

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Aplicação de técnicas de aprendizado de máquina em um sistema de recomendação de jogos eletrônicos

Deborah de Azevedo Salomão Condé

JUIZ DE FORA
DEZEMBRO, 2023

Aplicação de técnicas de aprendizado de máquina em um sistema de recomendação de jogos eletrônicos

DEBORAH DE AZEVEDO SALOMÃO CONDÉ

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Orientador: Saulo Moraes Villela

JUIZ DE FORA
DEZEMBRO, 2023

APLICAÇÃO DE TÉCNICAS DE APRENDIZADO DE MÁQUINA
EM UM SISTEMA DE RECOMENDAÇÃO DE JOGOS
ELETRÔNICOS

Deborah de Azevedo Salomão Condé

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS
EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTE-
GRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE
BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Saulo Moraes Villela
D.Sc. em Engenharia de Sistemas e Computação (COPPE/UFRJ)

Eduardo Barrére
D.Sc. em Engenharia de Sistemas e Computação (COPPE/UFRJ)

Victor Ströele de Andrade Menezes
D.Sc. em Engenharia de Sistemas e Computação (COPPE/UFRJ)

JUIZ DE FORA
11 DE DEZEMBRO, 2023

Dedico este trabalho aos meus pais, Eduardo e Vânia, cujo apoio incansável e crença constante foram meus guias para que eu chegasse onde estou hoje. Expresso também minha dedicação aos meus filhos de estimação, aqueles que partiram e os que ainda caminham ao meu lado. Sua constante presença e amor incondicional têm sido uma luz nos meus dias mais sombrios.

Resumo

No universo dos jogos eletrônicos, uma indústria global que movimenta bilhões, existem diferentes tipos de jogador. Richard Bartle propôs uma classificação em quatro arquétipos: *Achievers* (Conquistadores), *Explorers* (Exploradores), *Killers* (Assassinos) e *Socializers* (Socializadores). Este trabalho busca empregar essas características para classificar jogos eletrônicos e, por meio de técnicas de aprendizado de máquina, formular recomendações personalizadas de jogos com base no perfil de cada jogador. A premissa é que jogos com perfis semelhantes ao do jogador têm maior probabilidade de serem apreciados por ele, fundamentando assim a criação de recomendações adaptadas e personalizadas. Este estudo visa, portanto, usufruir de conhecimentos sobre o usuário para induzir as escolhas de jogos, contribuindo para uma interação mais significativa e satisfatória no universo dos jogos eletrônicos.

Palavras-chave: Inteligência Artificial, Jogos Eletrônicos, Taxonomia de Bartle, Aprendizado de Máquina. Perfil do Jogador, Recomendação

Abstract

In the universe of electronic games, a global industry that generates billions in revenue, various player types exist. Richard Bartle proposed a classification into four archetypes: Achievers, Explorers, Killers, and Socializers. This work aims to leverage these characteristics to categorize electronic games and, through machine learning techniques devise personalized recommendations based on each player's profile. The premise is that games with profiles similar to the player's are more likely to be enjoyed by them, thus endorsing the creation of personalized recommendations. This study, therefore, seeks to benefit from knowledge about the user to induce game selection, contributing to a more meaningful and satisfying interaction within the realm of electronic gaming.

Keywords: Artificial Intelligent, Electronic Games, Bartle Taxonomy, Machine Learning, player's profile, recommendation

Agradecimentos

A minha família, por todo o apoio e companheirismo.

Aos professores do Departamento de Ciência da Computação, em especial meu orientador, Saulo Moraes Villela, pela orientação e paciência com minhas constantes perguntas, e ao professor Eduardo Barrére por ter me dado inspiração e apoio durante todo o curso.

Aos amigos de coração, esses vocês sabem quem são.

“A tecnologia move o mundo, mas é a humanidade que o transforma.- Desconhecido”.

Conteúdo

Lista de Figuras	7
Lista de Tabelas	8
Lista de Abreviações	9
1 Introdução	10
2 Fundamentação Teórica	12
2.1 Taxonomia de Bartle	12
2.2 Sistemas de Recomendação	14
2.2.1 Conteúdo	14
2.2.2 Colaborativa	14
2.3 Aprendizado de Máquina	15
2.3.1 <i>K-Means</i>	15
2.3.2 <i>K-Nearest Neighbour</i>	17
2.3.3 Máquina de Vetores Suporte	17
2.3.4 Árvore de Decisão	18
2.3.5 Floresta Aleatória	19
2.3.6 <i>K-Fold Cross-Validation</i>	19
3 Abordagem Proposta	21
3.1 Base de Dados	22
3.1.1 Jogadores	23
3.1.2 Jogos	25
3.2 Agrupamento	31
3.3 Classificação e Avaliação	33
3.4 Relação Jogo-Jogador	34
3.5 Geração de Novos Dados e Recomendação	34
4 Experimentos e Resultados	36
4.1 Jogadores	36
4.2 Jogos	37
4.3 Avaliação	38
4.4 Relação Jogo-Jogador e Grafo Resultante	39
5 Considerações Finais	44
A Representação dos Clusters	46
Bibliografia	55

Lista de Figuras

2.1	Plano cartesiano de Bartle.	13
2.2	Exemplo do <i>K-Means</i>	16
2.3	Exemplo de classificação utilizando algoritmo KNN	18
3.1	Perfil dos jogadores.	26
3.2	Conjuntos de gêneros	30
3.3	Perfil dos jogos	31
4.1	Coefficiente da Silhueta para Jogadores	36
4.2	Método do cotovelo para jogadores	37
4.3	Coefficiente de Silhueta para Jogos	37
4.4	Método do cotovelo para Jogos	38

Lista de Tabelas

3.1	Perguntas do questionário de Bartle	24
3.2	Features de Jogadores	33
3.3	Features de Jogos	33
4.1	Acurácia e desvio padrão dos algoritmos para jogadores	38
4.2	Acurácia e desvio padrão dos algoritmos para jogos	39
4.3	Exemplo de grafo gerado. <i>Clusters</i> de jogadores como linhas e de jogos como colunas.	40
4.4	Previsão de novos jogadores.	42
4.5	Distância entre jogador e clusters de jogo	42
4.6	Distancia do jogador para todos os jogos do cluster 3	42
4.7	Previsão de novos jogos	43
4.8	Distancia entre jogo e <i>clusters</i> de jogadores	43
A.1	Número de jogadores agrupados por cluster	46
A.2	Número de jogos agrupados por cluster	46
A.3	Dados de players já classificados	49
A.4	Dados de jogos já classificados	54

Lista de Abreviações

FPS	<i>First Person Shooter</i>
KNN	<i>K-Nearest Neighbour</i>
MMORPG	<i>Massive Multiplayer Online Role Playing Game</i>
MUD	<i>Multi-User Dungeon</i>
NPC	<i>Non-Playable Character</i>
RPG	<i>Role Playing Game</i>
SVM	<i>Support Vector Machine</i>
TPS	<i>Third Person Shooter</i>

1 Introdução

A indústria de jogos eletrônicos é uma das mais lucrativas do mundo e movimenta bilhões de dólares anualmente. Segundo o relatório “Global Games Market Report” da Newzoo, em 2022 o mercado global de jogos gerou uma receita de US\$ 184,4 bilhões e espera atingir US\$ 220,79 bilhões no ano de 2023¹.

No mercado nacional pesquisas apontam que 66,3% dos brasileiros jogam algum tipo de jogo eletrônico, o que equivale a cerca de 84,4 milhões de pessoas, e isso não deveria ser surpresa: videogames tem estado entre nós por décadas através das mais variadas plataformas, dos fliperamas e consoles caseiros até consoles portáteis e celulares; frequentemente são o que há de mais atual em tecnologia.

Os denominados “jogadores” podem ser classificados segundo diferentes tipos. (BARTLE, 1996) criou uma taxonomia tipológica de jogadores (arquétipos), chamada de Taxonomia de Bartle, para caracterizar participantes de jogos online, como MMORPGs (*Massive Multiplayer Online Role Playing Game*) e MUDs (*Multi-User Dungeons*). Sua abordagem foi tão reconhecida pela comunidade de jogadores que sua taxonomia é utilizada até hoje por *game designers* na criação de seus jogos (FERRO; WALZ; GREUTER, 2013)

O propósito deste trabalho é aplicar a Taxonomia de Bartle, que categoriza jogadores em quatro arquétipos distintos (*Achievers*, *Explorers*, *Killers* e *Socializers*), para classificar jogos. Utilizando técnicas de aprendizado de máquina, o objetivo é agrupar jogos e jogadores que compartilham características semelhantes e prever os grupos de jogos e jogadores futuros. Através desse agrupamento, pretende-se estabelecer relações entre cada jogo com cada jogador com base em seus perfis, os quais são formados pelos arquétipos de Bartle. Dessa forma, torna-se possível recomendar jogos a jogadores cujos perfis estejam próximos, proporcionando uma abordagem personalizada na recomendação de jogos eletrônicos.

Este trabalho é composto por 4 capítulos, além desta introdução. No Capítulo 2

¹<https://newzoo.com/games-market-reports-forecasts#get-in-touch>

são revisadas as teorias, conceitos, pesquisas e estudos relevantes sobre o tema em questão: quais foram os algoritmos de aprendizado de máquina implementados para a recomendação aos usuários. No Capítulo 3 é apresentado em detalhes quais estratégias serão usadas para realização do experimento, como algoritmos e métodos utilizados, além de fornecer uma compreensão mais aprofundada sobre a metodologia utilizada para coletar e analisar dados relacionados aos jogos e jogadores. No Capítulo 4 são explicitados os resultados obtidos e, em conclusão, o Capítulo 5 resume os resultados e indica desdobramentos para trabalhos futuros nesta mesma linha.

2 Fundamentação Teórica

Neste capítulo são apresentados os fundamentos teóricos que estabeleceram a base para este trabalho.

2.1 Taxonomia de Bartle

Em 1978, Richard Bartle e Roy Trubshaw, alunos da Universidade de Essex, na Inglaterra, criaram em colaboração um jogo de computador revolucionário que se tornou a base para os jogos de RPG que conhecemos hoje. Esse jogo foi chamado de “MUD” (Masmorra com Múltiplos Jogadores) (KYATRIC, 2021). Conhecido como “MUD1”, esse foi um dos primeiros jogos online baseados em texto, sem gráficos, mas com uma característica única para a época: permitia que os jogadores interagissem simultaneamente pela internet. O MUD1 desempenhou um papel fundamental como precursor dos jogos MMO (Massively Multiplayer Online). Embora os MUDs originais fossem jogos de aventura em texto, eles abriram caminho para uma nova forma de entretenimento interativo e social online. Baseado em suas criações de anos atrás, Richard Bartle escreveu o artigo “HEARTS, CLUBS, DIAMONDS, SPADES: PLAYERS WHO SUIT MUDS” (BARTLE, 1996) em 1996. Nesse artigo, o autor formulou uma teoria que classifica os jogadores de MUDs e, por extensão, os jogadores de MMORPGs, em quatro tipos principais:

- *Achievers* (Conquistadores): Esses jogadores estão focados em alcançar resultados, obter pontuações altas, subir de nível e conquistar áreas do mapa. Eles se importam com o tempo decorrido para chegar onde estão e a conquista é o que mais importa. Eles estão interessados na ação dentro do mundo do jogo;
- *Explorers* (Exploradores): Os exploradores se deleitam em descobrir os mecanismos do jogo. Eles exploram todos os cantos do mapa, procuram por novos itens, interagem com Personagens Não Jogáveis (*Non-Playable Characters* – NPCs) e resolvem quebra-cabeças criados pelos desenvolvedores. Sua maior satisfação está na descoberta e eles estão interessados na interação com o mundo do jogo;

- *Socializers* (Socializadores): Os socializadores estão interessados nas pessoas. Para eles, os jogos são apenas um meio para o propósito final, que é a troca de experiências e relacionamentos com outros jogadores. Eles valorizam a interação com os demais e até mesmo observar o comportamento dos outros é uma fonte de interesse para eles;
- *Killers* (Assassinos): Os assassinos são movidos pela competição e desejam lutar contra outros jogadores, bem como contra NPCs. Eles estão focados em derrotar todos os demais, liderar placares e produzir caos no jogo. Eles estão interessados em agir contra os outros jogadores.

Esses quatro tipos representam diferentes motivações e estilos de jogo, dentro dos MUDs e MMORPGs, e a teoria de Bartle tem sido amplamente discutida e aplicada no estudo da psicologia dos jogos.

