para se realizar o trabalho de NLP são necessárias as *features*, ou seja, variáveis com informações estruturadas. Para converter os textos, que são uma forma de informações não estruturadas, em informações estruturadas utiliza-se a *feature extraction,* ou seja, transformação do texto em informação numérica para que seja possível utilizar em um modelo.

Uma das técnicas mais comuns para se realizar a *feature extraction* é a *Bag-of-words* ou BoW, de acordo com Marhov e Larose [31], "a abordagem BoW tenta capturar a semântica do documento utilizando os termos contidos nos documentos como características descritivas e ignorando informações relacionadas às posições dos termos, ordenação ou estrutura, a única informação relevante para este fim é se o termo ocorre ou não nos documentos e a frequência de sua ocorrência".

FIGURA 13 - *Bag of Words. Fonte: Elaborado pelo autor.*

Outra técnica utilizada é o *TF-IDF (term frequency-inverse document frequency),* segundo Marhov e Larose [31], o TF-IDF é uma evolução do IDF que é proposto por Sparck Jones com a intuição heurística de que um termo de consulta que ocorre em muitos documentos não é um bom discriminador e deve receber menos peso.

Para o cálculo da importância das palavras é realizado utilizando duas métricas:

- ***Term Frequency*** (a frequência do termo), que mede a frequência com que um termo ocorre num documento;
- ***Inverse Document Frequency*** (inverso da frequência nos documentos), que mede o quão importante um termo é no contexto de todos os documentos.

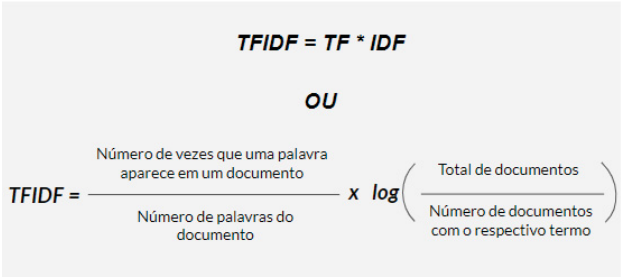


FIGURA 14 - Detalhamento de cálculo do TFIDF Fonte: Elaborado pelo autor.

III. MATERIAIS E MÉTODOS

Nesta Seção é descrita toda a metodologia utilizada neste trabalho para a classificação dos sentimentos dos *tweets* da base de dados. Na FIGURA 15, observa-se a metodologia adotada para classificação onde no tópico A detalha-se sobre o processo de coleta e estruturação dos dados brutos, no B aborda-se o processo de análise exploratória dos dados (EDA) e remoção de registros inconsistentes, no tópico C descreve-se toda parte de tratamento das informações e ajustes necessários para o pré-processamento dos dados, no D demonstra-se como os *Datasets* foram divididos entre parte para treinamento e validação, na seção E detalha-se a parte de extração das características e por último o F refere-se a como foi realizado o treinamento dos modelos.

FIGURA 15 - Metodologia Proposta. Fonte: Elaborado pelo autor.

A. COLETA / ESTRUTURAÇÃO DOS DADOS

A coleta dos dados foi realizada com informações do mundo real, com *tweets* capturados pela API oficial do Twitter.

Para captura dos *tweets* foram adicionados filtros para o mercado de CFT - (*cosmetics, fragrancies and toilletries*), colocando o nome de algumas empresas desse mercado e também alguns concorrentes diretos de presenteáveis. Os filtros utilizados foram:

- 'Boticario';
- 'Natura';
- 'Eudora';
- 'QDB';
- 'QuemDisseBerenisse';
- 'Vult';
- 'Sephora';
- 'BelezanaWeb';
- 'GrupoBoticario';
- 'TBB';
- 'TheBeautyBox';
- 'Toqueto';
- 'CacauShow';
- 'Loreal';
- 'Hinode';
- 'Loccitane';
- 'Avon';
- 'Herbalife';
- 'Unilever';
- 'OBoticario'.

O processo de captura foi realizado de 18 de setembro de 2021 a 28 de setembro de 2021, gerando um total de 21.083 *tweets* capturados.

TABELA III - Quantidade de registros por data dos *tweets*

Data	Quantidade
18/09	769
19/09	1.630
20/09	2.740
21/09	3.703
22/09	3.241
23/09	2.839
24/09	2.876
25/09	601
26/09	519
27/09	1.757
28/09	408
Total	**21.083**

Os *tweets* capturados foram armazenados no MongoDB Atlas, banco de dados em nuvem, não relacional que armazena os registros em documentos.

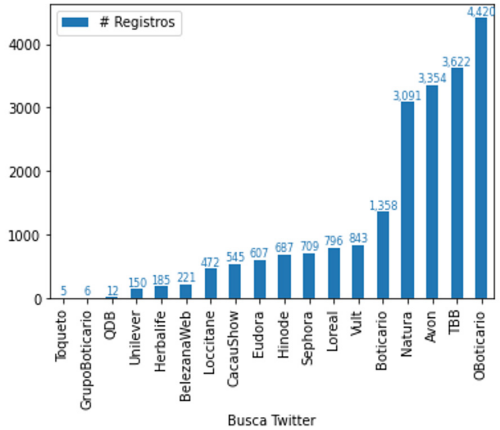


FIGURA 16 - Quantidade por palavra Pesquisada (21.083 registros)

B. EDA - Exploratory Data Analysis

Após a captação dos dados e armazenamento em um banco de dados (MongoDB), a próxima etapa é a Exploratory Data Analysis (EDA), ou seja, uma abordagem de exploração minuciosa dos dados que pretendemos analisar para extrair informações úteis com visões resumidas e sumarizadas.

Antes de iniciar a exploração dos dados, foi necessário realizar a exportação dos registros originados do Twitter e armazenados no MongoDB para o formato JSON. Após esse processo foi realizada a importação dos dados para o Python, utilizando a ferramenta Visual Studio para codificar e também as bibliotecas JSON para importar o arquivo e Pandas para trabalhar com os dados.

Os registros importados formam o *Dataset* Original com 21.083 registros.

A primeira análise a ser realizada foi identificar e retirar duplicidades da base. Foram identificados 3.681 *tweets* nessa condição. Além deste teste, também foi feita verificação de *retweets* (republicações de *tweets)* e 973 registros foram identificamos com essa característica. Ambos os casos foram retirados da base de dados totalizando 4.654 registros removidos do *dataset*.

