Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

VITOR SENNA SILVERIO

PROCESSAMENTOS DE TEXTOS PARA ANÁLISE DE SENTIMENTOS EM LÍNGUA PORTUGUESA

Assis/SP

Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

VITOR SENNA SILVERIO

PROCESSAMENTOS DE TEXTOS PARA ANÁLISE DE SENTIMENTOS EM LÍNGUA PORTUGUESA

Trabalho de Conclusão de Curso apresentado ao curso de Ciência da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e da Fundação Educacional do Município de Assis – FEMA, como requisito à obtenção do Certificado de Conclusão.

Orientando(a): Vitor Senna Silverio
Orientador(a): Dr. Almir Rogério Camolesi

Assis/SP

FICHA CATALOGRÁFICA

SILVERIO, V. S.
PROCESSAMENTOS DE TEXTOS PARA ANÁLISE DE SENTIMENTOS EM LÍNGUA PORTUGUESA/ Vitor Senna Silverio. Fundação Educacional do Município de Assis –FEMA – Assis, 2023.
Número de páginas. 49

Orientador: Dr. Almir Rogério Camolesi

1. Análise de Sentimentos. 2.Inteligência Artificial. 3. Python

CDD:
Biblioteca da FEMA

PROCESSAMENTO DE TEXTOS PARA ANÁLISE DE SENTIMENTOS EM LÍNGUA PORTUGUESA

VITOR SENNA SILVERIO

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: _____
Dr. Almir Rogério Camolesi

Examinador: _____
Me. Fábio Eder Cardoso

Assis/SP

DEDICATÓRIA

Dedico esse trabalho a todos à minha família e amigos que acompanharam, apoiaram e me auxiliaram em minhas conquistas.

AGRADECIMENTOS

Primeiramente gostaria de agradecer a minha família pelo apoio e direcionamento em momentos de decisão, esses últimos 4 anos foram essenciais para mudar a minha vida para melhor.

Agradeço aos meus amigos pelos momentos de felicidade proporcionados e apoio durante todos esses anos.

Ao meu orientador, Dr. Almir Rogério Camolesi por todo seu conhecimento, dedicação e empenho em sempre fazer o melhor, não apenas por mim, mas por todos os alunos.

RESUMO

Com a crescente utilização de mídias sociais, viu-se a necessidade de as empresas adaptarem-se para alcançar mais clientes e manter um bom relacionamento com os mesmos. Com o acesso à internet e às redes sociais aumentando gradativamente, dificultou-se lidar com tantos dados de comentários dos clientes em seus sites de compras e *chats*. Por isso, o presente trabalho propõe investigar o que é o processo de análise de sentimentos, bem como este se relaciona com a inteligência artificial. Ademais, a pesquisa visa averiguar a forma que as empresas já utilizam esse recurso para obterem *feedback* do público-alvo, a fim de se captar dados e perquisições para a realização de um algoritmo que seja capaz de executar a análise de sentimentos.
Palavras-chave: Análise de sentimentos, Inteligência Artificial, Python

ABSTRACT

With the growing use of social media, there was a need for companies to adapt in order to reach more customers and maintain good relationships with them. As internet and social media access have been gradually increasing, dealing with such a large amount of customer comment data on their shopping websites and chats has become challenging. Therefore, this current work proposes to investigate the process of sentiment analysis and how it relates to artificial intelligence. Furthermore, the research aims to examine how companies are already using this tool to gather feedback from their target audience, in order to collect data and insights for the development of an algorithm capable of performing sentiment analysis.

Keywords: Sentiment analysis, Artificial Intelligence, Python

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

IA	INTELIGÊNCIA ARTIFICIAL
PLN	PROCESSAMENTO DE LINGUAGEM NATURAL
AS	ANÁLISE DE SENTIMENTOS

SUMÁRIO

1. INTRODUÇÃO

Desde os anos 2000, a área da tecnologia tem apresentado grandes avanços se comparado a outras. Cerca de vinte anos atrás era necessário realizar estudos, cálculos e as mais profundas análises praticamente de forma manual, já que o acesso a *softwares* de controle de informações, como o Excel, era limitado.

Uma área que vem ganhando destaque foi a AS[1]. Liu (2012) descreve essa área como uma forma de extrair e analisar sentimentos, avaliações, opiniões e/ou emoções em textos perante o seu contexto. Para desenvolver uma ferramenta que seja capaz de AS em textos, é necessário levar em consideração o cenário que se encontra a análise e a proveniência dos dados que serão utilizados. Para Silva (2016), não se deve aplicar métodos de AS sem que antes se considere os prós e contras com relação à fonte de dados que será utilizada para o processamento, pois extrair um sentimento de textos gera diversas questões a serem levadas em consideração, como: tamanho do texto, variações de ortografia, e definir o contexto.

Segundo Martins *et al.* (2020), a análise de dados aborda a capacidade que uma máquina tem de receber um ou mais dados, estruturá-los de forma que seja possível sua análise e, por fim, gerar um resultado preciso que seja legível a um ser humano. Supondo um cenário onde é preciso realizar uma AS em um perfil de uma rede social, seria necessário extrair os comentários de publicações, organizar esses dados e aplicar um algoritmo capaz de identificar o sentimento presente em cada frase para, enfim, gerar o resultado.

O presente trabalho expõe como objetivo apresentar a IA[2] e usar seus conceitos para realizar AS em textos de língua portuguesa, suas principais limitações e de que forma essa ferramenta pode ser benéfica para as empresas. Ao fim, é desenvolvido um algoritmo que realiza a AS em textos, de forma a demonstrar os conceitos explanados durante o trabalho.

[1] Análise de sentimentos
[2] Inteligência artificial

1.1. METODOLOGIA

Com o objetivo de proporcionar uma compreensão mais clara e ampliar o potencial de investigação, apresenta-se inicialmente um levantamento bibliográfico sobre a IA, AS, PLN[3] e a linguagem *Python*. A partir das bases teóricas em estudo, foram identificadas as principais técnicas e ferramentas disponíveis de IA e aplicadas na área de AS. Em seguida é conduzido um estudo de caso de acordo com a base teórica levantada que permitiu desenvolver um algoritmo para validar a pesquisa e exemplificar os conceitos descritos.

1.2. OBJETIVOS

Explorando dentro do campo da IA, despertou-nos os seguintes objetivos para essa pesquisa:

a) Investigar as técnicas de IA na AS em textos não estruturados, apoiando-nos em uma bibliografia acurada sobre ambos os assuntos e a importância desse tema para o contexto histórico atual.
b) Demonstrar como as empresas se beneficiam ou já utilizam a AS para lidar com grandes volumes de dados textuais, como por exemplo, a base de dados de *feedback* de clientes.
c) Desenvolver um algoritmo que realize a AS, a partir dos conceitos e técnicas de IA apresentados.

1.3. JUSTIFICATIVAS

É notável a crescente busca por profissionais especializados em ciência de dados pelas empresas. Devido à alta demanda por essa mão de obra especializada, há um grande déficit na contratação desses profissionais. Segundo o site G1 (2021), a demanda por esses profissionais aumentou cerca de 500% no ano de 2021, o que demonstra como há mercado disponível para pessoas que se interessem por essa área. Dessa forma, esse trabalho busca contribuir com a comunidade que se interessa por IA e análise de dados, apresentando os conceitos teóricos e um algoritmo de forma prática.

[3] Processamento de linguagem natural

Ademais, o trabalho em questão pode ser utilizado como uma fonte de inspiração para futuros estudos que visem à análise de textos em diferentes contextos, oferecendo um modelo para o desenvolvimento de soluções na área de ciência de dados. Com isso, espera-se que esse estudo ajude a promover a importância da especialização de profissionais em ciência de dados e inspire novas pesquisas e iniciativas nessa área que vem conquistando cada vez mais espaço no mercado.

1.4. MOTIVAÇÃO

A motivação do presente projeto consiste em dar continuidade aos estudos realizados na pesquisa cientifica pelo presente aluno no ano de 2021. Considerando a crescente demanda do mercado por profissionais e ferramentas especializadas em inteligência artificial, centramo-nos no propósito de oferecer uma contribuição à comunidade acadêmica e aos estudantes interessados em seguir carreira na área de análise de dados e IA.

1.5. PERSPECTIVAS DE CONTRIBUIÇÃO

A área de ciência de dados apresenta uma constante atualização e crescimento, sendo utilizada principalmente para lidar com grandes volumes de informações e realizar análises personalizadas de acordo com as necessidades de cada contexto. Dentre os principais usos da ciência de dados, podemos destacar previsões de mercado, análise de faturamentos e a automatização de processos, o que permite a tomada de decisões mais eficientes. Assim, torna-se clara a importância da ciência de dados para as empresas de pequeno ou grande porte, que desejam obter insights valiosos, a partir de seus dados, e se manterem atualizadas e fortes concorrentes no mercado atual.

