



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I - CAMPINA GRANDE
CENTRO DE CIÊNCIA E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

CARLOS EDUARDO HENRIQUE LACERDA

**UM MODELO DE APRENDIZAGEM DE MÁQUINA PARA ANÁLISE DE
SENTIMENTOS E USABILIDADE EM JOGOS ONLINE**

**CAMPINA GRANDE
2023**

CARLOS EDUARDO HENRIQUE LACERDA

**UM MODELO DE APRENDIZAGEM DE MÁQUINA PARA ANÁLISE DE
SENTIMENTOS E USABILIDADE EM JOGOS ONLINE**

Trabalho de Conclusão de Curso apresentado ao curso de Ciência da Computação da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Graduado em Ciência da Computação.

Área de concentração: Usabilidade e Fatores Humanos

Orientador: Prof. Dr. Daniel Scherer

CAMPINA GRANDE
2023

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

L131m Lacerda, Carlos Eduardo Henrique.
Um modelo de aprendizagem de máquina para análise de sentimentos e usabilidade em jogos *online* [manuscrito] / Carlos Eduardo Henrique Lacerda. - 2023.
61 p. : il. colorido.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2023.

"Orientação : Prof. Dr. Daniel Scherer, Coordenação do Curso de Computação - CCT. "

1. Aprendizagem de máquina. 2. Processamento de linguagem natural. 3. Usabilidade. 4. Jogos online. I. Título

21. ed. CDD 005.12

CARLOS EDUARDO HENRIQUE LACERDA

UM MODELO DE APRENDIZAGEM DE MÁQUINA PARA ANÁLISE DE
SENTIMENTOS E USABILIDADE EM JOGOS ONLINE

Trabalho de Conclusão de Curso apresentado
ao curso de Ciência da Computação da
Universidade Estadual da Paraíba, como
requisito parcial à obtenção do título de
Graduado em Ciência da Computação.

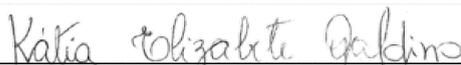
Área de concentração: Usabilidade e Fatores
Humanos

Aprovada em 28 de Junho de 2023.

BANCA EXAMINADORA



Prof. Dr. Daniel Scherer (CCT/UEPB)
Orientador (a)



Profa. Dra. Katia Elizabete Galdino (CCT/UEPB)
Examinador(a)



Profa. Dra. Kézia de Vasconcelos O. Dantas (CCT/UEPB)
Examinador(a)

RESUMO

O presente trabalho abordou uma discussão sobre o uso de um modelo de aprendizagem de máquina no campo de processamento de linguagem natural. Este modelo foi empregado no auxílio do processo de análise de alguns jogos através dos feedbacks dos jogadores, a fim de identificarmos as falhas de usabilidade presentes nos jogos escolhidos. Sendo de cunho bibliográfico e de pesquisa em sites especializados, esse projeto teve como objetivo principal desenvolver uma aplicação capaz de classificar o sentimento expresso pelos usuários em seus comentários, para que possamos analisar as avaliações negativas e identificar as falhas de usabilidade apontadas pelos jogadores. Para tanto, realizou-se um estudo inicial das técnicas envolvidas na construção de um modelo de aprendizagem de máquina, para que se fosse capaz de entender as diferenças entre os principais tipos de modelos: classificação, regressão e clusterização, e saber quais destes seriam aplicados no desenvolvimento desse projeto. Deste modo, considerar a experiência dos jogadores, ajudou a observar o controle da qualidade dos jogos estudados, uma vez que possibilitou a identificação das falhas que passaram despercebidas durante o processo de desenvolvimento destes jogos. Em nosso estudo, constatamos que os jogos escolhidos para análise apresentam erros semelhantes, o que pode estar associado à natureza desses jogos, uma vez que todos são voltados para o uso online. Além disso, verificamos que os usuários não se limitaram a reportar problemas de usabilidade em suas análises, mas sugeriram que os desenvolvedores incorporassem novas funcionalidades aos jogos como forma de solucionar erros e evitar outros possíveis problemas que pudessem surgir.

Palavras-chave: aprendizagem de máquina; processamento de linguagem natural; usabilidade; jogos online.

ABSTRACT

The present work addresses a discussion about the use of a machine learning model in the field of natural language processing. This model was used to help the process of analyzing some games through player feedback in order to identify the usability flaws present in the chosen games. Being of a bibliographic nature and requiring research on specialized sites, this project had as its main objective to develop an application capable of classifying the sentiment expressed by users in their comments so that we could analyze the negative reviews and identify the usability flaws pointed out by the players. To this end, an initial study of the techniques involved in the construction of a machine learning model was carried out so that it would be able to understand the differences between the main types of models: classification, regression, and clustering, and to know which of these would be applied in the development of this project. Thus, considering the experience of the players helped to observe the quality control of the games studied since it allowed the identification of the flaws that went unnoticed during the development process of these games. In our study, we found that the games chosen for analysis have similar errors, which may be associated with the nature of these games since they are all geared towards online use. In addition, we found that users did not limit themselves to reporting usability issues in their reviews but suggested that developers incorporate new functionality into games as a way to troubleshoot bugs and avoid other potential issues that might arise.

Key-words: machine learning; natural language processing; usability; online games.

LISTA DE FIGURAS

Figura 1 – Fluxograma.....	20
Figura 2 – Qrcode do código fonte.....	31
Figura 3 – Gráfico das amostras.....	36

LISTA DE QUADROS

Quadro 1 – Comentários removidos.....	26
Quadro 2 – Comentários específicos.....	37
Quadro 3 – Comentários generalizados.....	45
Quadro 2 – Comentários sugestivos.....	49

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
AWS	Amazon Web Service
CAPTCH	Completely Automated Public Turing test to tell Computers and A Humans Apart
CS:GO	Counter-Strike: Global Offensive
CSS	Cascading Style Sheets
DB	Database
FPS	Frames Per Second
HTML	HiperText Markup Language
HTTP	HyperTextTransferProtocol
IP	Internet Protocol
ISO	International Organization for Standardization
MMORPG	Massively Multiplayer Online Role-Playing Game
NaN	Not-A-Number
NLTK	Natural Language Toolkit
PNL	Processamento de Linguagem natural
PvP	Player vs Player
SSL	Secure Sockets Layer
TLS	Transport Layer Security
VAC	Valve Anti-cheat
XML	Extensible Markup Language

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Objetivo geral	11
1.2	Objetivos específicos	11
1.3	Organização do trabalho	11
2	METODOLOGIA	13
3	REVISÃO DA LITERATURA	14
3.1	Aprendizagem de máquina	14
3.2	Processamento de linguagem natural	15
3.3	Usabilidade	15
3.4	Ferramentas utilizadas	16
<i>3.4.1</i>	<i>Python</i>	<i>16</i>
<i>3.4.2</i>	<i>Jupyter Notebook</i>	<i>17</i>
<i>3.4.3</i>	<i>Bibliotecas</i>	<i>17</i>
<i>3.4.3.1</i>	<i>Pandas</i>	<i>18</i>
<i>3.4.3.2</i>	<i>NumPy</i>	<i>18</i>
<i>3.4.3.3</i>	<i>Scikit-learn ou Sklearn</i>	<i>18</i>
4	DESENVOLVIMENTO	20
4.1	Processo de aprendizagem de máquina	20
4.2	Apresentação do processo de coleta de dados	20
4.3	Apresentação do processo de limpeza e pré-processamento	21
4.4	Seleção de recurso e treinamento	21
4.5	Validação e implementação do modelo	21
4.6	Coleta de dados	22
4.7	Pré-processamento	25
4.8	Treinamento	30
4.9	Código fonte	31

5	ANÁLISE DOS RESULTADOS	32
5.1	Jogos analisados	33
<i>5.1.1</i>	<i>Counter-Strike Global Offensive</i>	<i>33</i>
<i>5.1.2</i>	<i>World of Warship</i>	<i>33</i>
<i>5.1.3</i>	<i>Star Trek Online</i>	<i>34</i>
<i>5.1.4</i>	<i>Paladins</i>	<i>35</i>
5.2	Problemas identificados	35
<i>5.2.1</i>	<i>Comentários específicos</i>	<i>36</i>
<i>5.2.1.1</i>	<i>Relação entre comentários específicos e usabilidade</i>	<i>43</i>
<i>5.2.2</i>	<i>Comentários genéricos</i>	<i>44</i>
<i>5.2.2.1</i>	<i>Relação entre comentários genéricos e usabilidade</i>	<i>48</i>
<i>5.2.3</i>	<i>Comentários sugestivos</i>	<i>49</i>
<i>5.2.3.1</i>	<i>Relação entre comentários sugestivos e usabilidade</i>	<i>51</i>
6	CONSIDERAÇÕES FINAIS	53
	REFERÊNCIAS	55
	APÊNDICE A - APRESENTAÇÃO DAS BIBLIOTECAS	59

1 INTRODUÇÃO

A discussão sobre o tema do presente trabalho surgiu enquanto estávamos cursando a disciplina de inteligência artificial, a ideia nasceu através da proposta de combinarmos o estudo de aprendizagem de máquina com uma de nossas áreas de interesse, os jogos eletrônicos. Deste modo, definimos que o nosso objeto de estudo seria o uso de um dos subcampos da aprendizagem de máquina, o processamento de linguagem natural, como sendo uma ferramenta facilitadora no processo de análise dos *feedbacks* dos jogadores a fim de identificarmos falhas de usabilidades nos jogos escolhidos. Pois o aumento da competitividade de mercado tem obrigado empresas e organizações a adotarem novas formas de amplificar a produtividade do desenvolvimento de softwares, com adesão de novas metodologias e técnicas para otimizar a produção das equipes.

Em muitos casos, a necessidade de cumprir os curtos prazos de entrega para atender a grande demanda, afetou diretamente a qualidade dos softwares desenvolvidos, o que gerou insatisfação dos clientes e a perda de credibilidade das empresas. Deste modo, surgiu a necessidade de se adotar práticas de controle de qualidade, tais como: automação de testes, revisão de código, análises estatísticas, entre outras. Embora estas práticas tenham ajudado a melhorar a qualidade dos softwares produzidos, é comum que as falhas ultrapassem essas etapas fazendo com que as empresas desenvolvam novas formas de melhorar seus produtos.

Nesse sentido, considerar a opinião dos jogadores sobre os jogos pode refletir diretamente na qualidade do produto bem como nos ganhos ou nas perdas econômicas para as empresas responsáveis por estes softwares, uma vez que um produto com uma boa usabilidade se torna mais fácil de se utilizar e passa a ser mais eficiente no cumprimento do objetivo para o qual foi desenvolvido, o que pode causar fidelização dos consumidores. Além de que um software bem avaliado pode conquistar um novo público, visto que ao garantir a satisfação dos usuários atuais podem os levar a recomendar o produto para mais pessoas.

Para as empresas, considerar e acompanhar as experiências dos jogadores pode ser extremamente importante para identificar falhas presentes no software, que não foram detectadas durante o processo de desenvolvimento, o que pode garantir um melhor tempo de resposta para solucionar tais problemas, obtendo vantagens com relação à concorrência.

Sendo de cunho bibliográfico e de pesquisa em sites especializados, o presente trabalho traz autores como: Aurélien Géron (2019) e Stephen Marsland (2015) e os sites: Alura, Steam, Techtudo e AWS (Amazon Web Services).

1.1 Objetivo Geral

Diante disso, o objetivo geral do nosso trabalho é criar um modelo de aprendizagem de máquina capaz de classificar o sentimento dos usuários expressos nos comentários, para que possamos analisar as avaliações negativas e identificar as falhas de usabilidade apontadas pelos jogadores.

1.2 Objetivos específicos

Os nossos objetivos específicos consistem em realizar um levantamento de uma base de dados de comentários, para que o modelo possa analisar cada um dos reviews feitos pelos usuários e atribuir a cada um deles um identificador, sendo este o número zero para sentimentos negativos e o número um para sentimentos positivos. Posteriormente, extrair os comentários negativos para identificar as características dos jogos que influenciaram os usuários a avaliarem negativamente e gerar ideias que poderiam aprimorar a experiência do usuário, bem como a qualidade dos softwares.

1.3 Organização do trabalho

O presente trabalho está estruturado em cinco capítulos, sendo estes: Metodologia, Revisão de Literatura, Desenvolvimento, Análise dos Resultados e as Considerações Finais.

No capítulo denominado de Metodologia abordamos os métodos utilizados para o desenvolvimento do nosso trabalho, nele descrevemos as etapas que seguimos durante o processo de construção do nosso modelo de aprendizagem de máquina, sendo elas: a coleta de dados, a limpeza e o pré-processamento, a seleção dos recursos, o treinamento, a validação e a implementação do modelo.

Na revisão de literatura expomos os principais conceitos para a compreensão do nosso projeto, tais como: o que é aprendizagem de máquina e quais são as suas aplicações, explanamos as definições de processamento de linguagem natural e de usabilidade, além de apresentarmos as ferramentas utilizadas na construção do nosso programa.

Já no desenvolvimento detalhamos os processos de implementação do código, mostramos as dificuldades encontradas durante a construção do programa, explicamos os processos utilizados para fazer a limpeza e o pré-processamento dos dados, além de discutirmos as funções utilizadas para construção e aplicação do modelo.

No capítulo análise dos resultados definimos a amostra que utilizamos, fazemos um introdutório explicando os jogos escolhidos e expomos os problemas de usabilidade que conseguimos identificar nos comentários feitos pelos usuários.

Por fim, nas considerações finais tecemos alguns comentários sobre as questões abordadas ao longo do trabalho, respondemos aos problemas propostos pela nossa pesquisa e apresentamos uma possível proposta de um trabalho futuro.

2 METODOLOGIA

Neste capítulo, apresentamos uma descrição dos processos realizados durante a nossa pesquisa para atendermos os objetivos propostos e para obtermos os resultados esperados.

Inicialmente, realizamos uma revisão bibliográfica, em livros e em páginas da internet, para definirmos conceitos relevantes para a compreensão do nosso trabalho. Por meio desta, obtivemos uma visão de outros autores sobre aprendizagem de máquina e usabilidade. Além disso, mapeamos e estudamos novas tecnologias para realizar a classificação e separação automatizada de comentários, através de técnicas de análise de sentimentos empregadas no processamento de linguagem natural.

Após realizarmos o embasamento teórico, a próxima etapa foi realizar um levantamento de uma base de dados de comentários, que foi separada em duas categorias: os dados que seriam utilizados no treinamento e no teste do modelo e os dados que seriam aplicados no modelo quando ele estivesse pronto, para que depois fosse possível realizar a análise dos comentários classificados como negativos. Como em nossa pesquisa estamos trabalhando com aprendizado de máquina supervisionado, os dados da nossa base de comentários que seriam destinados ao treinamento do modelo deveriam conter um indicador para identificar se o comentário é uma avaliação positiva ou negativa. Tendo isso em vista e os problemas encontrados nas tentativas de automatizar a extração dos dados, optamos por uma base de dados pronta.