A FIGURA 17, demonstra os registros duplicados no *Dataset*, pode-se observar que os textos da coluna ***text*** são iguais para todos registros e todas as linhas são do mesmo usuário, representado pela coluna **username**.

id	created_at	text	user_name
14414415…	Fri Sep 24	"A natura mihi	O Conselheiro
14398486…	Mon Sep 20	"A natura mihi	O Conselheiro
14393805…	Sun Sep 19	"A natura mihi	O Conselheiro
14408527…	Thu Sep 23	"A natura mihi	O Conselheiro
14410943…	Thu Sep 23	"A natura mihi	O Conselheiro
14401656…	Tue Sep 21	"A natura mihi	O Conselheiro
14404148…	Tue Sep 21	"A natura mihi	O Conselheiro

FIGURA 17 - Tweets duplicados. Fonte: Elaborado pelo autor.

Posteriormente, foram realizadas análises individuais dos textos dos *tweets* de casos aleatórios para cada uma das 18 marcas / palavras-chave pesquisadas para captura dos dados com objetivo de validar a semântica dos casos e identificar possíveis inconsistências dos dados ou textos que não remetiam ao mercado de cosméticos. Como resultado desta análise foi identificado que para a palavra pesquisada "TBB", que tinha o objetivo de capturar *tweets* com textos remetentes a marca "The Beauty Box" (empresa pertencente ao Grupo Boticário), a palavra também era utilizada como uma abreviação da palavra "também", com isso destoando do objetivo do projeto de avaliar o mercado de CFT. Dado que o contexto dos *tweets* destoavam do objetivo da pesquisa, todos os registros que foram capturados pela palavra pesquisada "TBB" foram retirados do *dataset*, totalizando 3.539 registros.

Na próxima análise, foi realizado um agrupamento pelo nome dos usuários que realizaram (coluna UserName do *Dataset*) a postagem do *tweet*, após esta sumarização, foi feita uma ordenação por ordem decrescente com objetivo de analisar os usuários com maior quantidade de postagens capturadas no *DataSet* e avaliar a semântica dos *tweets,* nessa análises objetivamos identificar vínculo destes usuários com a marca em questão e também avaliar se as postagens eram alguma forma de propaganda da marca ou se não remetia a contexto de Cosméticos (CFT) que a pesquisa estava buscando.

Foram identificados 11 usuários com vínculo com a empresa, as postagens referentes a eles totalizavam 1.794 registros conforme informado na *TABELA I*V.

TABELA IV - Usuários com maior quantidade de postagens durante o período observado

Id	User Name	Qtde
1	user 1	10
2	user 2	28
3	user 3	51
4	user 4	54
5	user 5	54
6	user 6	67
7	user 7	71
8	user 8	110
9	user 9	148
10	user 10	395
11	user 11	806
	TOTAL	**1.794**

Nesta mesma análise dos maiores usuários, foi identificado um usuário (FIGURA 17), com 112 *tweets* capturados durante o período observado pela palavra-chave de busca igual a "VULT". Os textos deveriam fazer alguma relação com a empresa do Grupo Boticário, mas remetiam ao "Deus VULT", um grito de guerra da igreja Católica associados ao período das Cruzadas.

TABELA V - Usuários com postagens fora do contexto de Cosméticos

Id	User Name	Qtde
1	User 12	112
	TOTAL	**112**

Todas os *tweets* dos usuários mencionados na *TABELA IV* (1.794 registros) e na *TABELA* V (112 registros) foram retirados do *DataSet* com objetivo da base de dado conter apenas *tweets* com sentimentos dos consumidores.

Ao final destas análises e retirada de diversos casos conforme descritos acima, o *Dataset* final ficou com **10.984** registros, conforme ilustrado na FIGURA 18. Um repositório com *tweets* que representam as dores ou elogios dos consumidores dentro do mercado de Cosméticos (CFT), sem propagandas ou contas de usuários das marcas, sem duplicidade e sem republicações.

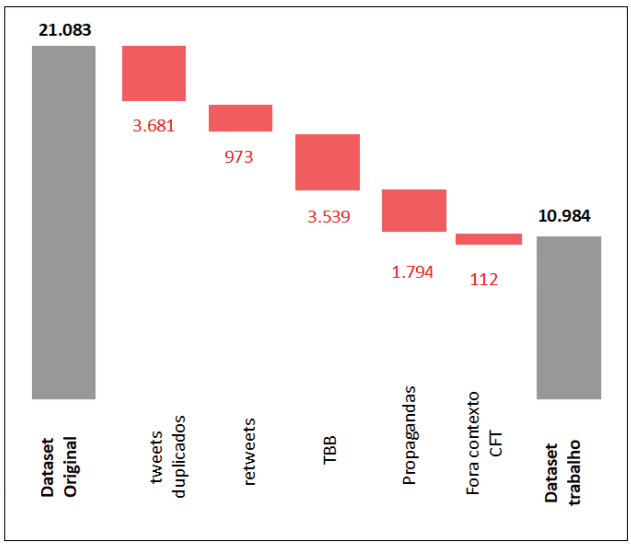


FIGURA 18 - Processo de limpeza de Registros. Fonte: Elaborado pelo autor.

Após a análise dos dados, a próxima etapa foi enriquecer o *dataset* com a informação de sentimento (Positivo, Negativo ou Neutro) de cada *tweet.*

Para esta etapa foi necessário realizar um análise e preenchimento de sentimento individual de cada registro para posterior treinamento supervisionado do modelo.

Dado ao alto volume de dados para análise manual, foi realizada uma amostragem aleatória de 5.000 registros do *dataset* utilizando a função *sample* com *seed* = 1 e exportada para um arquivo CSV (*comma separated values*).

Após gerada a amostra, foi utilizada a ferramenta Microsoft Excel para importação do arquivo CSV para que fosse possível realizar a análise individual e também para o preenchimento dos sentimentos de forma manual pelo próprio autor.

O resultado e a distribuição dos preenchimentos são demonstrados na TABELA VI.

TABELA VI - Enriquecimento da amostra com sentimento dos tweets

Sentimento	Qtde	Share
Positivo	1.446	28,92%
Neutro	3.028	60,56%
Negativo	526	10,52%

A distribuição da amostragem permaneceu em linha com a base original. Pode-se observar pela FIGURA 19 que informa o percentual que cada marca representa na base, comparando a base original vs a amostra gerada aleatoriamente.