1.6. ESTRUTURA DO TRABALHO

O trabalho está estruturado em sete seções principais. A seção de Introdução (1) contextualiza o tema abordado, apresentando seus objetivos, justificativas, motivação, perspectivas de contribuição e metodologias utilizadas. Na seção de Análise de Sentimentos (2), é explicado o conceito de AS, os principais desafios enfrentados nesse tipo de análise, como o mercado utiliza essa ferramenta e quais são as principais

ferramentas disponíveis. Na seção de *Python* (3), é destacado por que essa linguagem é a mais indicada para o estudo, suas principais características e as bibliotecas que realizam análise de dados. Na seção de Estudo de Caso (4), são apresentados a proposta do trabalho e o processo de desenvolvimento. A seção de Conclusões (5) apresenta os resultados obtidos e as conclusões. A seção Trabalhos Futuros (6) contém algumas observações feitas durante o desenvolvimento do algoritmo e algumas sugestões de trabalhos futuros que podem ser realizados a partir deste. Por fim, a seção de Referências lista todo o referencial teórico utilizado no trabalho.

2. ANÁLISE DE SENTIMENTOS

Nesse tópico, são demonstrados o que é a AS, os principais problemas que podem ser enfrentados em seu desenvolvimento, como o mercado já utiliza esse tipo de análise e as principais ferramentas disponíveis.

2.1. CONCEITO DE AS

AS é uma vertente da IA e PLN que visa identificar e classificar se o sentimento presente em um texto é positivo, negativo ou neutro. Conforme afirmado por Netto e Maciel (2021), a análise de sentimentos é uma técnica utilizada para analisar textos não estruturados, como comentários em redes sociais, *feedbacks* de clientes, entre outros, a fim de realizar estudos sobre os resultados obtidos.

2.2. PRINCIPAIS PROBLEMAS PARA ANÁLISAR TEXTOS EM LÍNGUA PORTUGUESA

Realizar uma AS na língua portuguesa de forma precisa é uma tarefa complexa que exige a definição específica do contexto em que será aplicada. Seja em um simples comentário em uma rede social ou em um texto mais elaborado, o algoritmo utilizado deve ser capaz de lidar com diversas situações e particularidades do idioma, como: variações linguísticas, ambiguidades, ironias, dados falsos, estrutura textual, figuras de linguagem e, principalmente, desvios gramaticais.

Conforme destacado por Silva (2016), a AS, em um ambiente onde os usuários possuem liberdade para expressar suas ideias de maneira livre e pouco estruturada, é um desafio complexo a ser trilhado. Nesse tipo de contexto, a busca por resultados precisos exige a utilização de técnicas de PLN capazes de lidar com determinadas situações em seus respectivos cenários. Por isso, é importante a adoção de metodologias adequadas para garantir a confiabilidade dos resultados e a entrega de informações precisas para aqueles que buscam análises precisas em seus resultados.

2.3. DE QUE FORMA O MERCADO JÁ UTILIZA A ANÁLISE DE SENTIMENTOS

De acordo com Liu (2012), a AS tornou-se um campo ativamente estudado a partir dos anos 2000. Foi justamente nesse período que sistemas comerciais começaram a se tornar mais comuns nos estabelecimentos e também se popularizaram nas redes sociais. Isso foi um fator chamou a atenção das empresas, já que elas teriam que se adaptar ao novo cenário e estudar a possibilidade de utilizá-lo a seu favor.

Dessa maneira, as instituições começaram a perceber o potencial dessas plataformas como ferramenta de *marketing* e proximidade com seus clientes. Com a popularização de redes sociais como o *Facebook*, as empresas passaram a utilizá-las para divulgar seus produtos e serviços, criar campanhas de *marketing*, oferecer suporte ao cliente e até mesmo realizar pesquisas de mercado.

Com o tempo, novas redes sociais surgiram – como o *Linkedin*, *Instagram* e *Twitter* – e foi necessária uma adaptação para que cada uma delas pudesse continuar alcançando seu público-alvo. Por conseguinte, as redes sociais atualmente são consideradas uma parte fundamental da estratégia de marketing de muitas empresas em todo o mundo.

Segundo um artigo publicado na *Forbes* por Karra (2023), as empresas têm utilizado as mídias sociais para atingir e interagir com seus clientes em razão da crescente demanda dos consumidores pelo varejo online. Esse uso de mídias sociais permite que as empresas tenham um contato direto e pessoal com seus clientes, possibilitando a obtenção de feedbacks valiosos e a criação de um relacionamento próximo e duradouro.

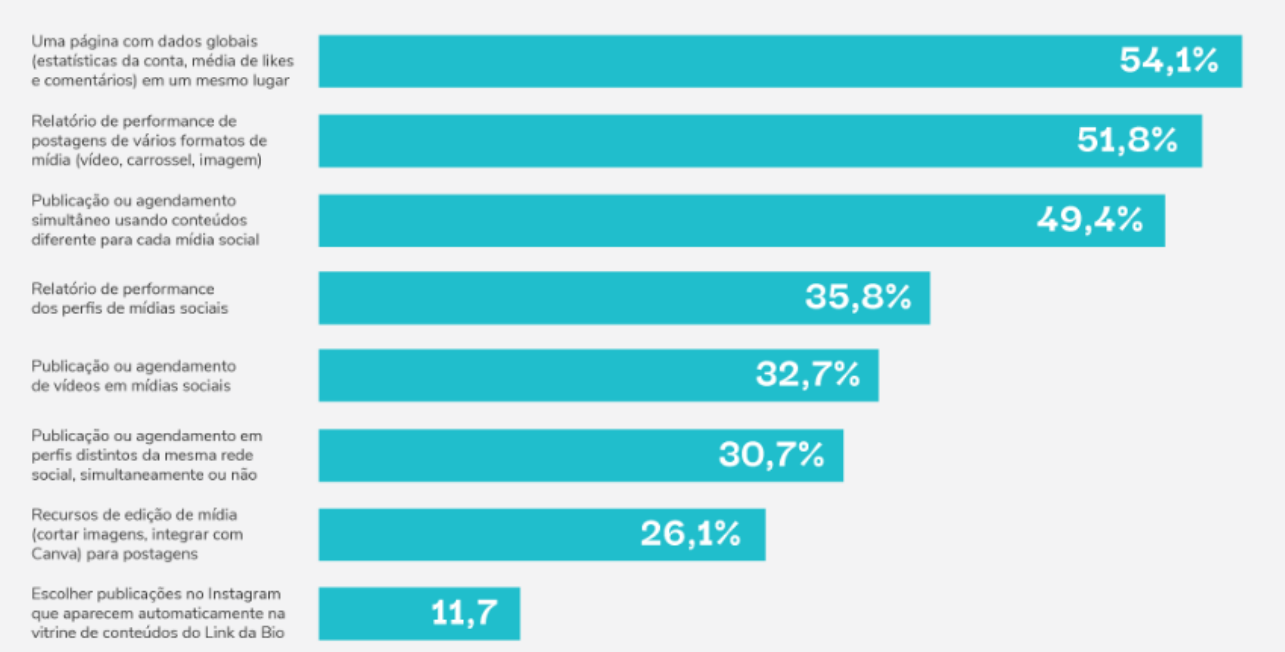

Figura 1: gráfico que demonstra como as empresas precisam da análise de dados (In: RODRIGUES, 2022)

Conforme demonstrado por Rodrigues (2022) na figura 1, em uma pesquisa na *RD Station* com cerca de 300 profissionais de *marketing*, é evidente a necessidade de uma ferramenta que seja capaz de centralizar os dados da empresa em relação as suas redes sociais, com estatísticas de comentários, likes, performance das postagens e/ou do perfil em geral.

2.4. PRINCIPAIS FERRAMENTAS QUE UTILIZAM ANÁLISE DE SENTIMENTOS

Atualmente, várias empresas utilizam ferramentas próprias para AS em seus devidos contextos. Dentre as principais, temos:

IBM Watson: Ferrucci (2012) explica em seu trabalho "*This is Watson*" que essa é uma ferramenta de que utiliza técnicas de análise sintática e semântica, onde várias tecnologias de PLN foram integradas para que fosse possível lidar com as demais necessidades na análise de dados.

