

**UNIVERSIDADE DE CAXIAS DO SUL
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E
ENGENHARIAS**

GABRIEL FERNANDO HANSEN

**SISTEMA DE MATCHMAKING HÍBRIDO PARA VALORANT:
OTIMIZAÇÃO E BALANCEAMENTO COM CLUSTERING
K-MEANS E CLASSIFICAÇÃO RANDOM FOREST**

CAXIAS DO SUL

2025

GABRIEL FERNANDO HANSEN

**SISTEMA DE MATCHMAKING HÍBRIDO PARA VALORANT:
OTIMIZAÇÃO E BALANCEAMENTO COM CLUSTERING
K-MEANS E CLASSIFICAÇÃO RANDOM FOREST**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharelado em Ciência da Computação na Área do Conhecimento de Ciências Exatas e Engenharias da Universidade de Caxias do Sul.

Orientador: Prof. Dra. Helena Graziottin Ribeiro

CAXIAS DO SUL

2025

GABRIEL FERNANDO HANSEN

**SISTEMA DE MATCHMAKING HÍBRIDO PARA VALORANT:
OTIMIZAÇÃO E BALANCEAMENTO COM CLUSTERING
K-MEANS E CLASSIFICAÇÃO RANDOM FOREST**

Trabalho de Conclusão de Curso
apresentado como requisito parcial à
obtenção do título de Bacharelado em
Ciência da Computação na Área do
Conhecimento de Ciências Exatas e
Engenharias da Universidade de Caxias
do Sul.

BANCA EXAMINADORA

Prof. Dra. Helena Graziottin Ribeiro
Universidade de Caxias do Sul - UCS

Prof. Dra. Elisa Boff
Universidade de Caxias do Sul - UCS

Prof. Dr. Marcelo Luís Fardo
Universidade de Caxias do Sul - UCS

RESUMO

O presente trabalho aborda o desenvolvimento de um sistema de *matchmaking* (*Matchmaking* é o processo de conexão e agrupamento de jogadores em partidas online, buscando formar equipes equilibradas com base em critérios como habilidades, desempenho e outros fatores comportamentais) para jogos de tiro em primeira pessoa (FPS) baseados em equipe, como *Valorant*, utilizando técnicas de *machine learning* para melhorar o equilíbrio das partidas. O objetivo principal é propor uma solução híbrida, combinando algoritmos de agrupamento (*K-Means*) para segmentação de jogadores e modelos preditivos (*Random Forest*) para avaliar a qualidade das partidas. Para fazer isso, o trabalho segue um processo que inclui coletar, preparar e analisar dados retirados do jogo *Valorant*. Os experimentos realizados demonstraram que o modelo desenvolvido é capaz de formar partidas equilibradas, considerando métricas como impacto dos jogadores, diversidade de funções e equilíbrio tático entre os times.

Palavras-chave: Machine Learning. Matchmaking. Clusterização. Modelo Preditivo. Sistema Competitivo. Balanceamento.

ABSTRACT

This work presents the development of a matchmaking system for team-based first-person shooter (FPS) games, such as Valorant, using machine learning techniques to improve match balance. The main goal is to propose a hybrid solution by combining clustering algorithms (K-Means) to segment player profiles and predictive models (Random Forest) to evaluate match quality. The methodology includes collecting, preprocessing, and analyzing gameplay data extracted from Valorant. The experiments showed that the proposed model is capable of forming balanced matches by considering metrics such as player impact, diversity of roles, and tactical balance between teams.

Keywords: Machine Learning; Matchmaking; Clustering; Predictive Model; Competitive System; Balance.

LISTA DE FIGURAS

Figura 1 – Funcionamento de um algoritmo de classificação	16
Figura 2 – Funcionamento do algoritmo K-means	17
Figura 3 – Imagem do site <i>Tracker.gg</i>	27
Figura 4 – <i>Knowledge Discovery in Databases</i> (KDD)	33
Figura 5 – Banco de dados	43
Figura 6 – <i>Clusters</i> PCA	47
Figura 7 – <i>Clusters</i> T-sne	48
Figura 8 – Exemplo de partidas geradas	51

LISTA DE ALGORITMOS

Algoritmo 1	Código em Python para K-Means	17
Algoritmo 2	Código em Python para AgglomerativeClustering	19
Algoritmo 3	Código em Python para Random Forest	20
Algoritmo 4	Exemplo de código Python utilizado para Kmeans	45
Algoritmo 5	Exemplo de código Python utilizado para Random Forest	49

LISTA DE ABREVIATURAS E SIGLAS

ML	<i>Machine Learning</i>
FPS	<i>First Person Shooter</i>
MMR	<i>Matchmaking Rating</i>
MOBA	<i>Multiplayer Online Battle Arena</i>
SBMM	<i>Skill-Based Matchmaking</i>
SVM	Máquina de Vetores de Suporte
KDD	<i>Knowledge Discovery in Databases</i>
PCA	Análise de Componentes Principais
ANN	Redes Neurais Artificiais
IA	Inteligência Artificial
WCSS	<i>Within-Cluster Sum of Squares</i>
DBI	Índice de <i>Davies-Bouldin</i>

SUMÁRIO

	Abstract	4
1	INTRODUÇÃO	10
1.1	Objetivos	11
1.2	Estrutura do trabalho	12
2	<i>MACHINE LEARNING APLICADA A JOGOS</i>	14
2.1	Aplicação de inteligência artificial e <i>machine learning</i> nos jogos	14
2.2	Métodos supervisionados e não supervisionados	15
2.2.1	Aprendizado Supervisionado	15
2.2.2	Aprendizado Não Supervisionado	16
2.3	Análise de dados	18
2.3.1	Uso de <i>clustering</i> para segmentação dos jogadores	18
2.3.2	Modelos preditivos	19
2.3.2.1	<i>Random forest</i>	20
2.3.3	Aprendizado supervisionado e não supervisionado nos jogos	21
2.4	Matriz de confusão	21
2.5	Métricas de avaliação para <i>machine learning</i>	22
2.6	MÉTRICAS DE AVALIAÇÃO PARA <i>CLUSTERING</i>	23
3	MATCHMAKING NOS JOGOS COMPETITIVOS	24
3.1	Sistema competitivos funcionais	25
3.1.1	Atributos	26
3.1.2	Tipos de <i>matchmaking</i>	27
3.2	Artigos relacionados	28
4	INTEGRAÇÃO DE <i>MACHINE LEARNING</i> E SISTEMAS DE <i>MATCHMAKING</i> EM JOGOS COMPETITIVOS	30
4.1	A importância da análise de dados	30
4.2	Segmentação de jogadores	31
4.3	Previsão de resultados	32
4.4	O futuro do <i>matchmaking</i> em jogos competitivos	32
4.5	Metodologia: o processo KDD	33
4.5.1	Seleção de dados	34
4.5.2	Pré-processamento e limpeza de dados	34
4.5.3	Transformação e redução de dados	34
4.5.4	Mineração de dados	34

4.5.5	Interpretação e avaliação	34
5	PROPOSTA: SISTEMA DE <i>MATCHMAKING</i> BASEADO EM ALGORITMOS DE <i>MACHINE LEARNING</i> HÍBRIDO COM <i>CLUSTERING</i> E MODELOS PREDITIVOS	36
5.1	Segmentação de jogadores por clustering	36
5.2	Modelo preditivo para avaliar a qualidade da partida	37
5.3	Recursos	38
5.3.1	Linguagem de Programação	38
5.3.2	Bibliotecas e Pacotes	38
5.3.3	Ferramentas de desenvolvimento e ambiente	39
5.4	Metodologia	39
6	IMPLEMENTAÇÃO DA SOLUÇÃO COM <i>MACHINE LEARNING</i> COM BASE NA METODOLOGIA KDD	42
6.1	Seleção e Coleta de dados	42
6.2	Pré-processamento e Transformação	43
6.3	Mineração de Dados: Clusterização com K-Means	45
6.4	Geração de Dados Supervisionados	48
6.5	Modelagem Preditiva com <i>Random Forest</i>	49
6.6	Avaliação e Geração de Partidas	50
6.7	Resumo do Experimento Realizado	52
7	CONCLUSÕES	53
7.1	Avaliação dos objetivos propostos	53
7.2	Aspectos Positivos e limitações	54
7.3	Trabalhos Futuros	54
	REFERÊNCIAS	56

1 INTRODUÇÃO

Nos últimos anos, os jogos competitivos online têm se tornado uma forma de entretenimento extremamente popular, atraindo milhões de jogadores ao redor do mundo. Um dos elementos centrais para a experiência do jogador nesses jogos é o sistema de *matchmaking*, que é responsável por parear jogadores de habilidades semelhantes para garantir partidas equilibradas e justas. No entanto, o *matchmaking* tradicional enfrenta diversos desafios que podem comprometer a qualidade das partidas e a satisfação dos jogadores.