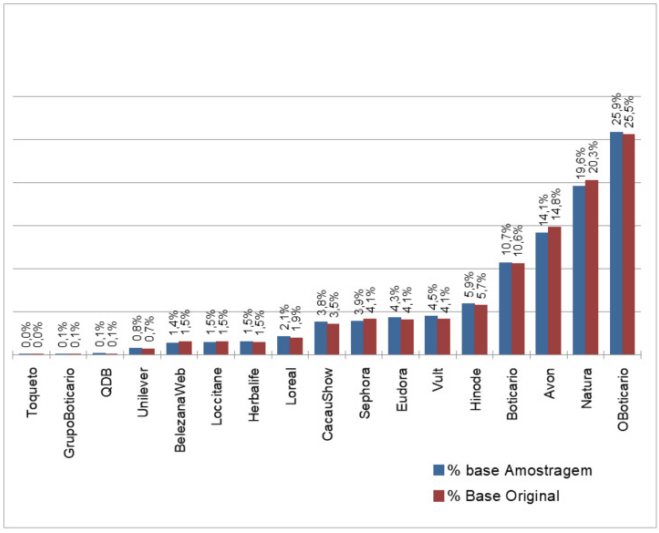


FIGURA 19 - Distribuição da base original vs Amostra de 5k. Fonte: Elaborado pelo autor.

C. PRÉ-PROCESSAMENTO DOS DADOS

Após a seleção da base a ser trabalhada, amostragem de 5.000 registros e enriquecimento manual com o sentimento de cada *tweet,* temos a parte de pré-processamento da informação, etapa que tem o objetivo de tratar os registros para que possa transformar de dados brutos em dados prontos para aplicação dos modelos.

Esta etapa foi realizada utilizando *Python* na ferramenta *Visual Studio Code* para toda parte de importação e tratativa dos dados.

Foi realizada a importação do arquivo Excel com os *tweets* enriquecidos com os sentimentos das postagens e foram realizadas diversas tratativas no *Dataset* para remover ruídos e informações em excesso que pudessem interferir no resultado do modelo.

A FIGURA 20, informa o fluxo adotado para realizar o tratamento dos dados.

FIGURA 20 - Fluxo de tratamento dos registros. Fonte: Elaborado pelo autor.

O primeiro processo realizado foi a criação de um dicionário para converter abreviações em palavras e deixar os textos dos *tweets* padronizados. Exemplo do dicionário criado:

TABELA VII - Exemplo do dicionário

de:	para:
mto	muito
oq	o que
adm	administrador
cmg	comigo
boti	boticário
agr	agora

Depois da criação do dicionário foi realizada a substituição das palavras no *dataset*.

Próximo passo foi a normalização do texto para minúsculo, substituindo todo conteúdo da *feature* com os textos das postagens para um conteúdo com todas as letras convertidas em minúsculas.

Depois da conversão em minúsculo, foi necessário remover diversos ruídos do *Dataset:*

- Retirada de menções de outros usuários nos textos dos *tweets*;
- Remoção de URL;
- Remoção de Hashtags;
- Retirada de caracteres especiais (! " # $ % & \ ' ( ) * + , - . / : ; < = > ? @ [\ ] ^ _ ` { | } ~ '.) ;
- Remoção de letras repetidas ;
- Remoção de espaços em branco em sequência ;
- Remoção da pontuação.

Próximo passo foi o de converter um *emoji* em um texto para que fosse possível utiliza-lo na análise de sentimento.

O Twitter é uma rede social onde *emojis* são muito utilizados, segundo a Olhar Digital [30], mais de 20% dos *tweets* contém pelo menos um dos mais de 3.500 emojis disponíveis na plataforma.

Para ter mais insumos para conseguir classificar sentimentos, a conversão dos *emojis* dos *tweets* capturados em representação textual é essencial.

Para a conversão foi utilizada a função *demojize* da biblioteca *emoji* e pode-se observar o exemplo na FIGURA 21.

```
':grinning:': '😀',
':smiley:': '😃',
':smile:': '😄',
':grin:': '😁',
':laughing:': '😆',
':satisfied:': '😆',
':sweat_smile:': '😅',
':rofl:': '🤣',
':joy:': '😂',
':slightly_smiling_face:': '🙂',
':upside_down_face:': '🙃',
```

FIGURA 21 - Tratativa de Emojis

Última tratativa do pré-processamento foi a remoção das de diversas palavras comuns que aparecem com frequência que geralmente apresentam pouco conteúdo lexical e sua

presença nos textos não seve para a diferenciar sentimentos são as *Stopwords*.

Para remoção foi utilizado o corpus da *Natural Language Toolkit (*NLTK) em português, que é um repositório com palavras que aparecem com alta frequência. Pode-se observar a lista de *stopwords* utilizada no estudo na TABELA VIII.

TABELA VIII - lista de stopwords NLTK

a, à, ao, aos, aquela, aquelas, aquele, aqueles, aqui, aquilo,
as, às, até, com, como, da, das, de, dela, delas, dele, deles,
depois, disse, do, dos, e, é, ela, elas, ele, eles, em, então,
entre, era, eram, éramos, és, essa, essas, esse, esses, esta,
está, estamos, estão, estas, estava, estavam, estávamos,
este, esteja, estejam, estejamos, estes, esteve, estive,
estivemos, estiver, estivera, estiveram, estivéramos,
estiverem, estivermos, estivessem, estivéssemos, estou,
eu, foi, fomos, for, fora, foram, fôramos, forem, formos,
fosse, fossem, fôssemos, fui, há, estivesse, haja, hajam,
hajamos, hão, havemos, havia, hei, houve, houvemos,
houver, houvera, houverá houveram, houvéramos,
houverão, houverei, houverem, houveremos, houveria
houveriam, houveríamos, houvermos, houvesse,
houvessem, houvéssemos, isso, isto, já, lhe, lhes, mais,
mas, me, mesmo, meu, meus, minha, minhas, muito, na
não, nas, nem, no, nos, nós, nossa, nossas, nosso, nossos,
num, numa, o, os, ou, outro, para, pela, pelas, pelo, pelos,
por, pra, qual, quando, que, quem, são, se, seja, sejam,
sejamos, sem, ser, será, serão, serei, seremos, seria,
seriam, seríamos, seu, seus, só, somos, sou, sua, suas,
também, te, tem, tém, têm, temos, tenha, tenham,
tenhamos, tenho, ter, terá, terão, terei, teremos, teria,
teriam, teríamos, teu, teus, teve, tinha, tinham, tínhamos,
tive, tivemos, tiver, tivera, tiveram, tivéramos, vos,
tiverem, tivermos, tivesse, tivessem, tivéssemos, tu, tua,
tuas, um, uma, você, vocês

Fonte: Elaborado pelo autor.

Após toda parte de tratamento dos registros, conforme ilustrado na FIGURA 20, houve uma mudança muito grande nas informações, na TABELA IX mostra um comparativo entre a quantidade de caracteres dos *tweets* do *dataset* original e a quantidade de caracteres após as tratativas realizadas no pré-processamento.