Após aplicarmos os dados no modelo e extrairmos os comentários negativos, procedemos a um estudo de casos baseados nesses comentários, visando identificar as relações dos resultados com problemas de usabilidade, por meio dos relatos dos usuários sobre os problemas que encontraram ao interagirem com os jogos e como isso impactou a sua satisfação com o produto.

3 REVISÃO DA LITERATURA

Neste capítulo abordaremos os conceitos necessários para a compreensão do trabalho, além de apresentarmos as ferramentas utilizadas durante o desenvolvimento da aplicação. Inicialmente discutiremos sobre o que é aprendizagem de máquina e as suas aplicações, em seguida, apresentaremos uma de suas vertentes, o processamento de linguagem natural. E por fim falaremos sobre a usabilidade.

3.1 Aprendizagem de máquina

Géron (2018, p. 4) define a aprendizagem de máquina como sendo uma ciência da programação, na qual o computador deve adquirir o conhecimento para realizar uma determinada tarefa através do conjunto de dados. Em outras palavras, trata-se de um campo de estudo nos quais programas são desenvolvidos para identificar padrões e relações entre os dados de entrada para que, então, consigam cumprir atividades específicas. Dentre as mais comuns, podemos destacar as previsões de resultados, a classificação de dados e o reconhecimento de padrões.

O aumento significativo da produção e do armazenamento de dados junto com o barateamento de computadores com maior poder de processamento amplificou o interesse das empresas e organizações em aplicarem algoritmos de aprendizagem de máquina com intuito de extrair informações relevantes para seus negócios. Dentre tais aplicações destacamos:

- No diagnóstico médico – utilizada na detecção de doenças e na análise de imagens de exames;
- No marketing – empregada na recomendação de produtos com base no conteúdo pesquisado pelo usuário; e
- Na análise financeira – existem modelos que são capazes de determinar as vantagens ao fazer um investimento e os possíveis riscos que esta transação pode acarretar.

Na aprendizagem de máquina existem diferentes tipos de abordagens que podem ser adotadas para treinar o modelo, o que vai determinar qual delas deve ser empregada é o tipo de problema o qual se deseja solucionar, um outro fator que deve ser levado em consideração é o tipo de dados com que se está trabalhando. Segundo Marsland (2014, p. 5-6) os principais tipos de abordagens são:

- Aprendizagem supervisionada – o conjunto de dados é fornecido junto com as respostas para que assim a máquina possa usá-las como base para determinar a solução correta para os próximos dados de entrada;
- Aprendizagem não supervisionada – neste caso, as respostas não são fornecidas, o computador tenta reconhecer similaridades entre os dados de entrada para, então, os categorizarem;
- Aprendizagem de reforço – vista como sendo a abordagem intermediária entre o aprendizado supervisionado e o não supervisionado. Nesta abordagem é possível apontar quando o algoritmo chega a uma solução incorreta, com base nesta informação a máquina deverá testar outras possibilidades até conseguir a resposta correta;
- Aprendizagem profunda – usada em diversas áreas, tais como: reconhecimento de fala, visão computacional e processamento de linguagem natural. Trata-se de um modelo com diversas camadas, baseado no funcionamento de redes neurais profundas.

3.2 Processamento de linguagem natural

De acordo com o site da Alura (2020) o Processamento de Linguagem Natural (PLN) é um subcampo da aprendizagem de máquina que visa estudar a interação entre o ser humano e o computador através da linguagem natural, em outras palavras, é o estudo das técnicas usadas para tornar a linguagem do cotidiano das pessoas compreensível para as máquinas.

Essas técnicas são aplicadas nas mais variadas áreas, na maioria das vezes, elas servem ao propósito de extrair informações relevantes de textos que agreguem algum valor de negócio para empresas e organizações. Dentre estas aplicações podemos citar: avaliação do sentimento de um cliente sobre sua experiência com determinado produto ou serviço – analisa os feedbacks dos consumidores, caso os resultados desta avaliação sejam negativos a empresa responsável poderá extrair os dados necessários para reverter esta situação; identificar tendências – permite realizar pesquisas de mercado através de padrões de consumo, desta forma é possível oferecer um produto de acordo com a demanda; e análise de documentos – pode ser usada para catalogar e resumir grandes volumes de textos.

3.3 Usabilidade

De acordo com a ISO 9241-11 (ABNT, 2002) usabilidade é uma medida utilizada para avaliar a interação entre um usuário específico e um produto, seja este físico ou abstrato

(software, por exemplo), na qual se analisa a capacidade do usuário em utilizar o produto para o propósito que este foi desenvolvido. Nesta avaliação existem alguns critérios que devem ser considerados, dos quais destacamos: eficácia – a precisão que o usuário atinge, ao cumprir o objetivo específico; eficiência – a produtividade obtida pelo usuário na conclusão da tarefa; e a satisfação – sentimento positivo do usuário sobre sua experiência com o produto.

A má usabilidade de um produto pode acarretar consequências negativas, entre elas perdas econômicas. Em um software, por exemplo, é comum que consumidores abandonem a ideia de utilizar um serviço por considerá-lo difícil de se manusear ou por constar que este não cumpre o seu propósito com eficiência.

Dessa maneira, para garantir a usabilidade de um software é importante realizar testes com participação dos usuários buscando identificar problemas presentes na interação destes com o produto, além de observar suas sugestões para possíveis melhorias. Nesses testes, recomenda-se avaliar se os softwares apresentam os seguintes critérios:

- Simplicidade – possui uma interface intuitiva, fácil de se compreender?
- Eficiência – não requer muito esforço para realizar uma tarefa?
- Acessibilidade – o software é inclusivo, pode ser utilizado por pessoas com deficiências físicas ou cognitivas?
- Fornece feedback – após o usuário realizar uma ação o software fornece uma resposta clara?

Todas estas questões devem ser levadas em consideração para que se tenha uma boa usabilidade do software em questão.

3.4 Ferramentas Utilizadas

Nesta seção serão discutidas as principais tecnologias envolvidas no desenvolvimento deste TCC.

3.4.1 Python 3.10

De acordo com o site AWS, (s/d) *Python*¹ é uma linguagem de programação *open source* (código aberto) criada pelo holandês Guido van Rossum, que se popularizou devido a sua simplicidade e facilidade de se aprender, pois possui uma sintaxe que se assemelha ao inglês. Em 1991 foi lançada a primeira versão da linguagem, o *Python 0.9.0*, desde então ela recebeu diversas atualizações com acréscimo de novos recursos e correções de erros. Para o

¹ Para mais informações acesse o site oficial disponível em: <https://www.python.org/>

nosso projeto utilizamos a versão 3.10, até então a mais recente, disponibilizada em outubro de 2021.

Devido à grande variedade de ferramentas e de bibliotecas que possui, o *Python* é utilizado para realizar as mais diversas tarefas, tais como: desenvolvimento *web* – usada na implementação de aplicações no lado servidor, pois disponibiliza uma série de recursos que auxiliam na execução de funções complexas do *back-end*; automação – através de pequenos trechos de códigos é possível automatizar tarefas repetitivas do cotidiano; e aprendizado de máquina – existem bibliotecas que auxiliam na visualização e no pré-processamento de dados, além de permitirem a criação e o treinamento de modelos de classificação.

3.4.2 Jupyter Notebook

Para executarmos o *Python* utilizamos o *Jupyter Notebook*² que é uma aplicação *web*, bastante utilizada no desenvolvimento de programas na área de ciência de dados, pois permite a visualização simultânea dos resultados, sejam estes cálculos matemáticos, gráficos estatísticos ou tabelas usadas para análise de dados.

De acordo com o site da documentação oficial o *Jupyter Notebook* (s/d) trata-se de um ambiente virtual no qual desenvolvedores podem realizar a combinação entre documentos de texto e *scripts* de programa, sem que haja interferência na execução do código, deste modo, é possível criar programas mais dinâmicos.

A aplicação funciona por meio de um servidor que pode ser local ou remoto, desta maneira é possível acessar determinado projeto de qualquer máquina desde que a pessoa esteja conectada à internet, além disso, o *Jupyter* dispõe de um *kernel* que permite rodar *scripts* em várias linguagens de programação sem a necessidade de fazer instalações, sendo algumas destas: *Julia*, *Python*, *Ruby*, *R*, entre outras.

3.4.3 Bibliotecas

Nesta seção apresentaremos as bibliotecas do *Python* que foram utilizadas na implementação do nosso trabalho. Essas ferramentas contribuíram para a execução dos diferentes estágios que envolvem a construção de um modelo de aprendizagem de máquina. O *Pandas*, por exemplo, auxiliou no processo de análise da base de dados e junto com o *NumPy* contribuiu para a realização da limpeza e do pré-processamento. Por outro lado, o *Scikit-learn*

² Disponível em: <https://jupyter.org/>

permitiu criar o modelo e validá-lo. Para obter informações mais aprofundadas sobre o emprego dessas bibliotecas, consulte o Apêndice A.

3.4.3.1 *Pandas*

Para manipularmos a nossa base de dados utilizamos o *Pandas* que, segundo o site da Alura (2023), é bastante utilizado em projetos da área de ciência de dados, pois este pode ser integrado a outras bibliotecas que são comumente empregadas nesse tipo de projeto, das quais podemos destacar a *Numpy* e a *Scikit-Learn*, que foram aplicadas no desenvolvimento do nosso trabalho.

Na aprendizagem de máquina o *Pandas* é utilizado para auxiliar a manipulação dos dados durante as etapas de limpeza e de pré-processamento pois este dispõe de um recurso chamado *DataFrame*, no qual é possível organizar os dados relacionais nas linhas e nas colunas de uma tabela, além de conter uma série de ferramentas que podem ser usadas para selecionar, filtrar e indexar os dados.

3.4.3.2 *NumPy*

Frequentemente empregada em projetos de aprendizagem de máquina sendo muitas vezes combinada com outras bibliotecas de alto nível, como *Pandas* e *Sklearn*. A *NumPy*, de acordo com o site *Haver* (s/d), auxilia no processamento dos dados através de métodos prontos que ajudam a resolver operações matemáticas e estatísticas que envolvem *arrays* multidimensionais.

O *n-array* é uma das principais estruturas de dados do *NumPy*, sendo este um *array* homogêneo de *n*-dimensões capaz de armazenar dados na memória do computador de forma contínua, deste modo, a biblioteca é capaz de acessar e modificar os dados de maneira eficiente.

3.4.3.3 *Scikit-learn ou Sklearn*

Segundo o site *Didática Tech* (s/d), a *Sklearn* permite criar modelos nas três principais subáreas da aprendizagem de máquina, sendo estas: a classificação – determina a qual classe um dado pertence, para isso é preciso que todas as classes sejam pré-definidas; clusterização – agrupa os elementos de acordo com os aspectos em comum; e a regressão – modelo baseado em uma técnica da estatística, na qual é possível prever o valor de uma variável dependente a partir do conhecimento do valor de uma variável independente, através desse método,

pode-se, por exemplo, determinar a quantidade de chuva para uma cidade, em particular, com base em dados climáticos de anos anteriores.

A *Sklearn* possui uma série de recursos que auxiliam durante as principais etapas de construção do modelo, o pré-processamento e a validação. Através da *Sklearn* é possível fazer a normalização dos dados, separá-los entre os conjuntos de teste e de treino, além de permitir avaliar a taxa de precisão do modelo através do seu módulo de métricas.

4 DESENVOLVIMENTO

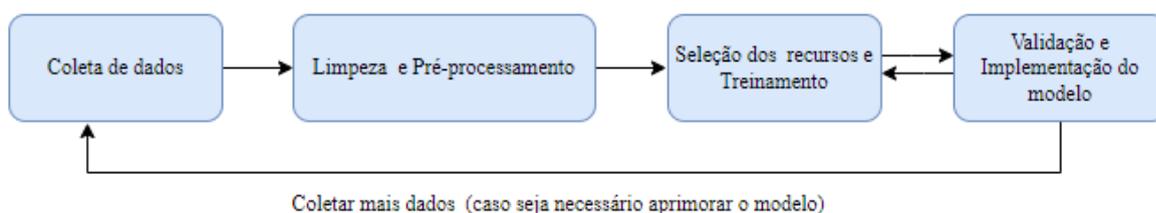
Nos tópicos seguintes detalharemos o processo de desenvolvimento da nossa aplicação, abordando as dificuldades encontradas no desenvolvimento do projeto e as técnicas utilizadas durante os estágios de construção do modelo. E por fim, explicaremos as funções utilizadas na implementação do nosso código.

4.1 Processo de aprendizagem de máquina

O desenvolvimento de um modelo de aprendizagem de máquina, assim como a maioria dos softwares, segue um fluxograma contendo as etapas que devem ser seguidas sequencialmente para a sua construção, dentre elas podemos destacar: a coleta de dados, a limpeza e o pré-processamento, a seleção dos recursos, o treinamento, a validação e a implementação do modelo. Deste modo, através desta seção abordaremos a metodologia utilizada no desenvolvimento deste trabalho.

As fases que aplicamos na implementação do nosso modelo, as quais descrevemos a seguir, podem ser observadas na Figura 1:

Figura 1 – Fluxograma



Fonte: Elaborado pelo autor, 2023.

4.2 Apresentação do processo de coleta de dados

Antes do processo de extração dos dados foi necessário definir o objeto de estudo do nosso trabalho, uma vez determinado que criaríamos um modelo de aprendizado de máquina para aplicarmos no controle de qualidade de jogos eletrônicos a fim de identificar problemas de usabilidade, definimos que os dados coletados deveriam ser constituídos por relatos dos

usuários para que assim pudéssemos avaliar o grau de insatisfação destes com relação ao software.

Os dados utilizados no desenvolvimento do nosso projeto foram extraídos de uma plataforma de competição de aprendizagem de máquina, a Kaggle³, que disponibiliza bases de dados prontas e em formato de planilha, destinadas para o treinamento de modelos de aprendizagem supervisionada.

4.3 Apresentação do processo de limpeza e pré-processamento

As etapas de limpeza e de pré-processamento são essenciais quando se está trabalhando com processamento de linguagem natural pois são através delas que os dados são organizados e preparados de forma que a máquina consiga compreendê-los com maior facilidade. Após obtermos nossa base de dados, tivemos de realizar uma análise para identificar problemas presentes no conjunto de dados para que estes não afetassem a performance do nosso modelo. Dentre os problemas encontrados destacamos: os erros de ortografia, inconsistências, duplicidade de informações, elementos inválidos, entre outros. Uma vez identificados os problemas, aplicamos, então, técnicas para solucioná-los.

4.4 Seleção de recurso e treinamento

Para definirmos quais ferramentas seriam aplicadas no treinamento do nosso modelo nos baseamos em uma videoaula disponibilizada no YouTube (Live #6: Análise de sentimentos utilizando dados do Twitter). Foi através dos métodos abordados nela que conseguimos separar o nosso conjunto de dados de treino e de teste e convertê-los para um formato que fosse compreensível para a máquina. Uma vez que a conversão foi concluída, aplicamos os dados de treinamento no nosso modelo para seguirmos para a etapa de validação.

