

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CAMPUS TIMÓTEO**

Marco Aurélio Ferreira Moreira

**ANÁLISE DE SENTIMENTOS EM *REVIEWS* DE JOGOS
UTILIZANDO PROCESSAMENTO DE LINGUAGEM NATURAL -
ESTUDO DE CASO**

Timóteo

2024

Marco Aurélio Ferreira Moreira

**ANÁLISE DE SENTIMENTOS EM *REVIEWS* DE JOGOS
UTILIZANDO PROCESSAMENTO DE LINGUAGEM NATURAL -
ESTUDO DE CASO**

Monografia apresentada à Coordenação de Engenharia de Computação do Campus Timóteo do Centro Federal de Educação Tecnológica de Minas Gerais para obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Luciano Nascimento Moreira
Coorientadora: Andressa Oliveira Souza

Timóteo

2024

**ANÁLISE DE SENTIMENTOS EM REVIEWS DE JOGOS
UTILIZANDO PROCESSAMENTO DE LINGUAGEM NATURAL -
ESTUDO DE CASO**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, campus Timóteo, como requisito parcial para obtenção do título de Engenheiro de Computação.

Trabalho aprovado. Timóteo, 14 de fevereiro de 2025:

Documento assinado digitalmente
 **LUCIANO NASCIMENTO MOREIRA**
Data: 20/02/2025 12:46:57-0300
Verifique em <https://validar.iti.gov.br>

Prof. Luciano Nascimento Moreira
Orientador

Documento assinado digitalmente
 **ANDRESSA OLIVEIRA SOUZA**
Data: 20/02/2025 12:49:23-0300
Verifique em <https://validar.iti.gov.br>

Prof. Andressa Oliveira Souza
Coorientadora

Documento assinado digitalmente
 **MARCELO DE SOUSA BALBINO**
Data: 20/02/2025 16:48:09-0300
Verifique em <https://validar.iti.gov.br>

Prof. Marcelo de Sousa Balbino
Professor Convidado

Dedico este trabalho aos meus pais,
que, em troca de suas noites de sono,
me permitiram sonhar.

Agradecimentos

A conclusão deste trabalho representa o fim de uma importante etapa da minha vida acadêmica, e foi possível graças ao apoio e incentivo de muitas pessoas, às quais expresso aqui minha mais profunda gratidão.

Aos meus amigos, que estiveram ao meu lado ao longo desta jornada, oferecendo palavras de encorajamento e momentos de descontração nos momentos mais desafiadores, meu muito obrigado.

À minha família, pelo apoio incondicional, pela compreensão e pelo carinho, que foram essenciais para que eu pudesse me dedicar a este projeto com tranquilidade e confiança. Vocês são a base de tudo.

Não poderia deixar de expressar minha gratidão especial a orientação da professora Andressa. Sua orientação foi muito além das expectativas. Sua paciência, compreensão e o tempo dedicado ao meu projeto foram fundamentais para o desenvolvimento deste trabalho. Cada conselho, revisão e crítica construtiva contribuíram significativamente para o amadurecimento das minhas ideias e para o resultado deste trabalho. Agradeço profundamente por todo o conhecimento que você compartilhou comigo e pela confiança depositada em meu potencial.

A todos, meu mais sincero agradecimento.

*“Nós somos o que repetidamente fazemos.
A excelência, portanto, não é um ato, mas um hábito.”
Aristóteles*

Resumo

Com o avanço da internet e das mídias sociais, tornou-se possível para os usuários não apenas consumir, mas também criar e compartilhar suas opiniões em tempo real. Este trabalho explora a aplicação de técnicas de Análise de Sentimentos para classificar avaliações de jogos como positivas ou negativas, utilizando técnicas de Processamento de Linguagem Natural (PLN) e aprendizado de máquina. Foram avaliados três algoritmos principais: *Multinomial Naïve Bayes* (MNB), *Random Forest* (RF) e *Decision Tree* (DT), aplicados a uma base de dados com 21.747.371 avaliações (antes da subamostragem), contendo avaliações de diferentes jogos e idiomas. A base foi balanceada e reduzida para 93.746 amostras para lidar com o desbalanceamento, sendo que, antes do balanceamento, havia 790.015 avaliações positivas e 46.873 negativas. Os resultados, obtidos por meio de validação cruzada com 10 *folds*, mostraram que o MNB obteve a maior acurácia média de 87,31%, com sensibilidade de 87,07%, especificidade de 87,56%, precisão de 87,52% e F1-score de 87,29% para avaliações negativas. Compreender as avaliações negativas é crucial para entender a qualidade e aceitação do projeto, uma vez que elas podem impactar negativamente nas vendas. Tanto o classificador RF quanto o DT apresentaram acurácia média inferior, com valores de 86,55% e 81,61%, respectivamente. Apesar dos desafios enfrentados, como a necessidade de subamostragem, que foi essencial para obter os resultados devido às limitações de hardware e ao desbalanceamento da base, o estudo demonstrou o desempenho dos classificadores escolhidos na análise de sentimentos dos dados textuais presentes nas avaliações dos jogos.

Palavras-chave: Análise de Sentimentos, Processamento de Linguagem Natural, Aprendizado de Máquina, Avaliações de Jogos, Classificadores.

Abstract

With the advancement of the internet and social media, it has become possible for users not only to consume but also to create and share their opinions in real-time. This work explores the application of Sentiment Analysis techniques to classify game reviews as either positive or negative, using Natural Language Processing (NLP) and machine learning techniques. Three main algorithms were evaluated: *Multinomial Naïve Bayes* (MNB), *Random Forest* (RF), and *Decision Tree* (DT), applied to a database with 21.747.371 reviews (before subsampling), containing reviews of various games and languages. The dataset was balanced and reduced to 93,746 samples to address the imbalance, with 790,015 positive reviews and 46,873 negative reviews before balancing. The results, obtained through 10-fold cross-validation, showed that MNB achieved the highest average accuracy of 87.31%, with a sensitivity of 87.07%, specificity of 87.56%, precision of 87.52%, and an F1-score of 87.29% for negative reviews. Understanding negative reviews is crucial for assessing the quality and acceptance of a project, as they can negatively impact sales. Both RF and DT classifiers showed lower average accuracies, with values of 86.55% and 81.61%, respectively. Despite the challenges faced, such as the need for undersampling, which was essential for obtaining the results due to hardware limitations and dataset imbalance, the study demonstrated the performance of the chosen classifiers in analyzing the sentiment of textual data present in game reviews.

Keywords: Sentiment Analysis, Natural Language Processing, Machine Learning, Game Reviews, Classifiers.

Sumário

1	INTRODUÇÃO	9
1.1	Objetivos	10
2	TRABALHOS CORRELATOS	11
3	FUNDAMENTAÇÃO TEÓRICA	13
3.1	Processamento de Linguagem Natural	13
3.1.1	Análise fonológica	14
3.1.2	Análise morfológica	14
3.1.3	Análise lexical	14
3.1.4	Análise sintática	15
3.1.5	Análise semântica	16
3.1.6	Análise pragmática	17
3.2	Aprendizado de Máquina	17
3.3	Algoritmos classificadores	18
3.3.1	Multinomial Naïve Bayes	18
3.3.2	Random Forest	19
3.3.3	Decision Tree	22
3.4	Undersampling	24
3.4.1	Random undersampling	24
3.5	Métricas	24
3.5.1	Acurácia	24
3.5.2	Precisão	25
3.5.3	Sensibilidade	25
3.5.4	Especificidade	25
3.5.5	F1-Score	25
4	MATERIAIS E MÉTODOS	26
4.1	Base de Dados	26
4.2	Pré-processamento da Base de Dados	27
4.3	Escolha dos Algoritmos Classificadores	27
4.4	Treinamento da Base de Dados	27
4.5	Avaliação dos Resultados	27
5	METODOLOGIA EXPERIMENTAL	29
5.1	Leitura e análise da base de dados	29
5.2	Tratamento e balanceamento da base de dados	30
5.3	Classificação	31
6	RESULTADOS	32

6.1	Comentários positivos	32
6.2	Comentários negativos	36
6.3	Analisando os resultados	37
7	CONSIDERAÇÕES FINAIS	38
	REFERÊNCIAS	40

1 Introdução

A indústria de jogos ao longo dos últimos anos passou por um grande impulsionamento e isso fez com que novos fãs e entusiastas de jogos surgissem por todo o mundo. Somente no ano de 2021, o mercado de jogos movimentou cerca de 178 bilhões de dólares e ultrapassou o mercado de música e cinema (VIANA, 2022). Em 2022, a Pesquisa Game Brasil (2022) divulgou uma estimativa de que cerca de 74,5% da população brasileira joga algum jogo.

Paralelamente a isso, o avanço da internet acabou permitindo que as pessoas não apenas consumam conteúdo, mas também o criem e o compartilhem instantaneamente por meio de plataformas de mídia social, como Facebook, Twitter e Instagram (STROPPIA; ROTHENBURG, 2015). Com isso, formas de interação direta entre as empresas de jogos e o público foram criadas, possibilitando o compartilhamento de *reviews* (do inglês, avaliações) relacionadas aos jogos, sejam eles negativos ou positivos, em *sites* como o Reddit. Essas avaliações permitem que as empresas desenvolvedoras tomem conhecimento sobre a aceitação do seu produto pela comunidade, o que é algo importante já que a aceitação pela comunidade interfere diretamente nas vendas.

De acordo com um estudo realizado por Zhu e Zhang (2006), que analisou as vendas globais de 159 jogos de Xbox e PlayStation 2, observou-se que, em média, os comentários positivos obtidos do site GameSpot.com impulsionavam as vendas em 4%, enquanto os comentários negativos poderiam levar a declínios nas vendas superiores a 4%. Além disso, outro estudo conduzido por Li, Xiaolin, Chaojiang Wu e Feng Mai (LI; WU; MAI, 2019) avaliou o impacto dos comentários online nas vendas de tablets em *e-commerces* e concluiu que avaliações positivas podem aumentar as vendas em até 26,7%.

Um dos problemas de avaliar e classificar essas avaliações manualmente é a grande quantidade de conteúdo. Por exemplo, apenas no jogo “Counter-Strike: Global Offensive” a plataforma *Steam* conta com mais de 7 milhões de avaliações (Valve Corporation, 2023). Outro problema enfrentado é que os textos encontrados em redes sociais apresentam um alto grau de informalidade, o que dificulta a classificação automática (ROSA, 2015). Diante desse contexto, surge a necessidade de processar e classificar essas avaliações de forma automática, a fim de se obter um entendimento mais rápido e eficiente do conteúdo transmitido.

A área de pesquisa em Análise de Sentimentos, também conhecida como Mineração de Opinião (do inglês, *Opinion Mining*), é uma referência na literatura para lidar com análise de opiniões, sentimentos, avaliações, atitudes e emoções em relação a entidades como produtos, serviços, organizações, indivíduos, questões, eventos, tópicos e seus atributos, sendo assim um amplo campo de pesquisa (LIU, 2012). Além disso, esse campo de pesquisa está relacionado aos campos de Linguística Computacional e Mineração de Texto, buscando estudar e entender os elementos subjetivos presentes na Linguagem Natural (MEJOVA, 2009).

A Linguagem Natural (LN) refere-se aos idiomas utilizados diariamente pelos seres humanos, como o português e o hindu, e assim como as linguagens de programação, está em

constante evolução, dificultando a definição de regras explícitas (BIRD; KLEIN; LOPER, 2009), sendo frequentemente falada e escrita de forma incorreta.