Conforme TABELA IX, na base original, existem em média 86 caracteres por *tweet* observando a visão Total geral, sendo que a marca Grupo Boticario é a que tem a maior quantidade de caracteres média por *tweet* com 133 caracteres em média e a marca Toqueto é a que apresenta a menor quantidade média com 58 caracteres. Com as tratativas realizadas conseguiu-se reduzir muito do ruído dos textos, houve uma redução de 15% ou 13 caracteres em média na visão consolidada do *dataset*. Maior queda observada pelo percentual foi da marca Vult com uma queda de 36,3%, saindo de uma média de 80 para 51 caracteres. Quando observamos pelo prisma de quantidade a marca Grupo Boticario foi a que teve a queda mais expressiva, saindo de 133 em média para 85, uma redução de 48 caracteres por *tweet.*

TABELA IX - Comparativo dataset original vs após tratativas pré-processamento

	Qtde média de caracteres *por tweets*		Redução %	Redução Qtde
	Origem	Final		
Avon	98	84	-14,3%	-14
BelezanaWeb	94	71	-24,1%	-23
Boticario	95	88	-7,1%	-7
CacauShow	95	74	-22,3%	-21
Eudora	86	83	-3,6%	-3
GrupoBoticario	133	85	-36,2%	-48
Herbalife	102	96	-6,1%	-6
Hinode	84	70	-16,6%	-14
Loccitane	104	85	-18,7%	-19
Loreal	89	74	-16,3%	-14
Natura	91	80	-12,1%	-11
OBoticario	69	55	-19,9%	-14
QDB	70	63	-11,0%	-8
Sephora	89	83	-6,6%	-6
Toqueto	58	45	-22,4%	-13
Unilever	114	89	-22,5%	-26
Vult	80	51	-36,3%	-29
Total geral	**86**	**73**	**-15,0%**	**-13**

Na TABELA X, ilustra como os textos dos *tweets* ficaram após as tratativas de pré-processamento, compara o texto original com o texto após as tratativas,

TABELA X - Comparativo entre Tweets originais vs após tratativas pré-processamento

Texto Original	Texto Tratado
@oBoticario tudo né amg??	tudo ne amiga
@oBoticario Eu usava o macherrie (nem sei como escreve) e na adolescência o capricho	eu usava o macherrie nem sei como escreve e na adolescencia o capricho
@oBoticario Boti mandando no quadradinho.😍	quadradinho :smiling_face_with_heart-eyes:
@oBoticario Chama mimo 👀🔥	chama mimo :eyes: :fire:
Aí você compra um chocolate e como ele vem 🤮🤮🤮 Uma vergonha @cacaushow @brasilcacaucvel https://t.co/IOLp4eyEIB	ai voce compra um chocolate e como ele vem :face_vomiting: :face_vomiting: :face_vomiting: uma vergonha

D. SEPARAÇÃO DOS DADOS

A separação dos dados em conjuntos de treinamento e testes é uma etapa muito importante do processo de

treinamento e validação do modelo. Neste artigo foram divididos em 80% dos registros foram utilizados para o treinamento do modelo e 20% para validação. Foi utilizada a função *train_test_split* da biblioteca *SKlearn* com o parâmetro *random state* igual a 100 para todos os modelos testados para ter uma consistência de resultados nos diferentes algoritmos testados.

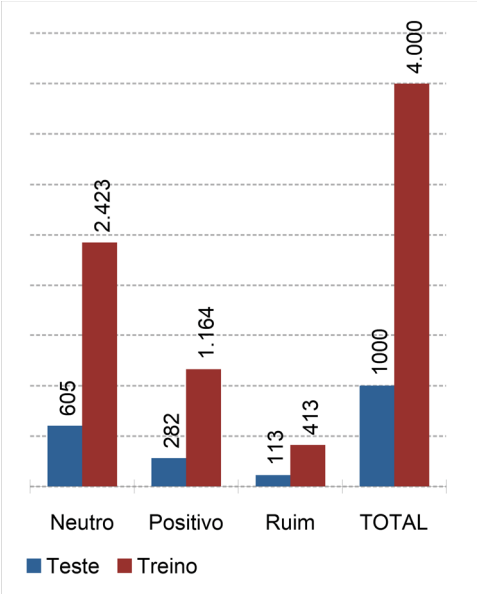


FIGURA 22 - Distribuição entre Treino e Teste - Quantidade. Fonte: Elaborado pelo autor.

A FIGURA 23 mostra que após a distribuição entre treino e teste a base manteve a mesma proporção entre sentimentos Neutro, Positivo e Ruim.

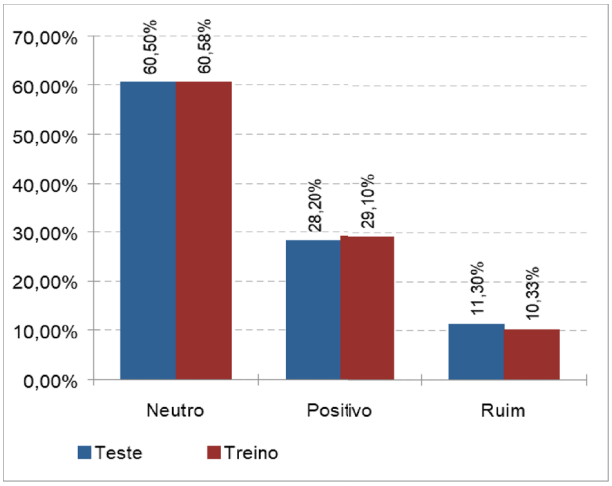


FIGURA 23 - Distribuição entre Treino e Teste - Percentual. Fonte: Elaborado pelo autor.

E. EXTRAÇÃO DAS CARACTERÍSTICAS

Conforme citado na seção 2.M, em trabalhos de NLP é necessário a utilização de *feature extraction,* para converter os textos em *features.*

Para este trabalho foram utilizadas as ferramentas *CountVectorizer e TfidfVectorizer*, fornecidas pela biblioteca *scikit-learn*.

Foram criadas matrizes esparsas, matrizes com muita ocorrência de elementos com valores zero, com 5k de linhas e 8.465, uma coluna para cada palavra contida no texto, já retirando todas as tratativas realizadas no Pré-processamento (seção 3.C).

Conforme ilustrada na TABELA XI, a *CountVectorizer* (CV) foi utilizada para criação da *Bag of Words*, onde para cada palavra contida no texto é gerada uma coluna, informando a quantidade de vezes que a palavra apareceu.