Como demonstrado pela Figura 2.1, tem-se dois eixos onde suas extremidades representam dados distintos. No eixo que podemos chamar de “X” temos jogadores em uma ponta e o mundo (do jogo) em outra; no eixo “Y” temos interação e ação. Os quadrantes resultantes são os arquétipos de Bartle.

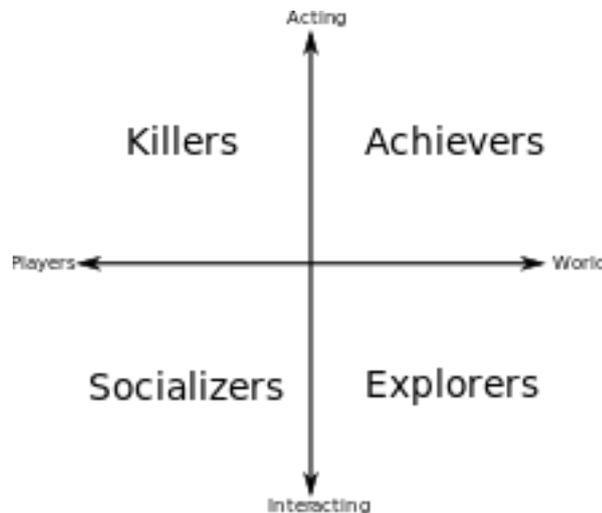


Figura 2.1: Plano cartesiano de Bartle.

Esse trabalho propõe utilizar a taxonomia citada para classificar jogadores do formato “Single Player”, ou “único jogador”. Além disso, busca-se aplicar a taxonomia para classificar os próprios jogos de videogame, buscando assim uma similaridade jogador a partir de características, comportamentos e estilo de jogo.

2.2 Sistemas de Recomendação

Na internet, onde a quantidade de opções é tão intensa, existe um grande número de aplicações web que trabalham com previsões, chamadas de Sistemas de Recomendação. Existem dois grandes exemplos desse tipo de sistema: Recomendação Baseada em Conteúdo, e Recomendação Baseada em Filtragem Colaborativa: (MEDEIROS, 2012)

Estas aplicações oferecem ao usuário, direta, ou indiretamente, meios para a tomada de decisão com base nas preferências de cada um, normalmente obtendo dados de navegação via *cookies*. (TIMES, 2021). Deste modo, os usuários podem receber recomendações mais assertivas sobre as suas preferências ou visualizar anúncios mais específicos ao navegar na web.

2.2.1 Conteúdo

A filtragem baseada em conteúdo é fundamentada na similaridade entre os itens recomendados; se um usuário gosta de um determinado item, também gostará de um item semelhante. Para essa recomendação é necessária uma representação para o conteúdo, como a data de uma notícia, categoria de um produto e/ou gênero de um livro.

Este sistema normalmente opera a partir de dados fornecidos pelo próprio usuário em forma de classificação e pode separar dois conjuntos: os interesses e os desinteresses do usuário. Dessa forma, o sistema deverá ser capaz de identificar a qual conjunto pertence os itens que ainda não foram avaliados pelo usuário, por meio de algoritmos de Aprendizado de Máquina.

2.2.2 Colaborativa

Próximo das recomendações baseadas em similaridade, a filtragem colaborativa tem o princípio básico de identificar usuários semelhantes. Como exemplo, um usuário deve indicar itens, positiva ou negativamente, permitindo a recomendação de algo que outro usuário aprovou ou desaprovou por ter uma opinião, ou perfil, próximo do seu.

Um exemplo real desse tipo de sistema é o Netflix (NETFLIX, 2021), onde os usuários podem opinar, dentre os programas assistidos, se recomendariam ou não tal

atração.

2.3 Aprendizado de Máquina

Aprendizado de Máquina é o nome dado à tecnologia capaz de reconhecer padrões em um grande volume de dados e criar previsões a partir de informações pré existentes.

Existem três tipos de algoritmos de aprendizado de máquina: os algoritmos supervisionados, aqueles não supervisionados e os reforçados.

Como o nome indica, os algoritmos supervisionados dependem de intervenção humana, onde “O aprendizado a partir de um conjunto de dados pode induzir um viés — as máquinas podem repetir preconceitos humanos. Já teve caso de uma máquina dizer que um homem negro era um gorila porque o banco de dados analisado não tinha diversidade étnica. A ciência de dados é muito importante, porque a forma como você estabelece os dados é o que define os resultados” (Steven Choi, 2019).

Nos algoritmos não supervisionados, a máquina começa a analisar por si os dados e a identificar os padrões — aprendendo a separar o que é uma lata de uma garrafa, por exemplo. Como é a máquina aprendendo por si só conceitos que nunca viu antes, o processo é mais demorado e, portanto, não tão popular.

Por fim, os algoritmos reforçados são particularmente relevantes nos ambientes nos quais o “agente”, a máquina, interage com um sistema dinâmico. No aprendizado reforçado, o agente toma decisões sequenciais com o objetivo de maximizar uma recompensa cumulativa ao longo do tempo. Esse paradigma é inspirado na forma como os seres humanos aprendem por tentativa e erro, ajustando seu comportamento com base nas consequências de suas ações. Os algoritmos de aprendizado reforçado têm sido aplicados com sucesso em uma variedade de domínios, incluindo jogos eletrônicos.

2.3.1 *K-Means*

O algoritmo *K-Means* desempenha um papel fundamental na classificação de informações com base nos próprios dados. Ele realiza essa classificação por meio da análise e comparação dos valores numéricos presentes nos dados. Uma das principais vantagens neste

caso é que ele pode fornecer uma classificação sem a necessidade de intervenção humana, o que torna-o um algoritmo não supervisionado.

O algoritmo recebe como entrada um conjunto de dados C e o número desejado de partições k . A partir daí ele lê os dados como uma tabela, identifica as classes (*clusters*) e quais as linhas da tabela pertencem a cada classe. O valor de k fornecido é utilizado para separar os dados em k classes, o que originou o nome do algoritmo.

Para gerar as classes e classificar as ocorrências, o algoritmo compara cada valor de cada linha usando uma medida de distância, geralmente a distância euclidiana, e esta comparação determina a proximidade entre as ocorrências. Em seguida são calculados os centroides para cada uma das classes. Conforme o algoritmo itera, o valor de cada centroide é refinado pela média dos valores de cada atributo das ocorrências que pertencem a este centroide. Dessa forma, o algoritmo gera k centroides e posiciona as ocorrências na tabela com base em sua distância em relação aos mesmos. (AARSETH, 2001)

Como resultado do processo, cada objeto é atribuído a um único *cluster*, permitindo a classificação eficiente dos dados. O algoritmo *K-Means* é amplamente utilizado em aplicações de mineração de dados, análise de agrupamentos e segmentação de dados, proporcionando *insights* valiosos sobre a estrutura subjacente dos conjuntos de dados.

A Figura 2.2 ilustra, à esquerda, os dados antes da etapa de clusterização; à direita, após a classificação dos dados.

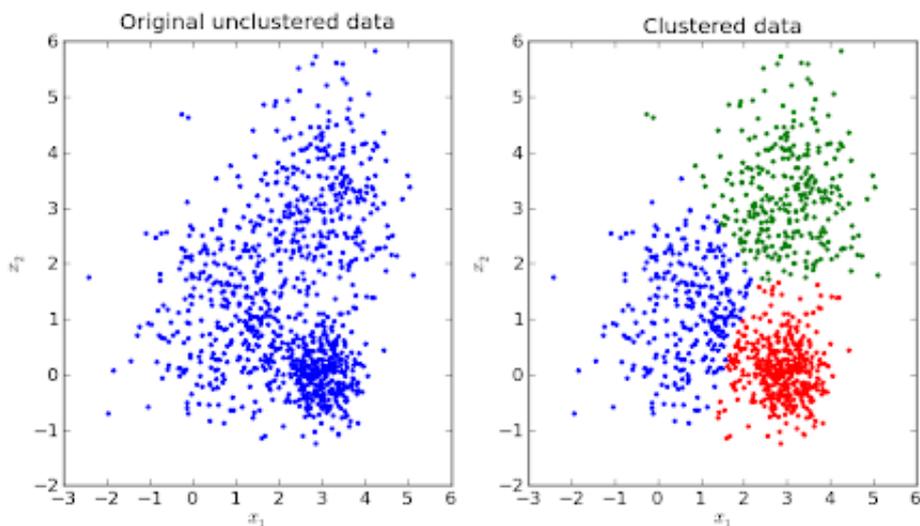


Figura 2.2: Exemplo do *K-Means*

2.3.2 *K-Nearest Neighbour*

O *K-Nearest Neighbour* (KNN) é um algoritmo de classificação supervisionado que se baseia na premissa de que objetos semelhantes estão próximos uns dos outros. Ele captura esta ideia de similaridade calculando a distância entre os objetos ou vértices de um grafo.

A entrada do algoritmo consiste em um conjunto de elementos a serem avaliados, chamado de “data”. Além disso, é fornecido um valor constante k , que representa o número de vizinhos desejado. Por exemplo, em modelos binários geralmente temos $k = 2$. Também é enviado um exemplo de consulta, chamado de “query”. Para cada item no conjunto de dados, calculamos a distância entre a *query* e o item atual, adicionando-os a uma coleção. No final, essa coleção é ordenada com base no valor de k especificado.

No entanto, o algoritmo KNN possui algumas limitações e desafios. Um deles é a determinação do valor de k de forma manual, o que pode exigir ajustes experimentais para obter resultados satisfatórios. Além disso, a influência da distância euclidiana na medida de similaridade pode ter impacto no desempenho do algoritmo. Por exemplo, o algoritmo assume que os atributos são numéricos, o que pode ser problemático com atributos categóricos em problemas reais. Além disso, a escala utilizada por cada atributo pode influenciar a medida de distância e, conseqüentemente, os resultados do algoritmo.

Apesar dessas limitações, o algoritmo KNN é amplamente utilizado em problemas de classificação, reconhecimento de padrões e sistemas de recomendação. Ele se destaca por sua simplicidade e interpretabilidade, permitindo uma abordagem intuitiva para a análise de dados e a tomada de decisões com base na proximidade e similaridade entre os objetos.

Na Figura 2.3 é possível visualizar que, em sua maioria, objetos similares estão localizados nas proximidades uns dos outros.

2.3.3 Máquina de Vetores Suporte

Máquina de Vetores Suporte (*Support Vector Machine* – SVM) é um meio de aprendizado de máquina supervisionado que pode ser usado para classificação ou regressão. Seu foco maior é no treinamento e classificação de um grupo de dados.

A ideia principal por trás do SVM é criar um hiperplano de decisão, que maximize

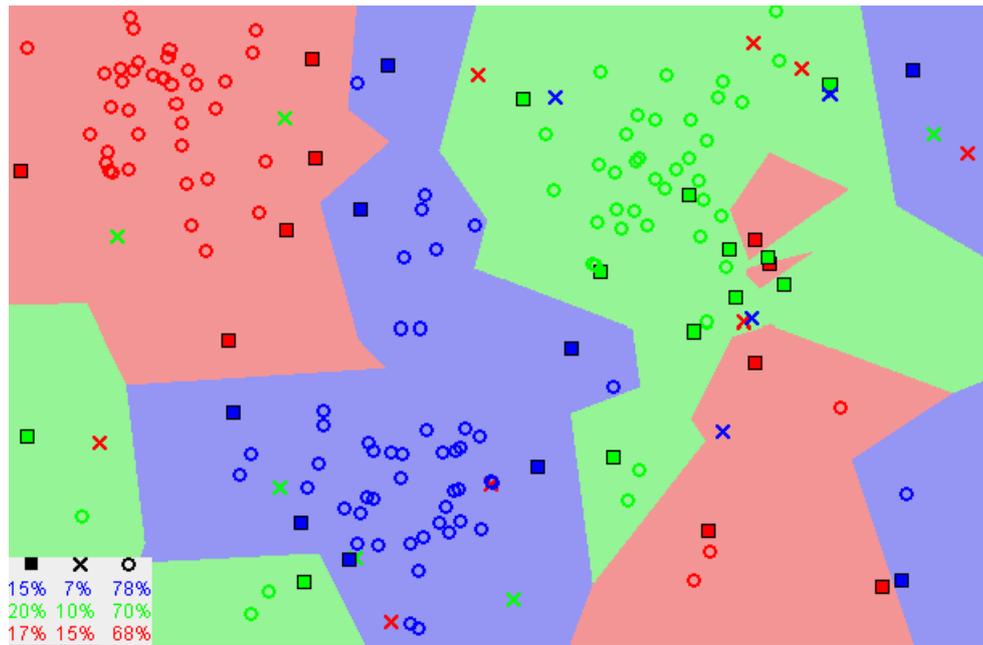


Figura 2.3: Exemplo de classificação utilizando algoritmo KNN

a margem entre os pontos de dados de diferentes classes. Essa margem é definida como a distância entre o hiperplano e os pontos de dados mais próximos, chamados vetores de suporte. O objetivo é encontrar o hiperplano que separa as classes de forma mais eficiente possível, minimizando erros de classificação e maximizando a generalização para novos dados.