Compreender a linguagem é entender não apenas as palavras, mas os conceitos e como eles se relacionam para criar significados (BARBOSA et al., 2017). Assim, o Processamento de Linguagem Natural é um campo da Ciência da Computação com o objetivo de estudar o desenvolvimento de *softwares* que analisam, reconhecem e/ou geram textos de linguagens naturais, ou humanas (VIEIRA; LOPES, 2010).

Dessa forma, uma hipótese levantada por esse trabalho é de que a utilização de técnicas de Análise de Sentimento (AS) possa categorizar avaliações de jogos em positivas ou negativas com uma boa precisão nos resultados.

Com isso, este trabalho visa responder a seguinte pergunta de pesquisa: é possível classificar essas avaliações de forma automática utilizando técnicas de Análise de Sentimentos com uma boa precisão?

Nos próximos capítulos, serão apresentados os trabalhos correlatos e a fundamentação teórica que serviram para compreender e embasar o desenvolvimento deste trabalho. Em seguida, é detalhado o roteiro do projeto e a base de dados utilizada para o processamento de linguagem natural, incluindo as tratativas necessárias para garantir a obtenção de resultados confiáveis. Por fim, serão apresentados os resultados do projeto e as conclusões obtidas a partir deles.

1.1 Objetivos

O objetivo geral deste trabalho é explorar a utilização de técnicas de Processamento de Linguagem Natural para classificar avaliações da plataforma de jogos *Steam* como positivas ou negativas. Para alcançar esse objetivo, são propostos os seguintes objetivos específicos:

1. Realizar a revisão bibliográfica, para melhor entendimento e compreensão em relação ao campo de Análise de Sentimento;
2. Investigar o desempenho dos algoritmos classificadores mais utilizados nos trabalhos analisados na base de dados pública *Steam*.

2 Trabalhos Correlatos

Para realizar o embasamento teórico deste projeto, foi necessário estudar trabalhos que abordam a utilização da análise de sentimentos aplicada a produções textuais redigidas em Linguagem Natural. Alguns dos trabalhos estudados utilizam algoritmos como *Random Forest*, *Naïve Bayes*, *Decision Tree*, *Multilayer Perceptron* (MLP) e *Support Vector Machine* (SVM), juntamente com as métricas e técnicas apresentadas anteriormente, para atingir os resultados desejados.

Malta e Kuroiva (2019), desenvolveram um trabalho cujo objetivo era lidar com comentários de cunho ofensivo na plataforma Empurrando Juntos, que prejudicava a experiência dos usuários. Os autores propuseram criar uma aplicação capaz de receber comentários e classificá-los em positivos ou negativos. Para isso, utilizaram o banco de *stopwords* da biblioteca do NLTK em português e para o banco de dados utilizaram o banco de dados do OffComBr. Foram realizados testes utilizando algoritmos como *Stochastic Gradient Descent*, *Random Forest*, *MultiNB*, *Decision Tree*, *MLP*, *Custom Classifier* e *SVC* com o intuito de avaliar a precisão e a sensibilidade em casos de discurso de ódio e discurso comum. O algoritmo *Custom Classifier B* apresentou um bom equilíbrio entre precisão e sensibilidade. Além disso, alguns outros algoritmos também tiveram resultados interessantes: o *Random Forest* favorecia a precisão e o *MultiNB* e *Decision Tree* favoreciam a sensibilidade.

Wongkar e Angdresey (2019) desenvolveram uma análise de sentimentos utilizando um *Data Crawler* para obter comentários da rede social Twitter. Após a coleta, os dados foram processados e por meio dos algoritmos como SVM, KNN e Naïve Bayes realizaram a classificação desses textos de LN em positivo ou negativo. Os dados recolhidos são referentes ao período eleitoral de 2019 na Indonésia que decidiria o presidente do período de 2019-2024. As métricas observadas foram a sensibilidade e a acurácia. Os três métodos tiveram valores próximos em relação à acurácia e sensibilidade, porém o Naïve Bayes se sobressaiu em algumas situações.

No estudo realizado por Nair, G e Vinayak (2021), foram utilizados *tweets* relacionados à COVID-19 coletados do Twitter. No intuito de auxiliar na obtenção dos dados, utilizou-se as “hashtags” que abordavam o assunto e assim conseguiram reunir uma base de dados contendo 25.000 *tweets*. Inicialmente, foi feito um pré-processamento dos dados em que foi realizado a retirada das *stopwords*, pontuações e em sequência foi realizado a tokenização das sentenças. Com isso, para os classificadores utilizados, obtiveram os seguintes resultados: para o *Logistic Regression* tiveram uma acurácia de 83%, uma precisão de 84%, uma sensibilidade de 83% e um F-score de 83%, para o *Bidirectional Encoder Representations* (BERT) obtiveram uma acurácia de 82%, uma precisão de 90%, uma sensibilidade de 94% e um F-score de 92%, já para o *Valence Aware Dictionary for Sentiment Reasoning* obtiveram uma acurácia de 88%, uma precisão de 94%, uma sensibilidade de 93% e um F-score de 88%.

Zhen Zuo (2018) realizou um estudo utilizando dois algoritmos de aprendizado de má-

quina, *Decision Tree* e *Gaussian Naïve Bayes*, para classificar as avaliações obtidas da plataforma de jogos Steam, nos quais continham a quantidade de horas investidas no jogo, a data de postagem, a classificação pelos próprios usuários da plataforma, o perfil do autor e a avaliação do próprio autor. Para obter as reviews, ele utilizou a API (Application Programming Interface) da própria Steam e, após a obtenção, com a ajuda de algumas técnicas, realizou o tratamento dos dados. Durante o desenvolvimento do estudo ele observou que algumas reviews que eram muito curtas e palavras que eram muito frequentes ou infrequentes poderiam trazer ruídos para a aplicação, portanto, foram ignoradas. No total, foram desenvolvidas 196 aplicações, nas quais em 192 foi utilizada a técnica de TF-IDF e nas outras 4 foi o utilizado o IG (Information Gain), o IG foi menos utilizado devido aos autores terem um poder computacional limitado. Considerando o resultado das 196 aplicações utilizando o dataset da Steam, o algoritmo *Decision Tree* teve um desempenho superior ao *Gaussian Naïve Bayes*, no qual o primeiro alcançou uma acurácia de 75%.

Gözde Merve Demirci, Şeref Recep Keskin e Gülüstan Doğan (2019) conduziram um estudo de análise de sentimentos visando analisar os *tweets* turcos que cercavam a data do dia 15 de julho de 2016, na qual houve uma tentativa de golpe de estado na Turquia. Para coletarem os dados do Twitter utilizaram a biblioteca de linguagem chamada Twint e filtraram usando a hashtag “#15Temmuz”, assim, obtiveram 3000 *tweets*, sendo deles, 2500 para treinamento e 500 para teste. Para realizar o PLN em turco foram seguidas as seguintes etapas: *deasciifier* (processo de identificar e converter textos em turco escritos em ASCII para o turco apropriado, por exemplo, *dusunuyorum* para *döşönöyorum*), tokenização, remoção de pontuações, remoção de *stopwords* e *stemming*. Utilizaram para a incorporação de palavras (do inglês, *word embeddings*) *Bag of Words*, TF-IDF e *Word2Vec*, e para o aprendizado de máquina foram utilizados os seguintes algoritmos: *Logistic Regression*, SVM e *Random Forest*.

Assim, a escolha dos classificadores *Decision Tree* (DT), *Random Forest* (RF) e *Multinomial Naïve Bayes* (MNB) é fundamentada em estudos e resultados apresentados por Malta e Kuroiva (2019), Wongkar e Angdresey (2019), Nair, G e Vinayak (2021), Zhen Zuo (2018) e Gözde Merve Demirci, Şeref Recep Keskin e Gülüstan Doğan (2019). Esses trabalhos destacam a eficácia desses algoritmos em diferentes contextos, proporcionando uma base sólida para a seleção.

3 Fundamentação Teórica

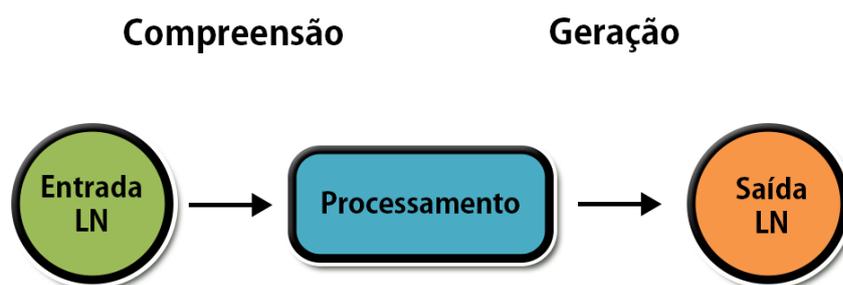
Este capítulo visa explorar os conceitos e tópicos principais utilizados neste trabalho, relacionados ao aprendizado de máquina, incluindo a utilização e entendimento dos classificadores, métricas de avaliação, técnicas avançadas e a aplicação do Processamento de Linguagem Natural (PLN) em problemas de Linguagem Natural.

Na primeira seção, é apresentado o conceito de aprendizado de máquina e uma introdução sobre os tipos de aprendizado de máquina. Na segunda seção, são definidos o que são classificadores e quais foram utilizados no presente trabalho. Em seguida, serão apresentadas as métricas utilizadas para avaliar os resultados e uma das técnicas empregadas no processamento da base de dados. Posteriormente, é conceituado o processamento de linguagem natural, juntamente com as abordagens para trabalhar com o PLN. Por fim, são apresentados os trabalhos correlatos que serviram de apoio para o desenvolvimento deste trabalho.

3.1 Processamento de Linguagem Natural

A Linguagem Natural (LN) é a linguagem falada pelas pessoas e o Processamento de Linguagem Natural (PLN) é tudo que é necessário para o computador receber uma entrada em LN e fornecer uma saída em LN. A PLN é um subcampo da Inteligência Artificial (IA) e linguística cujo objetivo é fazer com que o computador entenda as declarações ou palavras escrita pelas pessoas (CHOPRA; PRASHAR; SAIN, 2013).

Figura 1 – Processamento de Linguagem Natural



Fonte: Imagem elaborada pelo autor

As fases da análise de LN é dividida em vários níveis como: fonológico, morfológico, léxico, sintático, semântico, pragmático e análise do discurso.

3.1.1 Análise fonológica

Fonologia é a análise da linguagem natural falada. Portanto, ela lida com o reconhecimento e geração de discursos. A tarefa principal de reconhecimento e geração de recursos da análise fonológica é a de receber ondas acústicas como entrada e transformar isso em uma saída contendo várias palavras. A análise fonológica faz parte da análise de linguagem natural (KUMAR, 2013).

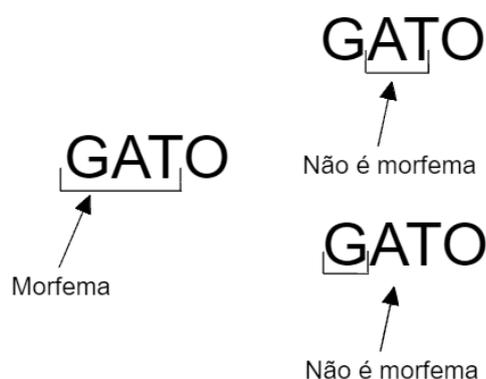
A fonologia busca estudar os sistemas de sons de uma determinada língua, em outras palavras, estuda o comportamento dos fonemas. Esse campo de pesquisa, tem por objetivo fornecer suporte às pesquisas de compreensão da língua que tem por finalidade permitir a comunicação de forma oral com os computadores (BARROS, 1996).