4.5 Validação e Implementação do modelo

Com o conjunto de dados de teste realizamos a validação do nosso modelo por meio de um comparativo entre as respostas previstas pelo modelo e os resultados esperados. Desta forma, conseguimos medir a sua taxa de acerto. Observamos que o valor obtido nesta etapa serviria ao propósito do nosso projeto. Logo após concluir a validação, separamos uma nova

³ Disponível em: <https://www.kaggle.com/>

base de dados para aplicarmos no modelo. Para isto tivemos que executar as mesmas técnicas usadas nas etapas de limpeza e pré-processamento. Depois que esse novo conjunto de dados foi classificado pelo nosso modelo, extraímos os comentários dos usuários que foram categorizados como negativos para então realizarmos a análise dos resultados.

4.6 Coleta dos dados

A coleta de dados é uma das fases mais importantes para a aprendizagem de máquina, uma vez que é a partir do conjunto de dados que um computador consegue adquirir a experiência necessária para realizar uma determinada tarefa. Com a ausência desses dados, a máquina não seria capaz de realizar a sua função, visto que, não possuiria as informações necessárias para aprender e então desenvolver suas habilidades.

Em conformidade com o que afirmam Farias e Angelo (s/d, p. 11) “A fase de seleção de dados é a primeira no processo de descobrimento de informação e possui impacto significativo sobre a qualidade do resultado final”, nesta etapa é feita a seleção dos dados para que possamos aplicá-los ao nosso modelo de aprendizagem de máquina.

Em um primeiro momento, havíamos idealizado que as extrações dos dados seriam realizadas de maneira automatizada, usando ferramentas de *Web Scraping*⁴, para que então pudéssemos montar um *database* (DB) com o *feedback* dos usuários. A coleta seria realizada nas plataformas mais populares, conhecidas por oferecerem serviços de distribuição de aplicativos e de jogos digitais, tais como: *Google Play*, serviço da *Google* nativo em dispositivos com sistema operacional *Android*, a *Steam*, *software* de gestão de direitos digitais criada pela *Valve Corporation*, e a *Xbox Game Store*, plataforma de distribuição digital utilizada pela *Microsoft* nos consoles *Xbox One* e *Xbox 360*. Como o nosso trabalho tinha como intuito inicial utilizar múltiplas plataformas para realizar a seleção dos dados, esta tarefa apresentaria uma maior complexidade, pois como destacam Faria e Angelo (s/d, p.12) “os dados podem vir de uma série de fontes e podem possuir os mais diversos formatos.”

Durante as tentativas iniciais de se fazer uma aplicação que fosse capaz de efetuar a extração dos dados, realizamos diversos testes com algumas ferramentas, das quais a que mais se destacou para nosso propósito foi a *Beautiful Soup*. Uma biblioteca da linguagem de programação Python responsável pela extração de dados em HTML (Hypertext Markup Language) e XML (Extensible Markup Language), através de um interpretador (o parser), ela

⁴ De acordo com o site Canaltech (2021) *web scraping* é um processo de extração de dados estruturados da *web*, de forma automatizada.

é capaz de navegar, buscar e modificar a árvore de análise (BEAUTIFUL SOUP, s/d). Entretanto, a *Beautiful Soup* possui algumas desvantagens, pois apesar de ser uma biblioteca bastante poderosa ela não é capaz de fazer extrações dinâmicas, ignorando assim, todos os conteúdos da página que estejam em JavaScript. Dessa forma, para que pudéssemos fazer a nossa aplicação foi preciso complementá-la usando a API (Application Programming Interface) Selenium, responsável por automatizar a interação com o navegador, com isso o Selenium seria encarregado por renderizar os dados em JavaScript para que então a Beautiful Soup pudesse extraí-los (MEDIUM, 2021).

Uma das dificuldades que encontramos ao trabalhar com a *Beautiful Soup*, foi que ela utiliza elementos do HTML e do CSS (Cascading Style Sheets) para realizar as buscas, deste modo, a cada página seria preciso realizar revisões ou até mesmo alterações no código do programa para que assim pudéssemos obter os comentários. Além disso, muitas plataformas possuem mecanismos de detecção e proteção contra ferramentas de Scraping. O Google por exemplo, que apesar de não adotar medidas legais para resguardar os dados de interesse da empresa, adotou outros métodos de proteção. Segundo o site ApifyDocs (2022), as técnicas anti-scraping mais populares utilizadas pelos websites são:

- Bloqueio baseado em endereço de IP (Internet Protocol) – esses sistemas consistem no bloqueio de intervalos de IP ao qual seu endereço pertença, caso seja detectado um fluxo de tráfego de maneira automatizada seu acesso ao site em questão será interrompido.
- Limitação de taxa de IP – os sites são capazes de registrar o número de requisições que são feitas ao acessá-los, em programas de Web Scraping o fluxo de requisições é bem maior do que o fluxo realizado por um ser humano, desta forma se o número de requisições exceder o limite estipulado, o site usará o sistema de bloqueio baseado em endereço de IP ou realizará um teste CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) que consiste em uma verificação para distinguir entre pessoas e computadores.
- Análise de solicitações HTTP (HyperText Transfer Protocol) – quando um cliente realiza uma solicitação HTML ao servidor, ele está enviando uma série de informações ao servidor, que são imperceptíveis ao usuário, dentre as quais podemos destacar: o cabeçalho do HTML, o endereço de IP do cliente, versão SSL/TLS (Secure Sockets Layer/ Transport Layer Security), entre outras. É através do cabeçalho que o

site é capaz de identificar se a solicitação veio de um navegador real ou por meio de script.

- Análise de comportamento do usuário – Os sites são capazes de monitorar o comportamento dos usuários por um longo período. Observando o seu tempo de permanência em uma página, a movimentação do mouse, a ordem de acesso das páginas e a velocidade em que os formulários são preenchidos, dessa forma o site pode identificar quando o acesso está sendo realizado por um script para que então possa reagir, interrompendo o tráfego ou através de um CAPTCHA.
- Impressão digital do navegador – existem técnicas que podem ser usadas por um site para identificar se o acesso está sendo feito ou não por um ser humano, para isto é preciso extrair informações do navegador, tais como: a versão do programa, as extensões instaladas, o fuso horário, o número de visitas realizadas ao site, entre outros. À combinação dessas informações dá-se o nome de impressão digital do navegador.

Devido às técnicas de anti-scraping tivemos que adotar uma outra abordagem para realizarmos a coleta dos dados. Nesta nova etapa encontramos uma API, a Google-Play-Scraper, que propunha fazer Web Scraping no site da Google Play Store, desta forma poderíamos coletar os comentários com êxito. No entanto, identificamos que essa API não serviria ao nosso propósito devido ao fato de possuir limitações, visto que, segundo o site Octoparse (2022), a API estabelece que as requisições deveriam ser limitadas a uma quantia de sessenta, durante o prazo de uma hora. Deste modo, para que pudéssemos coletar mais dados, seria preciso esperar o tempo estipulado.

Em razão dessas adversidades e tendo em vista que ainda seria necessário classificar manualmente os comentários dos usuários, para que pudéssemos gerar o nosso DB e assim utilizá-lo para o treinamento do nosso modelo de aprendizado de máquina, decidimos por mudar a nossa abordagem para uma proposta que seria mais viável. Sendo assim, optamos por pesquisar um DB já pronto que atendesse às nossas necessidades. Em nossas buscas, encontramos o Kaggle⁵ que, segundo o blog Hashtag (2022), é uma plataforma popular entre a comunidade que tem interesse em aprender sobre a área de Data Science (Ciência de Dados), pois esta é repleta de uma grande variedade de recursos sobre o tema, tais como: tutoriais, campeonatos, cursos, dicas, fóruns, *datasets* e muito mais⁶. O Kaggle é a maior comunidade

⁵ Mais detalhes no site: <https://www.hashtagtreinamentos.com/kaggle>.

⁶ O Dataset pode ser encontrado no site: <https://www.kaggle.com/datasets>

da internet na área de Data Science, que até o ano de 2022 contava com mais de 536 mil membros ativos. Este número cresce diariamente com a entrada de novos participantes.

Um dos pontos positivos de se usar o *dataset* do Kaggle em nosso trabalho é que está disponível de forma gratuita para qualquer usuário e sem a necessidade de efetuar qualquer tipo de cadastro. Além disso, o seu DB é composto por dados limpos, completos, padronizados, formatados e sem erros, já com o propósito de serem utilizados para o treinamento de um modelo de aprendizagem de máquina. Deste modo, poderíamos pular algumas fases do pré-processamento e da limpeza dos dados, assim, conseguiríamos fazer uma análise mais aprofundada na lógica do programa.

O *dataset* que encontramos é formado por dois arquivos com formato CSV, no qual o primeiro seria utilizado para o treinamento do nosso modelo, contendo 17490 avaliações de 28 jogos, nos quais cada comentário era classificado como 1, para aqueles que expressavam sentimentos positivos, e 0 para os sentimentos negativos. Já o segundo arquivo é formado por 8044 comentários de 21 jogos, distintos dos que havia no primeiro arquivo, sendo que este não continha a classificação de sentimentos. Este último arquivo deveria ser destinado a etapa de teste do nosso modelo, na qual seria verificada a sua taxa de acerto, no entanto, optamos por dividir o primeiro arquivo para estas duas etapas, visto que ele já continha a classificação dos sentimentos e seria mais fácil verificar a acurácia do nosso algoritmo. Uma vez concluída a fase de teste, aplicamos o segundo arquivo no modelo já pronto para então podermos analisar os resultados em busca de identificar problemas de usabilidade nos relatos feitos pelos usuários.

4.7 Pré-processamento e limpeza

De acordo com Farias e Angelo (s/d, p. 12), é por meio da limpeza e do pré-processamento que são aplicadas as técnicas para a eliminação de dados redundantes, de inconsistências e de discrepâncias; além de realizar a recuperação de dados incompletos. É por meio destes processos que é determinada a qualidade dos dados, os quais são cruciais para definir a eficiência do algoritmo.

Como mencionado no tópico anterior, por termos utilizado o arquivo do DB disponibilizado pelo Kaggle, fomos poupados de realizar uma limpeza aprofundada nos dados, tendo apenas que fazer alguns ajustes. Por exemplo, em nosso conjunto de dados não

Para a limpeza de nossos dados realizamos as seguintes etapas: a remoção de acentos e de dados redundantes, a normalização e a remoção de *stop words* (palavras de paradas). As *stop words* são palavras usadas com frequência nos textos e que não agregam valor ao processamento de linguagem natural, e é por isso que podem ser removidas com segurança e sem prejuízo ao significado da frase. A remoção destas palavras proporciona uma redução no tempo de processamento, além de liberar espaço na base de dados. Apesar de não haver uma lista de *stop words* geral, é possível criar a sua própria ou até mesmo utilizar uma lista pré-definida, foi o que fizemos em nosso projeto, por meio da biblioteca NLTK¹¹ (Natural Language Toolkit) que nos permitiu fazer o download de uma lista pronta para que então pudéssemos usá-la.

De acordo com o site da Alura (2021), uma das maiores adversidades presentes quando estamos trabalhando com o processamento de linguagem natural é a simplificação do nosso conjunto de dados, para que assim sejamos capazes de extrair as informações necessárias para atingirmos o nosso objetivo. Um desses processos de simplificação se dá por meio da eliminação dos radicais das palavras. No PLN existem duas técnicas de normalização de textos que são conhecidas por obter esse resultado, o Stemming e a Lemmatization (stemização e lematização, no português). Vimos que seria necessário aplicar uma destas técnicas em nosso projeto uma vez que, de acordo com o site Wizard (2017) o inglês, assim como muitos outros idiomas, é constituído por palavras de diferentes origens, sendo que algumas delas são resultantes da combinação e da derivação de palavras já existentes.

Para podermos decidir qual das duas técnicas iríamos aplicar em nosso projeto, primeiramente precisaríamos entender suas diferenças. De acordo com a página da Alura (2021), embora ambas as técnicas tenham como objetivo encontrar as raízes das palavras, a principal diferença entre elas é que na stemming o processo de redução da palavra é feito por meio da eliminação dos últimos caracteres, o que pode resultar em uma palavra gramaticalmente incorreta, já na lemmatization é realizada uma análise buscando o radical correto para cada palavra. Um dos aspectos que deve ser levado em consideração na hora de fazer a escolha é qual objetivo se pretende alcançar, uma vez que a lematização leva maior tempo de processamento, sendo a mais indicada para quando a linguagem for importante durante o processo de desenvolvimento da aplicação. Já a stemming é a mais indicada quando se está trabalhando com problemas como a análise de sentimento, uma vez que é importante obter a base da palavra para que esta seja classificada como sendo positiva ou negativa.

¹¹ É uma das principais bibliotecas do Python para o estudo com o processamento de linguagem natural, pois nela contém algoritmos essenciais para o pré-processamento de textos.

Embora o nosso *database* seja constituído, em sua maior parte, por comentários na língua inglesa, que comumente não apresenta o uso de palavras acentuadas, sendo usado apenas em palavras com influência estrangeira, optamos por realizar a remoção dos acentos por meio da *string.punctuation* que é uma *string* utilizada como uma constante, contendo um conjunto de caracteres de pontuação. Em nossa implementação realizamos uma varredura em cada comentário, fazendo um comparativo com este conjunto. Desta forma, caso houvesse alguma palavra com acentuação, a letra na qual esta se encontraria seria substituída pelo mesmo caractere não acentuado. Contudo, por não haver dados redundantes, no nosso projeto não foi necessário a remoção de tais dados.

É bem comum em algumas linguagens de programação se fazer distinção entre o uso de caracteres maiúsculos e minúsculos, a chamada case-sensitive. Por exemplo, quando uma palavra é escrita usando apenas caracteres em minúsculo e esta mesma palavra é copiada, mas dessa vez mudando uma das letras para maiúsculos, ambas são interpretadas pelo programa como sendo palavras diferentes, por isso se faz necessária a normalização, para que não exista essa distinção. Em Python existem dois métodos internos que podem ser usados para resolver esse problema, *.upper()* e o *.lower()*, que são métodos usados para manipular *string*, sendo que o primeiro muda os caracteres que constituem a *string* para caixa alta e o segundo os mudam para caixa baixa. Em nosso projeto utilizamos o comando *.lower()* para realizamos a normalização de todos os comentários.

Uma vez concluída a normalização seguimos para as próximas etapas, nas quais iríamos realizar a preparação dos dados para aplicarmos ao nosso modelo. Primeiramente tivemos que realizar a tokenização dos comentários que, segundo o site Alura (2021), consiste na quebra das frases, na qual cada uma de suas palavras são transformadas em pequenos segmentos, os chamados de tokens. Para isso, utilizamos um dos módulos presentes na biblioteca NLTK, o *tokenize*, foi através de um de seus métodos, o *word_tokenize()*, que fomos capazes de realizar a segmentação dos comentários, uma vez que este realiza uma verificação a fim de identificar caracteres que são usados para delimitar o fim de uma palavra. Por exemplo, os espaços em branco e as pontuações, quando um destes é identificado o *word_tokenize* realiza uma quebra, acrescentado cada palavra e pontuação a uma lista de *substring*.