Google Cloud Natural Language API: de acordo com Dale (2018), a *Google Cloud Natural Language API* se trata de um conjunto de ferramentas de análise de sentimentos, análise sintática e processamento de linguagem natural em nuvem da *Google* com poder de classificação de dados utilizando aprendizagem de máquina.

Microsoft Azure Text Analytics: Dale (2018) define o *Microsoft Azure Text Analytcs* como uma ferramenta de análise de linguagem natural na nuvem que permite aos desenvolvedores realizem os diversos processamentos de texto, como a “tokenização”, detectar o idioma, reconhecer palavras-chave e até mesmo gerar respostas automáticas a partir de textos não estruturados.

Chat GPT: Deng e Lin (2023) conduziram um estudo sobre os benefícios e desafios envolvendo o *Chat GPT*, onde eles explicam que ele se trada de um modelo de linguagem natural que gera respostas em tempo real de acordo com a interação do usuário. Por utilizar técnicas de aprendizagem de máquina, ele se possui a capacidade de se aprimorar sozinho, reduzindo a necessidade de realizar atualizações em seu modelo manualmente.

Logo, observa-se a importância da utilização de tecnologias como a inteligência artificial e análise de dados. As ferramentas disponíveis no mercado possibilitam uma maior eficiência na gestão de dados e processos, bem como oferecem uma vantagem competitiva significativa para as empresas que as utilizam.

3. CIÊNCIA DE DADOS E SUAS TECNOLOGIAS

Nesse tópico, é apresentada a ciência de dados com seu conceito e suas definições. Também é abordado o PLN, enfatizando como este se relaciona com a ciência de dados. Outrossim, é disposta ainda uma seção sobre a linguagem *Python*, sua história de origem, principais conceitos, características e as principais bibliotecas para AS e manipulação de dados.

3.1. CIÊNCIA DE DADOS

A ciência de dados é uma subárea da IA, que visa lidar com massivas quantidades de dados (*Big Data*). Rautenberg e Carmos (2019) afirmam que a ciência de dados na prática pode ser definida por uma construção de métodos que são responsáveis por transformar dados em informações, permitindo a tomada de decisões.

Figura 2: composição de conhecimentos para análise de dados (In: NETTO; MACIEL, 2021)

Abrangendo matemática, estatística, aprendizado de máquina, inteligência artificial e programação, pode-se criar um certo receio de ingressar na área, porém, atualmente, há diversas formas de trabalhar com análise de dados sem necessariamente ser um especialista em tudo que ela abrange. Analisando diagrama adaptado da figura 2 (Netto, Maciel, 2021), é possível observar que não é somente de matemática e estatística que se forma um cientista de dados, uma vez que compreender o problema em questão e ter familiaridade com programação facilitam na solução do problema.

Netto e Maciel (2021), portanto, reforçam que, para se compreender a área de ciência de dados e aprendizagem de máquina, a fim de desfrutar de maneira mais abrangente o que essas áreas tem a oferecer, é necessário sim possuir uma certa familiaridade com matemática e estatística, não de forma que isso seja um limitante, mas sim algo que deve ser tratado com mais atenção.

3.2. PROCESSAMENTO DE LINGUAGEM NATURAL

Perna, Delgado e Finatto (2010) afirmam que PLN trata-se de uma área da ciência da computação que visa investigar métodos e melhorias para que uma máquina seja capaz de analisar e interpretar com eficiência e precisão textos de origem humana e/ou linguagem natural. Assim, o PLN não se restringe apenas à análise de textos, mas também de vídeos, áudios e/ou gestos, de forma que dê a capacidade de interpretação de tomada de decisão à máquina.

Por mais que o PLN tenha tido notoriedade recentemente, foi por volta de 1950 que foi dado o primeiro passo por Alan Turing. Martins *et al* (2020) referenciam que o seu mais famoso trabalho, conhecido atualmente como “teste de Turing”, avalia a capacidade que uma máquina tem de analisar e demonstrar um comportamento semelhante à inteligência humana. Para que isso aconteça, o PLN passa por algumas etapas.

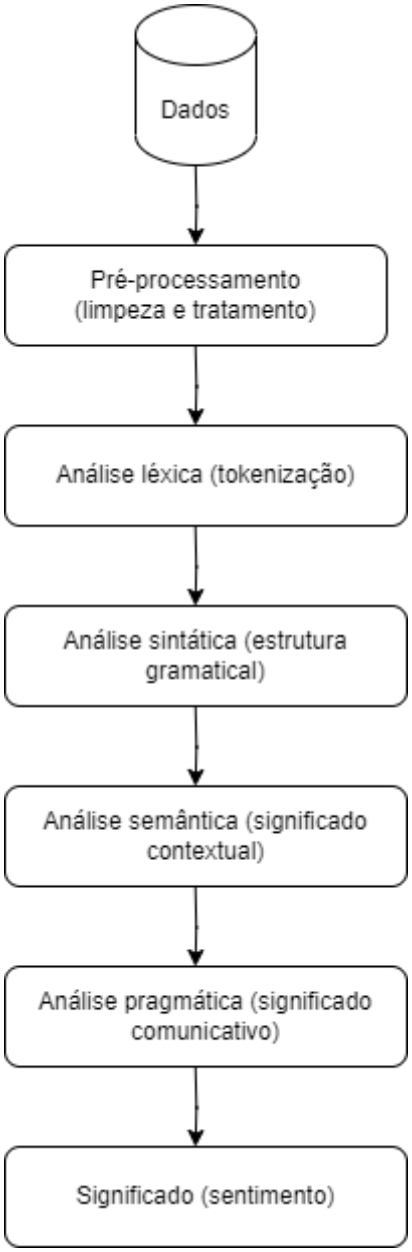

Figura 3: diagrama adaptado sobre os processos da realização da PLN (In: DALE, 2010)

O processo da realização do PLN, na figura 3 do diagrama adaptado (Dale, 2010), dá-se por ter uma base de dados para análise. A partir dos dados, são realizados pré-processamentos, como limpeza de símbolos, caracteres e palavras que não importam para a análise. Em seguida, ocorre o procedimento de análise léxica, que separa os textos em unidades menores, ou seja, em palavras. Após isso, realiza-se a análise semântica, que fornece o mapeamento dos textos ao seu significado dentro de um contexto, e, por fim, a análise pragmática, que explicita o resultado do que o texto deseja comunicar e expressar.

Logo, observar-se que o PLN é utilizado no dia a dia da sociedade, visto que as comunicações interpessoais e a necessidade de compreender um problema ou situação enquadram-se nessa área.

Conforme Oliveira (2017), a forma como pessoas buscam o entendimento entre si, tomando como objetivo melhorar seus conhecimentos a partir de *feedbacks* e novas experiências, acaba por fortalecer a rede de aprendizagem e melhorar a capacidade cognitiva. Dessa forma, é cristalino como essa área se encaixa perfeitamente no cenário atual, onde pessoas e empresas conectam-se por redes sociais e buscam por melhores resultados e maiores porcentagens de aprovação, alcançando assim, maior visibilidade.

Como as necessidades de automação de processos vêm aumentando exponencialmente nos últimos anos, há uma cobrança forte para o avanço da PLN. Basta observarmos como o lançamento da ferramenta *Chat GPT* compele o mercado a avançar na área de IA, fazendo com que grandes empresas, como a *Google*, movimentem-se e lancem algo relacionado para competir no mercado. De acordo com o que foi noticiado pela *Forbes* por Pacete (2023), a grande repercussão da ferramenta da *OpenAI* fez com que as grandes empresas de tecnologia se sentissem ameaçadas e forçassem investimentos cada vez maiores em IA para não ficarem atrasadas em relação ao mercado.

3.3. PYTHON

Lima (2021) conta em seu artigo que *Python* surgiu durante a década de 80. Seu criador, Guido Van Rossum trabalhava no desenvolvimento de uma linguagem chamada *ABC*, que tinha como foco diminuir a complexidade de códigos escritos em outras linguagens da época, como *Pascal* e *Basic*. Embora ela atendesse tal requisito, infelizmente possuía vários problemas de comunicação com o sistema operacional. Dessa forma, durante uma busca por uma linguagem alternativa ao *ABC*, *Rossum* teve a ideia de criar uma linguagem baseada em scripts que fosse capaz de solucionar os problemas com a *ABC*, resultando, assim, na sua primeira versão, lançada em 1991, com o nome de *Modula-3*, a qual futuramente viria a ser a tão conhecida linguagem *Python*.

Desde seu início, *Python* surgiu como uma forma de simplificar problemas e algoritmos que acabavam se tornando complexos em outras linguagens, o que levou a sua popularidade para os cientistas de dados e pesquisadores anos depois.