3.1.2 Análise morfológica

A análise morfológica é a fase mais elementar do PLN. Nessa fase é tratado a estrutura das palavras, mais especificamente, as palavras são analisadas conforme os seus “morfemas”. Portanto, alguns caracteres, como pontuações, são separados das palavras. Morfema é a estrutura mais básica de uma palavra (KUMAR, 2013).

O estudo da estrutura da palavra é a morfologia, e no processamento de linguagem natural têm-se a análise morfológica. A tarefa de transformar uma palavra em seu morfema é chamada de *morphological parsing* (da tradução literal, análise morfológica). Um morfema é a menor unidade significativa de uma palavra e não pode ser dividido em partes menores (KUMAR, 2013), por exemplo, o morfema “gat” da palavra “gato”, é possível dividi-lo em mais partes como “g” e “at”, mas essas partes não contêm significado, ou seja, a menor parte significativa é “gat”, como pode ser observado na Figura 2.

Figura 2 – Morfemas e não morfemas da palavra gato.



Fonte: desenvolvida pelo autor desse trabalho.

3.1.3 Análise lexical

O *lexicon* de uma linguagem é o seu vocabulário, isso inclui suas expressões e palavras (CHOPRA; PRASHAR; SAIN, 2013). Em outras palavras, *Lexicon* quer dizer dicionário. Em PLN, o primeiro estágio de processamento dos textos é analisar e identificar todas as in-

formações linguísticas que pode conter algum significado relevante. Nessa etapa é analisado os detalhes das palavras, como se é um advérbio ou um verbo, por exemplo (KUMAR, 2013).

No *lexicon* cada termo pode estar associado às características morfológicas, sintáticas e semânticas. A representação desse é realizada de acordo com representação gramatical do sistema em questão. A Figura 3 mostra a representação de duas palavras no *lexicon* representado em PATR-II (BARROS, 1996 apud SHIEBER, 1984):

Figura 3

```
mesa
  <categoria> = substantivo
  <gênero> = feminino
  <número> = singular

comprou
  <cat> = verbo
  <tempo> = pretérito-perfeito
  <número> = singular
  <peessoa> = 3
  <arg1> = SN
  <arg2> = SN
```

Fonte: (BARROS, 1996)

3.1.4 Análise sintática

A sintaxe é o conjunto das principais sequências de palavras gramaticalmente aceitas por pessoas fluentes em uma língua. Um dos conceitos básicos da sintaxe é o *part-of-speech* (da tradução literal, parte do discurso) no qual se refere ao papel de cada palavra em uma sentença (EISENSTEIN, 2019). Sintaxe se refere ao estudo do relacionamento formal entre palavras. Nessa fase é checado se as sentenças estão de acordo com as regras gramaticais. Para realizar essa etapa com maestria é necessário ter conhecimento de gramática e análise técnica (KUMAR, 2013).

As regras sintáticas, se baseando na categoria sintática, determinam a ordem linear dos constituintes na frase, de forma que os constituintes de uma frase compõem uma relação hierárquica entre si. Dessa forma, uma frase é composta por constituintes (por exemplo, SN e SV), que, por sua vez, são compostos por constituintes de ordem inferior (por exemplo, SP e SAdv), assim até chegar nas categorias mais básicas (por exemplo, preposição, substantivo e advérbio). A Figura 5 e 4 ilustram uma análise sintática da frase “A menina quebrou o jarro azul”.

Figura 4

Gramática

F → SN SV

SN → Det Subs

SN → Det Subs Adj

SN → Subs Adj

SV → V SN

Det → a

Subs → menina

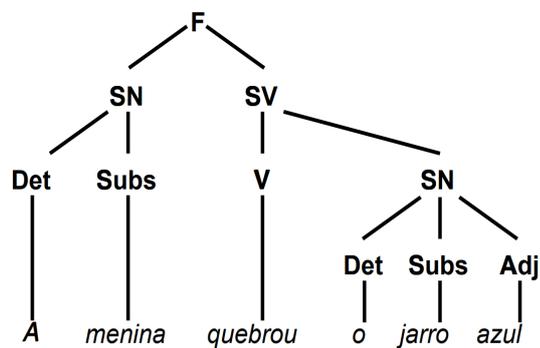
Subs → jarro

Adj → azul

V → quebrou

Fonte: (BARROS, 1996)

Figura 5



Fonte: (BARROS, 1996)

3.1.5 Análise semântica

A semântica é o significado exato da palavra no contexto ou o seu significado abstrato (CHOPRA; PRASHAR; SAIN, 2013). A análise semântica é o processo de converter a linguagem natural em sua representação do significado e a lógica semântica tem de seguir alguns critérios (EISENSTEIN, 2019).

- A representação do significado não pode ser ambíguo, tem de ter apenas um significado por afirmação.
- Tem de fornecer uma maneira de relacionar uma linguagem com ações, conhecimentos e observações.

- Deve suportar inferência computacional, de forma que possa derivar novos conhecimentos pela combinação dos significados.
- Deve ser expressivo o suficiente para abranger todas as coisas que as pessoas conversam na linguagem natural.

3.1.6 Análise pragmática

A pragmática busca abstrair ou derivar o real sentido do uso da linguagem, principalmente em situações que é necessário retirar a ambiguidade de um trecho, por exemplo, a entrada “*closethewindow?*” deve ser interpretada como uma requisição ao invés de uma ordem (CHOPRA; PRASHAR; SAIN, 2013). A pragmática específica como a informação do contexto pode ser melhor interpretada para obter melhor sentido entre diferentes semânticas e identificar ambiguidade (CAMBRIA; WHITE, 2014).

De acordo com Muller (2002), essa análise extrapola à estrutura de uma única frase. Ela busca em outras frases a compreensão do contexto que falta à frase em análise. Em geral, as estruturas não têm pré-definições que atendam a uma representação de problema de referências pronomiais (por exemplo, os pronomes *la*, *seu* e *o* no contexto: *João pegou a rosa. Ao pegá-la, seu espinho o espetou*), análise de discurso e coerência textual.

3.2 Aprendizado de Máquina

Originalmente o Aprendizado de Máquina (AM) nasceu como um campo da Inteligência Artificial na década de 60 com objetivo de aprender padrões a partir de uma base de dados. Porém, após os anos 90 essa área expandiu e passou a se estabelecer como um campo por si mesma. Em particular, as aplicações de AM começaram a ter relações com as de estatística (IZBICKI; SANTOS, 2020).

O termo “Aprendizado de máquina” é comumente usado para abranger procedimentos computacionais automáticos que se baseiam em operações lógicas ou binárias e que aprendem uma tarefa a partir de uma série de exemplos (MICHIE; SPIEGELHALTER; TAYLOR, 1994). Em outras palavras, AM diz respeito ao desenvolvimento de algoritmos que permitem que um computador aprenda. Os algoritmos de AM são classificados pela taxonomia, baseada no tipo de saídas que eles fornecem (AYODELE, 2010), alguns exemplos são:

- Aprendizado supervisionado: o algoritmo gera uma função que mapeia as entradas para as saídas desejadas. Um exemplo comum de tarefas de aprendizado supervisionado é um problema de classificação em que o algoritmo é requisitado a aprender uma função que mapeia um vetor em uma ou diversas classes por meio de exemplos de entradas-saídas (AYODELE, 2010).
- Aprendizado não-supervisionado: o algoritmo modela um conjunto de entradas sem ter acesso a exemplos previamente classificados (AYODELE, 2010).

- Aprendizado semi-supervisionado: combina exemplos classificados e não classificados para gerar uma função ou classificador adequado (AYODELE, 2010).
- Aprendizado por reforço: onde o algoritmo aprende uma política de como agir com base em uma observação do mundo. Cada ação têm uma reação e o ambiente fornece um *feedback* que guia o algoritmo (AYODELE, 2010).

3.3 Algoritmos classificadores

Os algoritmos classificadores são algoritmos de aprendizagem supervisionada que a partir de uma entrada composta por certos atributos tem por objetivo prever a qual classe ou rótulo essa pertence. A princípio, o algoritmo é treinado utilizando uma base de dados em que esses já se encontram previamente classificados em duas classes (classificação binária) ou em diversas classes (classificação multiclasse) (FONTANA, 2020).

As próximas subseções apresentam os algoritmos classificadores utilizados neste trabalho: o Multinomial Naïve Bayes que é um algoritmo feito a partir do algoritmo de Naïve Bayes; e os algoritmos Decision Tree e Random Forest, que utilizam o conceito de árvores de decisão para realizar as predições.

3.3.1 Multinomial Naïve Bayes

Os métodos baseados na abordagem Naïve Bayes são um conjunto de algoritmos em que a partir de um conjunto de atributos é determinado a probabilidade de um certo elemento pertencer a um grupo e essa probabilidade é calculada tendo como base o Teorema de Bayes (FONTANA, 2020). O Teorema de Bayes é um simples modelo matemático utilizado para calcular probabilidades condicionais. Esse teorema desempenha um papel proeminente em abordagens subjetivas ou Bayesianas para estatísticas, epistemologia e lógica indutiva (JOYCE, 2003).

O Multinomial Naïve Bayes (MNB) refere-se a uma aplicação multinomial do Naïve Bayes, que é usada para modelar probabilidade de ocorrência de múltiplas classes ou categorias em um conjunto de dados discreto. Assim, conforme os fundamentos do MNB, a probabilidade de um documento d pertencer à uma classe c é dado por:

$$P(c | d) = P(c) \prod_{i=1}^{|d|} P(w_i | c)^{f_i^d} \quad (3.1)$$

Em que $P(c)$ é probabilidade a priori da classe c , $P(w_i | c)$ é a condicional probabilística da palavra w_i pertencer à classe c e f_i^d é o número de ocorrências da palavra w_i no documento d . Essas probabilidades normalmente são estimadas tendo como base um *dataset* D com classes rotuladas em uma base de treinamento (WAGNER et al., 2015).

Para representar a estimativa pode ser usado o acento circunflexo, como \hat{P} , e assim a probabilidade a priori da classe $P(c)$ pode ser estimada por meio fração do conjunto de documentos de treinamento pertencentes à classe c . Em que N_c é o número de documentos

em \mathcal{D} pertencentes a classe c e $|\mathcal{D}|$ é o número total de documentos em \mathcal{D} (WAGNER et al., 2015).

$$\hat{P}(c) = \frac{N_c}{|\mathcal{D}|} \quad (3.2)$$

A probabilidade condicional $P(w_i | c)^{f_i^d}$ pode ser estimada através da frequência relativa da palavra $w_i \in V$ nos documentos da classe c :

$$\hat{P}(w_i | c) = \frac{N_{ic}}{\sum_{j=1}^{|V|} N_{jc}} \quad (3.3)$$

Na equação acima, N_{ic} é o número de ocorrências da palavra w_i nos documentos com rótulos c em \mathcal{D} , V é o vocabulário no conjunto de treinamento \mathcal{D} . Para as palavras que são desconhecidas durante a predição, ou seja, não presentes em V , é aplicado a correção de Laplace e assim o valor de probabilidade dessa é inicializado como $1/|V|$ (WAGNER et al., 2015).

A partir das fórmulas acima é possível entender as quantidades necessárias para estimar a probabilidade priori da classe $\hat{P}(c)$ e as estimativas condicionais da classe das palavras $\hat{P}(w_i | c)$ são os contadores de probabilidade priori da classe N_c e os contadores de probabilidade da palavra-classe N_{ic} , onde $w_i \in V$, $c \in C$ são todos calculados a partir do conjunto de treinamento \mathcal{D} . Assim, o MNB convencional, estático, utiliza do conjunto inteiro de treinamento \mathcal{D} para calcular esses contadores (WAGNER et al., 2015).