Um dos principais problemas é a dificuldade em criar partidas equilibradas para jogos competitivos baseados em equipe, onde normalmente ocorre uma disputa entre dois times tendo 5 integrantes que podem possuir características ou objetivos individuais específicos, especialmente em jogos com uma grande base de jogadores e uma ampla variação de habilidades (YUKSEL, 2024). Sistemas de *matchmaking* baseados em regras simples ou em classificações de habilidade muitas vezes falham em considerar fatores contextuais importantes, como o desempenho recente do jogador, a sinergia entre membros da equipe e até mesmo o comportamento dos jogadores em partidas anteriores. (YUKSEL, 2024; RÜTTGERS; KUHL; PAASSEN, 2024)

Para melhorar os sistemas de *matchmaking*, diversas abordagens têm sido exploradas nos últimos anos. As propostas vão desde adaptações de modelos de classificação, como *Elo* e *Glicko*, a métodos mais sofisticados de *Machine Learning* (ML).

Uma das técnicas tradicionais adaptadas para melhorar o *matchmaking* é o sistema Glicko-2, que é uma evolução do sistema Elo. O sistema Elo é um método de classificação de jogadores baseado em estatísticas, amplamente utilizado em jogos competitivos para estimar a habilidade relativa dos jogadores. O Glicko-2 aprimora o sistema Elo ao incorporar uma variável de volatilidade, que ajusta o desempenho dos jogadores com mais precisão. Essa volatilidade representa a incerteza na avaliação da habilidade de um jogador, permitindo que o sistema seja mais responsivo a mudanças no nível de habilidade dos jogadores e oferecendo uma adaptação mais dinâmica nas classificações. Em outras palavras, o Glicko-2 não apenas avalia a habilidade de um jogador, mas também a consistência dessa habilidade ao longo do tempo. No entanto, seu uso em jogos de equipe é limitado devido à falta de consideração da sinergia entre jogadores (GLICKMAN, 2001a; YUKSEL, 2024).

Outra abordagem mais recente é a aplicação de modelos baseados em aprendizado supervisionado, como Redes Neurais Artificiais (ANN), para prever o equilíbrio das partidas com base em dados históricos de desempenho dos jogadores e características contextuais, como comunicação e habilidades individuais. Redes neurais permitem ao sistema identificar padrões complexos de desempenho que algoritmos mais simples não captam, promovendo partidas mais equilibradas e personalizadas (REZAPOUR; FATEMI; NEMATBAKSH, 2023; STONEMAN; MILLER;

COOPER, 2022).

Além disso, os modelos de *clustering* não supervisionado, como o *K-Means* e o *Agglomerative Clustering*, têm se mostrado eficazes na segmentação de jogadores de acordo com seus estilos de jogo. Essa técnica permite uma classificação mais detalhada, que leva em conta habilidades específicas e comportamento coletivo, aprimorando a composição de equipes equilibradas (RÜTTGERS; KUHL; PAASSEN, 2024; YUKSEL, 2024; JAIN; MURTY; FLYNN, 1999a).

Essas estratégias mostram um panorama diverso de iniciativas, desde aprimoramentos em sistemas clássicos de *rating* até o uso de técnicas avançadas de *machine learning*, destacando-se como ferramentas promissoras para o desenvolvimento de sistemas de *matchmaking* que consideram fatores dinâmicos e complexos na criação de partidas equilibradas. Para abordar esses desafios, este trabalho propõe o uso de técnicas de ML no desenvolvimento de sistemas de *matchmaking*. O ML, com sua capacidade de analisar grandes volumes de dados e identificar padrões complexos, oferece uma abordagem promissora para melhorar a precisão e a eficiência do *matchmaking*. Ao utilizar algoritmos de aprendizado de máquina, é possível criar modelos que considerem uma variedade de fatores e ajustem dinamicamente as partidas para maximizar a equidade e a satisfação dos jogadores.

1.1 OBJETIVOS

O objetivo deste trabalho é desenvolver e validar uma abordagem inteligente de *matchmaking* para jogos competitivos online utilizando técnicas de ML, capaz de formar partidas mais equilibradas por meio da análise do desempenho dos jogadores, de suas funções táticas e de seus perfis comportamentais identificados por meio de clusterização. Para tanto, foi realizado um estudo de caso, permitindo a análise prática e validação dos resultados obtidos. Especificamente, os objetivos deste trabalho são:

1. Propor uma melhoria para sistemas de *matchmaking* com base em técnicas de ML, considerando critérios significativos como desempenho recente dos jogadores, sinergia entre membros da equipe, comportamento em partidas anteriores e variabilidade de habilidades entre os jogadores.
2. Definir critérios significativos para avaliação da qualidade dos sistemas de *matchmaking*, como precisão no pareamento, equilíbrio nas partidas e satisfação dos jogadores, analisando fatores que afetam diretamente a experiência de jogo e o nível de competitividade.
3. Explorar e selecionar algoritmos de ML apropriados para o contexto de *matchmaking*, comparando diferentes modelos quanto à precisão, eficiência computacional e adaptabilidade em cenários de jogos de equipe com ampla variabilidade de habilidades.
4. Desenvolver um protótipo de sistema de *matchmaking* utilizando ML, integrando os algoritmos selecionados e ajustando-os para otimizar a formação de partidas equilibradas.

5. Validar a eficácia da proposta por meio de um estudo de caso, aplicando o sistema em um conjunto de dados ou ambiente de teste e avaliando seu impacto na qualidade das partidas com base nos critérios definidos.

1.2 ESTRUTURA DO TRABALHO

O presente trabalho está organizado da seguinte forma:

- O Capítulo 2 descreve os conceitos fundamentais de ML e sua aplicação no contexto de jogos. Serão abordados os princípios do ML, incluindo os métodos supervisionados e não supervisionados, com exemplos de como essas técnicas são utilizadas para aprimorar a experiência de jogo. Além disso, serão discutidas as técnicas de análise de dados e as métricas de avaliação de modelos de ML, fornecendo uma base teórica para o desenvolvimento do sistema de *matchmaking* proposto.
- O Capítulo 3 descreve os sistemas competitivos utilizados em jogos, com foco no processo de *matchmaking*. Serão apresentados os principais tipos de *matchmaking*, como o *Skill-Based Matchmaking* (SBMM), e os fatores considerados para o balanceamento de partidas, como o *Matchmaking Rating* (MMR). Também serão discutidos os desafios e limitações dos sistemas de *matchmaking* tradicionais e a importância da aplicação de ML para otimizar esse processo.
- O Capítulo 4 descreve a integração de técnicas de ML em sistemas de *matchmaking* para jogos competitivos. Serão exploradas as vantagens dessa integração, como a capacidade de analisar grandes volumes de dados e considerar múltiplos fatores para o balanceamento de partidas. Serão discutidos exemplos de como o ML pode ser utilizado para segmentar jogadores, prever resultados de partidas e personalizar a experiência de jogo, resultando em partidas mais justas e equilibradas.
- O Capítulo 5 descreve a proposta do trabalho, que consiste em um sistema de *matchmaking* híbrido que combina algoritmos de *clustering* e modelos preditivos. Serão detalhadas as etapas do sistema, incluindo a segmentação de jogadores por *clustering* (*K-Means*) e a previsão da qualidade da partida usando um modelo preditivo supervisionado (*Random Forest*). Serão apresentados os recursos necessários para a implementação do sistema, como linguagem de programação, bibliotecas e ferramentas de desenvolvimento, além da metodologia utilizada para o desenvolvimento e avaliação da proposta.
- O Capítulo 6 detalha a aplicação prática da proposta do trabalho, com base na metodologia KDD. São apresentadas as etapas de coleta e tratamento dos dados extraídos do jogo *Valorant*, incluindo pré-processamento, normalização e engenharia de atributos comportamentais. Em seguida, é descrita a aplicação do algoritmo *K-Means* para segmentação

dos jogadores em *clusters* de perfis semelhantes, e a construção de um modelo preditivo *Random Forest* para avaliar a qualidade das partidas. O capítulo também apresenta a geração de um *dataset* supervisionado, a definição de critérios heurísticos para balanceamento e a implementação de duas funções para criação automática e personalizada de times equilibrados.