TABELA XI - Bag of Words utilizando *CountVectorizer*

Palavra	CV
balck_small_square	4
warning	2
avon	1
feed	1
link	1
...	...
empregada	0
empreendimento	0
empreender	0
empreendedor	0

Para criação do TF-IDF foi utilizada a função *TfidfVectorizer*, como mostra a TABELA XII onde cria uma medida estatística para informar a importância daquela palavra no documento.

TABELA XII - TF-IDF

Palavra	idf_weights
natura	2.835285
boticario	2.874906
avon	3.125476
perfume	3.930145
hinode	3.960565
...	...
francisca	8.824246
framboesa	8.824246
fraldas	8.824246
fragancias	8.824246

F. TREINAMENTO DO MODELO

Para o artigo foi foram feitas diversas combinações de técnicas e análises. Uma das derivações das análises foi testar abordagens de Unigrama ou Bigrama, ou seja, uma sequência de um ou dois itens dentro de uma frase. Para gerar as matrizes de características foram utilizadas três formas de *features extraction*, as duas principais foram a *CountVectorizer e TfidfVectorizer*, detalhadas na seção 2.M e

a terceira foi o *Embedding* da biblioteca *Keras*, utilizada como uma das camadas do algoritmo LSTM.

Para este estudo, foram utilizados 7 algoritmos que foram selecionados com base nos trabalhos relacionados informados na seção II.D, são eles:

- Árvore de Classificação;
- Random Forest;
- Nayve Bayes;
- Gradient Boosting;
- SVM;
- Regressão Logística;
- LSTM

Alguns testes com *GridSearch* da biblioteca *scikit-learn* também foram realizados com o objetivo de maximizar o resultado com a automação do processo de ajustes de parâmetros dos algoritmos.

As parametrizações dos testes realizados podem ser observadas na TABELA XIII, que informa todas as combinações e abordagens testadas nesse artigo.

TABELA XIII - Combinações de análises

Cód	Algoritmo	Critérios	Feature extrac-tion	Grid Searc h	Outros parâmetros
1	Arvore de Classifica-ção	Unigrama	CV	Não	random state = 100
2	Arvore de Classifica-ção	Unigrama	TFIDF	Não	random state = 100
3	Arvore de Classifica-ção	Bigrama	CV	Não	random state = 100
4	Arvore de Classifica-ção	Bigrama	TFIDF	Não	random state = 100
5	Random Forest	Unigrama	TFIDF	Não	bootstrap=True, criterion='gini', max_features='auto', random_state=100
6	Random Forest	Bigrama	TFIDF	Não	bootstrap=True, criterion='gini', max_features='auto', random_state=100
7	Random Forest	Unigrama	TFIDF	Sim	Parâmetros Grid_Search: número interações = 10, cv = 3, verbose=2, random_state=100, n_jobs = -1, n_estimators = [int(x) for x in np.linspace(start = 200, stop = 2000, num = 10)], max_features = ['auto', 'sqrt'], max_depth = [int(x) for x in np.linspace(10, 110, num = 11)], max_depth.append(None), min_samples_split = [2, 5, 10], min_samples_leaf = [1, 2, 4] e bootstrap = [True, False]
8	Random Forest	Unigrama	CV	Não	bootstrap=True, criterion='gini', max_features='auto', random_state=100
9	Nayve Bayes	Unigrama	CV	Não	
10	Nayve Bayes	Unigrama	TFIDF	Não	
11	Nayve Bayes	Bigrama	CV	Não	
12	Nayve Bayes	Bigrama	TFIDF	Não	
13	Gradient Boosting	Unigrama	TFIDF	Sim	Parâmetros Grid_Search: Número interações=10, scoring='accuracy', cv=3, verbose=1, random_state=100, n_estimators = [200, 800], max_features = ['auto', 'sqrt'], max_depth = [10, 40], max_depth.append(None), min_samples_split = [10, 30, 50], min_samples_leaf = [1, 2, 4], learning_rate = [.1, .5], subsample = [.5, 1.]
14	Gradient Boosting - GS	Unigrama	TFIDF	Sim	ShuffleSplit(n_splits=3, test_size=.33, random_state=100) Parâmetros Grid_Search: scoring='accuracy', cv=gcb_cv_sets, verbose=1,max_depth = [5, 10, 15], max_features = ['sqrt'], min_samples_leaf = [4], min_samples_split = [30, 50], n_estimators = [800], learning_rate = [.1, .3] e subsample = [0.5]
15	Gradient Boosting - Best Estimator	Unigrama	TFIDF	Sim	Grid_Search : best_estimator_
16	SVM	Unigrama	TFIDF	Sim	Parâmetros Grid_Search: { 'C': [0.1, 1, 10, 100, 1000], 'gamma': [1, 0.1, 0.01, 0.001, 0.0001], 'kernel': ['rbf'] }, refit=True, verbose=3
17	SVM	Bigrama	TFIDF	Sim	Parâmetros Grid_Search: { 'C': [0.1, 1, 10, 100, 1000], 'gamma': [1, 0.1, 0.01, 0.001, 0.0001], 'kernel': ['rbf'] }, refit=True, verbose=3
18	SVM	Unigrama	CV	Sim	Parâmetros Grid_Search: { 'C': [0.1, 1, 10, 100, 1000], 'gamma': [1, 0.1, 0.01, 0.001, 0.0001], 'kernel': ['rbf'] }, refit=True, verbose=3
19	Regressão Logística	Bigrama	TFIDF	Não	
20	Regressão Logística	Unigrama	TFIDF	Não	
21	Regressão Logística	Unigrama	CV	Não	
22	Regressão Logística	Unigrama	TFIDF	Sim	Parâmetros Grid_Search: { 'penalty': ['l1', 'l2'], 'C': [0.001,.009,0.01,.09,1,5,10,25] } e scoring='accuracy'
23	Regressão Logística	Unigrama	CV	Sim	Parâmetros Grid_Search: { 'penalty': ['l1', 'l2'], 'C': [0.001,.009,0.01,.09,1,1.5,2,3,4,5,10,25, 50] } e scoring='accuracy'
24	LSTM		Keras Embed-ding	Não	max_fatures = 25000, embed_dim = 128, lstm_out = 300, batch_size= 32 model = Sequential(), model.add(Embedding(max_fatures, embed_dim,input_length= X.shape[1])) model.add(SpatialDropout1D(0.4))mo-del.add(LSTM(lstm_out,dropout=0.2, recurrent_dropout=0.2)) model.add(Dense(3,activation='soft-max')) model.compile(loss = 'categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

IV. RESULTADO E DISCUSSÕES

Esta seção apresenta uma análise dos resultados obtidos com a aplicação dos modelos de classificação com objetivo de identificar sentimentos em textos em Português originados do Twitter de empresas do mercado de cosméticos e presenteáveis.