A extensão típica para fluxos de dados, atualiza as contagens com base em novas instâncias do fluxo. Seja d um novo documento que entra do fluxo S com rótulo de classe c . Atualizar o modelo MNB significa efetivamente atualizar as contagens de classe e palavra-classe com base no documento que entra. Em particular, o número de documentos pertencentes à classe c é incrementado, ou seja, $N_{c+} = 1$. Da mesma forma, as contagens de classe-palavra para cada palavra $w_i \in d$ são atualizadas, ou seja, $N_{ic+} = f_i^d$, onde f_i^d é a frequência de w_i em d . Para palavras existentes no modelo, isso implica apenas em uma atualização de suas contagens. Para palavras até então desconhecidas, isso implica que uma nova entrada é criada para elas no modelo. Essas contagens cumulativas são usadas durante a previsão de polaridade nas Equações 3.2 e 3.3. Tal modelo, pode ser referido como *accumulativeMNB* (WAGNER et al., 2015).

3.3.2 Random Forest

Random Forests (para o português, florestas aleatórias) foi um algoritmo criado por Breiman em 2001 baseado no artigo de 1996 de Freund e Schapire que abordava o uso de algoritmos de *boosting* (FREUND; SCHAPIRE et al., 1996 apud BREIMAN, 2001). *Random Forests* (RF) são uma combinação de preditores de árvore de decisão (do inglês, Decision Tree), de modo que cada árvore depende dos valores de um vetor aleatório amostrado de forma independente e com a mesma distribuição para todas as árvores na floresta. O erro de generalização para florestas converge quase certamente para um limite à medida que o

número de árvores na floresta se torna grande. O erro de generalização de uma floresta de classificadores de árvores depende da força das árvores individuais na floresta e da correlação entre elas. O uso de uma seleção aleatória de características para dividir cada nó produz taxas de erro comparáveis, mas é mais robusto em relação ao ruído (BREIMAN, 2001).

A floresta aleatória, desenvolvida por Breiman, utiliza a randomização para criar uma grande quantidade de árvores de decisão. A saída dessas árvores é agregada em uma única saída usando votação para problemas de classificação ou média para problemas de regressão. A randomização é implementada de duas formas. Primeiramente, o conjunto de dados é amostrado com reposição (amostragem *bootstrap*). O processo de agregar uma nova amostra dessa forma é chamado de “agregação *bootstrap*” ou “*bagging*”. Por exemplo, se houver 6 sujeitos em um estudo, 6 lançamentos de dados podem ser usados para escolher a nova amostra randomizada. Não há garantia de que todos os sujeitos aparecerão na nova amostra e alguns podem aparecer mais de uma vez. (RIGATTI, 2017).

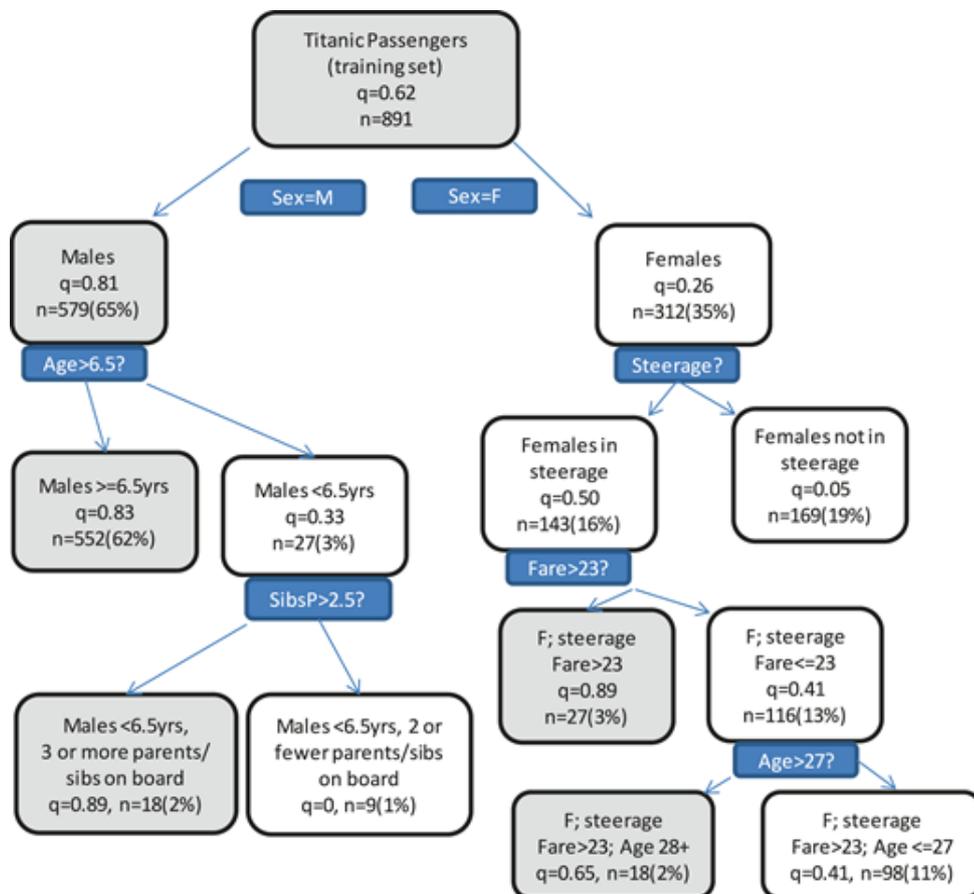
Em uma grande coleção de dados em que existe n sujeitos, a probabilidade de um ser deixado de fora de uma amostra *bootstrap* de tamanho n converge para $1/e$ ou cerca de 37%. Esses sujeitos deixados de fora ou “*out-of-bag*” compõem um conjunto útil de dados para testar a árvore de decisão desenvolvida a partir dos sujeitos na amostra. A segunda randomização ocorre nos nós de decisão. Em cada nó, um certo número dos preditores é escolhido. Para um conjunto com p preditores, um número típico é a raiz quadrada arredondada de p . O algoritmo então testa todos os limites possíveis para todas as variáveis selecionadas e escolhe a combinação variável-limiar que resulta na melhor divisão, por exemplo, a divisão que separa mais efetivamente os casos dos controles. Essa seleção aleatória de variáveis e teste de limites continua até que nós “puros” sejam alcançados (contendo apenas casos ou controles), ou algum outro ponto final predefinido. Todo esse processo de crescimento da árvore é repetido (seja 100, ou 1000 vezes) para crescer a floresta aleatória (RIGATTI, 2017).

Uma forma de entender as florestas aleatórias e pelo seguinte exemplo: em abril de 1912, o RMS Titanic, se chocou contra um iceberg e afundou na costa de Newfoundland. Ele transportava 2244 tripulantes, 1500 deles desapareceram. O intuito deste exemplo de análise de dados é desenvolver um algoritmo de previsão para classificar corretamente os sobreviventes daqueles perdidos no mar com base nas informações disponíveis no momento da partida. Os dados incluem um indicador de resultado (sobreviveu vs. perdido), a classe do passageiro (1, 2 ou 3, onde 3 = classe econômica), o ponto de embarque (S = Southampton, C = Cherbourg e Q = Queenstown), idade, sexo, preço do bilhete, o número total de irmãos/cônjuges do passageiro a bordo, o número total de filhos e pais do passageiro a bordo e o número da cabine (RIGATTI, 2017).

Inicialmente, é considerado que todas as mulheres sobrevivem e todos os homens não (“mulheres e crianças primeiro”). Isso resulta em uma precisão de previsão de 81%. Em seguida, é ajustado um modelo de regressão logística passo a passo regressivo. O resultado é um modelo com termos para idade, sexo, classe do passageiro, ponto de embarque e a variável de irmãos/cônjuges. Uma previsão usando este modelo é ligeiramente pior do que o modelo “todas as mulheres sobrevivem”, com uma precisão de previsão de 80%. Em seguida,

é ajustada uma única árvore de classificação. Isso utiliza idade, sexo, classe do passageiro, tarifa do bilhete e a variável de irmãos/cônjuges e assim melhora a precisão para 84%. Finalmente, é ajustada uma floresta aleatória usando 500 árvores, tentando 3 variáveis em cada nó e testando um máximo de 5 limites para cada variável. Isso melhora a precisão da previsão para 89%. O teste de importância das variáveis mostra que o sexo é a variável mais importante, seguido pela idade, classe do passageiro, tarifa do bilhete, pais/filhos a bordo, irmãos/cônjuges a bordo e o ponto de embarque. A implicação aqui é que, para uma tarefa de classificação, um grande grupo de árvores de decisão randomizadas pode superar uma única árvore “melhor” (RIGATTI, 2017).

Figura 6 – Ilustração do exemplo de análise utilizando *Random Forest*.



Fonte: (RIGATTI, 2017)

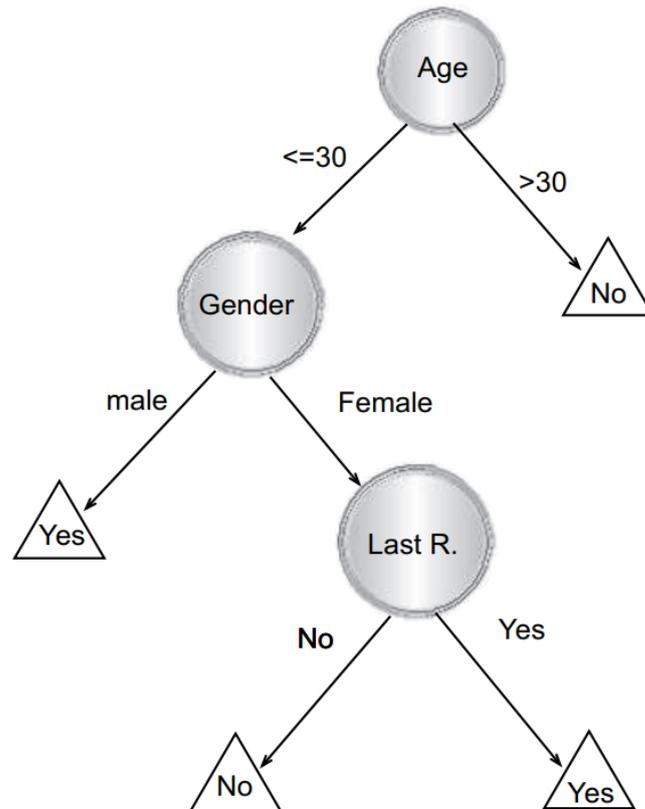
No exemplo acima, para cada ponto de decisão ou nó, a ramificação da esquerda representa um “Sim” e da direita representa um “Não” para a verificação feita em cada nó. As caixas sombreadas representam uma chance de mortalidade superior a 50% para o grupo das referidas caixas e dessa forma, nesse modelo, seria feito a predição de morte para esse grupo.

3.3.3 Decision Tree

O algoritmo de classificação de árvore de decisão (do inglês, *decision tree*) (DT) é caracterizado pelo fato de classificar uma amostra desconhecida usando uma ou várias funções de decisão de maneira sucessiva. Por motivos de processamento, a árvore é codificada como um conjunto de caracteres de forma que exista um relacionamento único entre os caracteres e a DT. Com isso, esses caracteres são decodificados pelo computador e ponteiros são definidos para definir o caminho de classificação apropriado para cada amostra (SWAIN; HAUSKA, 1977).