- O Capítulo 7 apresenta a síntese dos resultados obtidos no desenvolvimento do trabalho, avaliando o cumprimento dos objetivos propostos. São destacados os principais aspectos positivos, como a eficácia da abordagem baseada em *Machine Learning* para formar partidas equilibradas, e também as limitações encontradas, como a necessidade de grandes volumes de dados e a ausência de testes em ambiente real. Por fim, são sugeridas direções para trabalhos futuros, incluindo melhorias no modelo, testes em tempo real e a inclusão de variáveis subjetivas no processo de *matchmaking*.

2 MACHINE LEARNING APLICADA A JOGOS

O ML tem se consolidado como uma tecnologia transformadora em diversas áreas, e sua aplicação no desenvolvimento de jogos eletrônicos está ganhando cada vez mais relevância. Nos jogos competitivos, onde a experiência do jogador é fortemente influenciada por fatores como o equilíbrio entre as equipes e a precisão do *matchmaking*, as técnicas de ML têm o potencial de otimizar processos complexos de análise e tomada de decisão. Utilizando grandes volumes de dados gerados durante as partidas, o ML permite que sistemas inteligentes identifiquem padrões de comportamento, ajustem a dificuldade das partidas e melhorem a precisão na formação de equipes, proporcionando uma experiência de jogo mais satisfatória e justa. Com essa capacidade de adaptar-se dinamicamente, o ML se torna uma ferramenta essencial para o desenvolvimento de jogos mais interativos e personalizáveis, onde o desafio e a satisfação do jogador são maximizados.

2.1 APLICAÇÃO DE INTELIGÊNCIA ARTIFICIAL E MACHINE LEARNING NOS JOGOS

ML, ou Aprendizado de Máquina em português, é uma área de estudo da Inteligência Artificial que se concentra no desenvolvimento de algoritmos capazes de aprender e fazer previsões ou tomar decisões baseadas em dados. Essa tecnologia permite que sistemas melhorem seu desempenho na realização de tarefas específicas com o passar do tempo, sem que seja necessária a intervenção humana direta. O aprendizado de máquina utiliza métodos estatísticos para capacitar os computadores a 'aprender' com os dados, identificando padrões e comportamentos. Isso é possível através de modelos matemáticos que são treinados usando grandes conjuntos de dados. Esses modelos são então testados e ajustados até que possam realizar previsões ou julgamentos precisos. O ML está por trás de muitas das tecnologias que usamos diariamente, como motores de busca, recomendações de produtos, reconhecimento de voz e imagem, e muito mais. Ele tem aplicações em uma variedade de campos, incluindo medicina, finanças, engenharia e entretenimento, tornando-se uma ferramenta valiosa para inovação e avanço tecnológico.

A Inteligência Artificial (IA) e o ML são utilizados em jogos para criar ambientes dinâmicos, aumentar a imersão e proporcionar uma experiência desafiadora e personalizada ao jogador. Em jogos competitivos, as técnicas de ML, como os métodos preditivos, são aplicadas para o desenvolvimento de sistemas de *matchmaking*, balanceamento de dificuldade e análise do comportamento do jogador. Com isso, a IA consegue gerar respostas que são adaptadas em tempo real com base nas interações dos jogadores e no contexto das partidas, tornando os sistemas mais responsivos e capazes de fornecer uma experiência de jogo equilibrada.

Um recurso importante para sistemas de *matchmaking* que tem contribuição de ML é

a possibilidade de ajustes dinâmicos na definição de parceiros de jogos. O modelo preditivo, alimentado por dados em tempo real, pode permitir ao sistema modificar suas previsões e, eventualmente, sugerir ajustes no pareamento com base no comportamento recente dos jogadores (JUNIOR; CAMPELO, 2023). A coleta contínua de dados das partidas, incluindo métricas de desempenho individual e coletiva, pode permitir uma adaptação constante (NISHIHARA *et al.*, 2017), garantindo que o sistema permaneça relevante conforme os jogadores evoluem em suas habilidades e comportamentos. Esse ajuste dinâmico, baseado em *feedback*, pode garantir que o sistema se adapte rapidamente a novas tendências e estilos de jogo que podem surgir ao longo do tempo (MA *et al.*, 2017). A capacidade de reavaliar jogadores com base em dados de partidas recentes também pode evitar que o sistema fique defasado, oferecendo uma experiência de *matchmaking* cada vez mais refinada.

2.2 MÉTODOS SUPERVISIONADOS E NÃO SUPERVISIONADOS

Os métodos de *machine learning* podem ser divididos em duas categorias principais: supervisionados e não supervisionados.

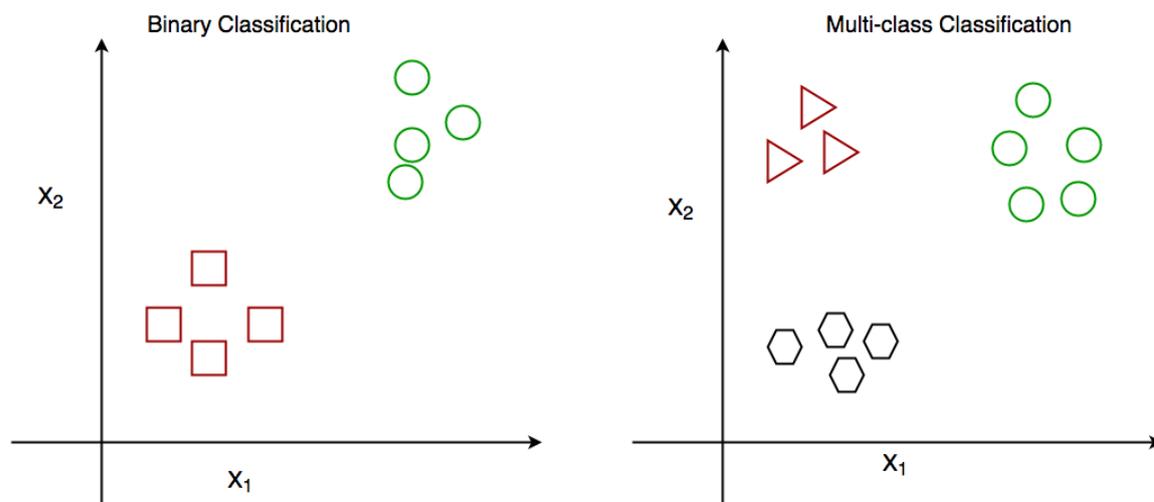
2.2.1 Aprendizado Supervisionado

No aprendizado supervisionado, o algoritmo é treinado com um conjunto de dados rotulados, onde as entradas correspondem a saídas conhecidas. O objetivo é aprender uma função que, dado um novo conjunto de dados, preveja a saída correta. Esse tipo de aprendizado é amplamente utilizado em jogos para classificação de habilidades de jogadores, como em sistemas de *matchmaking* onde a classificação dos jogadores é determinada a partir de seu desempenho. Uma técnica muito utilizada para avaliar o desempenho desses modelos, é a validação cruzada (BISHOP, 2006), a qual consiste em dividir o conjunto de dados em partes, onde cada parte é usada como conjunto de teste enquanto as outras servem para treinamento.

Exemplos de técnicas supervisionadas:

- **Classificação:** utilizada para categorizar jogadores em diferentes níveis de habilidade com base em métricas de performance. Um exemplo é o uso de *Random Forest* e ANN, que permitem identificar o nível de habilidade de um jogador e prever sua capacidade de colaboração com outros jogadores, como demonstrado com a figura 1 onde os objetos são classificados por seus formatos.
- **Regressão:** aplica-se para prever variáveis numéricas, como a pontuação provável de um jogador em uma partida futura, com base em seu histórico de desempenho.