Para a etapa seguinte tivemos que realizar a conversão de nossos dados, uma vez que o nosso modelo não seria capaz compreendê-los, pois as máquinas não conseguem trabalhar com dados em formato de texto apenas com valores numéricos. Para isso, utilizamos o *bag-of-words* (saco de palavra) que, de acordo com o site da Alura (2021), trata-se de uma

técnica bastante conhecida no processamento de linguagem natural o que permite criar uma representação numérica dos textos por meio de uma contagem contendo a frequência em que cada palavra se apresenta, desconsiderando a ordem em que esta aparece ou a sua estrutura no texto. A fim de realizarmos a conversão de nossos dados, optamos por utilizar a biblioteca *Scikit-learn*, também conhecida como *Sklearn*, a qual é direcionada especificamente para o desenvolvimento de aplicações na área de aprendizado de máquina. Foi por meio de um de seus módulos que fomos capazes de encontrar a função *CountVectorize()*, que nos permitiu criar um vocabulário através da conversão do nosso DB em uma matriz esparsa, a qual é estruturada da seguinte forma: cada coluna representa uma palavra presente nos textos, sendo tais palavras aquelas que aparecem com maior frequência¹² e as linhas são compostas pelo número de vezes em que estas palavras aparecem em cada comentário.

4.8 Treinamento

É nesta etapa que o conjunto de dados é inserido no modelo para que este possa aprender com os dados, a identificar padrões e fazer estimativas sem que seja diretamente instruído para isso.

Uma vez que conseguimos converter os dados para um formato que fosse compreendido pelo nosso modelo, a próxima etapa foi dividi-los entre os que seriam destinados à realização do treinamento do modelo e aqueles que seriam utilizados para testá-lo. Para isso utilizamos o método *train_test_split()*, disponível em um dos módulos da biblioteca *Sklearn*, que realiza a divisão dos dados em matrizes, nas quais os dados são distribuídos de forma randômica (aleatória). Por padrão, este método define um tamanho para os conjuntos de dados de teste e de treino, nos quais os dados são divididos entre 25% para o primeiro e 75% para o segundo. No entanto, para obter maior precisão do nosso modelo optamos por usar uma base de treinamento maior, constituída por 80% dos dados.

Durante a construção do nosso modelo de aprendizagem decidimos que iríamos utilizar um dos algoritmos de classificação com aplicações baseadas no teorema de *Naive Bayes*, sendo estes o *Multinomial*, o de *Bernoulli* e o de *Gaussian* presentes na biblioteca *Sklearn*. Para definirmos qual destes iríamos usar, realizamos um teste de desempenho, no qual instanciamos os três modelos passando o mesmo conjunto de dados de treino e de testes para cada um deles e, com o auxílio do método *accuracy_score()*, o qual realiza uma verificação entre a classificação verdadeira de cada texto e aquela que foi determinada pelo

¹² Determinamos que para esta etapa do nosso trabalho utilizamos as 10 mil palavras com maior frequência.

modelo para que, então, possa retornar o percentual de acertos, pudemos determinar qual apresentou maior precisão. A partir disso, observamos que o modelo *Multinomial* foi o que apresentou o melhor desempenho com taxa de acerto de 83,5% enquanto o *Bernoulli* ficou na média com 80,7%, já o *Gaussian*, apresentou o pior desempenho entre os três, com uma taxa de 78,2%.

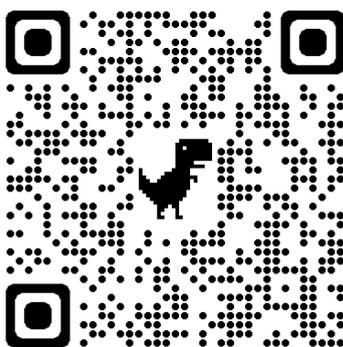
Uma vez concluída a fase de treino e de testes do nosso modelo, a próxima etapa seria realizar os mesmos processos de limpeza e de pré-processamento no outro *dataset* para que então pudéssemos realizar a análise dos resultados.

Após o modelo classificar os dados do nosso *dataset*, o próximo passo adotado foi realizar a separação dos comentários de acordo com o sentimento. Deste modo, inserimos uma nova coluna no *dataset* contendo um marcador indicando a classificação dada pelo modelo e por meio do método *query()*, que realiza uma busca por uma palavra ou termo específico. Com isso, conseguimos separar os comentários em duas listas, uma contendo 4439 comentários classificados como positivos enquanto a outra possuía 3571 comentários negativos. Dada a natureza do nosso trabalho, desconsideramos a primeira lista para dar enfoque nas reclamações dos usuários, deste modo, salvamos a lista com avaliações negativas em um arquivo com formato *xlsx* para podermos analisar os resultados.

4.9 Código Fonte

Para melhorar a legibilidade do nosso projeto optamos por hospedar o código fonte do programa no site do GitHub, que pode ser acessado através do seguinte Qrcode:

Figura 2 – Qrcode do código fonte



Fonte: Elaborada pelo autor, 2023.

5 ANÁLISE DOS RESULTADOS

Para a análise dos resultados agrupamos todos os dados que foram classificados como negativos pelo nosso modelo de aprendizagem de máquina. Com isso poderíamos analisá-los e verificar se era possível identificar problemas de usabilidade relatados pelos usuários.

No decorrer da nossa análise observamos que o grupo de dados podia ser categorizado das seguintes maneiras:

- genéricos – aqueles em que o usuário relata a existência de um problema, mas não o especifica, o que dificulta a identificação da falha para futuras correções;
- específicos – quando os usuários descrevem os problemas apresentados pelo software e como estes afetam sua experiência com o jogo;
- sugestivos – aqueles em que os usuários sugerem melhorias ou o acréscimo de novas funcionalidades, que não estão diretamente relacionadas a problemas.

O dataset, usado para realizar a análise dos resultados, continha 8044 comentários, dos quais 3571 foram apontados pelo modelo como sendo avaliações negativas. Selecionamos quatro dos vinte e um jogos presentes na nossa base de dados e, com isso, definimos que os jogos escolhidos seriam aqueles que já conhecíamos previamente. Assim, conseguiríamos compreender melhor as queixas dos usuários relacionadas às dinâmicas dos jogos. Mais especificamente, os jogos selecionados foram:

- Counter-Strike: Global Offensive (CS:GO);
- World of Warships;
- Star Trek Online;
- Paladins.

Com a classificação do modelo, observamos que alguns jogos continham quantidades menores de avaliações negativas do que os demais, alguns deles apresentando no máximo cem avaliações. Entre os jogos selecionados o CS:GO foi o jogo que apresentou o maior número de comentários negativos, a saber 291, enquanto o World of Warship continha 173, já Star Trek Online igualmente ao Paladins dispunha de cerca de 110 reviews. Tendo em vista estas variações nos números de comentários, definimos que limitaríamos a nossa amostra para que ela ficasse balanceada, contendo a mesma quantidade de comentários para cada um dos jogos; para isso, extraímos os cem primeiros comentários de cada um dos jogos selecionados.

5.1 Jogos analisados

Nos tópicos seguintes abordaremos os jogos selecionados, apresentaremos uma pequena descrição sobre cada um deles e exporemos os problemas que fomos capazes de identificar por meio dos feedbacks dos usuários. Para que haja melhor compreensão iremos traduzir para o português os reviews e, por fim, iremos suprimir todas as ofensas e palavras de baixo calão presentes nos mesmos.

5.1.1 *Counter-Strike Global Offensive*

Segundo os sites TechTudo (2022) e Betway Insider (2022), o CS:GO é um jogo gratuito lançado no ano de 2012 pelas empresas Valve Corporation e Hidden Path Entertainment. Que pode ser baixado para computadores com sistema operacional Windows, MAC OS e Linux, sendo distribuído pela Steam, e para os consoles de videogames da Sony, Playstation 3 (PS3) e da Microsoft, Xbox 360.

Trata-se de um jogo de tiro em primeira pessoa, que tem como premissa dividir 10 jogadores em duas equipes, os terroristas e os contraterroristas, em que cada equipe deve cumprir um objetivo definido. Existem diferentes modos de jogo, os três mais populares são:

- modo com a bomba – os terroristas têm que preparar e detonar os explosivos em locais específicos enquanto a outra equipe deve tentar impedir a detonação;
- com reféns – os contraterroristas devem tentar resgatá-los da outra equipe;
- o modo assassino – no qual os terroristas devem tentar matar um jogador específico enquanto a outra equipe deve impedir que isso aconteça.

Cada partida é constituída de 30 rodadas, com duração máxima de um minuto e cinquenta e cinco segundos, a partida se encerra ao fim das 30 rodadas ou se uma das equipes conseguir vencer primeiro um total de 16 rodadas.

5.1.2 *World of Warship*

De acordo com o Techtudo (2021), o *World of Warship* é um jogo online, gratuito, de tiro e com temática de guerra naval, lançado em 2015 pela empresa *Wargaming* para computadores com sistema operacional *Windows*. Só em 2018, devido a sua popularidade, foi que ele passou a ser distribuído para outras plataformas, sendo estas: *Playstation 4* e

Playstation 5 , *Xbox One*, *Xbox serie X* e *Xbox serie S*, além de ganhar uma versão *mobile* para celulares com sistema operacional *android* e *IOS*.

O site oficial *World of Warship* (s/d) separa a dinâmica do jogo em duas modalidades:

- batalha - no modo PvP (Player vs Player) os jogadores são divididos entre duas equipes constituídas por até doze pessoas cada, para que possam competir entre elas. Para vencer é preciso que uma das equipes obtenha mil pontos primeiro ou que consiga reduzir a pontuação dos seus adversários para zero, para isso é necessário destruir todas as embarcações inimigas. Já no modo cooperativo os jogadores devem se juntar para combater uma equipe formada por personagens controlados pela inteligência artificial do jogo;
- dominação - o mapa contém zonas especiais que as equipes devem disputar para conquistá-las e mantê-las sob o seu controle para que sua pontuação possa aumentar, a partida será encerrada quando uma das equipes atingir a pontuação determinada.

Em ambas as modalidades as partidas têm duração máxima de vinte minutos, caso nenhuma das equipes tenha conseguido a pontuação necessária para vencer antes do tempo limite acabar, aquela que estiver com a vantagem é declarada como sendo a vencedora.

5.1.3 Star Trek Online

Segundo os sites Techtudo (2016) e o Jogos MMORPG (s/d), trata-se de um subgênero do RPG (Role Playing Game) no qual vários jogadores interagem entre si e compartilham dos mesmos objetivos para avançar no jogo. Desenvolvido pela Cryptic Studios e lançado em 2010 para computadores com sistema operacional Windows, só no final de 2016 que passou a ser distribuído para o Playstation 4 e o Xbox One. Este jogo se baseia na famosa franquia de jornada nas estrelas (título em português) e se ambienta trinta anos após os eventos que se passaram no filme *Star Trek: Nemesis*, com o fim da aliança entre a Federação Unida dos Planetas e os Klingons.

No *Star Trek* online os jogadores devem personalizar os seus personagens escolhendo entre as facções: a Federação, os Klingons ou os Romulans. Cada facção possui um atributo diferente concedendo habilidades específicas ao jogador. Cada jogador pode obter sua própria espaçonave para realizar exploração e ainda realizar missões de combate. O objetivo principal do jogo é cumprir as missões disponíveis para aumentar a sua pontuação e então conseguir evoluir o personagem.

5.1.4 Paladins

O site Gamers & Games (2018) descreve Paladins como sendo um jogo de tiro em primeira pessoa online, no qual dez jogadores são divididos entre duas equipes que devem competir para determinar quem irá conseguir cumprir os objetivos primeiro. A mecânica do jogo pode ser separada em duas modalidades: mata-mata – os jogadores devem tentar eliminar seus adversários um determinado número de vezes para conseguir obter a vitória; cerco – as equipes devem transportar o mesmo veículo até a base inimiga aquela que conseguir avançar mais e capturando mais zonas ao longo do percurso será declarada como a vencedora.

Os jogadores poderão escolher seus personagens no início das partidas, sendo que estes são divididos em quatro classes que concedem habilidades e atributos específicos:

- tanques – combatentes da linha de frente que possuem grande resistência, cujo objetivo principal é proteger seus aliados;
- suportes – contêm a habilidade de curar membros da própria equipe, impedindo que eles sejam mortos durante o conflito;
- flanco – utilizado para eliminar inimigos das laterais este personagem é capaz de causar grande quantidade de danos e de se locomover com velocidade caso seja preciso recuar do combate; e o
- dano – capaz de efetuar mais dano do que as outras classes, seu principal objetivo auxiliar os seus companheiros eliminando o maior número de adversários possível.

De acordo com o site da Techtudo (2017), este jogo foi desenvolvido pela Hi-Rez que em setembro de 2016 o disponibilizou na versão beta para computadores, para que no ano seguinte fosse lançado para o *Playstation 4* e o *Xbox One*.

5.2 Problemas identificados

Primeiramente, extraímos nossa amostra contendo os cem comentários negativos de cada jogo, em seguida fizemos uma verificação para identificar se algum comentário havia sido classificado erroneamente como sendo negativo pelo nosso modelo. Observamos que em nossa amostra havia dois comentários que apresentavam este problema, estes foram substituídos por outros comentários pertencentes aos respectivos jogos.

Observamos então que muitos relatos feitos pelos usuários apresentavam aspectos de mais de uma categoria (genéricos, específicos ou sugestivos). Haviam comentários nos quais os problemas expostos pelos jogadores eram mais específicos além de outros problemas que

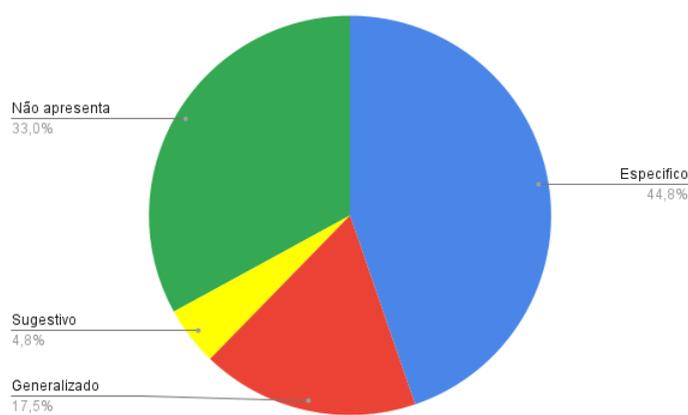
eram descritos de maneira mais genérica. Havia também aqueles que, além de expor o problema, ainda sugeriam melhorias e funcionalidades que poderiam ser acrescentadas. Desse modo, decidimos dividir em tabelas os comentários de acordo com a sua categoria, e aqueles que se encaixassem em mais de uma categoria seriam colocados em duas ou mais tabelas.

Depois dos processos de filtragem e de categorização dos dados observamos que os comentários que formavam a nossa amostra foram distribuídos da seguinte maneira:

- 33% destes não apresentavam problemas;
- 44,75% especificavam os problemas presentes no software;
- 17,5% abordavam as falhas de forma genérica; e
- 4,75% apresentaram sugestões de melhorias.

O gráfico da Figura 3 permite visualizar as porcentagens citadas anteriormente.