A principal métrica utilizada no artigo para avalição de resultado foi a acurácia, mas também foram utilizadas precisão, *recall*, *f1-score* e matriz de confusão da acurácia para melhor visualização dos resultados.

Foram feitas 24 comparações utilizando 7 algoritmos diferentes, 2 abordagens de n-gramas (unigrama e bigrama) e também 3 formas de executar *a features extraction,* em alguns casos foi utilizada *Grid Search* para otimizar os resultados.

Os resultados utilizando a técnica TF-IDF como ferramenta para gerar a extração das variáveis apresentou

melhor resultado do que o *Count Vectorizer* para maior parte dos algoritmos testados conforme demostrado na TABELA XIV, apenas para a Árvore de Decisão com abordagens Unigrama e Bigrama o *CountVectorizer* teve obteve melhor resultado e na utilização da Regressão Logística com Unigrama o resultado foi igual.

TABELA *XIV* - Comparação da acurácia das abordagens utilizando Count Vectorizer vs TF-IDF

Algoritmo - abordagem	CV	IDF	Melhor abordagem de *Feature extraction*
Árvore de Decisão - Bigrama	63,9%	62,0%	Count Vectorizer
Árvore de Decisão - Unigrama	68,0%	63,9%	Count Vectorizer
Nayve Bayes - Unigrama	66,2%	71,1%	TF-IDF
Nayve Bayes - Bigrama	54,0%	62,1%	TF-IDF
Random Forest - Unigrama	70,6%	71,9%	TF-IDF
Regressão Logística - Unigrama	73,4%	73,4%	-
SVM - Unigrama	72,9%	75,1%	TF-IDF

Na TABELA XV são apresentadas comparações das abordagens Unigrama vs Bigrama. Para este artigo de classificação de textos em português do Brasil a utilização de Unigrama demostrou melhor resultado em todas os algoritmos e técnicas testados.

TABELA XV - Comparação da acurácia das abordagens utilizando Unigrama vs Bigrama

Algoritmo - abordagem	Unigrama	Bigrama	Melhor abordagem de *Feature extraction*
Árvore de Decisão - CV	68,0%	63,9%	Unigrama
Árvore de Decisão - TF-IDF	63,9%	62,0%	Unigrama
Nayve Bayes - CV	66,2%	54,0%	Unigrama
Nayve Bayes - TF-IDF	71,1%	62,1%	Unigrama
Random Forest - TF-IDF	71,9%	63,8%	Unigrama
Regressão Logística - TF-IDF	73,4%	62,0%	Unigrama
SVM - TF-IDF	75,1%	63,3%	Unigrama
MÉDIA	69,94%	61,59%	

A abordagem Unigrama demostrou uma performance 13,5% (ou 8,3 pontos percentuais) melhor que a Bigrama em média conforme demostrado pela FIGURA 24.

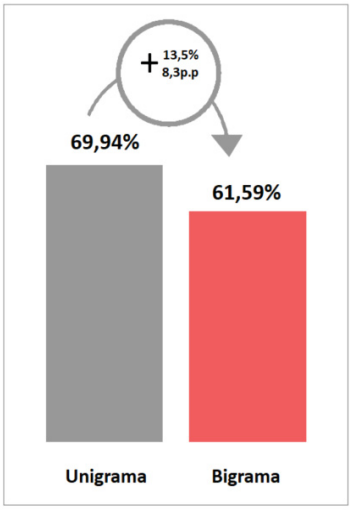


FIGURA 24 - Comparação média das acurácia das abordagens utilizando Unigrama vs Bigrama

TABELA XVI - Comparação da acurácia das abordagens utilizando Unigrama vs Bigrama

Cód	Algoritmo	Acurácia	F1 Score	Recall	Precisão
1	Arvore de Classificação	68,00%	66,77%	68,00%	66,33%
2	Arvore de Classificação	63,90%	63,17%	63,90%	62,57%
3	Arvore de Classificação	63,90%	56,49%	63,90%	62,39%
4	Arvore de Classificação	62,00%	54,73%	62,00%	57,87%
5	Random Forest	71,90%	67,01%	71,90%	72,06%
6	Random Forest	63,80%	55,32%	63,80%	57,31%
7	Random Forest	72,20%	67,23%	72,20%	76,08%
8	Random Forest	70,60%	67,04%	70,60%	68,35%
9	Nayve Bayes	66,20%	65,55%	66,20%	65,59%
10	Nayve Bayes	71,10%	65,17%	71,10%	65,13%
11	Nayve Bayes	54,00%	56,03%	54,00%	60,08%
12	Nayve Bayes	62,10%	49,54%	62,10%	61,59%
13	Gradient Boosting	71,50%	-	-	-
14	Gradient Boosting - GS	69,54%	-	-	-
15	Gradient Boosting - Best Estimator	70,80%	67,68%	70,80%	68,15%
16	SVM	75,10%	72,90%	75,10%	74,28%
17	SVM	63,30%	54,57%	63,30%	60,74%
18	SVM	72,90%	69,94%	72,90%	71,48%
19	Regressão Logística	62,00%	50,04%	62,00%	58,20%
20	Regressão Logística	73,40%	68,60%	73,40%	75,95%
21	Regressão Logística	73,40%	71,27%	73,40%	71,93%
22	Regressão Logística	74,20%	70,93%	74,20%	73,68%
23	Regressão Logística	72,50%	70,58%	72,50%	70,77%
24	LSTM	69,49%	-	-	-

Observa-se na TABELA XVI todos os 24 testes que foram realizados para construção deste artigo, também contém informações das abordagens utilizadas e os resultados das métricas de mensuração para cada um deles, informando acurácia, *F1-score*, *recall* e precisão.

Ao avaliar o resultado dos modelos, conclui-se o SVM (teste código 16), mostrou melhor resultado pelo acurácia, principal métrica avaliada pelo estudo, também obteve melhor resultado no *F1-score* e *recall*, ficando em segundo lugar

quando observa-se a Precisão, atrás do teste 20, Regressão Logística.

O teste 16, utilizou o algoritmo SVM, com a abordagem Unigrama e TF-IDF para gerar a extração das *features*, conseguiu chegar a uma acurácia de 75,10%.

TABELA XVII - Melhor acurácia para cada algoritmo

Algoritmo	Acurácia
SVM	**75,10%**
Regressão Logística	74,20%
Random Forest	72,20%
Nayve Bayes	71,10%
Gradient Boosting	71,55%
Árvore de Decisão	68,00%
LSTM	66,50%

Fonte: Elaborado pelo autor.