Uma das vantagens do SVM é sua eficácia em lidar com conjuntos de dados de alta dimensionalidade, além de sua capacidade em utilizar dados não linearmente separáveis usando *kernels* adequados. No entanto, o desempenho do SVM pode ser afetado pela escolha adequada de parâmetros, como o tipo de *kernel* e a configuração da margem flexível.

2.3.4 Árvore de Decisão

Este algoritmo se baseia na estrutura de uma árvore, tomando decisões sequenciais que dividem o conjunto de dados em subconjuntos cada vez mais homogêneos. Cada nó interno representa uma escolha baseada em uma característica específica, enquanto as folhas representam as decisões finais ou os resultados previstos. Durante o treinamento, o algoritmo busca otimizar a escolha das características e os pontos de divisão para maximizar a pureza dos subconjuntos resultantes.

É uma técnica poderosa para análise e classificação de dados, sendo especialmente eficaz em contextos nos quais a interpretabilidade do modelo é crucial. Ao construir a árvore, o algoritmo busca automaticamente as características mais discriminativas e os pontos de decisão ideais para guiar as escolhas durante a classificação. Essa estrutura hierárquica facilita a compreensão do processo decisório do modelo, permitindo *insights* claros sobre quais características são mais influentes na classificação final.

2.3.5 Floresta Aleatória

A Floresta Aleatória é um algoritmo que pertence à categoria de métodos *ensemble*, conhecidos por combinar múltiplos modelos para formar uma solução mais robusta e precisa. Este algoritmo é uma extensão da abordagem de árvore de decisão.

Ao invés de depender de uma única árvore de decisão, a Floresta Aleatória cria várias árvores de decisão durante o processo de treinamento. Cada árvore é construída com uma amostra aleatória dos dados de treinamento, e em cada decisão ao longo da árvore, um subconjunto aleatório das características é considerado. Esse processo introduz diversidade no treinamento, uma vez que cada árvore está sendo desenvolvida com uma perspectiva única dos dados. O poder da Floresta Aleatória surge quando essas árvores individuais são combinadas para formar uma decisão coletiva. Durante a fase de predição, cada árvore contribui com sua própria previsão, e a resposta final é determinada por meio de uma votação majoritária.

2.3.6 *K-Fold Cross-Validation*

Para avaliação dos modelos gerados por algoritmos de aprendizagem, é comum a utilização de um conjunto de dados (com rótulos conhecidos) não utilizados durante a fase de treinamento para avaliação do desempenho do modelo (FACELI K., 2011). Um dos métodos utilizados para essa avaliação é o *K-Fold Cross-Validation*. Neste método, o conjunto de dados é dividido em k partes (ou dobras) de tamanhos iguais, onde uma delas é utilizada como conjunto de teste e as outras $k-1$ partes são usadas como conjunto de treinamento. Esse processo é repetido k vezes, cada vez com uma parte diferente como conjunto de teste, e as métricas de desempenho são calculadas e posteriormente medidas.

Isso ajuda a avaliar a robustez do modelo, fornecendo uma estimativa mais confiável do seu desempenho em dados não vistos, minimizando o risco de superestimar a capacidade do modelo.

Uma variação deste método é chamada de *Stratified K-Fold Cross-Validation*. Nesta variação a divisão dos dados em subconjuntos é feita de forma a manter a proporção das classes em cada subconjunto. Isso significa que, se o conjunto de dados possui múltiplas classes, a estratificação garante que cada subconjunto contenha uma representação proporcional de cada classe, mantendo a mesma distribuição observada no conjunto de dados original. Essa técnica é essencial para a validação de modelos, ajudando a evitar o *overfitting* (ajuste excessivo) e fornecendo uma estimativa mais precisa do desempenho do modelo em dados reais. Ela é amplamente usada na prática de aprendizado de máquina e é um componente crítico na seleção de hiperparâmetros e na avaliação do desempenho de modelos de *machine learning*.

3 Abordagem Proposta

Este capítulo apresenta passo a passo a abordagem que será utilizada na realização do experimento proposto.

Em resumo: o primeiro passo consiste em aplicar a mesma técnica para classificar jogadores e jogos, utilizando a taxonomia de Bartle e as características presentes nessas duas entidades. Os dados classificados resultarão em dois datasets, que serão analisados e processados em um ambiente *Python*.

Dada as bases de dados resultantes, é necessário agrupar jogos e jogadores próximos, pois esta proximidade é a premissa para a recomendação. O algoritmo K-Means será utilizado para gerar o agrupamento. Com os dados agrupados, é possível treinar modelos de aprendizado de máquina para a classificação de novos dados.

Após o agrupamento, é criado um novo dataset, gerado pelas relações entre jogo/-jogador. Essa relação é dada pela interação de cada jogador com todos os jogos da base, e a interação se resume na multiplicação das colunas semelhantes entre jogo e jogador. São elas *Achiever*, *Explorer*, *Killer* e *Socializer*.

Com isso é possível gerar um grafo bipartido completo a partir dessas relações, onde os vértices do grafo serão as classes (grupos) de jogos e jogadores, e as arestas a relação entre os *clusters*.

Feito o grafo, são gerados novos jogos e jogadores e estes são classificados de acordo com os algoritmos previamente treinados, e inseridos nos seus respectivos *clusters*. Com isso, basta calcular a distância euclidiana entre vértices e encontrar o *cluster* de menor distância de cada novo jogador. O vértice encontrado será o mais próximo daquele usuário.

Basta agora recomendar os jogos presentes naquele *cluster* resultante da menor distância à aquele jogador.

Para realização dos passos citados, coleta e análise dos dados utilizados e o desenvolvimento das recomendações, foi empregado o ecossistema SciPy (JONES et al., 2001–2017), que engloba pacotes de software de código aberto para matemática, ciência

e engenharia. Destacam-se aqui os pacotes Scikit-learn, Pandas, Numpy e Matplotlib. A implementação foi realizada no ambiente Colab do Google².

A escolha desta linguagem de programação deve-se ao seu destaque na área de aprendizado de máquina, conforme mencionado em (RASCHKA; MIRJALILI, 2019).

3.1 Base de Dados

Esta sessão apresenta as bases de dados utilizadas para o experimento, uma vez que originaram do trabalho de Conde (2021) e foram reutilizadas como meio de persistência para novos dados coletados e resultados obtidos. No início, os dados foram mantidos independentes, uma vez que o objetivo central do trabalho consiste em estabelecer relações entre duas entidades distintas, jogador e jogo, a fim de torná-las comparáveis. Com essa finalidade foram mantidas as tabelas fundamentais: “Perguntas”, “Perfil_Jogador”, e “Perfil_Jogo”. A tabela resultante “Jogo_Jogador” foi recriada, uma vez que os resultados encontrados no trabalho original não fazem sentido nesta aplicação. A descrição detalhada de cada uma dessas tabelas segue:

A tabela “Perguntas” engloba um total de 30 perguntas provenientes do questionário de Bartle, que foi empregado como instrumento para a obtenção de dados relativos ao perfil do jogador.

A tabela “Perfil_Jogador”, abriga as respostas fornecidas por cada indivíduo participante do experimento.

A tabela “Perfil_Jogo” armazena informações referentes a 144 jogos distintos, compreendendo campos como “Título”, “Gênero”, e as quatro colunas correlacionadas aos arquétipos propostos por Bartle.

Por fim, a tabela “Jogo_Jogador” desempenha o papel crucial de estabelecer a relação entre as tabelas “Jogo” e “Jogador”, oferecendo uma indicação dos jogos mais recomendados para determinados tipos de jogadores com base nos resultados alcançados por meio do experimento.

²<https://colab.research.google.com>

3.1.1 Jogadores

Quando Richard Bartle redigiu seu artigo, também desenvolveu uma ferramenta de avaliação psicológica com o propósito de categorizar indivíduos de acordo com seus tipos de jogadores. Denominada “The Bartle Test of Gamer Psychology” (BARTLE, 2021), essa ferramenta consiste em uma série de perguntas e cenários relacionados às preferências e comportamentos em jogos. Os participantes são convidados a escolher suas respostas favoritas, classificando-as com base em suas inclinações pessoais. As respostas dadas auxiliam na identificação dos tipos de jogadores com os quais o indivíduo mais se identifica.

A abordagem proposta fará uso do mesmo questionário, que foi reproduzido utilizando a plataforma Google Forms³ e distribuído por e-mail para alunos dos cursos de ciências exatas da Universidade Federal de Juiz de Fora, bem como para colaboradores das empresas de tecnologia “CI&T” e “Thomson Reuters”. Essa amostra foi selecionada devido à sua proximidade com o campo tecnológico, o que provavelmente aumentaria a probabilidade de compreensão das perguntas apresentadas. Tais questões frequentemente envolvem terminologia específica, muitas vezes encontrada em contextos fantasiosos presentes em jogos eletrônicos.

A Tabela 3.1 apresenta as 30 perguntas do questionário de Bartle, utilizado para capturar os dados relativos ao perfil dos jogadores:

³forms.google.com

Perguntas	Opção A	Opção B
1. Você como um jogador fica mais confortável:	Conversando com amigos na taverna	Lá fora caçando Orcs pela experiência
2. Em uma aventura você prefere?	Se envolver na história	Colher as recompensas no final
3. Você prefere ser reconhecido	Por seu equipamento	Por sua personalidade
4. Com qual opção você se diverte mais em um jogo online?	Saber sobre a fofoca mais atual	Coletar itens novos
5. O que você como um jogador online preferiria ?	Um servidor privado onde você e seus amigos podem se comunicar	Sua própria casa que vale um milhão em moedas de ouro
6. O que você gostaria mais?	Comandar sua própria taverna	Fazer seus próprios mapas dos mundos e vendê-los
7. O que para você é mais importante em um jogo?	Número de jogadores	Número de lugares para explorar
8. O que para você é mais importante em um jogo?	A qualidade do "role-playing"	A qualidade dos recursos e das mecânicas únicas
9. Você está sendo perseguido por um monstro o que você faz?	Pede ajuda de um amigo para destruí-lo	Se esconde em algum lugar que o monstro não irá segui-lo
10. Você está prestes a entrar em um templo desconhecido e pode escolher mais uma pessoa para acompanhá-lo você escolhe	O bardo que é um bom amigo seu e está sempre animando você e seus amigos	Um mago que sabe identificar os itens que você encontrar
11. Você prefere	Conquistar seus inimigos	Convencê-los a trabalhar para você e não contra você
12. O que é mais empolgante?	Um cenário interativo	Uma batalha mortal
13. O que você gostaria mais	Ganhar um duelo contra outro jogador	Ser aceito dentro de um clã (grupo de outros jogadores)
14. É melhor ser	Amado	Temido
15. Você prefere	Ouvir o que o outro tem a dizer	Mostrar a ele a ponta afiada do seu machado
16. Você liberou uma nova área do mapa o que faz primeiro?	Explorar a nova área e descobrir sua história	Ser o primeiro a coletar os itens novos
17. Você prefere ser reconhecido como:	Alguém que conhece o mapa inteiro de cor e sabe chegar em qualquer lugar	Alguém que possui o melhor e único equipamento do jogo
18. Você prefere	Se tornar um herói mais rápido que seus amigos	Saber mais segredos que seus amigos
19. Você prefere	Saber onde achar itens	Saber como ter itens
20. Qual você prefere?	Resolver um mistério que ninguém mais conseguiu	Chegar em um alto nível de experiência mais rápido que outros jogadores
21. Você entraria em qual clã	Acadêmicos	Assassinos
22. Você preferiria ganhar	Um campeonato de trivia	Uma batalha de arena
23. Se você está sozinho em uma área você pensa	Que é seguro explorar	Que você precisará procurar sua vítima em outro lugar
24. Você descobre que outro jogador está planejando seu fim você:	Vai para uma área a qual sabe que seu oponente não tem prática e o espera lá	O ataca antes dele atacar você
25. Você conhece um novo jogador você acha que ele(a) é:	Alguém que aprecia o seu conhecimento do jogo	Ou uma vítima em potencial
26. Você em um jogo	Prefere ter uma espada duas vezes mais poderosa que qualquer outra	Ser o personagem mais temido
27. Em um jogo você se gabaria por	Quanto personagens você já matou	Seu equipamento
28. Você prefere ter	Um encantamento para machucar outros personagens	Um encantamento para aumentar a quantidade de pontos de experiência que você ganhará
29. Você prefere ganhar ao completar uma missão	Pontos de experiência	Uma varinha com três cargas de um ataque que deixe você controlar outros personagens contra a vontade deles
30. O que é mais divertido?	Ter a pontuação mais alta da lista	Ganhar do seu melhor amigo em um contra um

Tabela 3.1: Perguntas do questionário de Bartle

Um total de noventa e quatro pessoas responderam. Esses resultados foram exportados para um arquivo .csv e, em seguida, um código foi desenvolvido utilizando a linguagem C#.Net para calcular os pesos correspondentes a cada uma das respostas (CONDE, 2021). Posteriormente, esses resultados foram incorporados à tabela SQL, denominada “Perfil_Jogador”, e estão disponíveis para visualização abaixo:

A Figura 3.1 contém os dados dos respondentes com as colunas: Nome, *Achiever*, *Explorer*, *Killer* e *Socializer*. Registre-se que os valores somados dos pesos sempre dará 100%. O conteúdo da coluna Nome foi borrado a fim de preservar a privacidade dos respondentes, visto que são dados reais.