De forma geral, uma DT consiste de um nó raiz e vários nós internos e terminais. Os nós raiz e internos podem ser classificados no geral como nós não terminais, já que estão ligados em níveis de decisão e assim os nós terminais representam o final de uma classificação. Associado com o nó raiz está o conjunto inteiro de classes que uma amostra pode ser classificada. Os nós que estão a mesma “distância” do nó raiz compõem uma camada da árvore. Cada nó consiste em um conjunto de recursos a serem utilizados, um conjunto de classes a serem discriminadas e a regra de decisão para executar a classificação da forma correta (SWAIN; HAUSKA, 1977).

Na figura 7, tem-se uma *decision tree*, onde os nós são representados por círculos e as folhas (nós terminais) são representadas por triângulos. Cada nó representa uma característica, como, por exemplo, “Age”, podendo assumir valores numéricos e nominais. Em cada nó é realizado um teste, como, por exemplo, se “Age” é maior que 30. Após passar por todas as camadas da árvore e alcançar uma folha, é realizada a rotulação da entrada na árvore acima, em *No* ou *Yes* (ROKACH; MAIMON, 2005).

Figura 7 – Ilustração simples de uma *decision tree*.

Fonte: (ROKACH; MAIMON, 2005)

Os algoritmos de DT seleciona um atributo de particionamento e divide ele em subconjuntos a fim de criar um ramo para cada subconjunto. A cada novo subconjunto é realizada uma nova divisão e assim sucessivamente (GARCIA, 2003). De forma geral um algoritmo de árvore de decisão pode ser descrito pelo algoritmo abaixo em que S contém todos os exemplos de treino:

1. Se o atual conjunto de exemplos de S satisfazem um critério de parada, vai para o passo 2, se não, segue para o passo 3.
2. Após a criação de um nodo folha com algum nome classe é realizado a parada.
3. É realizado a seleção de algum atributo A que será utilizado como atributo de particionamento e após é criado um nodo com o nome do atributo de particionamento.
4. É escolhido algum teste sobre os valores do atributo A selecionado, com resultados coletivamente exaustivos R_1, \dots, R_k e mutualmente exclusivos, assim, é criado um ramo, a partir do nodo recém-criado, para cada teste.
5. É realizado o particionamento de S nos subconjuntos S_1, \dots, S_k , em que cada S_i e $i = 1, \dots, k$ contenha cada um dos exemplos em S com o resultado do teste R_i .
6. Assim, aplica-se este algoritmo recursivamente para cada conjunto S_i e $i = 1, \dots, k$.

Fonte: (GARCIA, 2003 apud FREITAS, 1998)

3.4 Undersampling

O termo *undersampling* é usado para se referir ao processo de reduzir a quantidade de amostras em um dado conjunto X_{maj} . Esse processo pode ser categorizado em 2 grupos, *fixed undersampling* e *cleaning undersampling*. O grupo de *fixed undersampling* refere-se aos métodos que utilizam o *undersampling* para alcançar uma proporção de equilíbrio apropriada $r_{X_{res}}$. Já o grupo de *cleaning undersampling* refere-se aos métodos que limpam o espaço das *features* baseado em algum critério empírico (LEMAÎTRE; NOGUEIRA; ARIDAS, 2017).

O *undersampling* é uma forma eficiente de lidar com um aprendizado onde há um desequilíbrio nas amostras. Esse método utiliza um subconjunto das classes majoritárias para treinar o classificador, resultando na omissão de alguns exemplos das classes majoritárias e, assim, tornando o conjunto de treinamento mais balanceado e o processo de aprendizado mais rápido. O principal ponto negativo é que amostras com informações úteis podem ser negligenciadas (GANGANWAR, 2012).

3.4.1 Random undersampling

O mais comum e melhor algoritmo simples para o *undersampling* é o random undersampling (MOHAMMED; RAWASHDEH; ABDULLAH, 2020 apud ESTABROOKS; JAPKOWICZ, 2001). O random undersampling é um algoritmo não heurístico que busca equilibrar as distribuições alvo (amostras) eliminando aleatoriamente instâncias da classe majoritária. Por meio dessa operação é possível balancear a base de dados, porém pode remover dados possivelmente valiosos que poderiam ser essenciais para modelos de classificação, mas é útil ao ter uma base de dados extensa.

3.5 Métricas

No contexto da classificação de textos, a avaliação desempenha um papel crucial, pois permite determinar a qualidade do classificador em análise (SOUZA, 2019). Nessa seção, será discutidas algumas das métricas utilizadas para avaliar um classificador.

3.5.1 Acurácia

Acurácia pode ser definido como a razão entre a quantidade de acertos de verdadeiros positivos e verdadeiros negativos pela quantidade total de predições (MENESES, 2021). Matematicamente, a acurácia é calculada como:

$$\text{Acurácia} = \frac{\text{Verdadeiros positivos} + \text{Verdadeiros negativos}}{\text{Total de predições}} \quad (3.4)$$

3.5.2 Precisão

A precisão representa a porcentagem de predições corretas entre o total de predições classificadas como positivas (SHALEV-SHWARTZ; BEN-DAVID, 2014). Também pode ser entendida como uma medida de relevância do resultado, ou seja, a proporção de resultados retornados que são relevantes (SOUZA, 2019). A fórmula para calcular a precisão é:

$$\text{Precisão} = \frac{\text{Verdadeiros positivos}}{\text{Verdadeiros positivos} + \text{Falsos positivos}} \quad (3.5)$$

3.5.3 Sensibilidade

Sensibilidade, também conhecida como *recall*, representa a proporção de predições corretas da classe positiva em relação ao total de positivos reais (MENESES, 2021). É calculado da seguinte forma:

$$\text{Sensibilidade} = \frac{\text{Verdadeiros positivos}}{\text{Verdadeiros positivos} + \text{Falsos negativos}} \quad (3.6)$$

3.5.4 Especificidade

Especificidade representa a proporção de predições corretas da classe negativa em relação ao total de negativos reais (MENESES, 2021). A fórmula para o cálculo da especificidade é:

$$\text{Especificidade} = \frac{\text{Verdadeiros negativos}}{\text{Verdadeiros negativos} + \text{Falsos positivos}} \quad (3.7)$$

3.5.5 F1-Score

O F1-Score é dado a partir da média ponderada entre as métricas de sensibilidade e precisão (SOUZA, 2019). Matematicamente o F1-Score é dado por:

$$\text{F1-Score} = \frac{2 \times \text{Precisão} \times \text{Sensibilidade}}{\text{Precisão} + \text{Sensibilidade}} \quad (3.8)$$

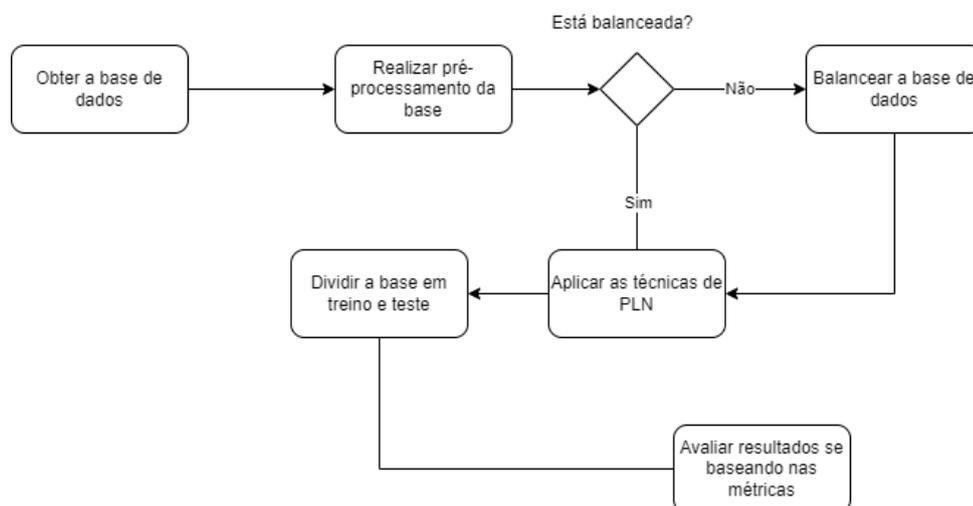
4 Materiais e Métodos

O presente trabalho caracteriza-se como uma pesquisa exploratória, conforme definido por (POLIT; BECK, 2004), pois parte de um fenômeno de interesse e busca compreender sua natureza completa, incluindo suas manifestações e os fatores que influenciam seus resultados. Além disso, trata-se de um estudo de caso, uma vez que foca em um contexto específico, analisando um único objeto ou um pequeno grupo de objetos de estudo. Esse tipo de abordagem visa compreender as circunstâncias do fenômeno investigado e, geralmente, é conduzido por meio de métodos qualitativos.

Neste estudo, foi utilizado Processamento de Linguagem Natural para classificar as *reviews* de jogos disponíveis na plataforma Steam como positivas ou negativas. Essa classificação foi realizada com base em uma revisão detalhada da literatura, a qual forneceu conhecimento sobre as métricas de avaliação e as técnicas utilizadas no desenvolvimento do trabalho.

A Figura 8 apresenta o esquema seguido ao longo deste estudo. Cada caixa de texto representa uma etapa do processo, as setas indicam a direção do fluxo e os quadrados representam as ações realizadas durante o desenvolvimento do projeto.

Figura 8 – Fluxograma de desenvolvimento do trabalho.



Fonte: elaborado pelo autor

4.1 Base de Dados

A base de dados utilizada foi obtida da plataforma pública Kaggle. A escolha dessa base foi feita considerando que as amostras já estavam rotuladas e devidamente classificadas em relação à língua de origem dos *reviews*, além de apresentarem uma variedade de amostras. Outro fator importante para a seleção dessa base foi sua disponibilidade pública. Para

este trabalho, foram utilizados apenas os *reviews* em português brasileiro.

A base de dados continha 23 atributos e 21.747.371 avaliações. Dentre esses atributos, 16 eram numéricos, 3 eram textuais e 4 eram booleanos. Entre os atributos textuais, um deles era o conteúdo da avaliação, e entre os booleanos, um indicava a classificação da avaliação, com *True* representando avaliações positivas e *False* representando avaliações negativas.

4.2 Pré-processamento da Base de Dados

Na sequência, foi realizado o pré-processamento da base de dados. Nesta etapa, foi avaliada a necessidade de remover os *reviews* muito curtos para evitar ruídos nos classificadores (ZUO, 2018). Além disso, considerou-se a aplicação de técnicas como *stemming*, tokenização e remoção de *stopwords* para preparar a base de dados adequadamente para os classificadores. Finalmente, após essa avaliação, foram removidos os *reviews* com conteúdo textual vazio e realizada a tokenização da própria classe *CountVectorizer*.

Os processos descritos acima, incluindo a separação da base de dados em conjuntos de teste e treino, foram implementados utilizando um algoritmo desenvolvido em Python. Considerando a extensão da base de dados e as necessidades de hardware, foi possível utilizar a totalidade das 93.746 amostras restantes após a subamostragem e balanceamento.

4.3 Escolha dos Algoritmos Classificadores

Com base na revisão bibliográfica e na análise de trabalhos com objetivos semelhantes, foram selecionados os algoritmos classificadores que melhor atenderam às necessidades da aplicação. A escolha foi baseada em uma análise comparativa, considerando as métricas (mencionadas na seção 3.5), com especial atenção para a precisão, que apresentou melhores resultados nos trabalhos correlatos estudados.