A TABELA XVII mostra um resumo da acurácia por tipo de algoritmo, agrupando as abordagens e utilizando apenas o melhor resultado dos testes que foram realizados.

TABELA XVIII - Métricas de performance SVM utilizando a abordagem Unigrama com o TF-IDF

Métrica	Performance
Acurácia	75,1%
F1-score	72,9%
Recall	75,1%
Precisão	74,3%

TABELA XVIII informa as métricas de performance da melhor abordagem do SVM.

		PREDITO			
		POSITIVO	NEUTRO	RUIM	Qtde
ATUAL	POSITIVO	179 (63,5%)	101 (35,8%)	2 (0,7%)	282
	NEUTRO	45 (7,4%)	548 (90,5%)	12 (1,9%)	605
	RUIM	9 (7,9%)	80 (70,8%)	24 (21,2%)	113
	Qtde	233	729	38	1.000

FIGURA 25 - Matriz de Confusão - SVM utilizando a abordagem Unigrama com o TF-IDF

Ao observar os resultados e também a Matriz de Confusão, FIGURA 25, o melhor resultado foi identificando sentimento Neutro com 90,5% de acerto, para sentimentos Positivos obteve 63,5% de acurácia e por fim, o ponto que precisa de mais atenção e velocidade nas respostas por parte das empresas, para as reclamações (sentimento Ruim), apenas 21,2% foram identificadas com sucesso. Provavelmente a baixa performance neste último sentimento, deve-se ao desbalanceamento da base, tema que será enderçado na seção V, considerações finais para trabalhos futuros.

Além disso, tanto para sentimentos Negativos, quanto para Positivos, a maior parte dos erros estão concentrados no sentimento Neutro.

Os *tweets* Positivos totalizam 282 registros na base de validação, observa-se na Matriz de Confusão que 179 foram classificados de forma correta e 103 classificações foram incorretas, destes 101 foram sinalizados como Neutro e apenas 2 como Negativos.

Para *tweets* que são reclamações ou sinalizam sentimento Ruim, boa parte dos erros estão concentrados no sentimento Neutro, de um total de 113 *tweets,* apenas 24 foram classificados como Ruim, 80 como Neutro e 9 como Positivos.

Tweets com classificações Neutras tinham participação da base, representavam cerca de 60% de toda base de treinamento e testes. Na base de validação eram 605 registros de um total de 1.000. Esse sentimento foi teve o melhor desembenho de classificação, com 90,5% (548 casos) de acurácia e apenas 57 registros com classificações incorretas, 45 classificando como Positivo e 12 como sentimento Ruim. Com esses baixos números de casos com sentimento Negativo, reforçam a necessidade de balanceamento da base para trabalhos futuros.

TABELA XIX - Exemplos da performance do SMV nos tweets com Classificações corretas

Tweet	Sentimento	Predito	Status Predição
eles tao se queimando, qual o problema pra cumprir o prazo dado por eles	Negativo	Negativo	OK
to louca para sentir o cheirinho dessa linha de bubbaloo :smiling_face_with_smiling_eyes: tem cheirinho de chiclete mesmo	Positivo	Positivo	OK
esses shampoos do boticario sao a melhor coisa do mundo, meu cabelo ta uma paina	Positivo	Positivo	OK
os de caneta da vult eu gosto muito	Positivo	Positivo	OK
o boticario e uma laranja	Neutro	Neutro	OK

TABELA XX - Exemplos da performance do SMV nos *tweets* com Classificações incorretas

Tweet	Sentimento	Predito	Status Predição
nao aguento a natura natura nem militar essa mulher sabe	Negativo	Neutro	Incorreto
esse e todo de bom.. mas pra ser sincero eu caio dentro do chocotone tambem :smirking_face:	Positivo	Neutro	Incorreto
perfeito acho maravilhoso quem consegue fazer degrade, mas tambem tem que ter esse bocao mara ne	Positivo	Negativo	Incorreto
eu nao posso ver o catalogo da boticario mds, fico louca	Positivo	Neutro	Incorreto

A TABELA XIX, demonstra alguns casos da performance da predição do algoritmo SVM em *tweets* da base de validação com classificação correta. Já a *TABELA* XX, mostra alguns casos em que a classificação do modelo não está de acordo com o desejado. Pode-se observar que boa parte dos erros estão concentrados no sentimento predito Neutro.

V. CONSIDERAÇÕES FINAIS

Este artigo teve como objetivo testar diversas abordagens para conseguir identificar sentimentos em *tweets* relacionados ao mercado de cosméticos e presenteáveis. Foram testados 7 algoritmos de classificação: Árvore de Classificação, *Random Forest*, *Nayve Bayes*, *Gradient Boosting*, SVM, Regressão Logística e LSTM também foram realizadas diferentes abordagens de *feature extractions* (*Count Vectorizer*, TF-IDF e *Embedding* da biblioteca Keras) e 2 abordagens de n-gramas (unigrama e bigrama), totalizando 24 diferentes testes realizados.

Dentre todos modelos e abordagens testados, o que melhor mostrou capacidade de classificação foi o SVM com Unigrama e TF-IDF, conseguiu obter uma acurácia de 75,1% o que valida a hipótese deste artigo de conseguir identificar sentimentos dos textos do Twitter, sendo que o sentimento Neutro mostrou melhor performance de 90,5% de acerto.

Dada a importância do atendimento mais próximo e respostas mais rápidas para os consumidores na atualidade, é muito importante que as empresas prestarem atenção nos sinais que são compartilhados nas redes sociais e consigam agir de forma rápida e assertiva. No cenário atual, com o alto volume de interações de clientes com empresas e com o histórico de crescimento exponencial dessas interações, é muito importante a utilização de ferramentas e técnicas de ciência de dados para apoiar a tomada de decisão e dar a velocidade necessária para identificar qual o sentimento de cada contato.

Para trabalhos futuros, será importante utilizar técnicas para balancear as amostragens (como Smote ou outras técnicas de reamostragem) e assim deixar uma volumetria de casos positivos, negativos e neutros mais parelhos, pelo desbalanceamento dos dados e concentração em sentimentos Neutros, acabou influenciando os erros de classificação. Também poderão ser utilizadas as informações de letras em maiúsculo, imagens, pontuação, letras repetidas pois podem conter informações relevantes e ajudar a melhorar a classificação. A expansão da base de dados utilizando número maior de *tweets* e um período maior de observação, para evitar alguma forma de sazonalidade.

REFERÊNCIAS

[1] ÍNDICE DE CONSUMO DE ECOMMERCE. Mccnet. 2021. https://www.mccenet.com.br/indice-de-vendas-online . Acesso dia 07/11/2022.