Em uma análise inicial é possível afirmar que, por mais que as pessoas tenham uma porcentagem de todos os arquétipos, um sempre será mais predominante. Isso tampouco quer dizer que podemos definir um jogador por um único tipo, mas sim que as pessoas são uma composição dos quatro, o que aumenta a complexidade de definição.

3.1.2 Jogos

Com os dados dos jogadores devidamente armazenados foi necessário coletar dados de diferentes jogos eletrônicos, uma vez ser o objetivo final a recomendação de jogos por similaridade. A forma encontrada para transformar duas entidades tão distintas como jogos e pessoas foi categorizar também os jogos segundo os quatro arquétipos de Bartle. Para isso primeiramente, foi feita uma análise pela autora baseando-se nos gêneros e na pesquisa feita por Bartle.

Jogos eletrônicos possuem diversos gêneros e subgêneros que podem variar de acordo com os aspectos de jogabilidade. Alguns dos gêneros mais populares de jogos eletrônicos são:

- Ação
- Aventura
- Ação/Aventura
- Sandbox / Mundo Aberto

DB Browser for SQLite - C:\Users\debor\source\repos\tcc_exatas\tcc_exatas\TccDB.db

Arquivo Editar Exibir Ferramentas Ajuda

Novo banco de dados Abrir banco de dados Escrever modificações Reverter modificações Abrir projeto Salvar projeto

Estrutura do banco de dados Navegar dados Editar pragmas Executar SQL

Tabela: Perfil_Jogador Filtrar em qualquer coluna

ID	Nome	Achiever	Explorer	Killer	Socializer
Filtro	Filtro	Filtro	Filtro	Filtro	Filtro
1		20.0	33.33333333333333	13.33333333333333	33.33333333333333
2		10.0	33.33333333333333	16.66666666666667	40.0
3		20.0	36.66666666666667	20.0	23.33333333333333
4		13.33333333333333	26.66666666666667	16.66666666666667	43.33333333333333
5		26.66666666666667	23.33333333333333	26.66666666666667	23.33333333333333
6		20.0	40.0	13.33333333333333	26.66666666666667
7		20.0	40.0	16.66666666666667	23.33333333333333
8		26.66666666666667	43.33333333333333	3.33333333333333	26.66666666666667
9		26.66666666666667	40.0	10.0	23.33333333333333
10		36.66666666666667	16.66666666666667	40.0	6.66666666666667
11		23.33333333333333	26.66666666666667	26.66666666666667	23.33333333333333
12		20.0	16.66666666666667	40.0	23.33333333333333
13		26.66666666666667	40.0	10.0	23.33333333333333
14		16.66666666666667	43.33333333333333	20.0	20.0
15		30.0	33.33333333333333	13.33333333333333	23.33333333333333
16		16.66666666666667	40.0	13.33333333333333	30.0
17		26.66666666666667	33.33333333333333	10.0	30.0
18		13.33333333333333	36.66666666666667	16.66666666666667	33.33333333333333
19		26.66666666666667	10.0	23.33333333333333	40.0
20		13.33333333333333	23.33333333333333	36.66666666666667	26.66666666666667
21		26.66666666666667	46.66666666666667	10.0	16.66666666666667
22		26.66666666666667	33.33333333333333	16.66666666666667	23.33333333333333
23		6.66666666666667	40.0	13.33333333333333	40.0
24		23.33333333333333	40.0	6.66666666666667	30.0
25		23.33333333333333	30.0	20.0	26.66666666666667
26		20.0	40.0	16.66666666666667	23.33333333333333
27		30.0	20.0	23.33333333333333	26.66666666666667
28		10.0	33.33333333333333	33.33333333333333	23.33333333333333
29		20.0	36.66666666666667	10.0	33.33333333333333
30		23.33333333333333	30.0	23.33333333333333	23.33333333333333

1 - 30 de 95 Ir para: 1

Figura 3.1: Perfil dos jogadores.

- Estratégia
- RPG (Role-playing game)
- Simulação

- Outros

Cada um desses gêneros estão descritos abaixo.

O gênero de jogos eletrônicos **Ação** é caracterizado por uma jogabilidade intensa. Esses jogos colocam os jogadores em situações desafiadoras e empolgantes, onde reflexos rápidos, habilidades estratégicas e precisão são essenciais para o sucesso. Diferente dos jogos de ação/aventura, eles não possuem aprofundamento narrativo ou elementos de exploração extensiva. Os jogos de ação pura geralmente envolvem combate direto, tiros, lutas corpo a corpo e resolução de quebra-cabeças mais voltados para desafios de habilidade. Eles podem ser apresentados em diferentes perspectivas, como visão em primeira pessoa (FPS), terceira pessoa (TPS) ou até mesmo visão lateral. Como exemplo, “Doom”. Trata-se de uma franquia de jogos de tiro em primeira pessoa conhecida por sua ação rápida e frenética, onde os jogadores enfrentam hordas de demônios em ambientes infernais.

Jogos do gênero **Aventura** são conhecidos por oferecerem uma experiência imersiva, cuja ênfase é no enredo e não na ação. São caracterizados pela exploração dos cenários, pelos enigmas e quebra-cabeças, pela interação com outros personagens e pelo foco na narrativa. Um exemplo são jogos de aventura em texto, como “Ace Attorney: Phoenix Wright”, onde você assume o papel de um advogado de defesa.

Existe um gênero híbrido amplamente aceito pela comunidade de jogadores, conhecido como **Ação/Aventura**. Esses jogos combinam elementos de ação intensa com uma narrativa envolvente, resultando em uma experiência que cativa os jogadores tanto pela jogabilidade emocionante quanto pela imersão na história. Um exemplo marcante desse gênero é “The Last of Us”, os jogadores são levados a um mundo pós-apocalíptico infestado por seres hostis. A jogabilidade é repleta de ação, com combates desafiadores e momentos de tensão, enquanto os jogadores lutam pela sobrevivência. Ao mesmo tempo, a narrativa é profunda e comovente, com personagens bem desenvolvidos e uma história cativante que mantém os jogadores envolvidos emocionalmente.

Os “**SandBox**”, ou jogos de mundo aberto, tem como uma principal característica a ausência de uma linearidade estrita na jogabilidade. Os jogadores tem liberdade para escolher o que fazer, para onde ir e como abordar as missões ou objetivos do

jogo. Essa não linearidade dá aos jogadores um senso de controle e autodeterminação, permitindo-lhes criar sua própria experiência de jogo única. Além da liberdade de escolha, os jogos de mundo aberto geralmente apresentam vastos mapas ou ambientes virtuais, ricos em detalhes e repletos de atividades e possibilidades. Os jogadores podem explorar cidades, vilas, florestas, montanhas, desertos e outras paisagens, encontrando personagens não jogáveis, animais selvagens, tesouros escondidos, segredos e eventos dinâmicos. É o caso do Minecraft, onde o jogador deve sobreviver no seu mundo gerado aleatoriamente, minerando e explorando cavernas.

O gênero **Estratégia** é popular em jogos eletrônicos que envolvem a tomada de decisões para alcançar objetivos específicos, desafiando os jogadores a pensar de forma estratégica, planejar com antecedência e tomar decisões táticas para obter a vitória. Eles geralmente apresentam elementos como gerenciamento de recursos, construção de bases, diplomacia, combate estratégico e resolução de quebra-cabeças. Esses jogos são apreciados por jogadores que desejam exercitar sua capacidade de planejamento, tomar decisões e enfrentar desafios complexos. Eles oferecem uma experiência de jogo profunda e envolvente, onde o pensamento estratégico e a adaptabilidade são fundamentais para o sucesso. Além disso, muitos jogos de estratégia oferecem modos *single* e *multiplayer*, permitindo que os jogadores joguem contra a inteligência artificial ou desafiem outros jogadores de todo o mundo. “Pikmin” é um exemplo de jogo de estratégia onde o jogador controla ordas de seres alienígenas (pikmins) para realizar tarefas outrora impossíveis, como obter tesouros, congelar rios, e matar monstros dentro de um limite de tempo.

Os jogos eletrônicos do gênero **RPG** (Role-Playing Game), ou jogo de interpretação de papéis, são conhecidos por oferecerem aos jogadores a oportunidade para assumir o controle dos personagens em um mundo fictício, no qual eles podem embarcar em aventuras, explorar ambientes, interagir com personagens não jogáveis e enfrentar desafios diversos. No RPG eletrônico os jogadores geralmente têm a liberdade para personalizar e evoluir seus personagens, escolhendo habilidades, características e equipamentos. Eles também podem tomar decisões que afetam a história e o desenrolar do jogo, permitindo uma experiência mais imersiva e interativa. Esses jogos podem ser divididos em diferentes subgêneros, como os RPGs de ação que combinam elementos de jogabilidade de

ação em tempo real com mecânicas de RPG; e os RPGs táticos, que enfatizam estratégias e combates baseados em turnos. Ex: “Skyrim”

O gênero **Simulação** é conhecido por oferecer aos jogadores a oportunidade de experimentar e simular atividades do mundo real em um ambiente virtual. Esses jogos são projetados para replicar ou simular aspectos da vida real, permitindo que os jogadores assumam o controle de situações e desafios que normalmente encontrariam fora do mundo virtual. Os jogos de simulação abrangem uma ampla variedade de temas e atividades, desde simulações de voo, simulações de veículos e de gestão até as simulações de vidas e construção de cidades. Esses jogos são projetados para fornecer uma experiência autêntica e realista, muitas vezes incorporando elementos como física realista, gráficos detalhados e mecânicas de jogabilidade complexas. Um exemplo é o “The Sims”, famoso por ter sido o precursor de simulação da vida real.

Outros foi uma categoria criada para conter jogos que não pertencem a nenhuma das previamente citadas, considerados jogos que saem da curva dos gêneros mais conhecidos, como jogos rítmicos (Guitar Hero, Just Dance), quebra-cabeças (Tetris) e *Party Games* (jogos para se jogar em grupo)

A Figura 3.2 ilustra o conjunto de gêneros. No interior do conjunto, encontram-se apresentados os subgêneros, os quais detalham os diferentes tipos de jogos pertencentes a cada gênero.

Levando em consideração os gêneros é possível classificar jogos e estabelecer uma comparação, por similaridade entre os arquétipos de Bartle e os gêneros. Jogos onde há muita exploração e segredos - como os ação/aventura - vão ter um peso maior no perfil de *Explorer* e *Achiever*; jogos em que há muita interação entre jogadores e entre NPCs são identificados com perfil *Socializer* e assim por diante.