Figura 1 – Funcionamento de um algoritmo de classificação



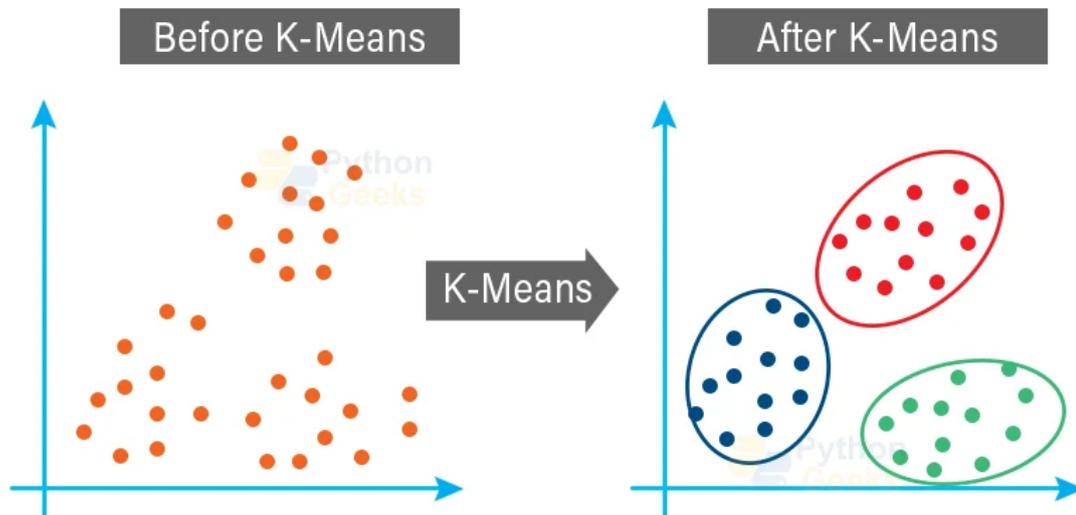
Fonte: Adaptado de (GEEKS, 2024b)

2.2.2 Aprendizado Não Supervisionado

O aprendizado não supervisionado é aplicado quando o modelo é treinado sem dados rotulados, ou seja, o sistema deve encontrar padrões ou agrupamentos nos dados sem conhecimento prévio das categorias. Nos jogos, essa abordagem é útil para segmentar jogadores de acordo com estilos de jogo e habilidades, permitindo que o sistema forme equipes equilibradas com jogadores que tenham sinergias específicas.

Clustering: uma técnica muito utilizada é o *K-Means*, representado pela figura 2 que é um algoritmo de *clustering* que segmenta dados em um número predefinido de grupos, ou *clusters*, com base em sua proximidade em um espaço dimensional. Ele é um método iterativo que seleciona pontos centrais (ou centróides) para cada *cluster*, calculando repetidamente a distância entre os dados e esses centróides, e os realocando para o *cluster* mais próximo até que não haja mais alterações significativas nos grupos formados (GUO *et al.*, 2004). Seu funcionamento pode ser dividido em 4 passos, sendo primeiramente a seleção do valor de K, onde K representa o número de vizinhos mais próximos que precisam ser considerados para fazer a previsão. O próximo passo é medir a distância euclidiana entre os dados de treinamento e destino, assim, encontrando os vizinhos mais próximos no passo três, que são os K com menor distância até o destino. Por fim, na classificação, é realizada a votação para a classificação, onde os rótulos de classe dos K vizinhos mais próximos são determinados. A classe com mais ocorrências entre os vizinhos se torna a classe prevista para o ponto de dados de destino conforme a figura 2. Já na regressão, o rótulo de classe é calculado tirando a média dos valores de destino dos K vizinhos mais próximos, e o valor médio calculado se torna a saída prevista para o ponto de dados de destino.

Figura 2 – Funcionamento do algoritmo K-means



Fonte: Adaptado de (GEEKS, 2024a)

Algoritmo 1 – Código em Python para K-Means

```
1 from sklearn.cluster import KMeans
2 import numpy as np
3
4 # Exemplo de dados de jogadores (habilidades, comunicacao, precisao)
5 dados_jogadores = np.array([
6     [5, 8, 7], [6, 9, 8], [7, 8, 9], [3, 5, 4], [4, 4, 3]
7 ])
8
9 # Inicializa o KMeans com o numero de clusters desejado
10 kmeans = KMeans(n_clusters=2, random_state=0)
11 # Ajusta o modelo aos dados e preve os clusters
12 clusters = kmeans.fit_predict(dados_jogadores)
```

Fonte: Documentação *scikit*

Redução de Dimensionalidade: algoritmos como Análise de Componentes Principais (PCA) reduzem a complexidade dos dados, permitindo uma análise mais eficiente de conjuntos com muitas variáveis e ajudando a identificar as características mais importantes para o desempenho no jogo. O PCA transforma variáveis possivelmente correlacionadas em um conjunto de componentes principais, que são combinações lineares das variáveis originais (BISHOP, 2006). O objetivo do PCA é capturar a maior variância dos dados em um menor número de dimensões, simplificando a análise e reduzindo a complexidade computacional.

Método *Elbow*: O método *Elbow* é uma técnica usada para determinar o número ideal de *clusters* em algoritmos de agrupamento como o *K-Means*. Ele avalia a soma dos erros quadrados

dentro dos *clusters* *Within-Cluster Sum of Squares* (WCSS) para diferentes números de *clusters*. À medida que o número de *clusters* aumenta, o WCSS diminui, pois os pontos ficam mais próximos de seus centróides.

O número ideal de *clusters* é encontrado no "cotovelo" da curva, onde a taxa de redução do WCSS começa a diminuir significativamente, indicando o ponto de equilíbrio entre complexidade e eficácia do modelo. Esse método ajuda a escolher o número de *clusters* de forma visual, fundamentando a escolha de uma configuração ideal no *K-Means*.

2.3 ANÁLISE DE DADOS

A análise de dados em *e-sports*, assim como nos esportes tradicionais, busca otimizar o desempenho competitivo e aprimorar as decisões estratégicas de equipes e jogadores. Com a crescente audiência e o valor econômico substancial dos *e-sports*, que atingiu aproximadamente US\$ 1,44 bilhão em 2023 (NEWZOO, 2023a), muitos campeonatos importantes de jogos como *League of Legends*, *Counter-Strike: Global Offensive* e *Valorant* são analisados por tecnologias avançadas para identificar padrões de jogo e antecipar comportamentos estratégicos. Este valor de mercado é sustentado principalmente por patrocínios e direitos de mídia, com patrocínios gerando US\$ 837 milhões em 2022 (STATISTA, 2023) e contribuindo para o crescimento contínuo da indústria.

Os eventos de *e-sports* atraem picos de audiência significativos; por exemplo, o Campeonato Mundial de *League of Legends* em 2023 alcançou 6,4 milhões de espectadores simultâneos (CHARTS, 2023), o que demonstra o grande interesse e o valor de *insights* detalhados sobre o desempenho dos times. Além disso, o número de fãs dos *e-sports* deve superar os 640 milhões até 2025 (NEWZOO, 2023b), aumentando a pressão sobre equipes para aplicar análise de dados em treinamento e competições.

No contexto estratégico, a análise de dados permite rastrear o desempenho de jogadores e equipes em tempo real, melhorando as habilidades e decisões táticas dos jogadores. Ferramentas avançadas de aprendizado de máquina e algoritmos de reconhecimento de imagem são empregados para mapear movimentos e prever lesões, o que ajuda a prolongar a vida útil dos atletas e otimizar o treinamento individual e coletivo. Assim, a aplicação de dados não só melhora a performance nos *e-sports*, mas também transforma o *e-sport* em um segmento cada vez mais rentável e influente globalmente.

Em resumo, a análise de dados aplicada aos *e-sports* é uma ferramenta valiosa que oferece *insights* quantitativos para melhorar o desempenho e a tomada de decisões.

2.3.1 Uso de *clustering* para segmentação dos jogadores

No contexto de jogos baseados em equipe, o uso de técnicas de *clustering* se mostra altamente eficaz para a segmentação de jogadores com base em seus estilos de jogo e habilidades

individuais (CHEN *et al.*, 2017). O *clustering* é uma técnica de aprendizado não supervisionado que agrupa dados semelhantes em *clusters*, permitindo a identificação de perfis de jogadores que compartilham características similares, como reflexos rápidos, habilidades de comunicação em equipe ou uma capacidade estratégica elevada (BISHOP, 2006). Ao aplicar essa técnica, é possível categorizar os jogadores em diferentes grupos de acordo com seus pontos fortes e fracos, facilitando o balanceamento de equipes em partidas competitivas. Isso pode contribuir significativamente para a formação de equipes mais homogêneas, em que as habilidades de cada jogador são equilibradas para proporcionar uma experiência de jogo mais justa e competitiva. Além disso, essa segmentação pode ajudar no desenvolvimento de estratégias personalizadas, otimizando o desempenho das equipes e aumentando a satisfação dos jogadores (JAIN; MURTY; FLYNN, 1999b) .

O *Agglomerative Clustering* é um método hierárquico que, não requer uma predefinição de número de *clusters*. Ele inicia com cada ponto de dado como um *cluster* individual e, a cada iteração, junta os dois *clusters* mais próximos. Esse processo continua até que um número específico de *clusters* seja alcançado ou até que a distância mínima seja excedida. A biblioteca *sklearn.cluster* disponibiliza o algoritmo de agrupamento hierárquico aglomerativo através da classe *Agglomerative Clustering*, permitindo sua aplicação em tarefas de *clustering*.