Figura 3 – Gráfico das amostras



Fonte: Elaborada pelo autor, 2023.

5.2.1 Comentários Específicos

Em nossa primeira investigação constatamos que nos relatos os usuários apontam problemas semelhantes entre os jogos analisados. Com exceção do Star Trek Online, vimos que uma das falhas mais recorrentes destacadas nos comentários categorizados como específicos estava relacionada à segurança dos softwares, principalmente relacionados à presença de jogadores que utilizam programas de trapaça nas partidas.

Observamos que no CS:GO, em particular, este problema era o de maior relevância na opinião dos usuários, tendo em vista que cerca de 73,4% dos reviews específicos apresentaram queixas sobre o assunto; além disso, vimos que as reclamações advindas deste problema se estendem por um longo período de tempo, sendo a primeira datada do ano de

2014, até o ano em que o *database* foi criado, 2018. No Quadro 2 é possível observar alguns exemplos de comentários específicos que puderam ser observados nos 4 jogos analisados.

Quadro 2 – Comentários específicos.

Ano	Nome do Jogo	Comentário
2014	Counter-Strike Global Offensive	Conserte este jogo sangrento, como a Valve espera que este seja um bom jogo competitivo, quando cada partida está cheia de trapaceiros que a Valve afirma ter consertado. Eles nem tentam esconder agora, porque o jogo está sempre à venda e eles podem apenas criar uma conta. Pare de atualizar os mapas e comece a consertar o jogo
2015	Counter-Strike Global Offensive	Bons gráficos, novos mapas, armas e modelos. Mas os desenvolvedores devem ouvir os clientes um pouco mais. Desenvolvedores, vocês estão muito focados em coisas que não são nada importantes. Você deve se concentrar em alterar a Tick Rate dos servidores de matchmaking para 128 e melhorar muito o VAC. Esses dois são o que os clientes realmente querem e você deve se concentrar. Não são adesivos, alterações de interface do usuário e HUD ou skins. E pare de mexer com as armas.
2018	Counter-Strike Global Offensive	[...] Há muitos hackers no jogo e não há anti-cheat, não, em vez disso, os desenvolvedores criaram overwatch onde os jogadores decidem se alguém está hackeando, retirando centenas de hackers e também banindo jogadores reais que não estão trapaceando, forçando-os a comprar o jogo novamente ou mudar para outro. Prime não funciona porque existem geradores de números de telefone online para que hackers e smurfs possam entrar no prime usando um número de telefone temporário e uma chave de verificação para obter o Prime e incomodar outros jogadores [...]
2017	World of Warship	É pagar para ganhar combinado com zero proteção contra trapaças. Não acredita em mim? Pesquise world of warships com qualquer mecanismo de pesquisa e, a partir do 4º resultado, haverá sites de aimbot/hack constantes para o jogo. É assim que a trapaça é comum.
2017	World of Warship	Eles tiraram a opção de denunciar pessoas para hackear, então agora está cheio de hackers. Eu vi o hacking aumentar 100% desde que eles fizeram isso. Os hackers podem matá-lo com 1 tiro quando você não tem dmg e vê caras com 6-7 mortes e não pode denunciá-los. Apenas por jogo ruim ou mau comportamento no chat. É estúpido. Este

		jogo está indo ladeira abaixo rapidamente com eles deixando os hackers correrem fazendo o que quiserem. Costumo colocar dinheiro neste jogo, mas não posso mais, a menos que resolvam esse problema.
2018	World of Warship	Agora, 90% dos jogadores estão trapaceando e os criadores não estão fazendo nada a respeito. [...] Eu vejo muitos grandes streamers jogando este jogo e eles ficam irritados por não ter algum Anticheat [...]
2017	Paladins	Gosto muito do jogo, quando os hackers não estão presentes. Há mais e mais hackers a cada dia. Drop hacking é comum. Já tive várias partidas travadas, como se a partida nunca tivesse acontecido. O bloqueio automático é muito comum. Finalmente desinstalei devido aos hackers...
2014	Paladins	Cheio de hackers inúteis que também têm a capacidade de atirar através das paredes, se você tentar jogar algo como bomba, ficará focado durante todo o jogo e se ousar sair do spawn, será morto por um jogador que joga um molly de inferno
2016	Paladins	[...] eu amo o jogo e pretendo jogá-lo muito mesmo depois de escrever isso. Mas caramba, é bug agora. Entendi, está em Beta. No entanto, tivemos um jogo mais estável cerca de 10 patches atrás. Eles continuam atualizando-o com novos campeões, novos cosméticos e algumas mudanças na jogabilidade... tudo ótimo, mas eles ainda precisam resolver os problemas que simplesmente o tornam impossível de jogar. Há um monte de exploradores/trapaceiros no jogo e o sistema "anti-cheat" parece não estar fazendo nada para detê-los. Precisa de trabalho.

Fonte: Extraído do *dataset* do Kaggle (2022, com adaptações).

Em nossa análise, pudemos observar que os jogos que possuem dinâmica de tiro e de combate entre os jogadores sofrem com a presença de trapaceiros. Muito usuários alegam que estes se utilizam de programas para obter vantagens nas partidas, tais como:

- auxílio de mira;
- uso de ferramentas que permitem aos jogadores ver os oponentes através das paredes, como é o caso do CS:GO e no Paladins;
- programas que aumentam a quantidade de dano infligido, que permite eliminar o adversário com apenas um disparo.

Enquanto nos reviews do World of Warship é apontada a falta de ferramentas para combater as falhas de segurança nos comentários do CS:GO, os usuários demonstram insatisfação com os métodos existentes para solucionar este problema. De acordo com os

usuários, em algumas ocasiões os métodos adotados além de serem ineficientes acabam resultando em novos problemas.

Um dos métodos que sofreram muitas críticas por não cumprir o seu propósito de maneira eficiente foi o sistema VAC (Valve Anti-Cheat, que é conhecido no português como Sistema Valve Antitrapaça). Segundo a página de suporte da Steam (s/d), trata-se de uma aplicação lançada no ano de 2015 com o propósito de detectar programas de trapaças que estejam instalados nos computadores dos usuários. O sistema VAC está conectado aos servidores oficiais da Valve para que possa realizar uma varredura, buscando identificar a presença de algum programa irregular, caso a varredura encontre alguma anomalia o jogador será banido, após o banimento é realizada uma validação para confirmar se houve algum engano, uma vez confirmado que o usuário estava trapaceando este não poderá mais recorrer, tendo assim, bloqueado o seu acesso ao servidor.

Para realizar a detecção o sistema deve verificar a presença de assinatura causada por programas de terceiros, caso estes estejam modificando o jogo para oferecer algum tipo de vantagem sobre outros jogadores. Qualquer tipo de alteração nos arquivos executáveis (EXEs) ou nas bibliotecas de ligação dinâmica (DLLs) fará com que o usuário seja adicionado à lista de banimento VAC.

Uma das medidas lançadas pela Valve no ano de 2015 (um ano após os primeiros relatos de problemas com segurança presentes no nosso *database*) para combater as trapaças no CS:GO, a qual também se tornou fonte das críticas nos reviews dos usuários ao longo dos anos, foi o software *overwatch*. De acordo com o site TechTudo (2020), o *overwatch* é um sistema de vigilância, no qual os jogadores mais qualificados, os investigadores, são responsáveis por analisar as denúncias a fim de identificar usuários que estejam usando programas de trapaça.

Para se tornarem investigadores os jogadores devem atender uma série de critérios, quais sejam: possuir um certo número de vitórias no modo competitivo, ter uma determinada quantidade de horas jogadas, ter um número limitado de denúncias e serem jogadores a bastante tempo. Na medida em que um jogador possui tais critérios avaliados positivamente, ele será selecionado e passará a receber casos para analisar, podendo ganhar ou perder pontos pelas análises feitas.

A pontuação atua como um sistema de recompensas com base no veredito dos jogadores. É no decorrer do processo de análise que é definido se os investigadores ganharão ou perderão pontos. Uma vez que estes emitam seus pareceres, aqueles cujos posicionamentos estejam de acordo com os da maioria receberão acréscimo na sua pontuação, caso contrário

sofrerá penalidade. O quantitativo de pontos gera uma hierarquia entre os investigadores, de modo que aqueles com maiores pontuações exercem uma certa influência sobre os demais na hora expor suas opiniões. Para preservar o seu cargo o jogador deve ter certo grau de envolvimento, o que pode resultar em uma compensação, ou seja, ele recebe um maior número de casos o que o possibilita obter mais pontos.

Para que um caso seja encaminhado para o processo de análise é necessário que haja uma série de denúncias. Quando as denúncias chegam às mãos dos investigadores é realizada uma revisão da partida para juntar as informações necessárias e assim formar um parecer. Com relação a esse parecer sobre um determinado caso, existem três possibilidades:

- o caso pode ser considerado inconclusivo, por falta de provas;
- o parecer final pode ser postergado;
- ou quando ocorre veredicto unânime de que há culpa, é aplicada uma penalidade de acordo com a gravidade do problema.

Uma das alegações feitas pelos usuários em seus comentários é de que muitas vezes eles são condenados equivocadamente não tendo como recorrer contra esta decisão. O que demonstra um problema causado por uma medida que deveria servir para corrigir falhas de segurança do jogo, como citado anteriormente.

Em nosso levantamento notamos que muitos usuários do CS:GO expressam sua insatisfação por terem que pagar por recursos que prometem eliminar os problemas com trapaceiros nas partidas, sendo estes recursos oferecidos pelo sistema de contas Prime e a ESEA. De acordo com o site Esports.Net (2022), o Prime é um sistema de assinatura lançado pela Valve no ano de 2016, cuja ideia era que os assinantes teriam à sua disposição uma série de privilégios sobre as demais contas, sendo estes: um sistema de *matchmaking* melhorado, jogar apenas com membros do Prime e a eliminação de partidas com *bots* e hackers.

Para os usuários que compraram o CS:GO, o Prime seria ativado na sua conta automaticamente como uma espécie de gratificação, uma vez que o jogo só passou a ser gratuito em 2018 seis anos após o seu lançamento. O Prime só não seria ativado para o caso em que o jogador tenha removido o número de telefone que estivesse vinculado a sua conta, visto que o status Prime não fica associado a conta e sim ao telefone.

Embora o Prime tenha prometido evitar jogadores que utilizam programas irregulares para obterem vantagens nos jogos, ficou evidente nas avaliações dos usuários que esta promessa nunca chegou a se concretizar, visto que muitos afirmam que os hackers conseguiram burlar o sistema usando números de telefones temporários gerados na internet e através de programas que criam chaves de validação que concedem acesso ao Prime.

Nas avaliações dos resultados notamos que um outro problema apontado pelos usuários está relacionado ao Tick Rate do CS:GO e do Paladins. Tais relatos são frequentes entre os comentários específicos, com uma incidência de 29,8%. De acordo com o site Pinnacle (2020), quando um jogador envia uma instrução ao servidor para que este a execute, é realizada uma coleta de informações necessárias para calcular como esta instrução irá afetar os eventos dentro do jogo, podendo atingir outros jogadores ou até mesmo objetos presentes nas partidas. Uma vez concluídos os cálculos, o servidor deverá enviar os resultados de volta aos jogadores, finalizando assim um Tick.

Ao expressarem suas insatisfações com o Tick Rate do CS:GO, os usuários não evidenciam quais são as falhas do sistema e como este está afetando sua experiência dentro do jogo, é apenas abordado como sendo um aspecto negativo e por meio de sugestões é afirmado que a empresa Valve deveria melhorar o sistema adotando a versão 128 ao invés da versão 64 utilizada até o ano do comentário, 2018. Para compreendermos as indignações dos usuários com a versão do sistema em vigor, realizamos uma busca em sites específicos, para entendermos as principais diferenças entre essas versões.

Nossas buscas nos levaram ao site Pinnacle (2020) que nos mostrou que o diferencial entre as versões está relacionado ao número de instruções que cada uma pode executar em um segundo. Os números 64 e 128 são os que definem a quantidade de instruções que cada servidor pode realizar em um segundo, sendo assim, a segunda versão oferece o dobro de poder de processamento em comparação com a primeira.

Vale destacar que o Tick Rate baixo, por exemplo, pode afetar negativamente a mecânica do jogo. O atraso do tempo de resposta pode levar a falta de precisão nos disparos e atraso na movimentação dos personagens, o que nos leva a uma outra falha apontada nos feedbacks, sendo que esta pode ou não vir a ser consequência do Tick Rate. Em cerca de 24,1% dos comentários do CS:GO e 36,3% nos do Paladins, os usuários apontam a existência de problemas com a mecânica de Hitbox que, segundo o site Game Vicio (2020), essa mecânica é realizada por meio de uma figura geométrica invisível aos olhos dos jogadores.

Em jogos com três dimensões muitas vezes essa figura assume a forma de ortoedro¹³, agindo como um campo que envolve os personagens e os objetos em um videogame, utilizado para detecção de colisões em tempo real. Em jogos de luta por exemplo, quando um personagem realiza a ação de chutar ou de socar um oponente, sua Hitbox entra em contato com a do outro jogador, havendo assim uma colisão que, uma vez registrada, é processada

¹³ De acordo com o site Maestrovirtuale (s/d) é uma figura geométrica tridimensional que se assemelha a um paralelepípedo retangular, cuja as seis faces são paralelas de dois a dois.

pelo servidor para que em seguida seja enviado um sinal como resposta, que varia entre uma animação do personagem ao ser atingido ou a quantidade de danos infligidos pelo impacto.

Em seus relatos muitos usuários apontaram que apesar de acertarem seus alvos, os danos não são registrados, como se não houvesse colisão entre a bala disparada e a Hitbox do outro jogador. Como explicado anteriormente, isso pode estar relacionado ao Tick Rate baixo, uma vez que o atraso no tempo de respostas pode causar a falsa impressão de que um jogador ainda esteja em uma localidade na qual ele já não se encontra mais.

Um problema similar que pode ser consequência da mesma causa, é o Peerker Advantage, que segundo o site 2 A.M. Gaming (sd), é um problema que ocorre quando um jogador, na busca por esconderijo, decide esconder-se atrás de um objeto para evitar ser atingido por um projétil disparado pelo oponente, mas, mesmo que em seu campo de visão tenha conseguido se abrigar, ainda sofre o impacto. Isso ocorre devido a um atraso de resposta uma vez que o oponente tem uma vantagem no tempo de processamento com relação ao jogador.

Um outro fator que podemos citar como sendo o responsável pelo surgimento desses problemas é o Ping alto, que aparece nos comentários de todos os jogos como sendo o principal responsável por gerar problemas de desempenho. De acordo com o site Kaspersky (s/d), os Pings alto e baixo são termos que aparecem com grande frequência na comunidade de jogos online para se referir a velocidade do Ping, uma unidade usada para medir a latência, ou seja, o tempo em milissegundos (ms) que um computador leva para enviar um sinal para o servidor e o tempo que este leva para responder de volta.