Uma vez que a autora tem uma lista de jogos cadastrada, que é a presente em (CONDE, 2021) para controle de sua coleção pessoal, foi criado um dataset a partir desse, contendo as informações relevantes para o estudo: Título, Gênero e as quatro colunas com os arquétipos (*Achiever*, *Explorer*, *Killer* e *Socializer*). Foram adicionados cento e quarenta e quatro jogos distintos. Baseando-se inteiramente nos gêneros, e na experiência da autora, foram quantificados os pesos de cada jogo. Uma vez que definir



Figura 3.2: Conjuntos de gêneros

perfil de jogos pode ser complexo e subjetivo, foram utilizados dois arquétipos principais por jogo, diferente do que foi feito para jogadores. Isso foi determinado pois, por mais que jogos tenham características pertencentes aos quatro arquétipos, ter dois principais resulta em uma melhor classificação e separação. A Figura 3.3 ilustra a tabela SQLite utilizada:

The screenshot shows a database browser window with the following table data:

ID	Título	Genero	Achiever	Explorer	Killer	Socializer
1	118 Age of Mythology	Estratégia	75.0	0.0	0.0	25.0
2	54 Animal Crossing New Horizons	Simulação	20.0	0.0	0.0	80.0
3	82 Apollo Justice: Ace Attorney	Simulação	0.0	70.0	0.0	30.0
4	50 Assassins Creed II	Ação / Aventura	60.0	40.0	0.0	0.0
5	51 Assassins Creed Origins	Ação / Aventura	60.0	40.0	0.0	0.0
6	43 Battlefield 1	Tiro em Primeira Pessoa	0.0	0.0	70.0	30.0
7	7 Bioshock	Tiro em Primeira Pessoa	70.0	0.0	0.0	30.0
8	8 Bioshock Infinite	Tiro em Primeira Pessoa	70.0	0.0	0.0	30.0
9	22 Bully	Ação / Aventura	40.0	0.0	60.0	0.0
10	123 Cadence of Hyrule	Música	50.0	50.0	0.0	0.0
11	34 Call of Duty Black Ops	Tiro em Primeira Pessoa	0.0	0.0	65.0	35.0
12	136 Call of Duty Ghosts	Tiro em Primeira Pessoa	0.0	0.0	65.0	35.0
13	42 Counter Strike	Tiro em Primeira Pessoa	0.0	0.0	70.0	30.0
14	75 Crash Bandicoot	Plataforma	60.0	40.0	0.0	0.0
15	88 Crash Bandicoot 4 It's About Time	Plataforma 3D	60.0	40.0	0.0	0.0
16	68 Crash Team Racing	Corrida	50.0	0.0	0.0	50.0
17	11 Dark Souls	RPG	50.0	50.0	0.0	0.0
18	95 Dead Space	Tiro em Primeira Pessoa	50.0	50.0	0.0	0.0
19	17 Demon Souls	RPG	50.0	50.0	0.0	0.0
20	16 Destiny	Tiro em Primeira Pessoa	0.0	0.0	50.0	50.0
21	131 Diablo	Estratégia	0.0	0.0	0.0	0.0
22	65 Disney's Lion King	Plataforma	50.0	50.0	0.0	0.0
23	133 Donkey Kong	Arcade	80.0	0.0	20.0	0.0
24	3 Donkey Kong Country	Plataforma	50.0	50.0	0.0	0.0

Figura 3.3: Perfil dos jogos

3.2 Agrupamento

Como os dados de jogos e jogadores não possuem um rótulo conhecido que auxiliasse na determinação de suas classes, foi inicialmente feita uma clusterização de ambos como forma de identificar jogos e jogadores semelhantes utilizando o algoritmo *K-Means*.

Como um pré-processamento dos dados, antes de inicializar o algoritmo para gerar o agrupamento, foi feita a padronização dos *datasets* para tratar problemas com a diferente escala das *features*. A padronização z-score, também conhecida como padronização de média zero e desvio padrão unitário, tem como funcionamento subtrair a média da coluna

de cada valor e depois dividir pelo desvio padrão da coluna. Isso coloca os dados em uma escala onde a média é 0 e o desvio padrão é 1:

$$Z = \frac{X - \mu}{\sigma}$$

onde:

- Z é o valor padronizado (z -score) do valor X .
- X é o valor original.
- μ é a média dos valores na população.
- σ é o desvio padrão dos valores na população.

Como *K-Means* necessita de um número K para fazer o agrupamento, foi mensurado o coeficiente de silhueta que é uma métrica útil para avaliar a qualidade da separação entre *clusters*, com valores mais próximos de 1 indicando uma separação clara dos dados:

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

onde:

- $S(i)$: Coeficiente de silhueta para o objeto i .
- $a(i)$: Média da distância entre o objeto i e os outros objetos no mesmo *cluster* (a medida de coesão).
- $b(i)$: Menor média da distância entre o objeto i e os objetos em outros *clusters* diferentes do seu próprio (a medida de separação).

Além do coeficiente de silhueta foi utilizada uma segunda métrica, conhecida como o “método do cotovelo”, que tem esse nome por causa do gráfico resultante. O ponto em que a curva começa a nivelar, lembrando um “cotovelo”, é frequentemente considerado um bom valor para K . Esses dois métodos se complementam e foram aplicados nos dois *datasets* como será mostrado no Capítulo 4.

Features	Valor
Nome	ex: João
Achiever	[0, 50]
Explorer	[0, 50]
Killer	[0, 50]
Socializer	[0, 50]

Tabela 3.2: Features de Jogadores

Features	Valor
Título	ex: Minecraft
Gênero	ex: Sandbox
Achiever	[0, 100]
Explorer	[0, 100]
Killer	[0, 100]
Socializer	[0, 100]

Tabela 3.3: Features de Jogos

Os campos pertencentes aos quais foram aplicada a padronização estão presentes nas Tabelas 3.2 e 3.3.

Para os jogos, antes de aplicar os métodos silhueta e cotovelo, foi necessário realizar um tratamento com as variáveis categóricas presentes na coluna “Gênero”. Elas foram convertidas em variáveis binárias usando o método “*one-hot encoding*” da função *get_dummies()* do Pandas. Isso permite que as informações sobre gêneros de jogos sejam representadas de forma numérica, o que é obrigatório para o algoritmo *K-Means*.

3.3 Classificação e Avaliação

Com as duas bases de dados agrupadas pela aplicação do *K-Means*, será possível utilizar outros algoritmos de aprendizado de máquina a fim de treiná-los para classificar novos jogos e jogadores. Foram avaliados seis algoritmos: KNN, SVM Linear, SVM Gaussiano, *Nearest Centroid*, Árvore de Decisão e Floresta Aleatória.

Os modelos foram treinados utilizando a técnica *Stratified K-Fold* com $k = 10$. Isso significa que o conjunto de dados foi dividido em 10 partições estratificadas, garantindo que cada subconjunto mantivesse a mesma proporção de classes que o conjunto de dados original.

3.4 Relação Jogo-Jogador

Para realizar recomendações de jogos é preciso primeiro estabelecer uma relação entre os dados coletados. Uma vez que as características (*features*) para jogos e jogadores serem iguais, será possível criar uma nova base de dados combinando esses conjuntos de dados. Nessa nova base, a relação entre todos os jogadores e todos os jogos é estabelecida pela multiplicação das colunas correspondentes. Ou seja, a relação se dá por um vetor de quatro posições, onde cada posição representa o peso de cada coluna. Por exemplo, se um jogador tem pontuações de 40, 30, 20 e 10 nas características (*Achiever*, *Explorer*, *Killer*, *Socializer*), e um jogo tem pontuações de 50, 50, 0 e 0 nas mesmas características, a relação jogo-jogador seria representada pelos valores resultantes da multiplicação, ou seja, um vetor $v = [2000, 1500, 0, 0]$, respectivamente.

Essa abordagem resulta em uma tabela que destaca a afinidade potencial entre cada jogador e cada jogo com base em suas características. Com isso, será possível utilizar uma simplificação do método definido em (SAWANT, 2013), construindo um grafo bipartido completo, onde os vértices são os *clusters* encontrados e as arestas são as relações entre os *clusters*. O peso das arestas é representado por vetores de quatro posições, onde cada posição representa o valor somado das relações jogo/jogador para cada uma das quatro características, como demonstrado no exemplo anterior. Estes valores foram posteriormente normalizados para garantir o padrão das relações.

Pode-se identificar os dois conjuntos de vértices do grafo como o conjunto E de *clusters* de jogadores, o conjunto P de *clusters* de jogos e A representando as arestas do grafo que conectam vértices de um conjunto no outro. Assim, tem-se o grafo $G = (E, P, A)$, um grafo bipartido completo onde existe relação de todos os *clusters* de jogador com todos os *clusters* de jogo.

3.5 Geração de Novos Dados e Recomendação

Com a relação entre *clusters* de jogo e jogador estabelecida pelo grafo resultante, bastará gerar novos jogos e jogadores, aplicar os modelos treinados para que encontrem seus respectivos *clusters* e, finalmente, fazer o cálculo da distância euclidiana entre o(s) joga-

dor(es) e os *clusters* de jogos:

$$D = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}. \quad (3.1)$$

A menor distância encontrada significa maior proximidade entre jogador e os jogos pertencentes ao *cluster*, o que indica uma recomendação.

4 Experimentos e Resultados

Dada a abordagem descrita no Capítulo 3, e considerando que os dados foram previamente processados e estão aptos para testes, é iniciado o experimento. O primeiro passo é encontrar o melhor K para aplicação do algoritmo K -Means. Para isso, calcula-se o coeficiente de silhueta entre $K = 2$ e $K = 10$. Aplica-se também a análise do gráfico do "método do cotovelo". Esse processo é realizado para ambos datasets, como visto a seguir.

4.1 Jogadores

Ao calcular o coeficiente de silhueta, o melhor valor de K encontrado foi $K = 2$, com coeficiente no valor **0.32**, Figura 4.1. Do "método do cotovelo" foi gerado o gráfico da Figura 4.2, e após análise, fica evidente que nos pontos 2, 4 e 5 há uma alteração na curva, resultando em $K = 2$, $K = 4$ e $K = 5$. Esses podem ser considerados bons valores para o agrupamento. Em um primeiro momento, o agrupamento entre 2 *clusters* é o mais indicado, mas esse gera um número muito baixo para fins de recomendação. Para evitar esse enviesamento do modelo, o número escolhido foi 5.

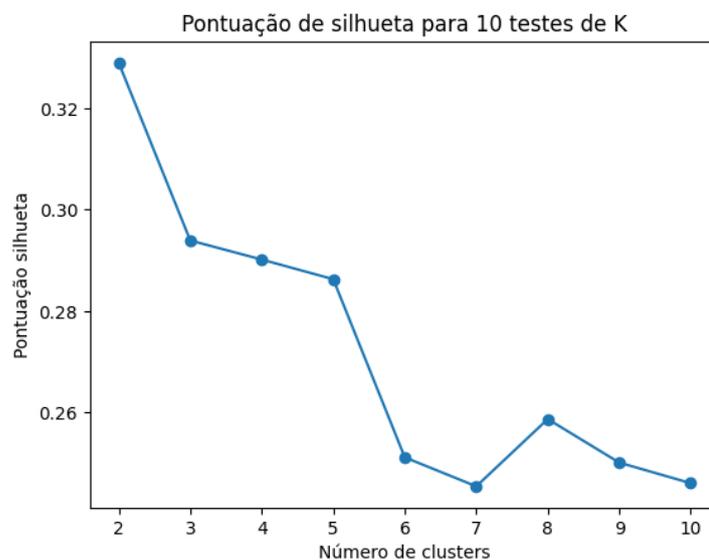


Figura 4.1: Coeficiente da Silhueta para Jogadores

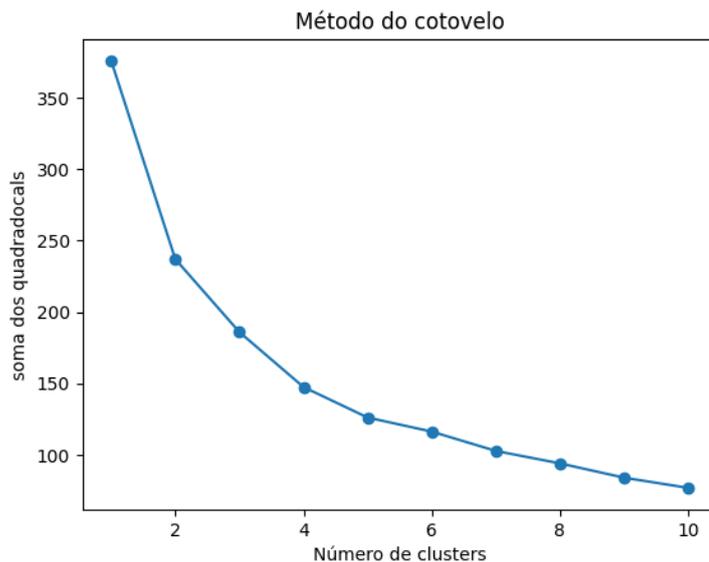


Figura 4.2: Método do cotovelo para jogadores

Com a quantidade de *clusters* definida, é aplicado o algoritmo *K-Means* com uma configuração de 5 *clusters*, 10 execuções com diferentes sementes e no máximo 300 iterações a cada execução, O resultado gerado é adicionado ao *dataset* de jogadores em uma nova coluna chamada “cluster” e pode ser visto no Apêndice.