Algoritmo 2 – Código em Python para *AgglomerativeClustering*

```
1 from sklearn.cluster import AgglomerativeClustering
2
3 # Inicializa o AgglomerativeClustering
4 agg_clustering = AgglomerativeClustering(n_clusters=2)
5 # Ajusta o modelo aos dados
6 clusters_agg = agg_clustering.fit_predict(dados_jogadores)
```

Fonte: Documentação *scikit*

2.3.2 Modelos preditivos

Os modelos preditivos desempenham um papel essencial na previsão dos resultados de partidas em jogos competitivos. Utilizando dados históricos e estatísticas detalhadas dos jogadores, como desempenhos anteriores, taxas de acerto, tempo de reação, entre outros fatores, esses modelos são capazes de prever com certa precisão os possíveis desfechos de partidas futuras (MURPHY, 2012). A implementação de algoritmos de regressão e classificação tem sido amplamente utilizada para essa finalidade, permitindo que os sistemas de *matchmaking* não apenas organizem as equipes, mas também façam projeções sobre as chances de vitória com base nos atributos dos jogadores. Esses modelos fornecem uma base sólida para a criação de equipes equilibradas e para a formulação de estratégias, influenciando diretamente a qualidade e o equilíbrio das partidas.

2.3.2.1 *Random forest*

O *Random Forest* é um conjunto de algoritmos de decisão (árvores) que usa uma abordagem de *bagging*, na qual várias árvores de decisão são treinadas com amostras diferentes dos dados. Cada árvore é treinada para fazer uma previsão, e a previsão final do conjunto é feita pela média (para regressão) ou voto majoritário (para classificação) dos resultados das árvores. O *Random Forest* é robusto a dados ruidosos e pode capturar bem a não linearidade dos dados, tornando-o útil para prever a qualidade da partida (BREIMAN, 2001).

Algoritmo 3 – Código em Python para Random Forest

```
1  from sklearn.ensemble import RandomForestClassifier
2  from sklearn.model_selection import train_test_split
3
4  # Dados de exemplo de entrada (habilidade media e colaboracao)
5  X = np.array([
6      [7, 9], [6, 8], [5, 6], [3, 5], [8, 10], [4, 7]
7  ])
8  # Saidas esperadas (1 = jogo equilibrado, 0 = jogo desbalanceado)
9  y = np.array([1, 1, 0, 0, 1, 0])
10
11 # Divide os dados em conjunto de treinamento e teste
12 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
13 random_state=0)
14
15 # Inicializa o RandomForestClassifier
16 modelo_rf = RandomForestClassifier(n_estimators=100, random_state=0)
17 # Ajusta o modelo aos dados
18 modelo_rf.fit(X_train, y_train)
19
20 # Previsoes
21 previsoes = modelo_rf.predict(X_test)
```

Fonte: Documentação *scikit*

Além do *Random Forest*, outros métodos de classificação são amplamente utilizados para diferentes finalidades em ML. Árvores de Decisão são o método básico para classificação, onde as decisões são feitas em cada nó com base em características específicas dos dados. Máquina de Vetores de Suporte (SVM) são eficazes para classificar dados de alta dimensionalidade, buscando uma linha de separação entre as classes. *K-Nearest Neighbors* classifica um ponto com base nas classes dos seus vizinhos mais próximos, enquanto *Naive Bayes* é um classificador probabilístico que assume independência entre as características, sendo útil em tarefas como classificação de texto. ANN, por sua vez, são ideais para capturar padrões complexos em dados com múltiplas camadas de neurônios.

Essas técnicas oferecem diferentes abordagens para classificação e podem ser escolhidas com base nas características específicas dos dados e nos objetivos do modelo.

2.3.3 Aprendizado supervisionado e não supervisionado nos jogos

No contexto de *matchmaking*, as técnicas de aprendizado supervisionado e não supervisionado oferecem abordagens complementares para o desenvolvimento de sistemas mais eficientes e precisos. Algoritmos de aprendizado supervisionado, como ANN e SVM, são amplamente aplicados para prever a compatibilidade entre jogadores e criar equipes balanceadas com base em dados rotulados. Esses modelos utilizam históricos de desempenho e outras métricas para treinar o sistema a identificar combinações ideais de jogadores, resultando em partidas mais equilibradas.

Por outro lado, o aprendizado não supervisionado, como o uso de algoritmos de *clustering*, pode ser empregado para descobrir padrões ocultos nos dados, revelando novas formas de compatibilidade entre jogadores que não são facilmente identificadas com métodos tradicionais. Essa abordagem pode ser especialmente útil em jogos *multiplayer* online, onde as interações e dinâmicas de equipe são complexas, permitindo a criação de equipes que maximizam o potencial coletivo sem depender exclusivamente de atributos individuais.

Dessa forma, a combinação de aprendizado supervisionado e não supervisionado fornece um conjunto robusto de ferramentas para aprimorar o processo de *matchmaking*, garantindo uma melhor experiência para os jogadores e um nível mais elevado de competitividade nas partidas.

2.4 MATRIZ DE CONFUSÃO

A matriz de confusão é uma ferramenta visual amplamente utilizada na avaliação de modelos de classificação. Ela permite comparar as classes reais com as classes previstas pelo modelo, organizando essa informação em uma tabela que facilita a identificação dos acertos e erros do modelo em cada classe.

Seu uso é fundamental por diversos motivos: Avaliação detalhada do desempenho: permite verificar como o modelo se comporta em cada classe específica, evidenciando quais classes são mais difíceis de prever.

Identificação de vieses: auxilia na detecção de possíveis desequilíbrios ou tendências do modelo em favorecer uma classe em detrimento de outra.

Otimização do modelo: fornece informações claras sobre os tipos de erro (falsos positivos e falsos negativos), possibilitando ajustes finos nos hiperparâmetros e na estratégia de modelagem.

Comparação entre modelos: ao permitir uma análise detalhada dos erros, a matriz facilita a escolha do modelo com melhor desempenho global ou específico para determinadas classes.

A matriz de confusão é composta por quatro elementos principais:

- Verdadeiros Positivos (VP): Casos em que o modelo previu corretamente a classe positiva.
- Verdadeiros Negativos (VN): Casos onde o modelo previu corretamente a classe negativa.
- Falsos Positivos (FP): Casos em que o modelo previu a classe positiva, mas o valor real era negativo.
- Falsos Negativos (FN): Casos em que o modelo previu a classe negativa mas o valor real era positivo.

A partir desses valores, são calculadas as principais métricas de avaliação, como Acurácia, Precisão, *Recall* e *F1-Score*, que traduzem numericamente o desempenho do modelo.

2.5 MÉTRICAS DE AVALIAÇÃO PARA *MACHINE LEARNING*

A avaliação de modelos de ML é essencial para verificar a qualidade do aprendizado e garantir a eficácia das previsões realizadas. Dependendo do tipo de problema (classificação, regressão ou *clustering*), algumas das principais métricas utilizadas para avaliar classificadores são:

- Acurácia: Mede a proporção de previsões corretas em relação ao total de exemplos avaliados. É uma métrica simples, mas pode ser enganosa em conjuntos de dados desbalanceados.
- Precisão: Indica a proporção de verdadeiros positivos em relação a todos os exemplos classificados como positivos. É útil em casos onde minimizar falsos positivos é crucial.
- *Recall*: Mede a capacidade do modelo em identificar corretamente os exemplos positivos em relação ao total de positivos reais no conjunto de dados. É essencial em cenários onde os falsos negativos têm alto custo, como na detecção de fraudes ou diagnósticos médicos.
- *F1-Score*: Uma métrica combinada que representa a média harmônica entre precisão e *recall*. É ideal para problemas onde um equilíbrio entre esses dois aspectos é necessário.
- Curva ROC e AUC: Avalia o desempenho do modelo em diferentes limiares de decisão, considerando a taxa de verdadeiros positivos e falsos positivos. É frequentemente usada para comparar modelos de classificação.

Essas métricas ajudam a avaliar não apenas a eficácia dos modelos, mas também sua capacidade de generalização em dados novos, permitindo ajustes e melhorias contínuas no processo de aprendizado de máquina.

2.6 MÉTRICAS DE AVALIAÇÃO PARA *CLUSTERING*

No caso de algoritmos de aprendizado não supervisionado, como a clusterização, não existe um rótulo prévio para comparar diretamente os resultados. Portanto, são utilizadas métricas específicas que avaliam a coesão e a separação dos grupos formados.