Após a seleção dos algoritmos, foi realizado um estudo técnico aprofundado para compreender suas funcionalidades e características específicas. Isso permitiu utilizar os algoritmos com maior habilidade e aproveitamento.

4.4 Treinamento da Base de Dados

Utilizando os algoritmos selecionados na etapa anterior, foi realizado o treinamento da base de dados. O treinamento foi feito com um algoritmo desenvolvido em Python, que recebeu como entrada os dados processados na seção 4.2. Esse algoritmo utilizou os classificadores escolhidos para realizar a predição dos resultados, sejam eles positivos ou negativos.

4.5 Avaliação dos Resultados

Após o treinamento da base de dados, foi realizada a avaliação dos resultados dos modelos treinados. Para isso, foi utilizada a parte da base de dados reservada para testes

e as seguintes métricas de avaliação: acurácia, precisão, sensibilidade, especificidade e F1-Score. Essa etapa possibilitou a análise e aferição da qualidade das previsões efetuadas pelos modelos na base de testes.

5 Metodologia Experimental

Este capítulo descreve os procedimentos realizados na análise de sentimentos, iniciando com a leitura e análise da base de dados obtida da plataforma *Kaggle*, utilizando *Python* e a biblioteca *Pandas* para avaliar sua estrutura e conteúdo. A base de dados foi analisada para identificar as colunas relevantes, como a linguagem das avaliações e o conteúdo textual. O tratamento dos dados envolveu a filtragem para manter apenas avaliações em português e a remoção de amostras com valores *NaN*, resultando em um conjunto de dados mais limpo e adequado para análise. Para enfrentar a desproporção entre avaliações positivas e negativas, foi aplicada a técnica de *undersampling*.

Na seção de classificação, os dados foram transformados para Processamento de Linguagem Natural (PLN) usando a classe *CountVectorizer* e a matriz *csr_matrix*. Os dados foram então divididos em conjuntos de treinamento e teste, e os modelos de *Decision Tree*, *Random Forest* e *Multinomial Naïve Bayes* foram treinados com validação cruzada k-fold (k=10). Esta técnica permite avaliar o desempenho dos modelos de forma robusta, e os resultados são analisados com base nas métricas descritas na seção 3.5.

5.1 Leitura e análise da base de dados

Após a obtenção do arquivo em formato de planilha com extensão “.csv”, através da plataforma *Kaggle*, o próximo passo foi realizar a leitura dos dados com o intuito de visualizar e avaliar a estrutura dos mesmos. Para isso, foi utilizada a linguagem de programação *Python* e a biblioteca *Pandas*¹ para fazer a leitura dos dados.

Figura 9 – Visualização da base de dados

Unnamed: 0	app_id	app_name	review_id	language	review	timestamp_created	timestamp_updated	recommended	
0	0	292030	The Witcher 3: Wild Hunt	85185598	schinese	不玩此生遗憾, RPG游戏里的天花板, 太吸引人了	1611381629	1611381629	True
1	1	292030	The Witcher 3: Wild Hunt	85185250	schinese	拔DIAO无情打桩机-杰洛特!!!	1611381030	1611381030	True
2	2	292030	The Witcher 3: Wild Hunt	85185111	schinese	巫师3NB	1611380800	1611380800	True

Fonte: elaborada pelo autor

A base de dados obtida estava distribuída em 21.747.371 milhões de linhas de avaliações, onde cada linha tinha 23 colunas. Através de cada linha de avaliação, era possível

¹ *Pandas* é uma biblioteca de código aberto usada para realizar análise de dados e assim dar à linguagem *Python* a capacidade de manipular, de forma livre e rápida, planilhas (CHEN, 2018).

observar a linguagem em que a avaliação foi escrita, como pode ser observado na coluna *language*. Na coluna *app_name*, é possível observar sobre qual jogo a avaliação foi feita. Em outra coluna, a *review*, é possível observar o conteúdo textual da avaliação. E, por fim, na coluna *recommended*, os rótulos das avaliações são indicados, onde o valor *True* corresponde a uma avaliação positiva e o valor *False* corresponde a uma avaliação negativa.

5.2 Tratamento e balanceamento da base de dados

Após a leitura e análise da base de dados, foi possível inferir que ela continha avaliações em diversas linguagens, como inglês e chinês. Como o objetivo do trabalho era desenvolver uma análise de sentimentos em português brasileiro, foi necessário realizar uma filtragem e considerar apenas as amostras nessa língua. Isso resultou em uma base contendo 837.524 amostras.

Em seguida à filtragem, observou-se que em algumas amostras, a coluna de *review*, mencionada anteriormente, estava preenchida com valores *NaN*, o que impossibilitou a utilização dessas amostras na análise. Tendo isso em mente, foi necessário remover essas amostras consideradas ruídos, resultando em uma base contendo 836.888 amostras.

Figura 10 – Amostra com a coluna *review* tendo o valor *NaN*.

Unnamed: 0	app_id	app_name	review_id	language	review
29	29	The Witcher 3: Wild Hunt	85177505	brazilian	Se um dia alguém falar que esse jogo é ruim na...
32	32	The Witcher 3: Wild Hunt	85176661	brazilian	NaN
34	34	The Witcher 3: Wild Hunt	85176249	brazilian	Obra prima!!!

Fonte: elaborada pelo autor

Posteriormente, observou-se que a base de dados estava extremamente desbalanceada, com 790.015 amostras classificadas como positivas e 46.873 como negativas. Essa desproporção entre as amostras prejudica a qualidade da análise. Assim, utilizando uma das técnicas de *undersampling*, através da classe *RandomUnderSampler* da biblioteca *imblearn*, realizou-se a subamostragem da base, resultando em uma base de dados final contendo 93.746 amostras.

5.3 Classificação

Por meio da classe *CountVectorizer* da biblioteca *scikit-learn*, é produzida uma representação esparsa das contagens utilizando a matriz *csr_matrix* da biblioteca *scipy.sparse* (SCIKIT-LEARN, 2024). Isso permite a transformação das avaliações para possibilitar o Processamento de Linguagem Natural (PLN). Com o auxílio do Pandas, os dados foram separados em conjunto de dados (x) e rótulos (y). Posteriormente, o conjunto de dados foi dividido em conjuntos de treinamento e teste.

Em seguida, utilizando a biblioteca *scikit-learn*, foram criados os classificadores *Decision Tree*, *Random Forest* e *Multinomial Naïve Bayes*. Estes foram treinados utilizando a técnica de validação cruzada k -fold, com $k=10$. Essa técnica permite a divisão dos dados em k -folds consecutivos para treinamento e teste, sem embaralhar os dados. Assim, a base de dados foi dividida em 10 partes, sendo 9 partes utilizadas para treinamento e 1 parte para teste. Esse processo foi repetido 10 vezes e, ao final, foi calculada a média dos resultados das 10 iterações.

6 Resultados

Os resultados da análise de sentimentos, realizada sobre as avaliações obtidas da base de dados disponível na plataforma *Kaggle* e processadas conforme os procedimentos descritos no capítulo 4, foram analisados e comparados. Esta análise foi conduzida utilizando os classificadores *Decision Tree*, *Random Forest* e *Multinomial Naïve Bayes*. A comparação dos desempenhos dos modelos foi fundamentada nas métricas apresentadas na seção 3.5. A execução foi realizada dividindo a base de dados balanceada em 10 *fold*s, ou seja, $k = 10$. No final da execução, são apresentadas as médias dos resultados das 10 execuções, nas quais o Random Forest (RF) teve uma acurácia média de 86,55%, o Multinomial Naïve Bayes (MNB) 87,31% e o Decision Tree (DT) 81,61%.

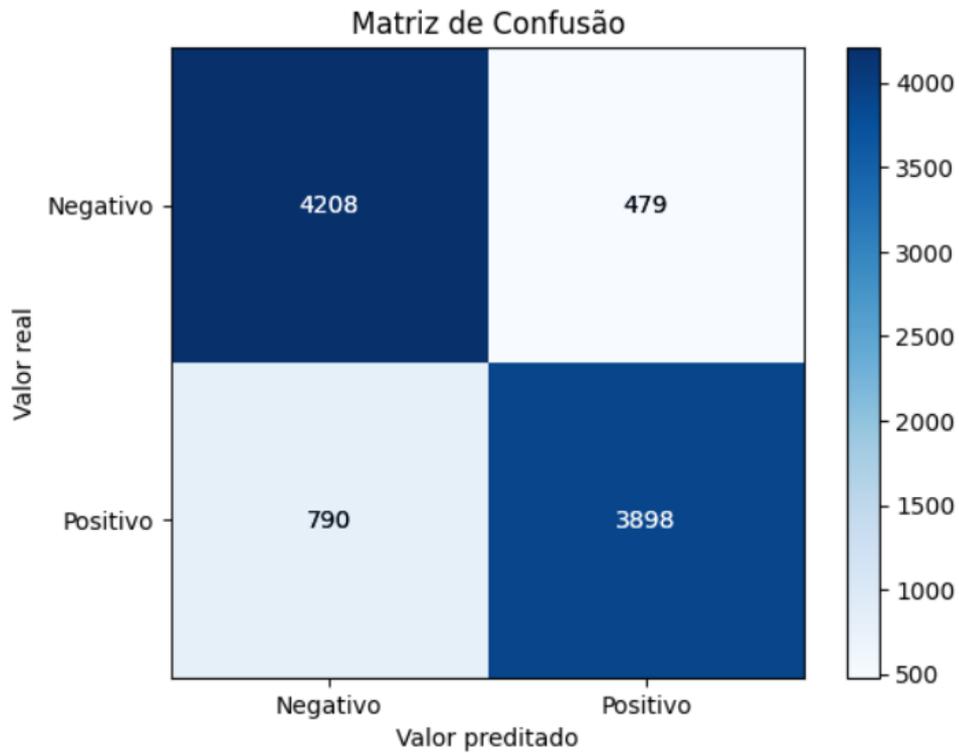
6.1 Comentários positivos

Tabela 1 – Décimo *fold* para os comentários positivos (3.3).