4.2 Jogos

Para jogos, foram aplicados os mesmos dois métodos - “silhueta” e “cotovelo”. Os resultados estão especificados nas Figuras 4.3 e 4.4

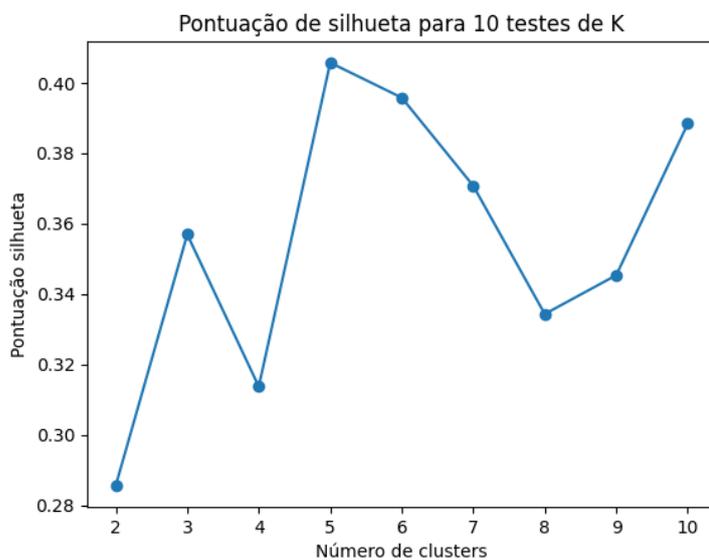


Figura 4.3: Coeficiente de Silhueta para Jogos

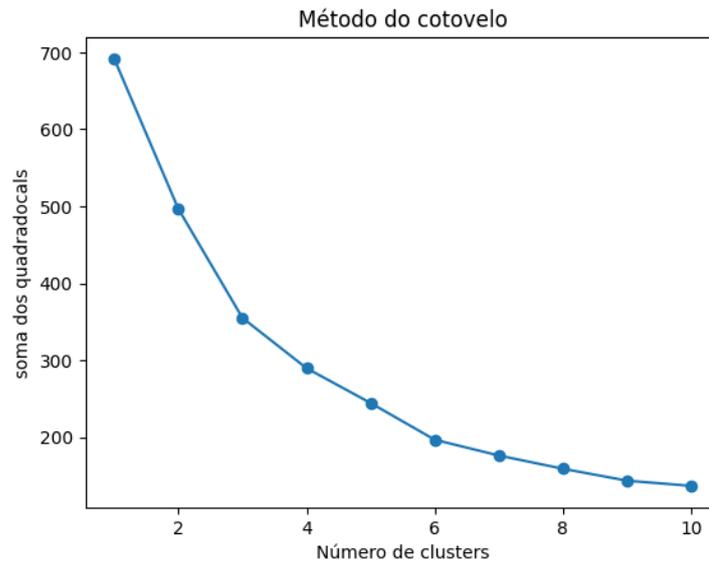


Figura 4.4: Método do cotovelo para Jogos

O “coeficiente de silhueta” indicou 5 como o melhor K , com valor médio **0.405**; o “método do cotovelo” indica um nivelamento entre 4 e 6. Com isso foi aplicado o *K-Means* com o número de *clusters* = 5 e a mesma configuração usada para jogadores. O resultado pode ser visto no Apêndice.

4.3 Avaliação

Com os dados devidamente agrupados, o próximo passo consiste em treinar os modelos. O treinamento é feito utilizando a técnica K-Fold com $K = 10$, onde 10 é o número de dobras (partições) dos dados. Os parâmetros necessários para o treino variam para cada algoritmo: KNN recebe $k = 3$ para o “número de vizinhos”. SVM foi treinado com ambos *kernels*, linear e gaussiano, e coeficiente $c = 1$. Para mensurar a acurácia dos modelos, é feito um comparativo entre as classes encontradas pelo teste, e a classe associada pelo *K-Means*.

Os valores de acurácia e desvio padrão, encontrados nas Tabelas 4.3 e 4.3 foram obtidos como resultado da validação cruzada (*cross-val-score*) entre os algoritmos.

JOGADORES	KNN	SVM Linear	SVM Gaussiano	NC	Árvore Decisão	Floresta
Acurácia média	0.946	0.925	0.936	0.989	0.893	0.946
Desvio Padrão	0.053	0.041	0.061	0.021	0.033	0.047

Tabela 4.1: Acurácia e desvio padrão dos algoritmos para jogadores

JOGOS	KNN	SVM Linear	SVM Gaussiano	NC	Árvore Decisão	Floresta
Acurácia média	0.964	1.0	0.964	0.971	0.978	0.976
Desvio Padrão	0.057	0.0	0.031	0.026	0.0175	0.0173

Tabela 4.2: Acurácia e desvio padrão dos algoritmos para jogos

Nota-se que, para jogos, o SVM linear obteve 100% de acerto, Isso pode ocorrer por alguns motivos, sendo um deles a estratégia usada pelo algoritmo como a busca por um “hiperplano” (uma espécie de linha ou plano) que separa de forma mais clara os diferentes tipos de jogos tendo por base suas *features*. O SVM linear é eficaz quando as características dos *jogos* podem ser separadas de maneira nítida por uma linha. Nesse caso, o modelo encontrou uma relação linear clara entre os tipos de jogos, o que justifica a acurácia encontrada.

Também é interessante notar que todos os modelos testados alcançaram uma acurácia superior a 96% na clusterização, o que significa uma alta confiabilidade nas previsões futuras.

Já para jogadores, comparando-os aos jogos, há uma acurácia menor entre os algoritmos, porém ainda alta e próxima dos 90%. Nesse caso, o mais preciso foi o *Nearest Centroid*. Novamente, isso pode ocorrer pela estratégia utilizada pelo NC, uma vez que o algoritmo busca o centroide de cada classe e o *K-Means* usa centróides como representantes de *clusters*. Dessa forma, as classes existentes no conjunto de treinamento servem como referências para o NC determinar a classe mais próxima.

O desempenho mais baixo nos outros algoritmos, principalmente SVM Linear, Gaussiano e Árvore de Decisão, pode ser explicada pelas medições obtidas do “coeficiente de silhueta”, uma vez que os valores foram baixos (entre 0.24 e 0.32 como vista na Figura 4.1) e foi feita a escolha de usar um K maior do que o sugerido. Nesse caso, as classes encontradas pelo K-Means estão próximas umas das outras e pode ser prejudicial para o aprendizado dos modelos gerando inconsistências entre a classe dada e a prevista.

4.4 Relação Jogo-Jogador e Grafo Resultante

No capítulo 3 é demonstrado como estabelecer relações entre jogos e jogadores, gerando o dataset resultante das interações. Dado estas relações, é possível criar o grafo bipartido

que será a base para as recomendações.

Nessa etapa é essencial dispor das relações entre jogos e jogadores para gerar o grafo corretamente. Como a classe de ambos já é conhecida, o processo envolve inicializar o grafo como uma estrutura tridimensional dimensionada com base no número de *clusters* de jogadores e *clusters* de jogos. A terceira dimensão é formada por um vetor de quatro posições, onde cada uma representa o valor de cada um dos atributos (Achiever, Explorer, Killer, Socializer), da relação. Esta representa as arestas do grafo.

Para preencher o grafo é realizada a iteração sobre as relações registradas previamente e cada jogo e jogador são posicionados nos seus respectivos vértices (*clusters*). Como o peso das arestas se dá pela relação entre jogo e jogador, a cada iteração as arestas são atualizadas. Por exemplo, a relação do jogador 1 com o jogo 1 é dada por $v = [2000, 1500, 0, 0]$, este vetor v encontrado previamente como resultante do dataset gerado pelas relações. O jogador 1 está classificado no *cluster* 0 de jogadores; o jogo 1 está classificado no *cluster* 0 de jogos; a aresta resultante da ligação entre *cluster* 0 de jogador e *cluster* 0 de jogo é $a = [2000, 1500, 0, 0]$. Se na próxima iteração o jogador 2, que está classificado também no *cluster* 0, tem relação com o jogo 1, então $v = [1500, 2000, 0, 0]$ e a aresta resultante agora é $[3500, 3500, 0, 0]$.

Isso se repete para todas as interações, posteriormente os valores das arestas são normalizadas. Do grafo bipartido resultante foi criada a Tabela 4.3:

	0	1	2	3	4
0	0.00 / 0.10 / 0.52 / 0.38	0.55 / 0.00 / 0.01 / 0.44	0.53 / 0.00 / 0.45 / 0.01	0.02 / 0.53 / 0.00 / 0.45	0.40 / 0.58 / 0.02 / 0.00
1	0.00 / 0.08 / 0.50 / 0.41	0.58 / 0.00 / 0.01 / 0.41	0.59 / 0.00 / 0.40 / 0.01	0.03 / 0.46 / 0.00 / 0.51	0.49 / 0.48 / 0.03 / 0.00
2	0.00 / 0.01 / 0.50 / 0.40	0.55 / 0.00 / 0.01 / 0.44	0.55 / 0.00 / 0.44 / 0.01	0.02 / 0.51 / 0.00 / 0.47	0.00 / 0.10 / 0.50 / 0.40
3	0.00 / 0.10 / 0.47 / 0.42	0.54 / 0.00 / 0.01 / 0.45	0.58 / 0.00 / 0.41 / 0.01	0.03 / 0.51 / 0.00 / 0.46	0.41 / 0.57 / 0.02 / 0.00
4	0.00 / 0.08 / 0.57 / 0.35	0.58 / 0.00 / 0.01 / 0.41	0.52 / 0.00 / 0.47 / 0.01	0.03 / 0.49 / 0.00 / 0.48	0.46 / 0.51 / 0.02 / 0.00

Tabela 4.3: Exemplo de grafo gerado. *Clusters* de jogadores como linhas e de jogos como colunas.

Cada *cluster* de jogador está associado com cada *cluster* de jogos por meio dos vetores apresentados na Tabela 4.3. Essa relação estabelece uma conexão direta entre os dois conjuntos, possibilitando a recomendação de jogos ao calcular a menor distância euclidiana entre o perfil de um jogador e todos os *clusters* de jogos. A menor distância reflete a proximidade entre os perfis de jogador e jogo, indicando que, para o jogador em questão, os jogos do *cluster* mais próximo são os mais indicados. Essa abordagem oferece uma maneira de personalizar recomendações com base nas preferências individuais dos

jogadores.

Dessa forma, no entanto, estamos relacionando um jogador a um único agrupamento de jogos. Uma vez serem vários os jogos possíveis, isto poderia enfraquecer a indicação. Uma solução plausível seria, após calcular a distância do perfil do jogador a todos os *clusters* de jogos, considerar o *cluster* de jogo encontrado, e calcular a distância do jogador para todos os jogos dentro desse *cluster*, o que possibilita atingir uma distância ainda menor intra-*cluster* e uma recomendação mais precisa.

Para analisar o experimento, uma vez obtido o grafo relacionado, foram criados 10 novos jogadores e 15 novos jogos, 10% da quantidade total de amostras treinadas. Esses novos dados são resultantes de um script criado que respeita os mesmos limites setados anteriormente para classificação de jogos e jogadores. Isso é importante para que não sejam criados dados que não sejam possíveis nos *datasets* originais, como valores negativos ou maiores que 100.

Uma vez que os modelos KNN, SVM linear, SVM Gaussiano, Árvore de Decisão e Floresta Aleatória foram previamente treinados, esses são aplicados para prever a classe desses novos dados. Os resultados das previsões estão apresentados nas Tabelas 4.4 e 4.7

Nota-se, na tabela 4.4, uma divergência significativa entre as classes previstas pelos algoritmos. Nesse caso, para definir à qual *cluster* cada jogador pertence, é realizada uma votação entre os algoritmos. Dentre os previstos, é selecionado o *cluster* que foi escolhido três ou mais vezes. Em caso de empate, é analisada a acurácia dos algoritmos para seleção definitiva.