As principais métricas aplicadas são:

- **Coefficiente de Silhueta:** Mede o quão bem um objeto está relacionado ao seu próprio *cluster* em comparação aos outros *clusters*. O valor varia entre -1 e 1, onde valores próximos de **1** indicam que os dados estão bem agrupados dentro do seu próprio *cluster* e distantes dos outros *clusters*, valores próximos de **0** indicam que o ponto está na fronteira entre dois *clusters* e valores negativos sugerem que os pontos foram agrupados no *cluster* errado (ROUSSEEUW, 1987).
- **Índice de *Davies-Bouldin*:** Avalia a média das razões entre a dispersão *intra-cluster* e a separação *inter-cluster*. Diferentemente do Coeficiente de Silhueta, para o Índice de *Davies-Bouldin* (DBI), valores menores indicam melhor qualidade do agrupamento, pois representam *clusters* mais compactos e bem separados. A fórmula é baseada na comparação de cada *cluster* com o *cluster* mais semelhante, considerando sua dispersão interna e a distância entre os centroides (DAVIES; BOULDIN, 1979).

3 MATCHMAKING NOS JOGOS COMPETITIVOS

A aplicação de ML no *matchmaking* de jogos competitivos, como *Valorant*¹ e *Counter Strike*², representa um avanço significativo na criação de partidas equilibradas e justas. O objetivo principal do ML no contexto de *matchmaking* é desenvolver sistemas que possam prever com precisão o resultado das partidas, levando em consideração a habilidade individual dos jogadores e a sinergia entre os membros da equipe. Estudos recentes, como o conduzido por (DEHPANAH *et al.*, 2021), demonstraram que é possível prever o resultado de jogos de *League of Legends*³ com uma precisão de 75.1% após a seleção dos campeões pelos jogadores, o que ocorre antes do início da partida. Embora este estudo seja focado em um *Multiplayer Online Battle Arena* (MOBA), os princípios podem ser adaptados para jogos *First Person Shooter* (FPS), como *Valorant* e *Counter Strike*.

Além da habilidade individual, o comportamento dos jogadores durante as partidas é um fator crucial que influencia o *matchmaking*. (DEHPANAH *et al.*, 2021) propuseram um sistema de avaliação comportamental dos jogadores em jogos de tiro competitivos online, onde características extraídas das estatísticas de jogo são utilizadas para modelar os jogadores e criar classificações que representem com precisão seu nível de desempenho. Isso sugere que, além da habilidade pura, aspectos comportamentais devem ser considerados para aprimorar a experiência de jogo.

A *Riot Games*, desenvolvedora do *Valorant*, implementa um sistema de *matchmaking* que leva em conta o tamanho do grupo de jogadores pré-formado e otimiza automaticamente a partida para favorecer tamanhos de grupo semelhantes na equipe adversária. Isso indica uma abordagem sofisticada que vai além da simples avaliação de habilidade, considerando também a dinâmica social e estratégica das equipes.

A integração de ML no *matchmaking* de jogos competitivos não apenas melhora a equidade das partidas, mas também contribui para uma experiência de jogo mais agradável e desafiadora. Ao analisar grandes volumes de dados de partidas anteriores, os algoritmos de ML podem identificar padrões e tendências que não são facilmente perceptíveis, permitindo ajustes mais refinados no sistema de *matchmaking*. Isso resulta em partidas mais equilibradas, onde jogadores de habilidades semelhantes são emparelhados, promovendo um ambiente competitivo saudável e estimulante.

Em resumo, o *Machine Learning* tem o potencial de revolucionar o *matchmaking* em jogos competitivos, tornando-o mais justo e agradável. À medida que a tecnologia avança e mais dados se tornam disponíveis, espera-se que os sistemas de *matchmaking* se tornem ainda

¹ <https://playvalorant.com/pt-br/>

² <https://www.counter-strike.net/cs2>

³ <https://www.leagueoflegends.com/pt-br/>

mais sofisticados, proporcionando uma experiência de jogo otimizada para todos os jogadores. (WU R.;MENG, 2024)

3.1 SISTEMA COMPETITIVOS FUNCIONAIS

Sistemas Competitivos Funcionais são uma área crucial no desenvolvimento e operação de jogos digitais. Estes sistemas são plataformas que oferecem uma variedade de serviços para suportar jogos online, incluindo hospedagem de servidores, *matchmaking*, análises de dados e ferramentas de operações ao vivo. Um exemplo proeminente é o *Steam*⁴, uma plataforma que revolucionou a distribuição digital de jogos. O *Steam* oferece uma interface amigável, uma vasta biblioteca de jogos e recursos comunitários robustos. No entanto, enfrenta críticas por seu modelo de receita e questões de segurança.

Outro sistema é o *PlayFab*⁵ da *Microsoft Azure*, que fornece uma gama completa de serviços de jogo, incluindo servidores *multiplayer*, análises em tempo real e ferramentas de *LiveOps*. É elogiado por sua escalabilidade e integração com outros serviços da *Azure*, mas os desenvolvedores podem encontrar desafios com a complexidade da configuração e custos potencialmente elevados.

*GameSparks*⁶ é uma plataforma que oferece *backend* como serviço, facilitando a criação de recursos de jogos como *leaderboards* e autenticação de jogadores. A plataforma é valorizada por sua flexibilidade e conjunto de ferramentas de teste, mas foi criticada por sua curva de aprendizado íngreme e suporte ao cliente após a aquisição pela *Amazon*.

Por fim, o *Amazon GameLift*⁷ é projetado para oferecer sessões de jogos de baixa latência e alta disponibilidade. Suas vantagens incluem a extensa infraestrutura da *Amazon* e a capacidade de escalar rapidamente. No entanto, os desenvolvedores podem enfrentar dificuldades com a complexidade da integração e a depuração de problemas de implantação.

Cada um desses sistemas contribui significativamente para o ecossistema de jogos digitais, oferecendo soluções que permitem aos desenvolvedores se concentrarem mais na criação de jogos envolventes e menos na manutenção de infraestrutura. As referências teóricas para essas informações podem ser encontradas em fontes acadêmicas sobre estratégias competitivas, literatura especializada em estratégia competitiva, análises de plataformas de jogos e documentação oficial dos serviços.

⁴ <https://store.steampowered.com/>

⁵ <https://playfab.com/>

⁶ <https://aws.amazon.com/pt/gamesparks/>

⁷ <https://aws.amazon.com/pt/gamelift/>

3.1.1 Atributos

O cálculo do MMR em jogos competitivos como *Valorant* e *Counter-Strike* é um processo complexo que leva em consideração diversos atributos para determinar a habilidade e o progresso dos jogadores (GLICKMAN, 2001b). No *Valorant*, o MMR é calculado usando um algoritmo que considera o desempenho individual, a performance da equipe, o tempo de jogo e o comportamento do jogador. O desempenho individual é avaliado através de métricas como taxa de eliminações, precisão, uso de habilidades e contribuição para os objetivos da partida. A performance da equipe também é crucial, pois vitórias e derrotas influenciam diretamente o MMR. O tempo de jogo pode afetar a confiabilidade do MMR, já que jogadores com mais partidas tendem a ter uma avaliação mais precisa. O comportamento do jogador, incluindo comunicação e colaboração, pode impactar positiva ou negativamente o MMR, dependendo da conduta durante as partidas. O *Valorant* também conta com classes de personagens que se classificam entre duelista, responsável pelo primeiro combate e ganho de espaço, iniciador, responsável por dar início as jogadas, auxiliando o duelista, controlador, que é capaz de controlar o espaço ganho pelo time, ou auxiliando a ganhar esse espaço fechando locais de avanço do time inimigo, e por fim, sentinela, que faz a função de jogar de forma mais lenta, precavendo cantos e costas dos jogadores.