Dado esse esclarecimento, observa-se que atualmente *Python* é a melhor escolha para análise de dados por possuir diversas bibliotecas que simplificam cálculos matemáticos e raciocínios estatísticos para extração e manipulação de dados, deixando o algoritmo muito mais legível e de fácil compreensão.

Conforme descrito na documentação oficial, *Python* (2023) é uma linguagem de programação interpretada de alto nível, dinamicamente tipada, *case sensitive,* orientada a objetos e interpretada com suporte a módulos, facilitando a implementação de novas ferramentas pela comunidade

Para programar com *Python* é possível utilizar tanto um ambiente local, instalando a linguagem e suas bibliotecas no computador, ou usá-lo na nuvem com uma ferramenta do *Google* gratuita, o *Colab*, onde já possui a maioria das bibliotecas disponíveis para a linguagem. Nele é possível compilar códigos em *Python* de forma totalmente online e gratuita.

3.3.1. BIBLIOTECAS DO PYTHON PARA PROCESSAMENTO DE DADOS E AS

Nessa seção, são listadas e descritas algumas das bibliotecas disponíveis em *Python* para AS. Contudo, é importante ressaltar que nenhuma biblioteca realiza a análise de sentimentos de forma perfeita, já que esta depende do contexto sob investigação e de como o desenvolvedor pode adaptar essas ferramentas.

NLTK: Perkins (2010) em seu livro “*Python Text Processing with NLTK 2.0 Cookbook*”, utiliza a *NLTK* e explica que ela se trata de uma ferramenta de processamento de textos com *Python* essa é uma das plataformas mais conhecidas para análise de textos, que contém a *WordNet*, que é um dicionário com palavras mapeadas preparadas para a realização de PLN.

TextBlob: Abiola *et. al* (2023) define *TextBlob* como um pacote para desenvolvimento de algoritmos com PLN, disponível para *Python* 2 e 3. Entre suas principais funcionalidades estão: marcação de texto, extração de textos, AS, classificação e identificação de idioma para tradução.

Vader Sentiment: Deo *et. al* (2020) descreve *Vader Sentiment* como um modelo capaz de identificar a polaridade de intensidade de um sentimento presente em um texto. Utiliza

análise preditiva aos dados que estão sendo utilizado, não sendo obrigatório utilizar uma base de dados de treinamento.

Scikit-Learn: em um estudo conduzido por Hao e Ho (2019) sobre modelos para treinamento de IA, destacam o *Scikit-Learn* como uma ferramenta que abrange diversos métodos de treinamento que pode ser escolhido de acordo com a necessidade de cada problema. Baseado em bibliotecas binárias que foram compiladas em inicialmente em *C*, *C*++ e *Fortran*, fazendo com que os cálculos realizados pela biblioteca sejam muito mais ágeis e precisos.

4. ESTUDO DE CASO

Nesse tópico, são apresentados a proposta e o processo de desenvolvimento do algoritmo. Vale destacar que esse trabalho não apresenta como objetivo ensinar *Python* ou os processos e meios de extração de dados, e sim, demonstrar a AS dentro do contexto de *feedback* de clientes para empresas. Assim, com o foco do trabalho bem definido, o mesmo poderá ser utilizado como inspiração e/ou modelo para futuros trabalhos em análise de dados em diferentes contextos.

4.1. PROPOSTA

Esse trabalho propõe o desenvolvimento de um algoritmo escrito em *Python* que seja capaz de analisar o sentimento presente em textos de língua portuguesa em uma base de dados fictícia, onde o aluno estruturará alguns comentários públicos em sites de varejo que oferecem algum produto ou serviço, no formado *CSV*.

Para auxiliar-nos no processo de elaboração, foi considerado como base as bibliotecas discorridas a seguir. Em um primeiro momento, utilizou-se a biblioteca *NLTK*, visto que é uma das ferramentas mais completas e abrangentes na análise de dados, principalmente por oferecer a *Word Net* como classificador de textos, predefinição de *stopwords* em português e métodos que auxiliam na construção da matriz de mapeamento dos sentimentos. Em conjunto, a *Scikit-Learn* foi utilizada para realizar o treinamento do modelo de IA. A biblioteca *RE*, oferecida pela própria linguagem, a partir da qual foi aplicada para interpretar expressões regulares em nosso código. E, por último, a biblioteca *Pandas*, que oferece ferramentas práticas de manipulação, visualização e análise de dados.

É imprescindível enfatizar ainda que os passos para a construção do algoritmo podem ser feitos tanto localmente, no computador, quanto na nuvem, utilizando o *Google Colab*. Para o presente estudo, será utilizado o ambiente do *Colab*.

4.2. DESENVOLVIMENTO DO ALGORITMO

Nessa seção, encontra-se o processo para o desenvolvimento do algoritmo, incluindo o código do mesmo. Para maior esclarecimento, apresentamos os trâmites por

meio do diagrama abaixo, o qual representa o fluxo de desenvolvimento e cada uma de suas etapas explicadas e detalhadas.

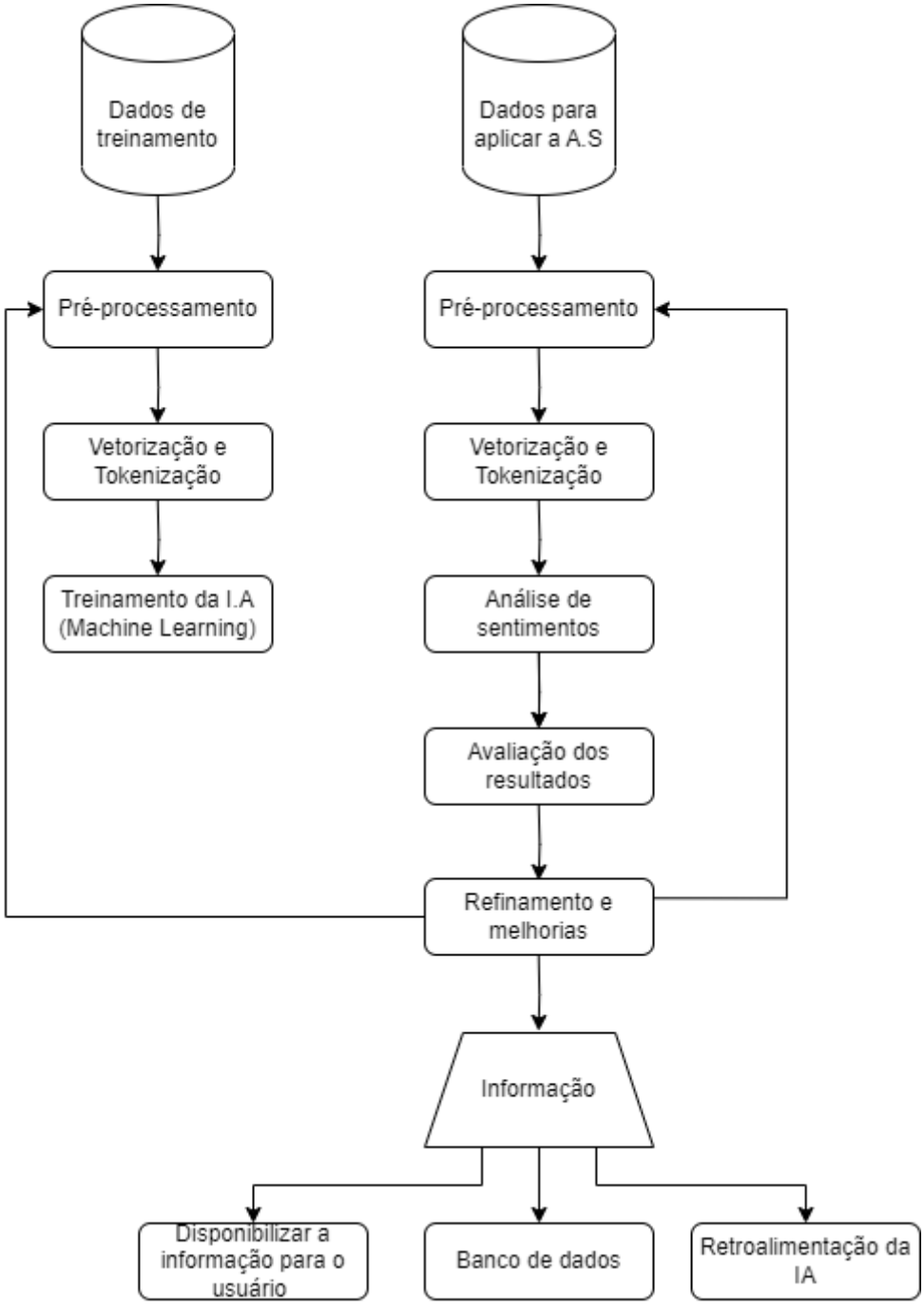

Figura 4: diagrama de organização e processos do desenvolvimento do algoritmo

Conforme o diagrama, há duas bases de dados utilizadas no algoritmo. A primeira foi nomeada como de base de dados de treinamento, a qual é uma base pública que pode ser encontrada na internet contendo textos já classificados como positivo e negativo, e é utilizada para treinar o modelo de IA, aplicando aprendizado de máquina. A

segunda base foi chamada de base de dados de aplicação, ou seja, os dados públicos que podem ser encontrados em páginas de varejo *on-line*, especificamente na área onde os clientes postam a sua opinião (*feedback*) sobre a compra de um determinado produto. Nesta base de aplicação, é aplicado o algoritmo treinado a partir da base de treinamento para realizar a análise de sentimentos.