Tomando o jogador de id 3 como exemplo, o *cluster* 0 é previsto por 4 dos 6 algoritmos, logo, é definido que este jogador pertence a este *cluster*. Esse processo se repete para todos os novos jogadores do dataset.

Uma vez definida a classe dos novos jogadores, é possível recomendar jogos a eles calculando a distância euclidiana do seu perfil aos *clusters* de jogos, utilizando o grafo previamente criado. Para fins de experimento foi escolhido aleatoriamente um jogador. O resultado segue:

No caso apresentado, o jogador aleatório foi classificado no *cluster* 3 pelo processo de votação e a menor distância euclidiana entre seu perfil e os *clusters* de jogos é

Jogador_Novo	Achiever	Explorer	Killer	Socializer	knn	svm_lin	svm_gauss	NC	Arvore	Floresta
0	36	44	14	6	3	3	3	3	3	3
1	6	7	42	45	0	0	0	0	0	0
2	41	23	17	19	4	4	3	4	4	4
3	21	12	35	32	0	1	0	4	0	0
4	47	24	4	25	1	4	3	4	4	4
5	29	3	29	39	4	1	0	4	0	0
6	19	18	23	40	1	1	1	1	1	1
7	6	43	34	17	3	3	3	3	3	3
8	16	23	31	30	1	1	0	1	0	0
9	32	32	12	24	3	3	3	3	3	3

Tabela 4.4: Previsão de novos jogadores.

Jogador Cluster 3

Cluster 0 Jogo	0.696
Cluster 1 Jogo	0.630
Cluster 2 Jogo	0.562
Cluster 3 Jogo	0.548
Cluster 4 Jogo	0.194

Tabela 4.5: Distância entre jogador e clusters de jogo

aproximadamente 0.19. Nesse caso é recomendado a ele jogos do *cluster 4*. Para uma recomendação mais precisa é possível calcular a distância dele com todos os jogos presentes no *cluster 4* como visto na tabela a seguir

	Jogador Cluster 3
Jogo 0	0.523
Jogo 1	0.538
Jogo 2	0.546
Jogo 3	0.760
Jogo 4	0.624
Jogo 5	0.886
Jogo 6	0.938
Jogo 7	0.993
Jogo 8	0.530
Jogo 9	0.523
Jogo 10	0.523
Jogo 11	0.530
Jogo 12	0.760
Jogo 13	0.760

Tabela 4.6: Distância do jogador para todos os jogos do cluster 3

Ao analisar as distâncias presentes na tabela 4.6, o jogador em questão mostra mais proximidade com os jogos de índices 1, 10, 11 (distância = 0.523). Apesar de os jogos do *cluster 4* serem próximos entre si conforme o agrupamento inicial, nem todos os jogos desse grupo estão próximos do jogador. Isso corrobora a necessidade de calcular

as distâncias euclidianas duas vezes: a primeira, para determinar o *cluster* de jogo mais próximo entre os possíveis; a segunda para identificar, dentro desse *cluster*, os jogos mais próximos ao jogador.

Para os novos jogos, o processo de votação se repete. Nesse caso, existem somente 3 jogos em que a previsão dada pelos modelos é divergente, são eles os ids : 6, 7 e 13. Por votação, as classes definitivas são dadas pelo *cluster* que foi escolhido três ou mais vezes.

Jogos_novos	Achiever	Explorer	Killer	Socializer	knn	svm_lin	svm_gauss	NC	Árvore	Floresta
0	0	58	0	42	4	4	4	4	4	4
1	34	0	0	66	2	2	2	2	2	2
2	0	74	0	26	4	4	4	4	4	4
3	74	26	0	0	0	0	0	0	0	0
4	13	0	0	87	4	4	4	4	4	4
5	34	0	66	0	3	3	3	3	3	3
6	0	0	12	88	4	4	4	4	1	4
7	0	89	11	0	4	2	4	4	0	0
8	27	73	0	0	0	0	0	0	0	0
9	81	0	0	19	2	2	2	2	2	2
10	68	0	0	32	2	2	2	2	2	2
11	0	38	62	0	1	1	1	1	1	1
12	0	86	0	14	4	4	4	4	4	4
13	0	88	12	0	4	2	4	4	0	0
14	0	0	31	69	1	1	1	1	1	1

Tabela 4.7: Previsão de novos jogos

Novamente, após definição das classes, realizou-se o cálculo da distância euclidiana, desta vez entre um novo jogo aleatório e os *clusters* de jogadores. No resultado apresentado na Tabela 4.8, observa-se que o *cluster* de jogador mais próximo desse jogo é o *cluster* 1. Portanto, torna-se possível recomendar este jogo a todos os jogadores pertencentes a esse *cluster*.

	Jogo Cluster 2
Cluster 0 Jogador	0.559
Cluster 1 Jogador	0.486
Cluster 2 Jogador	0.540
Cluster 3 Jogador	0.5043
Cluster 4 Jogador	0.575

Tabela 4.8: Distancia entre jogo e *clusters* de jogadores

Para validar o experimento, a autora escolheu uma pessoa conhecida, a quem foram apresentados três dos jogos mais próximos recomendados pelo sistema. Segundo ela, que já havia jogado todos, as recomendações foram válidas e positivas.

5 Considerações Finais

Observou-se que, treinando diferentes algoritmos de *machine learning* em ambas bases - jogos e jogador, obtém-se uma possível recomendação por similaridade, uma vez que foram usadas as mesmas características para classificá-los e existe uma relação direta entre jogos e jogadores.

Jogadores com um perfil mais conquistador e explorador, em comparação com os outros dois, foram recomendados jogos que tinham perfil também conquistador e explorador. Jogos esses presentes no *cluster* 0. Isso pode ser visto na Tabela 4.7, onde o jogo de id 8 que tem perfil 27% Achiever (conquistador) e 73% Explorer (explorador) foi previsto neste *cluster* por todos os modelos.

Do modo que o questionário de Bartle foi feito, com as trinta perguntas obrigatórias e somente 2 alternativas, os jogadores acabam tendo perfis próximos (o perfil dos jogos e jogadores pode ser visto no Apêndice), o que influencia o experimento. Se os jogadores estão próximos, a distância entre seus *clusters* é pequena, logo, a recomendação pode não ser tão assertiva.

Como um dos princípios da recomendação é sugerir algo que o usuário de fato goste, e como não há uma confirmação dos jogadores se a recomendação foi válida, não é possível medir quão efetiva foi a classificação ou se os jogos são relevantes para aqueles usuários.

Uma possível solução para isso seria receber *feedbacks* dos jogadores, similar ao que acontece em sistemas colaborativos, e adicionar aos datasets essa nova *feature* de “interesse” como um valor booleano, onde há indicação se o jogador gostou ou não daquela respectiva sugestão. Uma outra possibilidade seria o jogador dar uma nota entre 0 e 5, ao jogo sugerido. Isso daria aos modelos ainda mais informações sobre quais jogos interessam a quais jogadores, sendo possível fazer recomendações ainda mais assertivas.

Para futuras pesquisas relacionadas a este trabalho, seria interessante desenvolver um sistema web semelhante a uma rede social, permitindo que usuários compartilhem seus gostos e experiências. Durante o cadastro, os usuários responderiam ao questionário de

Bartle para determinar seu perfil de jogador. Dessa forma, recomendações personalizadas já poderiam ser fornecidas ao usuário ao ingressar na plataforma, uma vez que jogos e jogadores já foram classificados e inseridos no sistema como referência.

Após receber uma recomendação, o jogador teria a opção de aprovar ou desaprovar a indicação, semelhante a um botão “gostei”. Essa interação seria valiosa para aprimorar os modelos de inteligência artificial, permitindo sugestões mais precisas com base no *feedback* do usuário. À medida que os modelos aprendem com cada interação, jogadores com perfis semelhantes ao usuário que forneceu esse *feedback* também poderiam receber novas recomendações.

A Representação dos Clusters

Cluster	0	1	2	3	4
Número de Jogadores	26	25	10	21	12

Tabela A.1: Número de jogadores agrupados por cluster

Cluster	0	1	2	3	4
Número de Jogos	52	17	36	22	14

Tabela A.2: Número de jogos agrupados por cluster

ID	Achiever	Explorer	Killer	Socializer	clusters
1	20.00	33.33	13.33	33.33	1
2	10.00	33.33	16.67	40.00	1
3	20.00	36.67	20.00	23.33	3
4	13.33	26.67	16.67	43.33	1
5	26.67	23.33	26.67	23.33	4
6	20.00	40.00	13.33	26.67	3
7	20.00	40.00	16.67	23.33	3
8	26.67	43.33	3.33	26.67	3
9	26.67	40.00	10.00	23.33	3
10	20.00	23.33	26.67	30.00	0
11	36.67	16.67	40.00	6.67	2
12	23.33	26.67	26.67	23.33	0
13	20.00	16.67	40.00	23.33	0
14	26.67	40.00	10.00	23.33	3
15	16.67	43.33	20.00	20.00	3
16	30.00	33.33	13.33	23.33	3
17	16.67	40.00	13.33	30.00	1
18	26.67	33.33	10.00	30.00	1
19	13.33	36.67	16.67	33.33	1

ID	Achiever	Explorer	Killer	Socializer	clusters
20	26.67	10.00	23.33	40.00	4
21	13.33	23.33	36.67	26.67	0
22	26.67	46.67	10.00	16.67	3
23	26.67	33.33	16.67	23.33	3
24	6.67	40.00	13.33	40.00	1
25	23.33	40.00	6.67	30.00	3
26	23.33	30.00	20.00	26.67	0
27	20.00	40.00	16.67	23.33	3
28	30.00	20.00	23.33	26.67	4
29	10.00	33.33	33.33	23.33	0
30	20.00	36.67	10.00	33.33	1
31	23.33	30.00	23.33	23.33	0
32	23.33	43.33	10.00	23.33	3
33	20.00	30.00	20.00	30.00	1
34	20.00	26.67	16.67	36.67	1
35	20.00	33.33	26.67	20.00	0
36	13.33	33.33	33.33	20.00	0
37	13.33	36.67	26.67	23.33	0
38	23.33	26.67	26.67	23.33	0
39	23.33	26.67	13.33	36.67	1
40	23.33	40.00	13.33	23.33	3
41	16.67	30.00	13.33	40.00	1
42	13.33	23.33	26.67	36.67	0
43	23.33	43.33	13.33	20.00	3
44	33.33	43.33	13.33	10.00	3
45	10.00	30.00	26.67	33.33	0
46	20.00	26.67	33.33	20.00	0
47	6.67	30.00	33.33	30.00	0

ID	Achiever	Explorer	Killer	Socializer	clusters
48	26.67	26.67	13.33	33.33	1
49	23.33	20.00	43.33	13.33	2
50	30.00	23.33	36.67	10.00	2
51	20.00	23.33	30.00	26.67	0
52	20.00	30.00	26.67	23.33	0
53	16.67	26.67	36.67	20.00	0
54	16.67	40.00	23.33	20.00	3
55	16.67	36.67	20.00	26.67	1
56	20.00	43.33	0.00	36.67	1
57	16.67	33.33	10.00	40.00	1
58	30.00	30.00	33.33	6.67	2
59	13.33	26.67	40.00	20.00	0
60	23.33	36.67	10.00	30.00	1
61	23.33	20.00	43.33	13.33	2
62	20.00	33.33	13.33	33.33	1
63	3.33	26.67	16.67	26.67	0
64	30.00	26.67	16.67	26.67	4
65	26.67	46.67	3.33	23.33	3
66	26.67	30.00	13.33	30.00	1
67	36.67	26.67	26.67	10.00	2
68	30.00	13.33	43.33	13.33	2
69	16.67	36.67	13.33	33.33	1
70	23.33	36.67	10.00	30.00	1
71	40.00	16.67	36.67	6.67	2
72	26.67	36.67	10.00	26.67	3
73	26.67	46.67	13.33	13.33	3
74	26.67	23.33	26.67	23.33	4
75	23.33	26.67	26.67	23.33	0