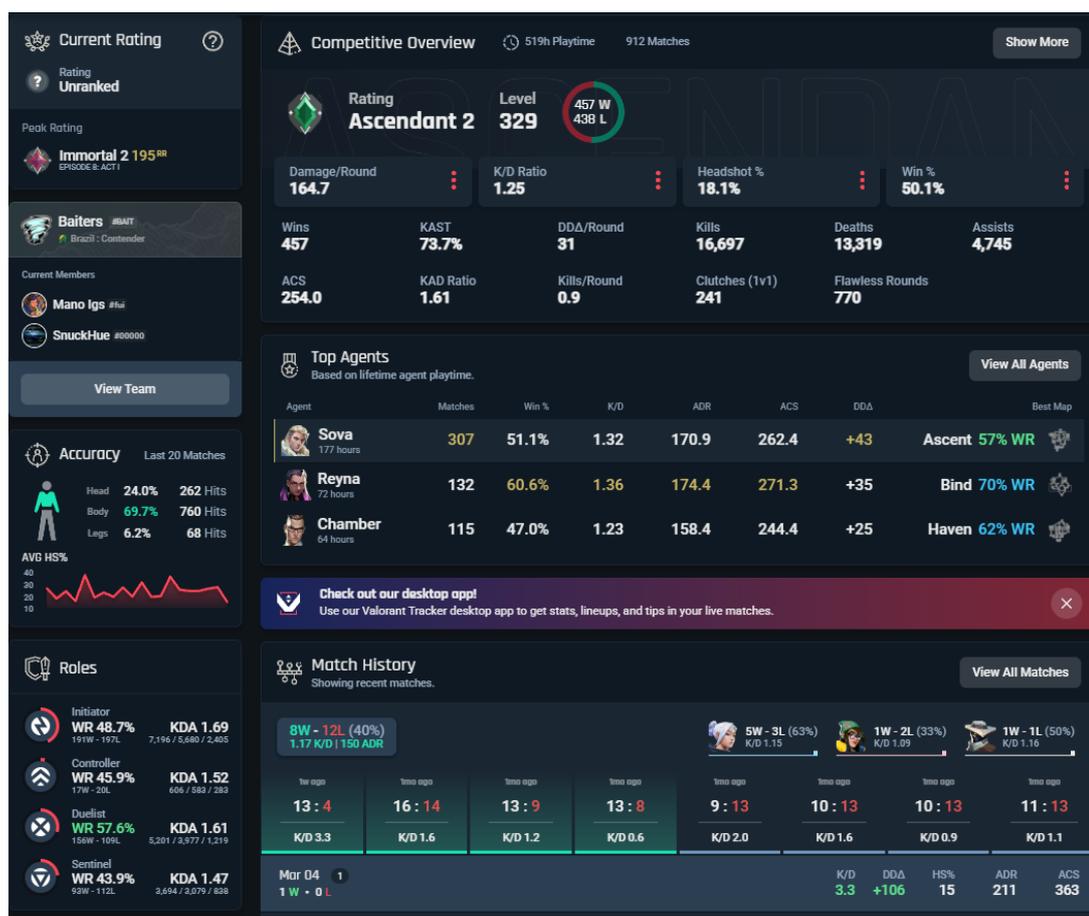
No *Counter-Strike*, embora os detalhes exatos do cálculo do MMR não sejam públicos, sabe-se que fatores como a vitória ou derrota da partida, a margem de vitória, o desempenho individual e o nível dos adversários são considerados. Jogadores que demonstram habilidades superiores e contribuem significativamente para o sucesso da equipe tendem a ganhar mais pontos de MMR. Além disso, comportamentos como comunicação eficaz e jogo em equipe são valorizados, podendo influenciar o MMR.

Para ambos os jogos, ferramentas externas como *Dak.gg*⁸ e *Tracker.gg*⁹ permitem que os jogadores acompanhem suas estatísticas e tenham uma ideia aproximada de seu MMR, mesmo que o valor exato não seja visível no jogo. Essas ferramentas analisam dados públicos das partidas e fornecem *insights* sobre a performance, o que pode ser útil para jogadores que buscam melhorar e subir de classificação. A figura 3 apresenta algumas informações competitivas de um jogador, como seus personagens mais usados, últimas partidas e desempenho individual na tela do *Tracker.gg*.

⁸ <https://dak.gg/>

⁹ <https://tracker.gg/>

Figura 3 – Imagem do site *Tracker.gg*



Fonte: Adaptado de *Tracker.gg*

3.1.2 Tipos de *matchmaking*

Existem diversos tipos de *matchmaking*, cada um com suas especificidades e aplicações. O *matchmaking* baseado em habilidade, conhecido como SBMM, é projetado para emparelhar jogadores de níveis semelhantes de habilidade, proporcionando uma competição justa e equilibrada (DOE; SMITH, 2020). Este sistema avalia o desempenho dos jogadores através de métricas específicas, ajustando o nível de desafio conforme a evolução do jogador.

O *matchmaking* por região, ou *Regional matchmaking*, visa reduzir a latência nas partidas ao conectar jogadores que estão geograficamente próximos (DOE; SMITH, 2020). Isso é particularmente importante em jogos online, onde o tempo de resposta é crítico para a experiência do usuário. Já o *matchmaking* baseado em preferências permite que os jogadores selecionem certos critérios, como modo de jogo, regras específicas ou tipos de oponentes, personalizando assim a sua experiência de jogo.

A aplicação desses sistemas de *matchmaking* em um projeto deve considerar o público-alvo e os objetivos do jogo ou plataforma. Por exemplo, um jogo competitivo pode se beneficiar mais do SBMM para manter o equilíbrio e a competitividade, enquanto uma plataforma so-

cial pode priorizar o *matchmaking* por região para fomentar comunidades locais. Além disso, é essencial que o sistema de *matchmaking* seja transparente e justo, evitando frustrações e garantindo a satisfação dos usuários.

Uma das principais limitações dos sistemas de classificação convencionais reside em sua abordagem unidimensional, na qual o desempenho do jogador é representado por um único valor numérico. Entretanto, em FPS com base em equipes, as habilidades dos jogadores devem ser avaliadas de forma mais abrangente, considerando aspectos como reflexos, capacidade de leitura do jogo e comunicação em equipe. Dessa forma, a construção de um perfil de jogador que inclua informações contextuais mais ricas pode contribuir significativamente para o equilíbrio das partidas, evidenciando que a classificação por uma única faceta é insuficiente (CHEN *et al.*, 2017).

3.2 ARTIGOS RELACIONADOS

Diversos estudos têm explorado o uso de *machine learning* para melhorar o balanceamento em jogos competitivos, abordando aspectos como análise de habilidades, comportamento de jogadores e previsões de resultados. Abaixo são apresentados alguns artigos relevantes para o tema deste trabalho.

Evaluating Team Skill Aggregation in Online Competitive Games (DEHPANAH *et al.*, 2021). Este estudo apresenta uma abordagem para avaliação das habilidades de equipe em jogos online competitivos, especialmente no contexto de jogos *multiplayer*. O artigo propõe um modelo usando sistemas de classificação como *Elo*, *Glicko* e *TrueSkill* que considera tanto as habilidades individuais quanto as habilidades agregadas da equipe, demonstrando que a sinergia entre os membros influencia diretamente os resultados das partidas. Os resultados indicam que a integração de habilidades coletivas nos sistemas de *matchmaking* pode proporcionar partidas mais equilibradas e melhorar a experiência dos jogadores.

Achieving Fairness in Team-Based FPS Games: A Skill-Based Matchmaking Solution (WU R.; MENG, 2024). O artigo explora uma solução de *matchmaking* voltada para jogos de FPS baseados em equipe. A proposta utiliza um modelo de ML baseado em árvores de decisão e *Random Forest* para ajustar dinamicamente o emparelhamento de acordo com habilidades individuais e coletivas dos jogadores, visando aumentar a equidade das partidas. Os resultados mostram que este método reduz a ocorrência de partidas desbalanceadas e melhora a retenção dos jogadores, especialmente em jogos com grande variação de habilidades.

Skill-Based Matchmaking for Competitive Two-Player Games (RÜTTGERS; KUHL; PASSEN, 2024). Yuksel explora o impacto de sistemas de *matchmaking* baseados em habilidades em jogos competitivos de dois jogadores. O estudo implementa diferentes abordagens de *matchmaking*, incluindo o uso de modelos estatísticos para prever o equilíbrio das partidas, considerando habilidades e experiências anteriores dos jogadores. Esse estudo destaca a impor-

tância da adaptação contínua dos sistemas de *matchmaking* e a necessidade de incorporar dados comportamentais para alcançar melhor balanceamento e satisfação dos jogadores.