Ambas as bases são estruturadas em *CSV*, que é um formato de arquivo que permite a representação de dados, onde cada dado é separado por um delimitador, formando uma estrutura muito similar a uma planilha. A escolha desse formato deve-se ao fato de sua fácil manipulação e modularidade para outros formatos de arquivos.

Em seguida, em ambas as bases são utilizadas técnicas de processamento e tokenização. A partir da base de dados de aplicação, onde é aplicada a análise de sentimentos, obtêm-se os resultados, o que nos permite verificar se foram satisfatórios. Portanto, com a análise de sentimentos aplicada com sucesso é possível auferir a informação obtida dos dados classificados.

4.2.1. Definindo a importação das bibliotecas necessárias

As bibliotecas importadas são: *Pandas*, *RE*, *NLTK*, *String* e *Scikit-Learn*.

```
import pandas as pd
import re
import nltk
nltk.download('stopwords')
nltk.download('rslp')
nltk.download('punkt')
nltk.download('wordnet')
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from nltk.stem import WordNetLemmatizer
wordnet_lemmatizer = WordNetLemmatizer()
```

Figura 5: definição das importações das bibliotecas para o algoritmo

A explicação do porquê e de como cada biblioteca está sendo utilizada pode ser conferida no presente estudo, no desenvolvimento da seção 4.

4.2.2. Importação das bases de dados

Nesse tópico, demonstra-se como ler e estruturar o consumo dos dados, no código em *Python*, os arquivos *CSV* de das bases de treinamento e de teste. Para isso, utilizou-se a biblioteca *Pandas*, que suporta diversos outros tipos de leituras de dados, como *JSON*, *HTML*, *XML*, *Excel*, *SQL*, entre outros.

Assim, é necessário atentar-se ao formato de texto da sua base de dados, pois, quando lidamos com dados em língua portuguesa, há acentos, símbolos e caracteres especiais que são amplamente presentes. Outrossim, se não for especificado o tipo de codificação da sua base, alguns textos podem ser interpretados incorretamente. Portanto, para que o Pandas consiga lidar com esses dados, informamos o parâmetro "*encoding*" com "*utf-8*", a fim de que todos os caracteres sejam reconhecidos corretamente, não interferindo de maneira negativa no treinamento do modelo de IA.

4.2.2.1. Base de dados de Treinamento

```
data_treinamento = pd.read_csv('drive/MyDrive/base_dados_treinamento.csv',
                               sep=';',
                               encoding='utf-8',
                               on_bad_lines='skip')
data_treinamento.head()
```

	texto	sentimento
0	@Tixaa23 14 para eu ir :)	Positivo
1	@drexalvarez O meu like eu já dei na época :)	Positivo
2	Eu só queria conseguir comer alguma coisa pra ...	Positivo
3	:D que lindo dia !	Positivo
4	@Primo_Resmungao Pq da pr jeito!!é uma "oferta...	Positivo

Figura 6: importação do CSV com dados de controle para treinamento da IA

Com o comando "*head*()" na nossa base de controle, conseguimos listar os cinco primeiros dados da tabela. Nela, notamos a presença de duas colunas: "texto" e "sentimento". Com os dados da coluna "texto" já classificados em "sentimento, aplicamos técnicas de pré-processamento convenientes ao nosso contexto de feedback de cliente.

4.2.2.2. Dados de aplicação

```
base_dados_aplicacao = pd.read_csv('drive/MyDrive/base_dados_analise.csv',
                                   sep=';',
                                   encoding='utf-8',
                                   on_bad_lines='skip')
base_dados_aplicacao.head()
```

	Feedback
0	Eu sempre quis esse livro, eu pedi também os o...
1	Amo essa série, é a minha preferida, a cada li...
2	Leitura rápida
3	Os livros de Morgan Rhodes são cheios de avent...
4	produto chegou em ótima qualidade. muito vale ...

Figura 7: importação do *CSV* com dados de aplicação

No caso acima, podemos verificar que a base de aplicação contém apenas uma coluna "*Feedback*". Diferente da base de treinamento, essa não tem as frases já classificadas, já que serão nelas que iremos aplicar o algoritmo para identificação dos sentimentos.

4.2.3. Definindo funções de pré-processamento

Como o contexto na análise encontra-se em textos de língua portuguesa, os enunciados apresentam caracteres especiais, como acentos e/ou espaços em branco desnecessários, que podem interferir negativamente na análise de sentimentos. Para isso, o pré-processamento dos dados torna-se uma etapa importante para a análise, uma vez que sua fonte é campos de texto em que as pessoas têm a liberdade de escrever qualquer coisa. Destarte, processar é um processo de limpeza capaz de remover do texto aquilo que não é importante para o treinamento da IA.

Nesse viés, a seguir foram definidas as funções utilizadas tanto na base de dados de treinamento, quanto na base de dados em que aplicamos a análise de sentimentos.

```
def to_lower_case(texto):
  return texto.lower()
```

Figura 8: função para transformar os textos em minúsculas

A primeira função tem o objetivo de transformar os textos das frases para minúsculas. É necessário realizar esse processo para que, quando os dados de treinamento forem mapeados para a IA, as sequências de palavras que determinam cada sentimento não sejam diferenciadas por conta de alguma letra maiúscula, padronizando o mapeamento. Por exemplo: se na base de treinamento temos “Hoje é um dia feliz” com sentimento “Positivo” definido para essa sequência de caracteres, e passarmos a frase para “hoje é um dia feliz”, a IA irá tratar as duas sequências como diferentes por conta da letra “H”, que se alterna em maiúscula e minúscula.

```
def limpar_texto(texto):
    texto = re.sub(r"http\S+", "", texto).replace('.','')
    texto = re.sub(r'^.+@[^\.].*\.[a-z]{2,}$', 'emailaddress', texto)
    texto = re.sub(r'^http\://[a-zA-Z0-9\-\.]+\.[a-zA-Z]{2,3}(/\S*)?$','webaddress', texto)
    texto = re.sub(r'£|\$', 'moneysymb', texto)
    texto = re.sub(r'^\(?[\d]{3}\)?[\s-]?[\d]{3}[\s-]?[\d]{4}$', 'phonenumbr', texto)
    texto = re.sub(r'\d+(\.\d+)?', 'numbr', texto)
    texto = re.sub(r'\s+', ' ', texto)
    texto = re.sub(r'^\s+|\s+?$', '', texto)
    return (texto)
```

Figura 9: função para remoção de caracteres e substituição de palavras desnecessárias

De acordo com a figura 9, a função “limpar_texto” é responsável por remover e substituir dos dados informações como *links*, endereços de *email*, números e valores monetários, espaços em branco seguidos e pontuações.

```
from nltk.corpus import stopwords
from string import punctuation
stopwords_custom = set(stopwords.words('portuguese') + list(punctuation))
stopwords_custom.remove('não')
def remover_stopwords(texto):
  palavras = [i for i in texto.split() if not i in stopwords_custom]
  return (" ".join(palavras))
```

Figura 10: função para remover as *stopwords* das frases

Já nessa função, definimos o pacote das *stopwords* para a língua portuguesa e concatenamos essas palavras junto a uma lista de pontuações da biblioteca *string*. Essa biblioteca contém todas as pontuações que podem ser encontradas em um texto, tais como vírgulas, pontos de exclamação e de interrogação, traços, barras, entre outras. A partir da remoção de todas essas palavras, ao realizarmos a vetorização, a matriz de palavras mapeadas aos seus respectivos sentimentos ficará menor, mais eficiente e precisa.