ID	Achiever	Explorer	Killer	Socializer	clusters
76	23.33	26.67	26.67	23.33	0
77	26.67	23.33	13.33	36.67	1
78	16.67	30.00	30.00	23.33	0
79	20.00	33.33	30.00	16.67	0
80	26.67	36.67	3.33	33.33	1
81	16.67	30.00	20.00	33.33	1
82	33.33	30.00	20.00	16.67	4
83	46.67	16.67	16.67	20.00	4
84	26.67	23.33	33.33	16.67	2
85	13.33	40.00	13.33	33.33	1
86	26.67	23.33	26.67	23.33	4
87	36.67	23.33	23.33	16.67	4
88	36.67	26.67	16.67	20.00	4
89	30.00	13.33	30.00	26.67	4
90	20.00	36.67	16.67	26.67	3
91	33.33	16.67	30.00	20.00	4
92	16.67	23.33	33.33	26.67	0
93	26.67	23.33	36.67	13.33	2
94	16.67	33.33	23.33	26.67	0

Tabela A.3: Dados de players já classificados

ID	Genero	Achiever	Explorer	Killer	Socializer	cluster
1	Plataforma	40.0	60.0	0.0	0.0	0
2	Plataforma	40.0	60.0	0.0	0.0	0
3	Plataforma	50.0	50.0	0.0	0.0	0
4	Plataforma	50.0	50.0	0.0	0.0	0
5	Plataforma	50.0	50.0	0.0	0.0	0
6	Tiro em Primeira Pessoa	30.0	0.0	70.0	0.0	3

ID	Genero	Achiever	Explorer	Killer	Socializer	cluster
7	Tiro em Primeira Pessoa	70.0	0.0	0.0	30.0	2
8	Tiro em Primeira Pessoa	70.0	0.0	0.0	30.0	2
9	Aventura	0.0	70.0	0.0	30.0	4
10	Aventura	0.0	60.0	0.0	40.0	4
11	RPG	50.0	50.0	0.0	0.0	0
12	Ação / Aventura	0.0	60.0	40.0	0.0	0
13	Plataforma	0.0	80.0	0.0	20.0	4
14	RPG	50.0	50.0	0.0	0.0	0
15	Survival Horror	0.0	50.0	50.0	0.0	1
16	Tiro em Primeira Pessoa	0.0	0.0	50.0	50.0	1
17	RPG	50.0	50.0	0.0	0.0	0
18	Plataforma	35.0	65.0	0.0	0.0	0
19	Ação / Aventura	50.0	50.0	0.0	0.0	0
20	Ação / Aventura	60.0	40.0	0.0	0.0	0
21	Ação / Aventura	40.0	60.0	0.0	0.0	0
22	Ação / Aventura	40.0	0.0	60.0	0.0	3
23	Ação / Aventura	50.0	0.0	50.0	0.0	3
24	Ação / Aventura	50.0	0.0	50.0	0.0	3
25	Luta	0.0	0.0	50.0	50.0	1
26	Luta	0.0	0.0	50.0	50.0	1
27	Luta	0.0	0.0	50.0	50.0	1
28	RPG	50.0	0.0	0.0	50.0	2
29	RPG	50.0	0.0	0.0	50.0	2
30	RPG	60.0	40.0	0.0	0.0	0
31	Aventura Gráfica	0.0	30.0	0.0	70.0	4
32	Tiro em Primeira Pessoa	50.0	0.0	50.0	0.0	3
33	Tiro em Primeira Pessoa	40.0	0.0	60.0	0.0	3
34	Tiro em Primeira Pessoa	0.0	0.0	65.0	35.0	1

ID	Genero	Achiever	Explorer	Killer	Socializer	cluster
35	Tiro em Primeira Pessoa	30.0	0.0	70.0	0.0	3
36	Tiro em Primeira Pessoa	30.0	0.0	70.0	0.0	3
37	Estratégia	65.0	0.0	0.0	0.0	2
38	Luta	0.0	0.0	50.0	50.0	1
39	Luta	0.0	0.0	50.0	50.0	1
40	Estratégia	0.0	45.0	0.0	55.0	4
41	Survival Horror	40.0	60.0	0.0	0.0	0
42	Tiro em Primeira Pessoa	0.0	0.0	70.0	30.0	1
43	Tiro em Primeira Pessoa	0.0	0.0	70.0	30.0	1
44	Tiro em Primeira Pessoa	40.0	0.0	60.0	0.0	3
45	Tiro em Primeira Pessoa	0.0	0.0	60.0	40.0	1
46	Ação / Aventura	40.0	40.0	20.0	0.0	0
47	Ação / Aventura	40.0	40.0	20.0	0.0	0
48	Ação / Aventura	50.0	0.0	0.0	50.0	2
49	Ação / Aventura	50.0	0.0	50.0	0.0	3
50	Ação / Aventura	60.0	40.0	0.0	0.0	0
51	Ação / Aventura	60.0	40.0	0.0	0.0	0
52	RPG	60.0	0.0	0.0	40.0	2
53	Simulação	25.0	0.0	0.0	75.0	2
54	Simulação	20.0	0.0	0.0	80.0	4
55	Simulação	15.0	0.0	0.0	85.0	4
56	Simulação	10.0	0.0	0.0	90.0	4
57	Estratégia	0.0	30.0	70.0	0.0	1
58	Esporte	50.0	0.0	50.0	0.0	3
59	Esporte	50.0	0.0	50.0	0.0	3
60	Esporte	50.0	0.0	0.0	50.0	2
61	Esporte	50.0	0.0	0.0	50.0	2
62	Esporte	60.0	0.0	0.0	40.0	2

ID	Genero	Achiever	Explorer	Killer	Socializer	cluster
63	Tiro em Primeira Pessoa	0.0	0.0	50.0	50.0	1
64	Ação	50.0	0.0	0.0	50.0	2
65	Plataforma	50.0	50.0	0.0	0.0	0
66	Plataforma 3D	70.0	30.0	0.0	0.0	0
67	Esporte	65.0	0.0	0.0	35.0	2
68	Corrida	50.0	0.0	0.0	50.0	2
69	Corrida	50.0	0.0	0.0	50.0	2
70	Corrida	70.0	0.0	0.0	30.0	2
71	Luta	40.0	0.0	40.0	20.0	3
72	Ação / Aventura	55.0	45.0	0.0	0.0	0
73	Música	80.0	0.0	0.0	20.0	2
74	Musica	80.0	0.0	0.0	20.0	2
75	Plataforma	60.0	40.0	0.0	0.0	0
76	RPG	40.0	40.0	20.0	0.0	0
77	RPG	60.0	40.0	0.0	0.0	0
78	RPG	50.0	0.0	50.0	0.0	3
79	RPG	50.0	50.0	0.0	0.0	0
80	RPG	50.0	50.0	0.0	0.0	0
81	Aventura Gráfica	0.0	75.0	0.0	25.0	4
82	Simulação	0.0	70.0	0.0	30.0	4
83	RPG	60.0	40.0	0.0	0.0	0
84	Luta	50.0	0.0	25.0	25.0	2
85	Ação / Aventura	65.0	35.0	0.0	0.0	0
86	Ação / Aventura	45.0	55.0	0.0	0.0	0
87	Estratégia	65.0	0.0	0.0	35.0	2
88	Plataforma 3D	60.0	40.0	0.0	0.0	0
89	Tiro em Terceira Pessoa	70.0	0.0	0.0	30.0	2
90	Ação / Aventura	45.0	55.0	0.0	0.0	0

ID	Genero	Achiever	Explorer	Killer	Socializer	cluster
91	Ação / Aventura	60.0	40.0	0.0	0.0	0
92	Ação / Aventura	55.0	45.0	0.0	0.0	0
93	Ação / Aventura	40.0	40.0	20.0	0.0	0
94	Estratégia	65.0	0.0	0.0	35.0	2
95	Tiro em Primeira Pessoa	50.0	50.0	0.0	0.0	0
96	Tiro em Terceira Pessoa	70.0	0.0	30.0	0.0	3
97	Tiro em Terceira Pessoa	70.0	0.0	30.0	0.0	3
98	Plataforma	60.0	40.0	0.0	0.0	0
99	Stealth	60.0	40.0	0.0	0.0	0
100	Stealth	50.0	50.0	0.0	0.0	0
101	Quebra-Cabeça	0.0	70.0	0.0	30.0	4
102	Quebra-Cabeça	0.0	75.0	0.0	25.0	4
103	Ação / Aventura	45.0	55.0	0.0	0.0	0
104	RPG	55.0	0.0	45.0	0.0	3
105	RPG	50.0	0.0	0.0	50.0	2
106	Esporte	30.0	0.0	0.0	70.0	2
107	Ação / Aventura	45.0	55.0	0.0	0.0	0
108	RPG	50.0	50.0	0.0	0.0	0
109	Beat em Up	80.0	0.0	0.0	20.0	2
110	RPG	35.0	65.0	0.0	0.0	0
111	Simulação	0.0	30.0	0.0	70.0	4
112	Plataforma	80.0	0.0	20.0	0.0	3
113	Plataforma	80.0	0.0	20.0	0.0	3
114	RPG	40.0	0.0	0.0	60.0	2
115	Plataforma	80.0	0.0	0.0	20.0	2
116	Plataforma	60.0	40.0	0.0	0.0	0
117	Ação	85.0	15.0	0.0	0.0	0
118	Estratégia	75.0	0.0	0.0	25.0	2

ID	Genero	Achiever	Explorer	Killer	Socializer	cluster
119	Estratégia	75.0	0.0	0.0	25.0	2
120	Quebra-Cabeça	80.0	0.0	0.0	20.0	2
121	Carta	0.0	30.0	0.0	70.0	4
122	Plataforma	75.0	25.0	0.0	0.0	0
123	Música	50.0	50.0	0.0	0.0	0
124	Aventura Gráfica	0.0	80.0	20.0	0.0	0
125	Ação	70.0	30.0	0.0	0.0	0
126	Ação / Aventura	50.0	0.0	0.0	50.0	2
127	Ação / Aventura	60.0	0.0	40.0	0.0	3
128	RPG	50.0	50.0	0.0	0.0	0
129	RPG	50.0	50.0	0.0	0.0	0
130	RPG	0.0	40.0	60.0	0.0	1
131	Estratégia	0.0	0.0	60.0	40.0	1
132	Corrida	60.0	0.0	0.0	40.0	2
133	Arcade	80.0	0.0	20.0	0.0	3
134	Arcade	80.0	0.0	20.0	0.0	3
135	RPG	60.0	0.0	0.0	40.0	2
136	Tiro em Primeira Pessoa	0.0	0.0	65.0	35.0	1
137	Simulação	60.0	0.0	0.0	40.0	2
138	Party	30.0	0.0	0.0	75.0	2
139	Party	30.0	0.0	0.0	70.0	2
140	RPG	60.0	40.0	0.0	0.0	0
141	Luta	0.0	0.0	50.0	50.0	1

Tabela A.4: Dados de jogos já classificados

Bibliografia

AARSETH, E. *Computer game studies, year one. Game Studies, 1(1)*. 2001. Disponível em: <http://www.gamestudies.org/0101/>.

BARTLE, R. Hearts, clubs, diamonds, spades: Players who suit muds. 1996.

BARTLE, R. *The Bartle Test of Gamer Psychology*. 2021. Disponível em: <https://matthewbarr.co.uk/bartle/>.

CONDE, D. de A. S. Sistema de recomendação de jogos baseado no perfil do jogador e aprendizado de máquina. 2021.

FERRO, L. S.; WALZ, S. P.; GREUTER, S. Towards personalised, gamified systems: an investigation into game design, personality and player typologies. p. 1–6, 2013.

JONES, E. et al. *SciPy: Open source scientific tools for Python*. 2001–2017. [Online; acessado em 24/11/2022]. Disponível em: <http://www.scipy.org/>.

KYATRIC. *Bartle's Taxonomy of Player Types (And Why It Doesn't Apply to Everything)*. 2021. Disponível em: <https://gamedevelopment.tutsplus.com/articles/bartles-taxonomy-of-player-types-and-why-it-doesnt-apply-to-everything--gamedev-4173>.

MEDEIROS, I. R. G. Estudo sobre sistemas de recomendação colaborativos. 2012. Disponível em: <https://www.cin.ufpe.br/~tg/2012-2/irgm.pdf>.

NETFLIX. *Como funciona o sistema de recomendação do netflix*. 2021. Disponível em: <https://help.netflix.com/pt/node/100639>.

RASCHKA, S.; MIRJALILI, V. *Python Machine Learning: Machine Learning and Deep Learning with Python*. [S.l.]: Packt Publishing, 2019.

SAWANT, S. Collaborative filtering using weighted bipartite graph projection: A recommendation system for yelp. 2013.

TIMES, N. *You're Tracked Everywhere You Go Online*. 2021. Disponível em: <https://www.nytimes.com/2019/11/24/smarter-living/privacy-online-how-to-stop-advertiser-tracking-opt-out.html>.