Para jogos de tiros como o CS:GO, World of Warship e Paladins que demandam precisão para efetuar disparos e que requer dos jogadores a capacidade de se locomoverem e de se posicionarem rapidamente dentro da partida, faz-se necessário que haja velocidade na comunicação entre o computador e o servidor, desta forma, é desejável que o Ping seja baixo, sendo os valores abaixo de 30 ms considerados como sendo ideais, já os valores que vão de 30 até 75 ms são razoáveis, apontados como na média, e acima disso são considerados indesejáveis, podendo afetar a jogabilidade, causando lentidão e fazendo com que o jogo trave.

De acordo com o site da HP (2022) existem diversos fatores que contribuem para o Ping alto, os mais comuns provêm dos componentes que os jogadores utilizam para se conectarem à internet, entre eles destacamos: a qualidade do provedor de serviço de internet, a velocidade da conexão, a configuração do *firewall* que realiza a verificação de cada pacote de dados quando é enviado ou recebido gerando atraso na comunicação, o tipo de roteador e de

computador que está sendo utilizado para rodar o jogo, entre outros. Existem também fatores que são de responsabilidade da desenvolvedora do jogo, por exemplo, a má distribuição geográfica dos servidores, que por serem distantes da localidade dos jogadores levam mais tempo para efetuarem a transmissão dos dados.

Além disso, temos também a configuração do jogo e como este é otimizado, o que pode gerar lentidão e atrasos no tempo de resposta. No caso do CS:GO estes relatos não são recorrentes nos reviews, aparecendo apenas em um ou dois comentários, deste modo, não podemos afirmar se de fato o problema em questão é resultado de uma falha no software ou se está relacionado aos usuários. Já para os jogadores de Star Trek Online, os problemas de atraso de comunicação e de conexão com os servidores são recorrentes, constituindo cerca de 78% dos relatos específicos. Muitos usuários afirmam que o seu jogo trava com muita frequência e que muitas vezes são desconectados no meio das partidas sem um motivo aparente, apenas mostrando uma mensagem dizendo que o servidor parou de responder.

Ainda neste contexto, fomos capazes de observar que os usuários culpabilizam as empresas desenvolvedoras dos jogos por descaso com relação às falhas dos softwares que perduram por muitos anos, além de acusá-las de serem coniventes com as falhas de segurança. No CS:GO, por exemplo, é apontado que muitos jogadores utilizam programas de trapaça abertamente e que a Valve tem pleno conhecimento disto, mas não adota medidas para a resolução deste problema. Nos comentários fica evidente a insatisfação com as atualizações realizadas que, ao invés de implementar correções, são usadas para acrescentar recursos que, na opinião dos jogadores, não agregam nenhuma melhora na sua experiência, que são vistos por muitos deles como sendo formas de a empresa arrecadar mais dinheiro, o que resulta no desejo de muitos usuários de abandonarem o jogo.

5.2.1.1 Relação entre comentários específicos e usabilidade

Começando pelas reclamações sobre uso de mecanismos que permitem trapaças, fica evidente que há desbalanceamento entre usuários, com os recursos para trapaça, enquanto que outros não possuem este recurso acaba por gerar insatisfação na forma de uso do(s) jogo(s). A satisfação fica comprometida de forma mais evidente quando os sistemas impostos pela empresa não se mostram eficazes para resolver a situação e, na verdade, aumentam mais ainda o abismo de jogabilidade entre os usuários.

Um recurso que foi adotado pelos usuários como alternativa para remediar os problemas com fraudes nos jogos é o ESEA que, segundo os sites ESPN (2015) e TeamPlay

(2010) trata-se de um serviço fundado pela Comunidade de Esportes Eletrônicos, a E-Sports Entertainment Association League. Este serviço está disponível em diversos países ao redor do mundo, incluindo o Brasil, e realiza o pareamento das partidas em seus servidores; deste modo, apenas os jogadores que dispõem desse recurso podem acessá-las. Além disso, cada partida é protegida por um software anti cheat, que foi reconhecido nos Estados Unidos como sendo um dos melhores no mercado, recebendo atualizações frequentemente para melhorar o seu algoritmo na detecção de novos programas hackers, além de manter uma equipe de programadores em tempo real realizando buscas por jogadores que utilizam recursos ilícitos.

Mesmo tendo este recurso (ESEA), a empresa detentora do jogo não o adota, preferindo manter um sistema que, pelos comentários, é pública e notoriamente defesado/ineficiente.

Outro problema de usabilidade apontado pelos usuários está relacionado aos termos: Tick rate baixo, problemas de ping, tempo de resposta, atraso de comunicação ou impossibilidade de conexão com o servidor. Este problema de *delay* de comunicação reflete em sentimento de injustiça quando o resultado final difere daquele que o usuário observou na ação (ele viu o dano aparecer no inimigo, mas o dano não foi aplicado pois na realidade o usuário não havia acertado realmente o inimigo).

Novamente temos uma situação de abismo entre jogadores, seja por:

- terem equipamentos melhores (diferenças nas configurações de hardwares utilizados pelos usuários);
- acesso a serviços de conexão de Internet melhores;
- maior proximidade com as localizações de servidores.

Cabe observar que tais problemas podem ser resolvidos, de forma mais simples, na medida em que as limitações mínimas de recursos para a jogabilidade sejam reais (e não fictícias); ou que a programação dos jogos considere este desbalanceamento; ou que a estrutura de servidores seja adequadamente dimensionada.

Em última análise, o usuário terá sua jogabilidade garantida.

5.2.2 Comentários Genéricos

Relatos de problemas com atualizações são recorrentes entre os comentários generalizados. Em tais relatos essas atualizações são apontadas como sendo as responsáveis por inserirem novas falhas no software. Nesse contexto, alguns usuários afirmam que os problemas das atualizações estão relacionados à falta de testes por parte dos desenvolvedores

durante o processo de implementação. Embora nesses comentários seja evidenciada a existência de tais problemas sua natureza não fica clara, desta forma não é possível fazer sua correção.

No Quadro 3 mostramos alguns exemplos de comentários, nos quais os problemas não são evidenciados.

Quadro 3 – Comentários generalizados.

Ano	Nome do Jogo	Problema
2018	Counter-Strike Global Offensive	Eu não recomendaria entrar nisso em seu estado atual. O CSGO atingiu o fundo do poço com as promessas perdidas da Valve de atualizações que mudam o jogo. Os bugs estão no jogo há anos e ainda não foram corrigidos. [...] Eu realmente não sei por que ainda me esforço para jogar este jogo, porque geralmente acaba me arrependendo. Acho que só gostaria de poder me divertir de alguma forma como costumava jogar este jogo, mas parece que esses dias já se foram.
2016	Counter-Strike Global Offensive	[...] A atualização em 25/10/2016 eles levaram longe demais. CS:GO sempre foi um tema cinza para mim, mas agora eu denunciei publicamente o CS:GO e qualquer um que seja um desenvolvedor, mod ou gerente dele. Vocês venderam suas almas para isso? Vocês são jogadores felizes? É isto o que você queria? 'Este jogo' é uma vergonha para o que o contra-ataque deveria ser. Desenvolvedores de CS:GO, o que vocês fizeram é imperdoável. Vamos apenas esperar que esse embaraço de um jogo seja esquecido em 2022. [...]
2018	World of Warship	Parece divertido na superfície (e está em níveis mais baixos porque as expectativas são baixas), mas uma vez que você entra nele (por exemplo, nível 5 passado), é muito frustrante para o prazer que dá. Há muita dinâmica obscura e o balanceamento acontecendo em segundo plano, o que torna o jogo muito inconsistente de sessão para sessão. Há muita trapaça acontecendo. Os visuais são bons, mas inconsistentes com o mecanismo (e o mapa), por isso geralmente é difícil apontar. Há MUITOS bugs desde a atualização passada.
2018	World of Warship	Sim, é um bom jogo, mas com esta atualização eu meio que me sinto desapontado, espero que o desenvolvedor conserte os bug para que eu possa atualizar uma nova revisão. recomende este jogo para o novo jogador, para que eles possam sentir o gosto da guerra

		naval
2015	Star Trek Online	Este jogo já foi muito bom. No entanto, na última atualização, o jogo parou de funcionar para mim. Cryptic se recusou (não por palavras, mas por ações), a consertar o bug que está afetando a mim e a tantos outros. Se eles consertarem esse jogo, eu ficaria feliz, mas não estou otimista
2015	Star Trek Online	Esse jogo era bom antes das atualizações, agora este jogo se tornou uma festa de bugs. Existem bugs em quase todos os lugares do jogo. Os desenvolvedores não se preocupam em corrigi-los. Eles fecham os olhos para o feedback da base de jogadores. Nos fóruns estão sendo bloqueados tópicos que fornecem feedback negativo sobre o jogo. Eles só se concentram em encher a própria carteira. Eles continuam adicionando conteúdo quebrado e a quantidade de bugs continua aumentando. Eles consertam 1 bug? 10 novos bugs aparecem para substituir o que eles corrigiram. [...]
2017	Paladins	costumava ser um bom jogo agora, honestamente, não posso recomendá-lo devido aos desenvolvedores se concentrarem cada vez mais em monetizar o jogo, em vez de corrigir os bugs e problemas que todas essas "coisas novas" estão causando. não apoie suas decisões horríveis, caso contrário, você terá desperdiçado seu dinheiro em um jogo que não poderá ser jogado no futuro.
2017	Paladins	Revisão da atualização: Então, a última vez que joguei paladins há 3 meses e foi divertido e dou uma recomendação para este jogo, mas depois de visitar paladins, acho que não posso recomendá-lo para novos jogadores porque é tão cheio de bugs e tantos recursos que o deixaram confuso , então eu não recomendei mais este jogo, e mais uma coisa, por que diabos eles fazem moeda VIP [...]

Fonte: Extraído do *dataset* do Kaggle (2022, com adaptações).

A natureza genérica dos comentários tornou difícil a compreensão dos problemas descritos nos mesmos, uma vez que os usuários não descrevem com exatidão quais são as falhas encontradas, além de não indicarem, em sua maioria, quais atualizações foram responsáveis por causarem problemas no software. Uma abordagem que pensamos em adotar para contornar isso seria fazermos um comparativo entre os comentários genéricos e os específicos, de mesma data, para que assim obtivéssemos um entendimento dos problemas.

No entanto, a nossa base de dados não especifica a data exata em que cada comentário foi realizado informando apenas o ano, o que poderia gerar uma análise imprecisa.

A limitação da nossa base de dados dificultou também a identificação das atualizações, as quais os usuários se referiam, uma vez que os jogos receberam diversas atualizações ao longo dos anos, sendo atualizado várias vezes em um mesmo ano. Em apenas um dos comentários conseguimos extrair a data da atualização que foi informada por um usuário do CS:GO, foi através dela que conseguimos realizar uma busca no blog oficial da empresa que desenvolveu o jogo. Através deste blog conseguimos identificar os detalhes da atualização e os possíveis problemas que foram relatados pelos usuários.

Um desses problemas está relacionado a taxa de quadros por segundo (frames per second – FPS) visto que, de acordo com alguns comentários, houve uma queda no valor da taxa fazendo com que este ficasse abaixo do desejável, causando problemas de desempenho no jogo. De acordo com o site AVG Signal (s/d), este problema ocorre quando o computador do usuário não consegue executar o jogo de maneira eficiente causando problemas de lentidão. Dentre os principais causadores deste problema destacamos os drivers desatualizados, o alto consumo da memória RAM e a placa de vídeo que esteja abaixo dos requisitos mínimos.

Nos comentários, os usuários afirmam que o jogo passou a consumir muita memória e que o FPS caiu mais da metade do valor que costumava ser. Em um dos relatos é destacado que o valor da taxa de quadros do CS:GO se limita a valores inferiores a 50 causando problemas no jogo, embora que para alguns jogos valores até 30 FPS não afetam a jogabilidade.

Outro problema que conseguimos identificar envolve o sistema *matchmaking*, este aparece também nos comentários do jogo World of Warship. De acordo com o site Canaltech (2022), trata-se de um sistema responsável por garantir o balanceamento das partidas por meio do gerenciamento dos jogadores envolvidos. O *matchmaking* define quais usuários estarão na mesma rodada com base em critérios definidos pelos desenvolvedores, sendo os dois mais comuns, o nível de experiência de cada jogador e a qualidade da sua conexão. Observamos, ainda, que em alguns comentários os usuários afirmam que não existe balanceamento nas partidas, muitas vezes eles são colocados para competirem com outros jogadores que possuem mais tempo de jogo, o que diminui suas chances de obterem vitórias. Isso acontece também nas formações das equipes quando jogadores de níveis diferentes são colocados no mesmo time, o que na opinião dos usuários resulta na decisão, por parte dos outros jogadores, de expulsar aquele com pior desempenho na partida.

Durante o processo de análise pudemos observar que uma das reclamações mais recorrentes do CS:GO não está diretamente ligada a um problema, mas ao comportamento dos outros jogadores, os quais, segundo a opinião dos usuários, possuem condutas antidesportivas, muitas vezes consideradas abusivas para com os outros. De acordo com os relatos é comum haver troca de ofensas através do chat do jogo, além de que essas ofensas muitas vezes têm teor discriminatório como: comentários racistas, xenofóbicos, machistas entre outros. É apontado ainda que o comportamento tóxico na comunidade do CS:GO não se limita apenas a ofensas, mas muitos jogadores tentam atrapalhar uns aos outros durante as partidas através da eliminação de membros da própria equipe ou pelo banimento dos jogadores que não apresentam um bom desempenho na partida.

5.2.2.1 Relação entre comentários genéricos e usabilidade

Como exposto, no caso dos comentários genéricos há uma dificuldade em se identificar sobre o que o usuário está reclamando, muitas vezes ficando em textos amplos que reclamam de questões, inicialmente, abstratas.

Queda de FPS apesar de genérica, denota que há uma inconstância na execução do jogo, o que obviamente impacta a jogabilidade, principalmente com o sentimento de lentificação das ações executadas pelo usuário e como seu avatar corresponde a estas ações. Cabe observar que aqui que a queda de FPS pode ser por motivos associados a infraestrutura de Internet e localização de servidores. Porém uma situação mais típica vem a ser a defasagem do próprio poder computacional do hardware do usuário. Novamente o jogo peca em não definir os requisitos mínimos para a adequada operação.

Quanto ao *matchmaking* ou mesmo reclamações sobre desbalanceamento. Novamente temos o resultado de programação mal feita. Seja por não considerar adequadamente as variáveis para a formação de times, seja por permitir o embate entre usuários com avatares complementamente diferentes em termos de poder de ataque ou mesmo por inclusão de novos recurso que trazem novas vantagens mas que em comparação com o que já existia destoam totalmente e desequilibram as forças dos avatares.

Por fim, uma das reclamações recorrentes, diz respeito à comunidade tóxica. Em, talvez, uma expectativa da dona do jogo de que haja auto regulação dos usuários ou que estes tenham (e exerçam) o bom senso, os desenvolvedores acabam não tomando providências quanto a estes problemas.