[2] MINISTÉRIO DAS COMUNICAÇÕES. Governo Federal, Ministério das Comunicações. Pesquisa mostra que 82,7% dos domicílios brasileiros têm acesso à internet. Gov.br. 2021. https://www.gov.br/mcom/pt-br/noticias/2021/abril/pesquisa-mostra-que-82-7-dos-domicilios-brasileiros-tem-acesso-a-internet . Acesso dia 07/11/2022.

[3] REPUTAÇÃO DIGITAL: QUAL A IMPORTÂNCIA?. Hawks, 2021. https://www.hawkz.com.br/reputacao-digital/.Acesso dia 07/11/2022.

[4] ONLINE REVIEWS STATS AND SURVEY. ReviewTrackers, 2021. https://www.reviewtrackers.com/reports/online-reviews-survey/.Acesso dia 20/05/2022.

[5] BLAKE MORGAN. The Top 20 Traits Of Customer Experience Leaders. Forbes, 2021. https://www.forbes.com/sites/blakemorgan/2018/03/30/the-top-20-traits-of-customer-experience-leaders/?sh=445538e56fb4 . Acesso dia 07/11/2022.

[6] CACHICH, Daniela. Mídia e Marketing - O poder está na mão do consumidor. Uol, 2020. https://economia.uol.com.br/noticias/redacao/2020/03/05/daniela-cachich-vp-de-mkt-da-pepsico-o-poder-esta-na-mao-do-consumidor.htm. Acesso dia 07/11/2022.

[7] PETRI Ivan, Como a visão 360º e em tempo real ajuda a impulsionar a experiência do cliente . Zendesk, 2022. https://www.zendesk.com.br/blog/os-br-como-a-visao-360o-e-em-tempo-real-ajuda-a-impulsionar-a-experiencia-do-cliente/.Acesso dia 07/11/2022.

[8] BLAKELY, Ford. *How to Improve Your Customer Service Response Times*. Customer Think 2018. https://customerthink.com/how-to-improve-your-customer-service-response-times/. Acesso dia 07/11/2022.

[9] KOTLER, Philip – Administração de Marketing – 10ª Edição, 7ª reimpressão – São Paulo: Prentice Hall, 2000

[10] KOTLER, P. Administração de marketing: Análise, planejamento, implementação e controle. 5. ed. São Paulo: Atlas, 1998.

[11] SAYCE, David. Twitter - The Number of tweets per day in 2020. Dsayse, 2022. https://www.dsayce.com/social-media/tweets-day/.Acesso dia 07/11/2022.

[12] ANÁLISE EXPLORATÓRIA DE DADOS. USP. http://www.each.usp.br/lauretto/SIN5008_2011/aula01/aula1.Acesso dia 07/11/2022.

[13] CHATFIELD, Chris. Exploratory data analysis, School of Mathematics, Bath University, Bath, Avon, BA 2 7A Y, United Kingdom. 1986.

[14] LIDDY, Elisabeth D. Natural Language Processing. In M. Drake (Ed.), Encyclopedia of Library and Information Science. 2003.

[15] COVINGTON, Michael, NUTE, Donald, VELLINO, André. Prolog Programming in Depth, Prentice-Hall, 1997.

[16] KANSAON, Daniel P. BRANDÃO, Michele A. PINTO, Saulo A. de Paula. Análise de Sentimentos em Tweets em Português Brasileiro, Pontifícia Universidade Católica de Minas Gerais (PUC-MG) - Belo Horizonte, MG, 2018.

[17] JUNQUEIRA, Kássio T. C., FERNANDES, Anita. Análise de Sentimento em Redes Sociais no Idioma Português com Base em Mensagens do Twitter. Universidade do Vale do Itajaí (UNIVALI), Itajaí, SC. 2016. ote

[18] HOCHREITER, Sepp. Untersuchungen zu dynamischen neuronalen netzen. Master's thesis, Institut fur Informatik, Universidade Técnica de Munique, Munique, Alemanha, 1991.

[19] BENGIO, Yoshua, SIMARD, Patrice, and FRASCONI, Paolo. Learning long-term dependencies with gradient descent is difficult. Neural Networks, 1994.

[20] SAK, Hasim, SENIOR, Andrew W., BEAUFAYS, Françoise. "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH), Estados Unidos, 2014.

[21] LINDEMANN, T. Müller, H. VIETZ, N. Jazdi, and M. Weyrich. A survey on long short-term memory networks for time series prediction. In Procedia CIRP, 2021.

[22] VAPNIK, V. N. The nature of statistical learning theory. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[23] ESCOVEDO, Tatiana & KOSHIYAMA Adriano S. "Introdução a Data Science — Algoritmos de Machine Learning e métodos de análise". São Paulo, Ed. Casa do Código, 2020.

[24] LIU, S.; JIANG, N. Svm parameters optimization algorithm and its application. In: Mechatronics and Automation, 2008. ICMA 2008. IEEE International Conference on. [S.l.: s.n.], 2008.

[25] SMOLA, A. J. e SCHOOLKPF, B. Learning with Kernels. MIT Press, 2002.

[26] QUINLAN, J. R. Induction of Decision Tress. In: Machine Learning, 1986.

[27] YIN Pei, CRIMINISI Antonio, WINN John, ESSA Irfan. Tree-based Classifiers for Bilayer Video Segmentation. In Proc. CVPR, 2007.

[28] SAFARIPOUR, Razieh. Machine Learning in Population Health: Frequent Emergency Department Utilization Pattern Identification. University of Saskatchewan, Saskatoon, Canada. 2021.

[29] HOSMER David W., LEMESHOW Stanley. Applied Logistic Regression, 2020.

[30] Olhar Digital - Uso de Emojis no Twitter - https://olhardigital.com.br/2021/07/14/internet-e-redes-sociais/uso-de-emojis-no-twitter-atinge-maior-patamar-da-historia/

[31] MARHOV, Zdravko, LAROSE, Daniel T., Data mining the web, wiley, 2007.

[32] MUKHERJEE, Saurabh, SHARMA, Neelam. Intrusion Detection using Naive Bayes Classifier with Feature Reduction, 2012.

[33] PANDEY, Pooja. Naive Bayes Classifier. 2020

[34] M. O. STITSON, J. A. E. WESTON, A. GAMMERMAN, V. VOVK, V. VAPNIK, London. Theory of Support Vector Machines. 1996

[35] QUINLAN J. R. Discovering Rules by Induction from Large Collection of Examples, in Expert Systems in Microelectronic Age, D. Michie (Ed.), Edinburgh University Press, Edinburgh, 1979.

[36] BREIMAN, Leo. Random Forests, 2001.