10º K-fold: Comentários positivos						
	Sensibilidade	Especificidade	Precisão	F1-score	VP	FN
RF	83,15%	89,78%	89,06%	86%	3898	790
MNB	86,71%	87,35%	87,27%	86,99%	4065	623
DT	82,76%	80,31%	80,78%	81,76%	3880	808

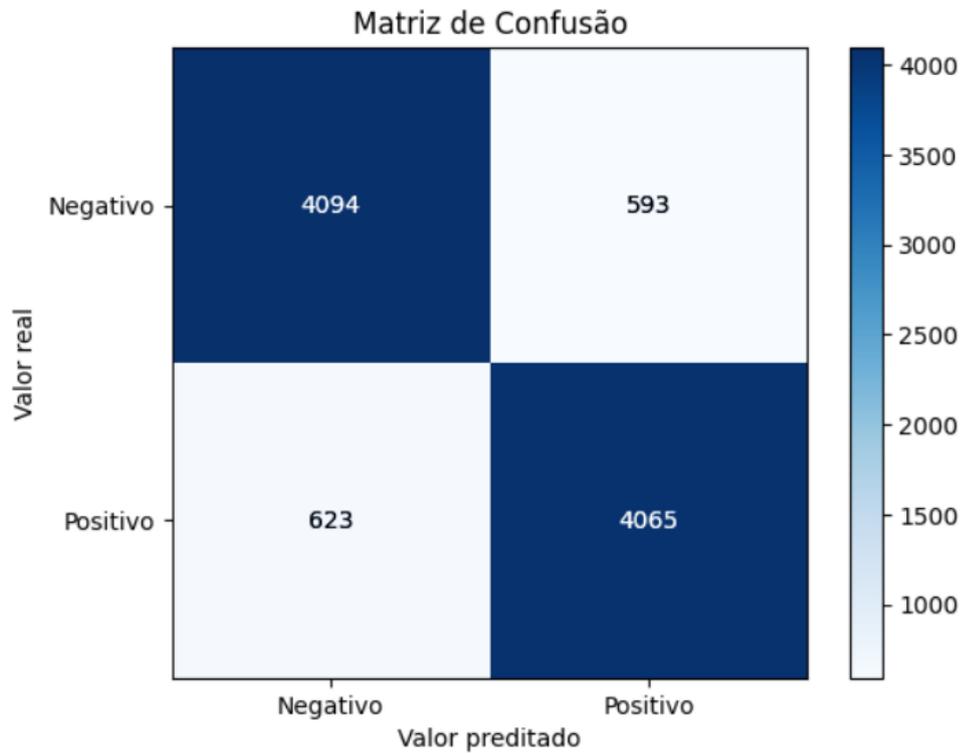
Fonte: elaborada pelo autor

Na tabela 1 é possível observar os resultados do cálculo das métricas para os comentários positivos no décimo *fold* da execução. Nessa execução, o RF atingiu 83,15% no cálculo da sensibilidade, 89,78% no cálculo da especificidade, 89,06% no cálculo da precisão e 86% no cálculo do F1-score. Como também é possível notar, para esse *fold*, o RF classificou 3898 avaliações como verdadeiros positivos (VP) e 790 como falsos negativos (FN), ou seja, classificou corretamente 3898 avaliações como positivas e errou 790 ao classificá-las como negativas. Esses resultados estão refletidos na matriz de confusão abaixo na figura 11, correspondente à décima execução do RF, na qual o valor previsto representa a classificação feita pelo classificador e o valor real indica a classificação feita pelos usuários da plataforma Steam.

Figura 11 – Matriz de confusão para o décimo *fold* do RF.

Fonte: elaborada pelo autor

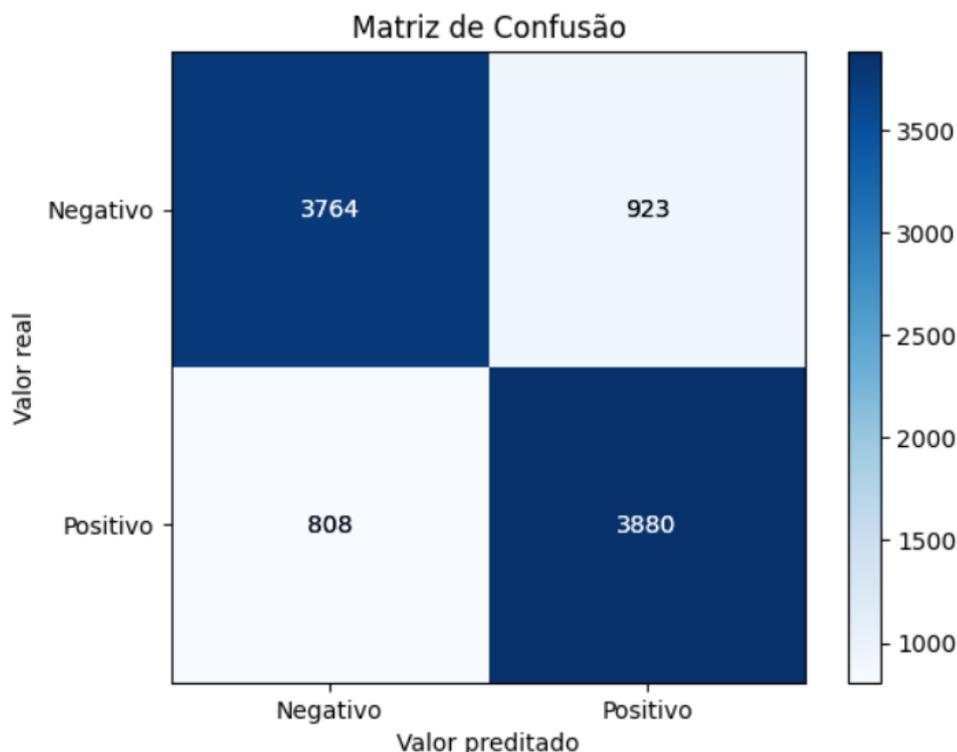
Ainda na tabela 1 é possível perceber que o MNB encontrou os seguintes valores: 86,71% para a sensibilidade, 87,35% para a especificidade, 87,27% para a precisão e 86,99% para o F1-score. No que diz respeito ao número de avaliações, 4065 foram classificadas corretamente, enquanto 623 foram classificadas incorretamente. Esses valores podem ser observados na matriz de confusão abaixo, referente à execução do décimo *fold* para o MNB:

Figura 12 – Matriz de confusão para o décimo *fold* do MNB.

Fonte: elaborada pelo autor

Ainda na tabela 1, o DT encontrou valores de 82,76% para a sensibilidade, 80,31% para a especificidade, 80,78% para a precisão e 81,76% para o F1-Score. Quanto aos números referentes à quantidade de avaliações, 3880 foram classificadas corretamente e 808 foram classificadas incorretamente. Abaixo está a matriz de confusão referente ao décimo *fold*:

Figura 13 – Matriz de confusão para o décimo *fold* do Decision Tree.



Fonte: elaborada pelo autor

Na tabela 2, é possível observar os resultados do cálculo das métricas para os comentários positivos, onde o RF atingiu 82,98% na sensibilidade, 90,12% na especificidade, 89,36% na precisão e 86,06% no F1-score. De maneira semelhante, o MNB obteve 87,56% na sensibilidade, 87,07% na especificidade, 87,13% na precisão e 87,34% no F1-score. Ainda na mesma imagem, o DT apresentou 82,60% na sensibilidade, 80,65% na especificidade, 81,02% na precisão e 81,80% no F1-score.

Tabela 2 – Média dos resultados das execuções para os classificadores citados na seção 3.3.

Média: Comentários positivos				
	Sensibilidade	Especificidade	Precisão	F1-score
RF	82,98%	90,12%	89,36%	86,06%
MNB	87,56%	87,07%	87,13%	87,34%
DT	82,60%	80,65%	81,02%	81,80%

Fonte: elaborada pelo autor.

6.2 Comentários negativos

Tabela 3 – Décimo *fold* para os comentários negativos (3.3).

10º K-fold: Comentários negativos						
	Sensibilidade	Especificidade	Precisão	F1-score	VP	FN
RF	89,78%	83,15%	84,19%	86,90%	4208	479
MNB	87,35%	86,71%	86,79%	87,07%	4094	593
DT	80,31%	72,76%	82,33%	81,30%	3764	923

Fonte: elaborada pelo autor

Na tabela 3 estão presentes os resultados do cálculo das métricas para os comentários negativos no décimo *fold* da execução. Nessa execução, o RF atingiu 89,78% no cálculo da sensibilidade, 83,15% no cálculo da especificidade, 84,19% no cálculo da precisão e 86,90% no cálculo do F1-score. Também é possível inferir para esse *fold* o RF classificou 4208 avaliações como verdadeiros positivos (VP) e 790 como falsos negativos (FN), ou seja, classificou corretamente 4208 avaliações como negativas e errou 790 ao classificá-las como positivas. Os valores citados anteriormente podem ser observados na matriz de confusão da figura 11.

Continuando a análise da tabela 3 é possível inferir que o MNB encontrou os seguintes valores: 87,35% para a sensibilidade, 86,71% para a especificidade, 86,79% para a precisão e 87,07% para o F1-score. No que diz respeito ao número de avaliações, 4094 foram classificadas corretamente, enquanto 593 foram classificadas incorretamente. Os valores mencionados acima podem ser conferidos na figura 12.

Da mesma forma, segundo a tabela 3, o DT encontrou valores de 80,31% para a sensibilidade, 72,76% para a especificidade, 82,33% para a precisão e 81,30% para o F1-Score. Quanto aos números referentes à quantidade de avaliações, 3764 foram classificadas corretamente e 923 foram classificadas incorretamente. Esses valores podem ser observados na figura 13.

De maneira semelhante à seção anterior, na tabela 4 é possível encontrar os resultados do cálculo das métricas para os comentários negativos, onde o RF atingiu 90,12% na sensibilidade, 82,98% na especificidade, 84,12% na precisão e 87,02% no F1-score. De forma análoga, o MNB obteve 87,07% na sensibilidade, 87,56% na especificidade, 87,52% na precisão e 87,29% no F1-score. Ainda na mesma imagem, o DT apresentou 80,65% na sensibilidade, 82,60% na especificidade, 82,26% na precisão e 81,45% no F1-score.

Tabela 4 – Média dos resultados das execuções para os classificadores citados na seção 3.3.

Média: Comentários negativos				
	Sensibilidade	Especificidade	Precisão	F1-score
RF	90,12%	82,98%	84,12%	87,02%
MNB	87,07%	87,56%	87,52%	87,29%
DT	80,65%	82,60%	82,26%	81,45%

Fonte: elaborada pelo autor.

6.3 Analisando os resultados

Os resultados apresentados ao longo desse capítulo mostraram que, entre os classificadores utilizados, o *Multinomial Naïve Bayes* (MNB) se destacou com a maior precisão média de 87,52% na classificação de comentários negativos. Este desempenho foi superior ao do *Random Forest* (RF), que obteve uma precisão média de 84,12%, e ao do *Decision Tree* (DT), que apresentou a menor precisão média entre os três, com 82,26%. A precisão na identificação de comentários negativos é particularmente importante, pois esses comentários podem influenciar negativamente a percepção do produto e, conseqüentemente, impactar as vendas.

Embora o *Random Forest* tenha mostrado uma sensibilidade superior, com uma média de 90,12%, as métricas de acurácia, especificidade e F1-score confirmaram a superioridade do MNB na classificação de avaliações negativas. Estes resultados ressaltam que o *Multinomial Naïve Bayes* é especialmente eficaz para tarefas de análise de sentimentos em dados textuais, especialmente quando se trata de identificar *feedbacks* negativos que podem afetar o desempenho comercial do produto.

7 Considerações finais

O presente trabalho teve como objetivo explorar e comparar a eficácia de três classificadores — *Decision Tree*, *Random Forest* e *Multinomial Naïve Bayes* — na tarefa de análise de sentimentos aplicada a avaliações de jogos disponíveis na plataforma Steam, utilizando um conjunto de dados extraído do Kaggle. Para alcançar esse objetivo, foi empregada a técnica de validação cruzada k-fold, com k=10, nas 93.746 amostras, o que se mostrou fundamental para assegurar uma avaliação mais precisa e robusta dos modelos, evitando viés e garantindo que cada parte do conjunto de dados fosse utilizada corretamente ao longo do desenvolvimento do trabalho.

Os resultados apresentados no capítulo 6 demonstraram que o *Multinomial Naïve Bayes* (MNB) apresentou o melhor desempenho na classificação de comentários negativos, com uma precisão média de 87,52%, superando o *Random Forest* (84,12%) e o *Decision Tree* (82,26%). Apesar da maior sensibilidade do *Random Forest*, as métricas de acurácia, especificidade e F1-score confirmaram a superioridade do MNB. Esses achados destacam sua eficácia na análise de sentimentos em dados textuais, especialmente na identificação de *feedbacks* negativos, que podem impactar a percepção e o desempenho comercial do produto.