5.2.3 Comentários Sugestivos

Um outro problema que está presente nos jogos World of Warship e o Star Trek Online está relacionado a mecânica de *grinding*, que segundo o site Plarium (2022), é o processo de ter que realizar uma tarefa repetidas vezes, nos jogos, para conseguir subir de nível ou ganhar algum atributo ou item. De acordo com os relatos dos jogadores, esse processo repetitivo é exaustivo e muitas vezes os levam a querer desistir de jogar. É possível notar o incômodo dos usuários com estes problemas, que por meio dos comentários sugestivos, recomendam a inclusão de recursos ou melhorias nos já existentes. No Quadro 4, destacamos alguns desses comentários.

Quadro 4 – Comentários Sugestivos

Ano	Nome do Jogo	Comentário
2018	Counter-Strike Global Offensive	Seria um jogo melhor se tivesse um sistema de report mais eficiente e específico contra toxicidade de outros jogadores e atitudes anti-jogo, ultimamente o server está um lixo, muita toxicidade jogadores que se você reclama uma coisa já é banido por motivo banal precisa melhorar muito, essa comunidade que está quebrada.
2015	Counter-Strike Global Offensive	[...] você verá um flamer ¹⁴ em seu time constantemente tentando chutá-lo ou apenas intimidá-lo para sair abusando do Chat de voz. As razões pelas quais você pode ser chutado em partidas competitivas são tão imaturas e tristes que se tornaram um problema sério: 1. É claro que se você tiver a pontuação mais baixa nas primeiras 4 rodadas, eles não hesitaram em votar em chutá-lo (mesmo que sua pontuação seja apenas 1 ou 2 abaixo deles). Deveria haver algum tipo de moderação. [...]
2016	Counter-Strike Global Offensive	Tentei explicar para mais de 1.000 jogadores que eu tinha um total de 4 horas, mas começaram a me chamar de ♥♥♥♥♥ e ♥♥♥♥♥♥. [...] Tentei ser sorrateiro virando as esquinas lentamente [...] foi dito "mova seu ♥♥♥♥♥♥♥♥ retardado" então assim que eu larguei o turno fui instantaneamente atingido por uma pistola do outro lado do mapa e, em seguida, fui expulso por ser novo no csgo. Adicione algo como um campo de treinamento no menu "play" para noobs vs noobs
2017	World of Warship	Não me interpretem mal, eu gosto deste jogo, mas este

¹⁴ A definição de flamer no dicionário é uma pessoa que envia mensagens de correio eletrônico abusivas. (Educalingo, s/d)

		jogo parece um simulador de grinding gigante. Especialmente quando você chega nos níveis médio-altos, isso pode tornar o jogo muito obsoleto e chato às vezes, eu gostaria de ver um pouco mais de experiência distribuída nas batalhas
2018	World of Warship	[...] Não há muitos jogos em que você tenha batalhas PvP na virada do século até os navios do final da Segunda Guerra Mundial. Se é isso que você quer, esta é uma maneira muito divertida de obtê-lo se você permitir "algumas" imprecisões históricas. Não é perfeito, matchmaking pode ser frustrante, equilíbrio é quase inexistente e o grinding é real, mas você não realmente precisa gastar dinheiro para se divertir, embora provavelmente ajude se você quiser pular alguns dos navios sem brilho em uma linha. [...]
2015	Star Trek Online	jogo este jogo desde antes de seu lançamento oficial e, honestamente, se não fosse pelo fato de ser o ÚNICO jogo de Star Trek que eu teria deixado há muito tempo. Mesmo antes da temporada Delta Rising, o grinding é bastante intenso apenas para obter equipamentos decentes ou itens de evento... é preciso mudar a dinâmica do para não ficar tão repetitiva [...]
2015	Star Trek Online	Um jogo que você realmente precisa passar a vida inteira trabalhando, a menos que gaste uma pequena fortuna. Para subir de nível ou obter itens decentes é preciso fazer as mesmas missões repetidas vezes [...] 99% é grinding 1% é gameplay

Fonte: Extraído do *dataset* do Kaggle (2022, com adaptações).

Em nossa análise, observamos que muitos relatos sobre o CS:GO, entre os anos de 2015 e 2018, apontavam que o comportamento prejudicial de alguns jogadores afetava a qualidade das partidas, de modo que, muitos usuários, sugeriram que a Valve deveria adotar um sistema que aplicasse penalidades a fim de corrigir estes problemas. Embora já exista a suspensão e o banimento do modo competitivo que, segundo a página suporte da Steam (s/d) e o site da Techtudo (2018), a partir do momento em que um jogador recebe uma série de denúncias sobre o seu comportamento cabe ao sistema de Overwatch averiguar a conduta deste jogador para que então seja aplicada a suspensão, que o impedirá de participar, temporariamente, dos jogos no modo competitivo.

Para o CS:GO existe um código de conduta a ser seguido, caso um jogador viole alguma de suas normas este será penalizado. A seguir destacamos alguns dos comportamentos

em que é cabível a suspensão: causar danos a um aliado, trapacear, se desconectar diversas vezes das partidas, expulsar muitos membros da sua equipe, se ausentar no meio do jogo entre outros. Em seus relatos alguns usuários afirmam que apesar do sistema atual denúncias lidar com os casos de trapaça e com o comportamento tóxico dos jogadores seria necessário criar um sistema específico para reportar esse e outros casos de má conduta¹⁵.

No nosso estudo observamos que uma outra funcionalidade sugerida pelos usuários surgiu como consequência das falhas do sistema de *matchmaking* e com o comportamento tóxico dos outros usuários. Uma vez que não há balanceamento no gerenciamento de partidas, nas quais jogadores com menor grau de experiência são colocados em times compostos por jogadores de níveis mais altos, os quais impõem pressão sobre aqueles que não possuem as mesmas habilidades. Deste modo, surgiu a necessidade de adicionar uma opção de treinamento para que os jogadores que estão começando possam aperfeiçoar suas habilidades para que, assim, possam melhor competir com os demais.

Em seus relatos os usuários sugerem que, para resolver problemas com *grinding* é preciso acrescentar algo mais a história do jogo, de modo que este tenha uma jogabilidade mais dinâmica para que não se torne algo tão monótono e cansativo. O site Medium (2019) sugere algumas abordagens que podem ser utilizadas para contornar este problema, primeiramente seria preciso tornar o jogo mais desafiador, mesmo que o enredo do jogo não seja diversificado. Ter desafios que aumentem gradativamente a dificuldade do jogo, testando as habilidades dos jogadores pode melhorar a experiência dos jogadores com o software, mantendo os usuários mais engajados. Outra abordagem seria fragmentar a jogabilidade em pequenos pedaços, ou seja, ao invés de fazer com que o jogador realize uma tarefa de uma só vez é possível reparti-la em partes menores e ir intercalando essas partes com outras missões. Em nossa pesquisa, observamos ainda que a amostra dos dados coletada sobre o jogo Paladins não continha comentários nos quais os usuários indicassem sugestões de acréscimo de novos recursos.

5.2.3.1 Relação entre comentários sugestivos e usabilidade

Comentários sugestivos, normalmente podem trazer a tona problemas já expostos de forma genérica (ex.: *matchmaking*, desbalanceamento, comunidade tóxica). Mas também podem ser observadas sugestões que podem ser incorporadas no jogo, tais como formas de

¹⁵ De acordo com o site Techtudo (2020) a Valve começou a implementar um sistema para combater o comportamento tóxico de jogadores no chat de voz. A nova funcionalidade tem como objetivo colocar os usuários que forem denunciados por cometer abuso no modo silencioso, os impedindo de falar durante as partidas.

melhoria para reduzir/evitar problemas de *grinding*, *matchmaking*, desbalanceamento. Muitas vezes os comentários sugestivos podem (devem) ser vistos como um *brainstorming* para novas funcionalidades e/ou soluções inovadoras para problemas antigos.

O sistema de comentários permite algo extra que é a discussão da ideia pela própria comunidade, que pode validar/identificar como interessante ou não.

6 CONSIDERAÇÕES FINAIS

A presente pesquisa trouxe uma descrição sobre as etapas do processo de desenvolvimento de uma aplicação de aprendizagem de máquina, empregada no processamento de linguagem natural para fazer a análise de sentimentos em textos. Neste sentido, o programa desenvolvido teve como finalidade identificar comentários negativos dos usuários a fim de determinar as falhas de usabilidades presentes nos jogos eletrônicos escolhidos para análise, desta forma, poderíamos apontar, de acordo com a visão dos usuários, como estas falhas afetam as experiências dos jogadores e as insatisfações que os levam à muitas vezes desistirem dos jogos ou os avaliarem mal.

Para a evolução do nosso trabalho foi importante a realização de um estudo inicial das técnicas envolvidas na construção de um modelo de aprendizagem de máquina, para que fossemos capazes de entender as diferenças entre os principais tipos de modelos: classificação, regressão e clusterização. Além disso, foi preciso compreender as diversas abordagens adotadas para realizar o treinamento do modelo, como por exemplo: o treinamento supervisionado, o treinamento não supervisionado e o treinamento de reforço. Uma vez que conseguíssemos entendê-los, saberíamos quais destes seriam aplicados no desenvolvimento do nosso projeto.

Durante a implementação do nosso trabalho, uma das principais dificuldades encontradas foi durante o processo de coleta de dados. Existem muitos fatores que devem ser considerados para se criar uma base de dados que será utilizada no treinamento de um modelo voltado para a análise de sentimentos, um deles é o volume de dados que pode afetar o desempenho do modelo, caso o conjunto de dados de treinamento seja pequeno. Além disso, é preciso verificar se este conjunto é bem balanceado, ou seja, se não existe uma quantidade maior de comentários com sentimento positivo do que negativo, e vice-versa, este problema pode afetar a precisão do modelo causando classificações erradas. Uma outra dificuldade encontrada durante a coleta de dados foi com relação aos mecanismos adotados pelos sites para bloquearem o uso de *scripts* que automatizam a extração dos dados, por isso, tivemos que mudar nossa abordagem optando por utilizarmos uma base de dados pronta.

Com a aplicação do nosso modelo conseguimos analisar os reviews negativos, o que nos permitiu identificar e extrair os problemas de usabilidades apontados pelos usuários, obtendo, assim, um resultado satisfatório para o objetivo proposto pelo nosso trabalho. Em nossa pesquisa, observamos que os jogos escolhidos para análise apresentam falhas similares, isto pode estar ligado a natureza desse jogos, visto que, por serem todos voltados para uso

online, é de se esperar que as queixas estejam diretamente relacionadas a problemas de desempenho causados por falha na conexão ou por atraso na comunicação com os servidores.

Em nossa pesquisa, observamos que os problemas de usabilidade expostos pelos usuários estavam diretamente relacionados ao desbalanceamento da jogabilidade, este problema surge através da aparente inabilidade das empresas responsáveis em combater os mecanismos de trapaças que concedem vantagens a um determinado grupo de jogadores. Este desbalanceamento se torna mais grave quando os desenvolvedores falham em criar um sistema de *matchmaking* que seja capaz de equilibrar as partidas, através da formação dos times compostos por usuários com o mesmo nível de experiência, causando assim, um sentimento de injustiça entre os jogadores. Essa sensação de injustiça é agravada pela lentidão causada por problemas no FPS e na conexão com os servidores, que aparentemente beneficiam os jogadores que possuem maior poder computacional.

Além disso, notamos que os usuários não se limitaram a expor problemas de usabilidade em suas análises, mas sugeriram que os desenvolvedores pudessem incorporar novos recursos aos jogos como forma de corrigir falhas e evitar outros possíveis problemas que possam surgir.

Durante o processo de análise dos resultados, observamos que a atividade de revisar os comentários para identificar quais apresentam problemas de usabilidade pode ser considerada uma tarefa muito mecânica, que requer muito tempo para fazer algo repetitivo, desta forma, em um possível projeto futuro poderíamos aplicar os dados extraídos dos resultados deste trabalho para treinar um outro modelo de aprendizagem de máquina para que este consiga identificar termos de usabilidade presentes nos comentários dos usuários.

REFERÊNCIAS

2 A. M. GAMING. **O que é tick rate e porque isso influencia o seu desempenho nos games online.** Disponível em:

<https://blog.2amgaming.com/2020/04/o-que-e-tick-rate-influencia-desempenho-games/>.

Acesso em: 30 de jan. 2023.

ABNT. **Requisitos Ergonômicos para trabalho de escritórios com computadores:** Parte 11 – Orientações sobre Usabilidade. ABNT/CB-21/SC-10 – Subcomite de Software ABNT/CE-21.101.08 – Comissão de Ergonomia de Software Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs)/ Part 11 – Guidance on usability.

Disponível em:

https://www.inf.ufsc.br/~edla.ramos/ine5624/_Walter/Normas/Parte%2011/iso9241-11F2

Acesso em: 7 fev. 2023.

ALURA. **Guia de NLP – conceitos e técnicas.** Disponível em:

<https://blog.2amgaming.com/2020/04/o-que-e-tick-rate-influencia-desempenho-games/>.

Acesso em: 04 fev.2023.

ALURA. **Letmmatication vs. Stemming:** quando usar cada uma? Disponível em:

<https://www.alura.com.br/artigos/lemmatization-vs-stemming-quando-usar-cada-uma#:~:text=Na%20stemming%20vamos%20analisar%20cada,com%20valor%20para%20nossa%20an%C3%A1lise>. Acesso em: 22 de jan. 2023.

ALURA. **Pandas:** o que é, para que serve e como instalar. Disponível em:

<https://www.alura.com.br/artigos/pandas-o-que-e-para-que-serve-como-instalar>. Acesso em: 19 fev. 2023.

ALURA. **PLN:** o que é Processamento de Linguagem Natural? Disponível em:

<https://www.alura.com.br/artigos/o-que-e-pln>. Acesso em: 22 fev. 2023.

APIFY DOCS. **Anti-scraping techniques.** Disponível em:

<https://www.octoparse.com/blog/google-play-scraper#:~:text=You%20can%20scrape%20the%20app,API%20might%20not%20be%20feasible>. Acesso em: 17 de jan. 2023.

AVG. **Como aumentar o FPS e otimizar o PC para jogos.** Disponível em:

<https://blog.counter-strike.net/pt-br/index.php/2016/12/17300/>. Acesso em: 07 fev. 2023.

AWS. **O que é Python?** Disponível em: <https://aws.amazon.com/pt/what-is/python/>. Acesso em: 16 fev. 2023.

BEAUTIFUL SOUP. **Beautiful Soup Documentation.** Disponível em:

<https://canaltech.com.br/seguranca/o-que-e-web-scraping/>. Acesso em: 08 fev. 2023.

BETWAY INSIDER. **O que é o CS:GO? Por que tanta gente ama esse jogo?** Disponível em:

<https://blog.betway.com/pt/esports/o-que-%C3%A9-o-csgo-por-que-tanta-gente-ama-esse-jogo/>. Acesso em: 21 de jan. 2023.

CANALTECH. **O que é matchmaking?** Disponível em:

<https://blog.counter-strike.net/pt-br/index.php/2016/12/17300/>. Acesso em: 07 fev. 2023.

CANALTECH. **O que é web scraping e como ocorre?** Disponível em: <https://canaltech.com.br/seguranca/o-que-e-web-scraping/>. Acesso em: 12 mar. 2023.