No entanto, há um ponto a se atentar ao usar as ferramentas de análise de dados oferecidas por bibliotecas como a *NLTK*. Por mais que atualmente podemos encontrar diversos algoritmos e bases de processamento prontas, é muito importante analisarmos se realmente o que a biblioteca oferece atende a nossa necessidade. O que torna um algoritmo eficiente não é a quantidade de ferramentas de processamento ou o tamanho da base de treinamento, e sim como podemos adaptá-las e torná-las ainda mais poderosas dentro do contexto desejado.

Na lista de *stopwords* na *NLTK*, está presente a palavra “não”. Ou seja, ao passarmos um texto pela função de remoção de *stopwords*, o “não” é eliminado, o que é extremamente prejudicial para o nosso algoritmo, visto que essa palavra é uma negação muito importante no significado e sentimento do texto. Por exemplo, as frases "Eu gosto desse carro" e "Eu não gosto desse carro" possuem significados completamente opostos. Portanto, fica claro que é necessário passarmos o comando “remove(‘não’)” na nossa lista de *stopwords* para que essa palavra não seja afetada pelo pré-processamento dos textos.

```
def pre_processar_dados(texto):
  texto = to_lower_case(texto)
  texto = limpar_texto(texto)
  texto = remover_stopwords(texto)
  return texto
```

Figura 11: função para reutilizar todas as funções acima descritas

Como forma de facilitar o uso das funções de limpeza em outras partes do código sem repetições, vamos aplicá-las a uma única função que recebe, como parâmetro, o texto que desejamos processar.

4.2.4. Aplicando o pré-processamento as bases de dados de treinamento e aplicação

```
textos_treinamento = []
for texto in data_treinamento['texto']:
  print('Antes: ' + texto)
  texto_pre_processado = pre_processar_dados(texto)
  print('Depois: '+ texto_pre_processado)
  textos_treinamento.append(texto_pre_processado)

classificacao_treinamento = data_treinamento['sentimento']
```

Figura 12: função que extrai os textos da base de treinamento e aplica o pré-processamento

No código especificado na figura 12, estamos acessando a coluna “texto” da base de treinamento e, para cada texto presente nas linhas, estamos passando a função de pré-processamento descrita na figura 11. Além do texto, estamos extraindo quais são os tipos de sentimentos que há na base. Como o nosso contexto está restrito a feedbacks de clientes, teremos apenas os sentimentos “positivos” e “negativos” na base.

```
Antes: Quem te pegaria curte aqui — 0 :( https://t.co/9nscfJk5uI
Depois: pegaria curte aqui — numbr :(
Antes: @lucaskaaue eu quero um coalaaa e um bicho preguiça :(((
Depois: @lucaskaaue quero coalaaa bicho preguiça :(((
Antes: @Jimultisa quero att :(( amei a au
Depois: @jimultisa quero att :(( amei au
Antes: Meninas que são abandonadas pela namorada no dia do aniversário de namoro :(
Depois: meninas abandonadas namorada dia aniversário namoro :(
Antes: @simaira_ Que aconteceu? :(
Depois: @simaira_ aconteceu? :(
Antes: Eu to muito sentimental hoje :(
Depois: to sentimental hoje :(
Antes: @Gabisalvis @PedrinZika1001 mals pedrinho :(
Depois: @gabisalvis @pedrinzikanumbr mals pedrinho :(
Antes: @anthousai_ Junte. Eu te ajudo. :( GRRR. ????
Depois: @anthousai_ junte ajudo :( grrr ????
```

Figura 13: antes e depois de pré-processar os textos da base de treinamento

Podemos reparar que vários caracteres, pontuações e *links* foram removidos. Vale ressaltar ainda que não necessariamente as funções de pré-processamento, definidas nas figuras anteriores do texto, vão melhorar o desempenho da análise de sentimentos em todos os contextos.

```
textos_aplicacao = []
for texto in base_dados_aplicacao['Feedback']:
  print('Antes: ' + texto)
  texto_pre_processado = pre_processar_dados(texto)
  print('Depois: '+ texto_pre_processado)
  textos_aplicacao.append(texto_pre_processado)
```

Figura 14: função que extrai os textos da base de aplicação e aplica o pré-processamento

Como a base de aplicação possui apenas uma coluna, estamos extraindo os textos, acessando "*Feedback*" e passando a função de pré-processamento para cada linha do *CSV*.

```
Antes: Eu sempre quis esse livro, eu pedi também os outros dois. Chegou bem antes do prazo, não demorou nem 3 dias
Depois: sempre quis livro, pedi outros dois chegou bem antes prazo, não demorou numbr dias
Antes: Amo essa série, é a minha preferida, a cada livro uma emoção. Recomendo.
Depois: amo série, preferida, cada livro emoção recomendo
Antes: Leitura rápida
Depois: leitura rápida
```

Figura 15: resultado do pré-processamento dos dados da base de aplicação

Na figura 15, está listado o antes e depois do pré-processamento em algumas das frases da base de dados de aplicação.

4.2.5. Definindo modelo para separação dos tokens e vetorização

A "tokenização" é um processo importante na AS e no PLN, visto que a maioria dos algoritmos não usufruem de sua máxima capacidade ao lidar com textos em sua forma bruta. A "tokenização" dos dados é processo que dá-se por dividir as frases em unidades menores, chamadas de *tokens*, as quais podem ser uma palavra, um caractere ou um conjunto de símbolos.

Como na pesquisa lidamos com textos informais – que podem conter erros, desvios gramaticais, emojis representados a partir de letras e pontuações, como: ":D", "=)", ":(", "<3", entre outras representações da polaridade do sentimento –, não é recomendado usarmos um "tokenizador" que não seja preparado para lidar com esse tipo de texto. Para isso, a *NLTK* oferece dois "tokenizadores" muito populares, um é o "*word_tokenizer*" e o outro é o "*TweetTokenizer*". No caso da primeira opção, esta vai considerar tokens tudo

aquilo que for separado por espaços e as pontuações serão consideradas *tokens* individuais.

```
from nltk.tokenize import word_tokenize
frase = 'Esse produto é muito ruim! @vendedor não retornou as minhas reclamações, desapontado :('
word_tokenize(frase)
```

```
['Esse',
 'produto',
 'é',
 'muito',
 'ruim',
 '!',
 '@',
 'vendedor',
 'não',
 'retornou',
 'as',
 'minhas',
 'reclamações',
 ',',
 'desapontado',
 ':',
 '(']
```

Figura 16: exemplo de “tokenizador” não preparado para textos informais

Podemos observar que “@vendedor” é uma forma de referenciar a pessoa que fez a venda, mas, ao aplicar a tokenização, “@” e “vendedor” tornaram-se *tokens* diferentes, sendo que o correto, nessa situação, seria considerá-los apenas um. Outro exemplo é a representação de um emoji “:(“ que simboliza tristeza, “:” e “(“ foram separados.

```
from nltk.tokenize import TweetTokenizer
text_tokenizer = TweetTokenizer()
frase = 'Esse produto é muito ruim! @vendedor não retornou as minhas reclamações, desapontado :('
text_tokenizer.tokenize(frase)
```

```
['Esse',
 'produto',
 'é',
 'muito',
 'ruim',
 '!',
 '@vendedor',
 'não',
 'retornou',
 'as',
 'minhas',
 'reclamações',
 ',',
 'desapontado',
 ':(']
```

Figura 17: exemplo de “tokenizador” preparado para textos informais

Nessa etapa, podemos reparar que os tokens “@vendedor” e “:(“ foram reconhecidos corretamente. Por mais que seu nome faça menção à rede social *Twitter*, a *TweetTokenizer* é uma dependência da *NLTK*, que foi feita e preparada para lidar com textos em que a escrita é livre e não possuem um padrão dentro da norma culta da língua portuguesa, portanto, ela pode ser usada em outras fontes de dados, não exclusivamente do *Twitter*. Por conseguinte, empregamos o *TweetTokenizer* como nosso “tokenizador” para realizar a vetorização dos dados.

Assim, será possível elaborar o modelo de “tokenização” e vetorização dos dados que será utilizado:

```
vetorizador = CountVectorizer(analyzer='word', tokenizer=text_tokenizer.tokenize)
```

Figura 18: definição do modelo de “tokenização” e vetorização

4.2.6. Treinamento da I.A, *Machine Learning* e *Bag of Words*

Nesse momento, treinamos um modelo de IA a partir dos dados da base de treinamento já processados. Para isso, é usado um conceito chamado *Bag of Words*, que é uma das formas de vetorização de dados muito eficiente na análise de sentimentos.