A experiência de conduzir este trabalho foi altamente enriquecedora, ao permitir um aprofundamento significativo nas técnicas de aprendizado de máquina aplicadas ao processamento de linguagem natural (PLN). Ao longo da pesquisa, foi possível observar que a fundamentação teórica estabelecida a partir da revisão de literatura foi essencial para a escolha dos classificadores e das técnicas de pré-processamento, evidenciando a importância de uma abordagem bem estruturada e baseada em evidências na aplicação de algoritmos de classificação.

Entretanto, ao longo do desenvolvimento do presente trabalho, foram encontrados diversos empecilhos. A manipulação de grandes volumes de dados representou uma dificuldade, uma vez que a base de dados utilizada era extensa e complexa, se fez necessário reduzir a amostragem devido às limitações no hardware utilizado. Da mesma forma, a necessidade de realizar subamostragem para lidar com o desbalanceamento dos dados foi uma das soluções adotadas, mas que também reduziu a diversidade de amostras e impôs limitações no processo.

Para futuras pesquisas, seria interessante expandir a análise para outros algoritmos de aprendizado de máquina, como redes neurais profundas, que têm mostrado grande potencial em tarefas de PLN e podem oferecer ainda mais precisão na análise de sentimentos. A inclusão de novas fontes de dados, como avaliações de outras plataformas de jogos, também poderia proporcionar uma comparação mais ampla e rica sobre as percepções dos usuários em diferentes contextos ou, ainda, acrescentar a implementação para ser possível a utilização de diferentes línguas, como o inglês.

Em resumo, ao desenvolver esse projeto, foi possível aprender e também contribuir com o campo da análise de sentimentos em jogos eletrônicos, destacando a relevância dos

algoritmos tradicionais de classificação e sugerindo novos caminhos a serem explorados em pesquisas futuras. Através da aplicação cuidadosa das técnicas de aprendizado de máquina e da avaliação rigorosa dos resultados, com a ajuda das métricas citadas na seção 3.5, foi possível obter *insights* valiosos sobre o desempenho dos classificadores e identificar oportunidades para avanços e melhorias nas metodologias empregadas.

Referências

- AYODELE, T. O. Types of machine learning algorithms. *New advances in machine learning*, InTech Rijeka, Croatia, v. 3, p. 19–48, 2010. Citado nas páginas 17 e 18.
- BARBOSA, J. et al. Introdução ao processamento de linguagem natural usando python. *III Escola Regional de Informatica do Piauí*, v. 1, p. 336–360, 2017. Citado na página 10.
- BARROS, J. R. Flávia de A. *Processamento de Linguagem Natural*. [s.n.], 1996. Disponível em: <<https://www.cin.ufpe.br/~fab/cursos/jai96/ProcessamentoDeLinguagemNatural.pdf>>. Citado nas páginas 14, 15 e 16.
- BIRD, S.; KLEIN, E.; LOPER, E. *Natural Language Processing with Python*. 1st. ed. [S.l.]: O'Reilly Media, Inc., 2009. ISBN 0596516495. Citado na página 10.
- BRASIL, P. G. *Pesquisa Game Brasil 9ª Edição Gratuita*. 2022. 76 p. Disponível em: <<https://materiais.pesquisagamebrasil.com.br/2022-painel-gratuito-pgb22>>. Citado na página 9.
- BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, p. 5–32, 2001. Citado nas páginas 19 e 20.
- CAMBRIA, E.; WHITE, B. Jumping nlp curves: A review of natural language processing research [review article]. *IEEE Computational Intelligence Magazine*, v. 9, n. 2, p. 48–57, 2014. Citado na página 17.
- CHEN, D. Y. *Análise de dados com Python e Pandas*. [S.l.]: Novatec Editora, 2018. Citado na página 29.
- CHOPRA, A.; PRASHAR, A.; SAIN, C. Natural language processing. *International journal of technology enhancements and emerging engineering research*, Citeseer, v. 1, n. 4, p. 131–134, 2013. Citado nas páginas 13, 14, 16 e 17.
- DEMIRCI, G. M.; KESKIN, S. R.; DOGAN, G. Sentiment analysis in turkish with deep learning. In: *2019 IEEE International Conference on Big Data (Big Data)*. [S.l.: s.n.], 2019. p. 2215–2221. Citado na página 12.
- EISENSTEIN, J. *Introduction to natural language processing*. [S.l.]: MIT press, 2019. Citado nas páginas 15 e 16.
- ESTABROOKS, A.; JAPKOWICZ, N. A mixture-of-experts framework for learning from imbalanced data sets. In: SPRINGER. *International Symposium on Intelligent Data Analysis*. [S.l.], 2001. p. 34–43. Citado na página 24.
- FONTANA, É. Introdução aos algoritmos de aprendizagem supervisionada. *Departamento de Engenharia Química, Universidade Federal do Paraná*, 2020. Citado na página 18.
- FREITAS, A. A. Generic, set-oriented primitives to support data-parallel knowledge discovery in relational database systems. 1998. Citado na página 24.
- FREUND, Y.; SCHAPIRE, R. E. et al. Experiments with a new boosting algorithm. In: CITESEER. *icml*. [S.l.], 1996. v. 96, p. 148–156. Citado na página 19.

- GANGANWAR, V. An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, v. 2, n. 4, p. 42–47, 2012. Citado na página 24.
- GARCIA, S. C. O uso de árvores de decisão na descoberta de conhecimento na área da saúde. 2003. Citado nas páginas 23 e 24.
- IZBICKI, R.; SANTOS, T. M. dos. *Aprendizado de máquina: uma abordagem estatística*. [S.l.: s.n.], 2020. ISBN 978-65-00-02410-4. Citado na página 17.
- JOYCE, J. Bayes' theorem. 2003. Citado na página 18.
- KUMAR, E. *Natural language processing*. [S.l.]: IK International Pvt Ltd, 2013. Citado nas páginas 14 e 15.
- LEMAÎTRE, G.; NOGUEIRA, F.; ARIDAS, C. K. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *The Journal of Machine Learning Research*, JMLR. org, v. 18, n. 1, p. 559–563, 2017. Citado na página 24.
- LI, X.; WU, C.; MAI, F. The effect of online reviews on product sales: A joint sentiment-topic analysis. *Information & Management*, Elsevier, v. 56, n. 2, p. 172–184, 2019. Citado na página 9.
- LIU, B. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, Morgan & Claypool Publishers, v. 5, n. 1, p. 1–167, 2012. Citado na página 9.
- MALTA, L. H. A.; KUROIWA, M. A. R. L. Aprendizado de máquina e processamento de linguagem natural aplicados à identificação de discurso de ódio. 2019. Citado nas páginas 11 e 12.
- MEJOVA, Y. Sentiment analysis: An overview. *University of Iowa, Computer Science Department*, 2009. Citado na página 9.
- MENESES, A. C. F. Predição de necessidade de uti hospitalar para pacientes internados utilizando métodos de aprendizado de máquina. 2021. Citado nas páginas 24 e 25.
- MICHIE, D.; SPIEGELHALTER, D.; TAYLOR, C. Machine learning, neural and statistical classification. p. 299, 02 1994. Citado na página 17.
- MOHAMMED, R.; RAWASHDEH, J.; ABDULLAH, M. Machine learning with oversampling and undersampling techniques: overview study and experimental results. In: IEEE. *2020 11th international conference on information and communication systems (ICICS)*. [S.l.], 2020. p. 243–248. Citado na página 24.
- MÜLLER, D. N. Compreensão da linguagem falada. *Porto Alegre: PPGC-UFRGS*, 2002. Citado na página 17.
- NAIR, A. J.; G, V.; VINAYAK, A. Comparative study of twitter sentiment on covid - 19 tweets. In: *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*. [S.l.: s.n.], 2021. p. 1773–1778. Citado nas páginas 11 e 12.
- POLIT, D. F.; BECK, C. T. *Nursing research: Principles and methods*. [S.l.]: Lippincott Williams & Wilkins, 2004. Citado na página 26.
- RIGATTI, S. J. Random forest. *Journal of Insurance Medicine*, American Academy of Insurance Medicine 1700 Magnavox Way, Fort Wayne, IN 46804, v. 47, n. 1, p. 31–39, 2017. Citado nas páginas 20 e 21.

- ROKACH, L.; MAIMON, O. Decision trees. *Data mining and knowledge discovery handbook*, Springer, p. 165–192, 2005. Citado nas páginas 22 e 23.
- ROSA, R. L. *Análise de sentimentos e afetividade de textos extraídos das redes sociais*. 2015. Tese (Doutorado) — Universidade de São Paulo, 2015. Citado na página 9.
- SCIKIT-LEARN. 2024. Disponível em: <<https://scikit-learn.org/>>. Citado na página 31.
- SHALEV-SHWARTZ, S.; BEN-DAVID, S. *Understanding Machine Learning: From Theory to Algorithms*. [S.l.]: Cambridge University Press, 2014. Citado na página 25.
- SHIEBER, S. The design of a computer language for linguistic information. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the tenth international conference on computational linguistics*. [S.l.], 1984. Citado na página 15.
- SOUZA, A. D. Mestrado acadêmico em ciência da computação. 2019. Citado nas páginas 24 e 25.
- STROPPA, T.; ROTHENBURG, W. C. Liberdade de expressão e discurso do Ódio: O conflito discursivo nas redes sociais. *Revista Eletrônica do Curso de Direito da UFSM*, Universidad Federal de Santa Maria, v. 10, 12 2015. ISSN 1981-3694. Citado na página 9.
- SWAIN, P. H.; HAUSKA, H. The decision tree classifier: Design and potential. *IEEE Transactions on Geoscience Electronics*, v. 15, n. 3, p. 142–147, 1977. Citado na página 22.
- Valve Corporation. *Counter-Strike: Global Offensive*. 2023. Steam. Acesso em: 20 maio 2023. Disponível em: <https://store.steampowered.com/app/730/CounterStrike_Global_Offensive/>. Citado na página 9.
- VIANA, B. de. *MERCADO DE JOGOS ELETRÔNICOS TOMOU IMPULSO NA PANDEMIA E CONTINUA EM ALTA*. 2022. Disponível em: <<https://www.insper.edu.br/noticias/mercado-de-jogos-eletronicos-tomou-impulso-na-pandemia-e-continua-em-alta/>>. Citado na página 9.
- VIEIRA, R.; LOPES, L. Processamento de linguagem natural e o tratamento computacional de linguagens científicas. *Em corpora*, p. 183, 2010. Citado na página 10.
- WAGNER, S. et al. Ageing-based multinomial naïve bayes classifiers over opinionated data streams. In: SPRINGER. *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part I 15*. [S.l.], 2015. p. 401–416. Citado nas páginas 18 e 19.
- WONGKAR, M.; ANGDRESEY, A. Sentiment analysis using naïve bayes algorithm of the data crawler: Twitter. In: *2019 Fourth International Conference on Informatics and Computing (ICIC)*. [S.l.: s.n.], 2019. p. 1–5. Citado nas páginas 11 e 12.
- ZHU, F.; ZHANG, X.; ZHANG, Z. *Association for Information Systems AIS Electronic Library (AISeL) The Influence of Online Consumer Reviews on the Demand for Experience Goods: The Case of Video Games Recommended CitationThe Influence of Online Consumer Reviews on the Demand for Experience Goods: The Case of Video Games*". 2006. 25 p. Disponível em: <<http://aisel.aisnet.org/icis2006/25>>. Citado na página 9.
- ZUO, Z. Sentiment analysis of steam review datasets using naïve bayes and decision tree classifier. 2018. Citado nas páginas 11, 12 e 27.