COUNTER STRIKE. **Detalhes de atualização.** Disponível em: <https://blog.counter-strike.net/pt-br/index.php/2016/12/17300/>. Acesso em: 07 fev. 2023.

DIDÁTICA TECH. **A biblioteca scikit-learn – Python:** o que é, para que serve. Disponível em: <https://didatica.tech/a-biblioteca-scikit-learn-pyhton-para-machine-learning/>. Acesso em: 19 fev. 2023.

EDUCALINGO. **Dicionário:** significado de “flamer”. Disponível em: <https://educalingo.com/pt/dic-en/flamer>. Acesso em: 08 fev. 2023.

ESPN. **‘CS:GO’:** ESEA volta ao Brasil com novidades. Disponível em: http://www.espn.com.br/noticia/564899_csgo-esea-volta-ao-brasil-com-novidades#:~:text=A%20ESEA%2C%20ou%20e%2DSports,n%C3%ADveis%20e%20software%20anti%2Dcheat. Acesso em: 26 de jan. 2023.

ESPORTS.NET. **CS:GO Prime:** o que é, como funciona e qual o valor? Disponível em: <https://www.esports.net/br/guias/csgo-prime/#:~:text=Em%20suma%2C%20o%20CSGO%20Prime,de%20n%C3%ADvel%20e%20dropar%20skins>. Acesso em: 24 de jan. 2023.

FARIA, Rafael; ÂNGELO, Celso; **Análise de informações:** mineração de dados. Disponível em: <https://www.grancursosonline.com.br/>. Acesso: 2 fev. 2023.

GAME VICIO. **Como as hitboxes funcionam.** Disponível em: <https://www.teamplay.com.br/noticias/esea/5139-entenda-mais-sobre-a-esea>. Acesso em: 30 de jan. 2023.

GAMERS & GAMES. **Quer jogar Paladins?** Confira nosso guia de iniciantes no jogo! Disponível em: <https://www.gamersegames.com.br/2018/06/12/quer-jogar-paladins-confira-nosso-guia-de-iniciante-no-jogo/>. Acesso em: 08 mar. 2023.

GÉRON, A. **Mãos à Obra:** Aprendizado de Máquina com Scikit-Learn & TensorFlow. Rio de Janeiro: Alta Books, 2019.

HASHTAG. **O que é o Kaggle?** Entenda e saiba como começar a usá-lo. Disponível em: <https://www.hashtagtreinamentos.com/kaggle>. Acesso em: 18 de jan. 2023.

HP. **Deu lag! Travou o jogo!** Os 5 principais motivos pelos quais o seu ping é tão alto. Disponível em: <https://blog.2amgaming.com/2020/04/o-que-e-tick-rate-influencia-desempenho-games/>. Acesso em: 01 fev. 2023.

JOGOS MMORPG. **Star Trek Online.** Disponível em: <http://www.jogos-mmorpg.com/mmorpg/star-trek-online/>. Acesso em: 07 mar. 2023.

JUPYTER NOTEBOOK. **The Jupyter Notebook.** Disponível em: <https://jupyter-notebook.readthedocs.io/en/stable/notebook.html>. Acesso em: 16 fev. 2023.

KAGGLE. **Kaggle**: your machine Learning and Data Science community. Disponível em: <https://www.kaggle.com/>. Acesso em: 08 fev. 2023

KASPERSKY. **Como reduzir o ping e otimizar o desempenho de jogos online**. Disponível em: <https://blog.2amgaming.com/2020/04/o-que-e-tick-rate-influencia-desempenho-games/>. Acesso em: 01 fev. 2023.

LIVE #6: ANÁLISE DE SENTIMENTO UTILIZANDO DADOS DO TWITTER (PASSO A PASSO), 2019. Vídeo (108 min). Publicado pelo canal stack. Disponível em: https://www.youtube.com/watch?v=Zy5sHkr5W7s&ab_channel=Stack. Acesso em: 10 jan. 2023.

MAESTROVIRTUALE. **Ortoedro**: fórmulas, área, volume diagonal, exemplos. Disponível em: <https://maestrovirtuale.com/ortoedro-formulas-area-volume-diagonal-exemplos/>. Acesso em: 15 mar. 2023.

MARSLAND, S. **Machine Learning**: An Algorithmic Perspective. 2. ed. Boca Raton: Chapman & Hall/CRC, 2015.

MEDIUM. **Os truques que grandes jogos fazem para superar o grinding**. Disponível em: <https://medium.com/level-ing/os-truques-que-grandes-jogos-fazem-para-superar-o-grinding-9289673e6647>. Acesso em: 09 mar. 2023.

MEDIUM. **Web scraping com Python para preguiçosos (unindo BeautifulSoup e Selenium)**. Disponível em: <https://medium.com/data-hackers/web-scraping-com-python-para-pregui%C3%A7osos-unindo-beautifulsoup-e-selenium-parte-2-8cfebf4f34e>. Acesso em: 08 fev. 2023.

OCTOPARSE. **Best Google Play Scraper to Extract Application Data and Reviews**. Disponível em: <https://www.octoparse.com/blog/google-play-scraper#:~:text=You%20can%20scrape%20the%20app,API%20might%20not%20be%20feasible>. Acesso em: 17 de jan. 2023.

PINNACLE. **Servidores com tick rate de 128 ou 64**: qual a diferença? Disponível em: <https://www.teamplay.com.br/noticias/esea/5139-entenda-mais-sobre-a-esea>. Acesso em: 30 de jan. 2023.

PLARIUM. **Mine, Level Up, Repeat**: The Ups and Downs Of Grinding Games. Disponível em: <https://plarium.com/en/blog/grinding-games/>. Acesso em: 09 mar. 2023.

STEAM. **CS:GO – Suspensões e banimentos do modo competitivo**. Disponível em: <https://help.steampowered.com/pt-br/faqs/view/7580-7BDB-1C9A-AC67>. Acesso em: 09 fev. 2023.

STEAM. **Sistema Valve Antitrapaça (VAC, do inglês Valve Anti-Cheat)**. Disponível em: <https://help.steampowered.com/pt-br/faqs/view/571a-97da-70e9-ff74>. Acesso em: 23 de jan. 2023.

TEAMPLAY. **Entenda mais sobre a ESEA**. Disponível em: <https://www.teamplay.com.br/noticias/esea/5139-entenda-mais-sobre-a-esea>. Acesso em: 26 de jan. 2023.

TECHTUDO. **MMORPG Star Trek Online será lançado para PS4 e Xbox One.**

Disponível em:

<https://www.techtudo.com.br/noticias/2016/05/mmorpg-star-trek-online-sera-lancado-para-ps4-e-xbox-one.ghtml>. Acesso em: 07 mar. 2023.

TECHTUDO. **Como funciona o Overwatch no CS:GO?** Entenda o sistema de vigilância.

Disponível em:

<https://www.techtudo.com.br/noticias/2020/08/como-funciona-o-overwatch-no-csgo-entenda-sistema-de-vigilancia-esports.ghtml>. Acesso em: 23 de jan. 2023.

TECHTUDO. **CS:GO ganha novo sistema para silenciar jogadores tóxicos**

automaticamente. Disponível em:

<https://www.techtudo.com.br/noticias/2020/02/csgo-ganha-novo-sistema-para-silenciar-jogadores-toxicos-automaticamente-esports.ghtml>. Acesso em: 09 fev. 2023.

TECHTUDO. **Download de Counter Strike: Global Offensive (CS:GO) é grátis para PC.**

Disponível em: <https://www.techtudo.com.br/tudo-sobre/counter-strike-global-offensive/>.

Acesso em: 21 de jan. 2023.

TECHTUDO. **O que é toxicidade?** Entenda o significado de postura proibida em jogos.

Disponível em:

<https://www.techtudo.com.br/noticias/2018/07/o-que-e-toxicidade-entenda-significado-de-postura-proibida-em-jogos-esports.ghtml>. Acesso em: 09 fev. 2023.

TECHTUDO. **Review Paladins.** Disponível em:

<https://www.techtudo.com.br/review/paladins.ghtml>. Acesso em: 08 mar. 2023.

WIZARD. **Os prefixos e sufixos em inglês.** Disponível em:

<https://www.wizard.com.br/idiomas/os-prefixos-e-sufixos-em-ingles/>. Acesso em: 22 de jan. 2023.

WORD OF WARSHIP. **Noções Básicas.** Disponível em:

<https://worldofwarships.com/pt-br/content/education/basics/>. Acesso em: 07 mar. 2023.

APÊNDICE A – APRESENTAÇÃO DAS BIBLIOTECAS

Pandas

A biblioteca *Pandas* requer instalação prévia para ser utilizada. Existem duas formas principais de realizar essa instalação: uma é através da distribuição Anaconda, utilizando o comando `conda install pandas`; outra é através do gerenciador de pacotes do Python, o pip, utilizando o comando `pip install pandas` na linha de comando. Para obter mais informações sobre a instalação do Pandas, recomenda-se consultar a documentação oficial da biblioteca, disponível em: https://pandas.pydata.org/docs/getting_started/index.html.

Para verificar se a instalação foi bem-sucedida, basta importar a biblioteca no código do programa utilizando o comando `import pandas as pd`. Se a instalação apresentar algum problema, o interpretador Python irá indicar que o módulo não foi encontrado; caso contrário, já será possível acessar todos os recursos disponíveis na biblioteca. Após a instalação e a importação da biblioteca, é possível carregar uma base de dados para um *DataFrame* utilizando o comando `df = pd.read_csv("arquivo.csv")`, no qual o método `read_csv` recebe como argumento o nome ou o caminho do arquivo CSV, e atribui o *DataFrame* à variável de nome `df` (*dataframe*).

O *DataFrame* é uma estrutura de dados tabular que permite trabalhar com os dados de forma eficiente e flexível. O *DataFrame* oferece uma série de funções que possibilitam realizar buscas, filtragens e manipulações das informações através de comandos como: `query()`, `iloc()` e `rename()`. Esses comandos recebem diferentes tipos de parâmetros para realizar operações específicas. Para mais informações sobre como usar esses comandos e muitos outros, pode-se consultar o site: https://pandas.pydata.org/docs/user_guide/basics.html.

NumPy

A biblioteca *NumPy* pode ser instalada por meio da distribuição Anaconda ou do gerenciador de pacotes pip, utilizando os mesmos comandos que se utilizaria para instalar o *Pandas*, alterando apenas os nomes das bibliotecas. Por exemplo: `conda install numpy` e `pip install numpy`.

Após a instalação do *NumPy*, ele estará disponível para uso. Para isso, é necessário importá-lo usando o `import numpy as np`. Depois de importar a biblioteca, é possível acessar todas as suas funções e classes usando o prefixo `np`. Por exemplo, para criar uma das principais estruturas presentes nessa biblioteca, o array multidimensional, pode-se usar o

comando `np.array()`, que recebe como parâmetro um vetor, uma lista ou uma sequência de objetos. A estrutura de arrays do *NumPy* pode ser aplicada no aprendizado de máquina para criar e modificar representações de textos, fazer diversas operações matemáticas avançadas entre arrays, entre outros. Para compreender melhor essa estrutura, pode-se consultar a documentação oficial no site: <https://numpy.org/doc/stable/user/whatisnumpy.html>.

A estrutura de arrays dessa biblioteca possibilita realizar diversas operações matemáticas, entre elas, operações aritméticas: como a soma, multiplicação, divisão e subtração. Usando os operadores aritméticos (+, -, /, *) pode-se fazer operações entre dois ou mais arrays. Essa operação é feita combinando cada elemento de um array com o elemento correspondente do outro array. Por exemplo, nos comandos `vetor = np.array([1,2,3])`, `vetor2 = np.array([4,5,6])`, `vetor + vetor2` são formados dois arrays numéricos que são atribuídos às variáveis `vetor` e `vetor2`, respectivamente. Em seguida, essas variáveis são somadas. Nesse caso, cada elemento do primeiro vetor é adicionado a um elemento do segundo vetor que ocupe a mesma posição.

Outro tipo de operação que podemos ter são os operadores trigonométricos, como: seno, cosseno, tangente, arco seno, entre outras. O *Numpy* oferece um conjunto de operadores que podem ser aplicados nos seus vetores de forma eficiente. É possível aprender mais sobre eles na página: <https://numpy.org/doc/stable/reference/routines.math.html>.

Scikit-learn

O *Scikit-learn* é uma biblioteca de aprendizado de máquina que contém uma ampla gama de algoritmos para diversas finalidades, tais como: pré-processamento, classificação, regressão, agrupamento, dentre outras. Para instalar o *Scikit-learn*, é necessário utilizar o comando `pip install -u scikit-learn`. Para verificar se a instalação foi realizada corretamente, podemos utilizar o comando `python -m pip show scikit-learn` que informará a versão e o local de instalação da biblioteca.

Neste trabalho, foi empregado um modelo de classificação supervisionado do *Scikit-learn*, mas há outras funcionalidades da biblioteca que podem ser consultadas na documentação oficial na página: <https://scikit-learn.org/stable/index.html>.

As ferramentas do *Scikit-learn* que se empregaram na construção do modelo foram: `train_test_split`, que permite realizar a divisão dos dados em conjuntos de treino e de teste, `MultinomialNB`, que possibilita treinar o modelo utilizando algoritmos baseados no teorema de Naive Bayes, e `accuracy_score`, que fornece uma medida de avaliação do modelo baseada

na proporção de acertos. Para utilizar essas ferramentas, é preciso importá-las com os seguintes comandos: `from sklearn.model_selection import train_test_split, from sklearn.naive_bayes import MultinomialNB e from sklearn.metrics import accuracy_score.`

A função para dividir os dados recebe os seguintes argumentos `x_train`, `x_test`, `y_train`, `y_test = train_test_split (dado_de_entrada, dado_de_saida, test_size = 0.20, random_state = 42)`, onde o `test_size` indica a proporção dos dados que serão usados para o teste (neste caso, 20%) e o restante será usado para o treino. O argumento `random_state` define como os dados serão embaralhados antes da divisão entre treino e teste. Além disso, a função recebe o `dado_de_entrada`, que consiste na base de dados já processada, e o `dado_de_saida`, que consiste no rótulo correto da classificação dos dados.

O passo seguinte é criar o modelo para que este seja treinado, atribuindo a função `MultinomialNB` a uma variável denominada `multi`, da seguinte forma: `multi = MultinomialNB()`. Após criar o modelo, deve-se treiná-lo utilizando o comando `multi.fit(x_train,y_train)`, no qual se passam os dados de entrada de treino (`x_train`) e os dados de saída de treino (`y_train`). Agora que o modelo foi treinado, deve-se validá-lo utilizando o comando `previsao = multi.predict (x_test)` para que o modelo classifique os dados de entrada de teste (`x_test`) e atribua o resultado a uma variável chamada `previsao`, que será utilizada como argumento para a função `accuracy_score()` juntamente com os dados de saída de teste (`y_test`), da seguinte forma: `acuracia = accuracy_score(previsao, y_test)`. Essa função irá comparar os dados previstos pelo modelo na variável `previsao` aos dados esperados na variável `y_test` e então retornará a proporção de acerto.