Para exemplificarmos, empregamos duas frases como forma de demonstração de como ocorre essa vetorização:

Base de dados de treinamento		
#	Frase	Sentimento
Frase 1	Esse produto me agradou bastante	Positivo
Frase 2	Esse produto é muito ruim	Negativo

Tabela 1: demonstração dos dados de treinamento

Ao aplicar a “tokenização”, temos:

Base de dados de treinamento		
#	Frase	Sentimento
Frase 1	[“Esse”, “produto”, “me”, “agradou”, “bastante”]	Positivo
Frase 2	[“Esse”, “produto”, “é”, “muito”, “ruim”]	Negativo

Tabela 2: demonstração da “tokenização”

Com os tokens, aplicamos o *Bag of Words*. Basicamente, cada *token* virará uma coluna de uma tabela, de forma que nenhum token será repetido na tabela:

Esse	produto	me	é	agradou	muito	Ruim	bastante

Tabela 3: demonstração do funcionamento do Bag of Words

A partir desse passo, faremos uma contagem com cada frase, isto é, qual a frequência que cada palavra da tabela aparece nas frases:

#	Esse	Produto	Me	é	Agradou	muito	ruim	bastante
Frase1	1	1	1	0	1	0	0	1
Frase2	1	1	0	1	0	1	1	0

Tabela 4: matriz binária obtida a partir da frequência de palavras na base de dados

Dessa forma, temos uma matriz binária que representa quantas vezes cada palavra aparece em toda a base. Essa contagem possibilitará, pois, mapear a frequência de cada sentimento já classificado na base de treinamento:

Frase1	Esse	produto	Me	é	Agradou	muito	ruim	Bastante	Sentimento
Frequência	1	1	1	0	1	0	0	1	Positivo

Tabela 5: mapeamento da matriz binária com seu respectivo sentimento

Para aplicarmos esse conceito na base de treinamento, é utilizado o “vetorizador” definido na figura 18:

```
frequencia_treinamento = vetorizador.fit_transform(textos_treinamento)
```

Figura 19: aplicando vetorização aos textos da base de treinamento

Com a matriz de vetorização aplicada e armazenada na variável “frequencia_treinamento”, criamos o modelo de vetorização mapeado com os sentimentos classificados da base de treinamento:

```
modelo = MultinomialNB()
modelo.fit(frequencia_treinamento, classificacao_treinamento)
```

Figura 20: criando modelo mapeado entre matriz de frequência e seus respectivos sentimentos classificados

Ao executar a ferramenta *MultinomialNB* fornecida pela biblioteca *Scikit-Learn*, realizamos esse mapeamento com a matriz “frequencia_treinamento” e a lista de classificação, “classificação_treinamento”, definidas respectivamente nas figuras 19 e 12.

Assim, com o nosso modelo de IA treinado e preparado, é possível aplicá-lo a nossa base de aplicação e analisarmos os resultados:

```
frequencia_aplicacao = vetorizador.transform(textos_aplicacao)
```

Figura 21: realizando vetorização dos dados de aplicação

Logo, o mesmo processo de vetorização realizado na base de treinamento é feito na base de aplicação:

```
modelo_treinado = modelo.predict(frequencia_aplicacao)
textos_analisados = zip(textos_aplicacao, modelo_treinado)
```

Figura 22: aplicando AS à base de aplicação a partir do modelo de treinamento

A partir do modelo mapeado de treinamento, os textos vetorizados da base de aplicação são calculados e retornam o sentimento presente no texto. Seguindo no exemplo da tabela 1 acima, vamos adicionar a frase “Ruim pelo preço cobrado” não classificada para compreender o que ocorre nesse trecho do algoritmo. Essa frase será “tokenizada”: “Ruim”, “pelo”, “preço”, “cobrado”. A partir disso, notamos que “Ruim” está presente na segunda frase da tabela 1 e a polaridade mapeada a ela é “Negativo”. A partir disso, o modelo de treinamento pode gerar como resultado que a frase “Ruim pelo preço cobrado” possui teor negativo.

4.2.7. Resultados obtidos

Vamos agora preparar as informações obtidas para convertê-las em uma visualização de tabela:

```
resultados = []
for texto,classificao_analisada in textos_analisados:
  resultado = {
      "Texto analisado": texto,
      "Classificação": classificao_analisada
  }
  resultados.append(resultado)
```

Figura 23: preparação das informações obtidas para visualização em tabela

Com o objeto “resultados” montado, ele será convertido em um *dataframe*:

```
dataframe_resultado = pd.DataFrame(resultados)
dataframe_resultado
```

Figura 24: convertendo objeto das informações para *dataframe*

Visualização dos resultados em uma tabela:

Texto analisado	Classificação
sempre quis livro, pedi outros dois chegou bem antes prazo, não demorou numbr dias	Positivo
amo série, preferida, cada livro emoção recomendo	Positivo
leitura rápida	Positivo
livros morgan rhodes cheios aventuras olhos corações leitores pulam alegrias páginas semeiam sonhos desejos neste livro não diferente tres reinos, tres soberanos, grupo jovens saber vidas esculpidas destino irá unir busca algo além sempre sonharam princesa insegura, principe ama irmã(mas irmã adotada não sabe) assim começa aventura fantástica mundo magistralmente criado morgan rhodes	Positivo
produto chegou ótima qualidade vale apena	Positivo
normalmente primeiro livro série fantasia costuma pouco lento, pois autor introduzindo leitor mundo criou, então demora leitor pegar ritmo	Positivo
não aconteceu queda reinos, história ritmo maravilhoso desde começo, mundo autora criou super interessante bem personagens	Positivo
li três primeiros livros atrás outro, tornou series queridinhas momento	Positivo
primeiras páginas fica presa história, amei, umas melhores leituras neste ano	Positivo
ótimo livro, história bem focada prende	Positivo
ótimo produto, pega não possuir bateria interna, caso queira comprar bateria, preço deveras salgado	Positivo
ponto gostei casa ficou tecnológica produto, toda família gostou, inicio certa resistência tecnologia alexa unico ponto preço elevado, mais, gosto experiência alexa, sendo todos dias deve aprender comigo família, tudo começa falando "alexa"	Positivo
funciona bem, simples, bom preço porém, toda vez vai usado som grita "playing from bla bla bla" simplesmente não dá pra desativar chatice sabe quanto irritando certeza atualização simples resolveria	Positivo
entregador deixou desejar, jogou pedido porta casa não soube esperar gente atender jogou pedido orientação proteção porta garagem tempo quebrar pedido	Positivo
simplesmente paga caro após numbr ano deixa funcionar, decepção tamanho	Negativo
frustrante, não possível interagir forma inteligente, precisa dar comandos sempre pré estabelecidos	Negativo
espaço guardar fio pequeno, quase não coube fonte entrou justa, inclusive arranhou toda caixa fonte colocar echo dot ficou apertado, suporte ficou bem torto não recomendo	Positivo
chegou quebrado precisei devolver material acabamento não ruins, parte dot fica encaixado, quebrou, fina não quebrado transporte pode quebrar facilmente encaixe faz alguma pressão	Negativo
qualidade projeto ruim	Negativo
bom, acho valor não corresponde qualidade produto, útil caso deixar echo parede, aparência melhor, fio aparente	Positivo
produto igual foto, porém não sustenta alexa, caiu tomada sozinho numbrx, não cabe direito cabo enorme echo dot decepção produto	Negativo
difícil instalar, principalmente embutir fio	Negativo
produto bom atende minimamente especificações versão anterior melhor ter espaço colocar fiação forma melhor	Positivo
fácil instalação	Negativo

Figura 25: resultado obtido após realizar a AS

Aqui podemos ver que o algoritmo atingiu bons resultados na assertividade dos sentimentos, apesar de haver algumas situações em que ele errou a classificação da frase.

5. CONCLUSÕES

Com o mercado de análise de dados em alta e ferramentas de inteligência artificial tornando-se cada vez mais comuns e necessárias para o nosso dia a dia, fica claro que não podemos simplesmente ignorar as mudanças e facilidades que isso nos trouxe.

Para iniciar nessa área, atualmente não é necessário começar com diversos conhecimentos avançados em estatísticas ou matemática, que são campos de extrema presença em análise de dados. Ferramentas como *Python* vieram com a proposta de abstrair conceitos que são complexos para facilitar o processo de análise de dados e torná-lo mais acessível para pessoas com diferentes formações acadêmicas. Além disso, existem diversas plataformas e cursos online que oferecem capacitação em análise de dados, permitindo que qualquer pessoa interessada possa se familiarizar com as técnicas e ferramentas utilizadas nessa área. Apesar disso, um bom conhecimento em estatística e matemática pode ser um diferencial importante para quem deseja se destacar nesse campo e explorar todo o potencial da análise de dados.

Vale ressaltar que ter uma base de dados de treinamento grande não garante que o algoritmo será mais eficiente na assertividade dos sentimentos. Deve-se atentar a cada contexto, preparar a base de dados, realizar o processamento pertinente aos resultados que se deseja alcançar. Por isso, durante a etapa de desenvolvimento, foi preciso realizar manipulações e alterações nas ferramentas prontas das bibliotecas para que o resultado fosse mais coerente.

6. TRABALHOS FUTUROS

Inicialmente, no processo de desenvolvimento do algoritmo e treinamento do modelo de IA, utilizamos uma base de dados de treinamento com mais de 800 mil dados de textos classificados com sentimentos positivos, negativos e neutros, porém o presente aluno reparou que o resultado estava pouco preciso. Por isso, reduzimos a quantidade de dados para treinamento e removemos os dados com classificação neutra, já que concluiu-se que, no contexto de *feedback* de clientes, esta não seria importante.

```
data_treinamento['sentimento'].value_counts()
```

```
Negativo    9824
Positivo    5883
Name: sentimento, dtype: int64
```

Figura 26: quantidade de dados da base de dados de treinamento

Podemos notar que houve quase o dobro de textos classificados como negativo se comparado com os positivos. Essa decisão foi tomada porque, na etapa de avaliação dos resultados, foi notado que o algoritmo estava com maior dificuldade de identificar os sentimentos negativos, portanto, a sua quantidade foi aumentada, proporcionando assim maior precisão aos resultados.

Diante disso, apontamos a seguir algumas sugestões para possíveis trabalhos futuros, como:

- Organizar a base de dados de treinamento de acordo com cada caso, já que uma base de dados grande não garante uma precisão maior no algoritmo.
- Ter cuidado ao utilizar algumas tratativas de, visto que em alguns casos isso pode atrapalhar nos resultados.
- Sempre buscar uma forma de adaptar as ferramentas disponíveis para análise de dados, não somente as bibliotecas de *Python*, mas em qualquer cenário. Elas vêm com a proposta de facilitar o trabalho com análise de dados e não para realizarem todo o processo de forma independente.

Assim, levando em consideração os pontos relevantes abordados na pesquisa sobre a análise de sentimentos a partir da IA, investigados teoricamente e na prática, possibilitará que futuros pesquisadores e/ou estudantes de tecnologia e outras áreas avancem mais profundamente no assunto, alicerçados pelo contínuo avanço tecnológico.

REFERÊNCIAS

ABIOLA, ODEYINKA.; ABAYOMI-ALLI, ADEBAYO; TALE, OLUWASEFUNMI AROGUNDADE. et al. **Sentiment analysis of COVID-19 tweets from selected hashtags in Nigeria using VADER and Text Blob analyser. Journal of Electrical Systems and Inf Technol**, v. 10, n. 5, 2023. Disponível em: https://doi.org/10.1186/s43067-023-00070-9. Acesso em: 5 jun. 2023.

DALE, Robert. **Classical approaches to natural language processing**. In: INDURKHYA, N.; DAMERAU, F. J. (Ed.) Handbook of natural language processing. 2. ed. Boca Raton: Chapman & Hall/CRC, 2010

DALE, Robert. **Text Analytics APIs, Part 1: The Bigger Players**. Natural Language Engineering, vol. 24, no. 2, 2018, pp. 317–324., doi:10.1017/S1351324918000013.

DENG, J.; LIN, Y. **The Benefits and Challenges of ChatGPT: An Overview**. Frontiers in Computing and Intelligent Systems, [S. l.], v. 2, n. 2, p. 81–83, 2023. DOI: 10.54097/fcis.v2i2.4465. Disponível em: https://drpress.org/ojs/index.php/fcis/article/view/4465. Acesso em: 7 ago. 2023.

DEO, Gouri Shashank. et al. **Predictive Analysis of Resource Usage Data in Academic Libraries using the VADER Sentiment Algorithm**. In: 12th International Conference on Computational Intelligence and Communication Networks (CICN), 2020, Bhimtal, India. Anais... Bhimtal: IEEE, 2020. p. 221-228. DOI: 10.1109/CICN49253.2020.9242575.

FERRUCCI, David A. **Introduction to "This is Watson**. IBM Journal of Research and Development, v. 56, n. 3.4, p. 1:1-1:15, maio-jun. 2012. DOI: 10.1147/JRD.2012.2184356.

G1. **Demanda por profissionais da área de dados cresce quase 500%; salários chegam a R$ 22 mil**. 05 jul. 2021. Disponível em < https://g1.globo.com/economia/concursos-e-emprego/noticia/2021/07/05/demanda-por-profissionais-da-area-de-dados-cresce-quase-500percent-salarios-chegam-a-r-22-mil.ghtml>. Acesso em: 11 nov. 2022.

HAO, Jiangang.; HO, Tin Kam. **Machine Learning Made Easy: A Review of Scikit-learn Package in Python Programming Language**. Journal of Educational and Behavioral Statistics. 2019, 44(3), 348–361. https://doi.org/10.3102/1076998619832248

KARRA, Sai. **How Social Media Is Changing Business Strategies**. Forbes. Disponível em: <https://www.forbes.com/sites/forbesbusinesscouncil/2023/01/19/how-social-media-is-changing-business-strategies/?sh=ba174962f5fe>. Acesso em: 14 fev. 2023.

LIMA, Guilherme. **Python: a origem do nome**. Alura. Disponível em: < https://www.alura.com.br/artigos/python-origem-do-nome>. Acesso em: 20 fev. 2023.

LIU, Bing. **Sentiment Analysis and Opinion Mining**, 1.ed. Editora Morgan & Claypool Publishers, 2012.

MARTINS, J.S.; LENZ, M.L.; SILVA, M.B.F.D.; AL., E. **Processamentos de Linguagem Natural**. Porto Alegre: Grupo A, 2020. 9786556900575. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9786556900575/. Acesso em: 08 jun. 2022

NETTO, Amilcar; MACIEL, Francisco. **Python para Data Science e Machine Learning Descomplicado**, 1. ed., Rio de Janeiro: Editora Alta Books, 2021.

PACETE, Luiz Gustavo. **Google x Microsoft: o que está em jogo na disputa entre Bard e ChatGPT**. Forbes. Disponível em: < https://forbes.com.br/forbes-tech/2023/02/google-x-microsoft-o-que-esta-em-jogo-na-disputa-entre-bard-e-chatgpt/>. Acesso em: 16 jun. 2023.

PERKINS, Jacob. **Python Text Processing with NLTK 2.0 Cookbook**, 1.ed. Editora Packt Publishers, 2010.

PERNA, Cristina Becker Lopes; DELGADO, Heloísa Orsi Koch; FINATTO, Maria José Bocornys. **Linguagens Especializadas em Corpora: Modos de Dizer e Interfaces De Pesquisa**. Porto Alegre: ediPUCRS, 2010.

PYTHON. **What is Python? Executive Sumary**. Disponível em: https://www.python.org/doc/essays/blurb/. Acesso em: 28 mai. 2023.

RAUTENBERG, S.; CARMO, P. R. V. do. **Big data e ciência de dados: complementariedade conceitual no processo de tomada de decisão**. Brazilian Journal of Information Science: research trends, [S. l.], v. 13, n. 1, p. 56–67, 2019. DOI: 10.36311/1981-1640.2019.v13n1.06.p56. Disponível em: https://revistas.marilia.unesp.br/index.php/bjis/article/view/8315. Acesso em: 28 maio. 2023.

RODRIGUES, Jonatan. **Pesquisa indica recursos mais relevantes de mídias sociais + 95 estatísticas de redes em 2022**. Resultados Digitais. Disponível em: <

https://resultadosdigitais.com.br/marketing/estatisticas-redes-sociais/#:~:text=Para%2065%25%20dos%20brasileiros%2C%20o,%2C%20com%2046%2C7%25>. Acesso em: 10 abr. 2023.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistemas de Bancos de Dados**. 5. ed. Tradução de Daniel Vieira. Rio de Janeiro: Editora Elsevier, 2006.

SILVA, Nadia Felix Felipe da. **Análise de sentimentos em textos curtos provenientes de redes sociais**. 2016. 138p. Tese (Doutorado) – Instituto de Ciências Matemáticas e de Computação – ICMC-USP/ Instituto de Ciências Matemáticas e de Computação, São Paulo, São Carlos, 2016.