Tabela 1 – Quadro Comparativo

Características	Evaluating Team Skill Aggregation in Online Competitive Games	Achieving Fairness in Team-Based FPS Games: A Skill-Based Matchmaking Solution	Evaluating Team Skill Aggregation in Online Competitive Games
Abordagem Principal	Avaliação das habilidades agregadas de equipe	Solução baseada em ML para FPS com ajustes dinâmicos	Matchmaking baseado em habilidades e experiências
Tipo de Jogo Focado	Jogos Multiplayer em geral	FPS baseados em equipe	Jogos competitivos de dois jogadores
Fatores Considerados	Habilidades individuais e sinergia coletiva	Habilidades individuais e coletivas, ajustes em tempo real	Habilidades, experiências anteriores e dados comportamentais
Resultados Principais	Aumento no equilíbrio das partidas e experiência do jogador	Redução de desbalanceamento e melhora na retenção de jogadores	Melhoria no balanceamento e satisfação dos jogadores
Método Utilizado	Modelos híbridos com análise estatística e comportamental	Modelos preditivos dinâmicos com Random Forest	Modelos estatísticos para prever equilíbrio

4 INTEGRAÇÃO DE *MACHINE LEARNING* E SISTEMAS DE *MATCHMAKING* EM JOGOS COMPETITIVOS

A utilização de técnicas de ML para aprimorar sistemas de *matchmaking* em jogos competitivos representa um avanço significativo na busca por partidas justas e equilibradas. A aplicação de algoritmos de aprendizado de máquina permite não apenas a análise de habilidades individuais dos jogadores, mas também a consideração de fatores contextuais que influenciam o desempenho nas partidas. Nesse sentido, é possível construir um sistema de *matchmaking* que não apenas classifique os jogadores com base em um único valor numérico, mas que também avalie a sinergia entre os membros da equipe e o comportamento dos jogadores durante as partidas (CHEN *et al.*, 2017) (WU R.; MENG, 2024).

4.1 A IMPORTÂNCIA DA ANÁLISE DE DADOS

A análise de dados desempenha um papel crucial na implementação de sistemas de *matchmaking* eficazes. A coleta e análise de informações sobre o desempenho dos jogadores em partidas anteriores são essenciais para a construção de modelos que prevejam resultados futuros. De acordo com (DEHPANAH *et al.*, 2021), o comportamento dos jogadores e suas características de desempenho devem ser modelados para criar classificações que reflitam com precisão o nível de habilidade. Essa modelagem permite que os sistemas de *matchmaking* não apenas considerem a habilidade média dos jogadores, mas também fatores como a eficácia da comunicação e a colaboração durante as partidas.

Além disso, a análise esportiva já é amplamente utilizada para otimizar o desempenho dos atletas em várias disciplinas, incluindo esportes tradicionais e *e-sports* (ALAMAR, 2013). Essa adaptação de métodos estatísticos e computacionais para o ambiente de jogos online é uma estratégia promissora, uma vez que possibilita a identificação de padrões de desempenho que podem ser utilizados para aprimorar a experiência de jogo. As abordagens que utilizam análise preditiva podem ajustar dinamicamente as partidas com base em dados coletados em tempo real, resultando em um sistema de *matchmaking* mais responsivo e preciso.

Um exemplo claro dessa abordagem é a implementação de sistemas de *feedback* em tempo real, que avaliam o desempenho dos jogadores durante as partidas. Essa capacidade de ajustar o *matchmaking* com base em informações atualizadas aumenta a equidade das partidas, contribuindo para a satisfação do jogador e a longevidade do jogo. Assim, a análise de dados não apenas fornece uma base sólida para a construção de sistemas de *matchmaking*, mas também transforma a forma como os jogos competitivos são estruturados e vivenciados pelos jogadores.

A implementação de sistemas competitivos funcionais, como os oferecidos por plataformas como *Steam* e *PlayFab*, tem demonstrado que a integração de serviços de hospedagem de

servidores, análises de dados e *matchmaking* é fundamental para criar uma experiência de jogo satisfatória (WU R.;MENG, 2024). Essas plataformas oferecem a infraestrutura necessária para o funcionamento dos jogos, permitindo que desenvolvedores se concentrem mais na criação de conteúdos envolventes do que na manutenção de infraestrutura.

Esses sistemas competitivos funcionais não apenas hospedam os jogos, mas também coletam dados valiosos que podem ser utilizados para aprimorar continuamente os algoritmos de *matchmaking*. Por exemplo, a integração de análises de comportamento permite que as plataformas ajustem as condições de *matchmaking* para evitar combinações de jogadores que possam resultar em partidas desbalanceadas. Com isso, há uma clara melhoria na experiência do usuário, uma vez que os jogadores são emparelhados com base em critérios que vão além das habilidades individuais.

Além disso, a escalabilidade dessas plataformas permite que elas se adaptem a uma base de usuários crescente, mantendo a qualidade do serviço mesmo em momentos de alta demanda. A utilização de dados em larga escala para alimentar algoritmos de ML oferece *insights* que podem ser usados para prever e otimizar o *matchmaking* de maneira eficiente. Portanto, a evolução e adaptação contínua desses sistemas são cruciais para a sustentabilidade e popularidade dos jogos competitivos.

4.2 SEGMENTAÇÃO DE JOGADORES

A aplicação de técnicas de *clustering* para segmentar jogadores com base em estilos de jogo e habilidades individuais tem se mostrado uma estratégia eficaz. O uso de *clustering*, uma técnica de aprendizado não supervisionado, permite identificar perfis de jogadores que compartilham características semelhantes, contribuindo para a formação de equipes equilibradas (JAIN; MURTY; FLYNN, 1999b). Isso é particularmente relevante em jogos de FPS, onde as habilidades exigidas variam amplamente entre os jogadores.

A segmentação dos jogadores não apenas melhora o balanceamento das equipes, mas também facilita a criação de estratégias específicas adaptadas a cada grupo. Por exemplo, jogadores que se destacam em comunicação podem ser agrupados com aqueles que possuem habilidades técnicas superiores, otimizando o desempenho geral da equipe. Além disso, o *clustering* pode revelar padrões de jogo que não são facilmente identificáveis por métodos tradicionais, permitindo uma compreensão mais profunda das dinâmicas de equipe.

Com o uso dessas técnicas, os desenvolvedores podem criar experiências de jogo mais personalizadas e adaptativas, onde os jogadores se sentem mais engajados e motivados a melhorar suas habilidades. Assim, a segmentação através de *clustering* não é apenas uma ferramenta para balanceamento, mas uma estratégia essencial para enriquecer a experiência competitiva.

4.3 PREVISÃO DE RESULTADOS

Os modelos preditivos desempenham um papel essencial na previsão dos resultados das partidas com base em dados históricos. A precisão na previsão é crucial para a eficácia do *matchmaking*, pois permite que o sistema forme equipes com chances semelhantes de vencer, promovendo uma competição justa (DEHPANAH *et al.*, 2021). A utilização de algoritmos de regressão e classificação para essa finalidade é amplamente aplicada, permitindo que os sistemas não apenas organizem as equipes, mas também façam projeções sobre as chances de vitória com base nas características dos jogadores.

Esses modelos preditivos utilizam dados como taxas de eliminação, precisão e contribuição para os objetivos da partida, entre outros fatores, para determinar a probabilidade de vitória. Ao incorporar variáveis comportamentais, como a comunicação e a colaboração entre os membros da equipe, os algoritmos se tornam mais eficazes na previsão de resultados, promovendo um ambiente de jogo mais equilibrado e agradável.

Adicionalmente, o uso de técnicas de aprendizado de máquina em modelos preditivos possibilita um ajuste contínuo do *matchmaking*, permitindo que as plataformas aprendam com os resultados passados e se adaptem às novas dinâmicas de jogo. Assim, a implementação de modelos preditivos não apenas melhora o equilíbrio das partidas, mas também contribui para a satisfação geral dos jogadores, que se beneficiam de uma experiência de jogo mais justa e envolvente.

4.4 O FUTURO DO *MATCHMAKING* EM JOGOS COMPETITIVOS

A evolução dos sistemas de *matchmaking* através da implementação de ML representa uma mudança paradigmática no design de jogos competitivos. À medida que mais dados se tornam disponíveis e a tecnologia avança, espera-se que os sistemas de *matchmaking* se tornem cada vez mais sofisticados, proporcionando um ambiente competitivo saudável e estimulante (WU R.; MENG, 2024).

No entanto, é importante destacar que o acesso a dados de partidas e de desempenho dos jogadores não é completamente aberto. A maioria das plataformas e jogos, incluindo o *Valorant*, oferece acesso a dados por meio de APIs públicas, porém com restrições. O acesso geralmente exige um cadastro na plataforma de desenvolvedores da *Riot Games*, mediante geração de uma chave de API, que possui limites de requisições por tempo e está sujeita a termos de uso específicos. Além disso, há regras que visam proteger a privacidade dos jogadores, restringindo o acesso a informações sensíveis e permitindo apenas dados relacionados a partidas públicas e informações estatísticas associadas a contas que estejam visíveis na plataforma.

Portanto, embora os dados sejam relativamente acessíveis para pesquisa e desenvolvimento, seu uso está condicionado a diretrizes e políticas de privacidade estabelecidas pela

desenvolvedora, garantindo que haja controle e responsabilidade na utilização dessas informações.

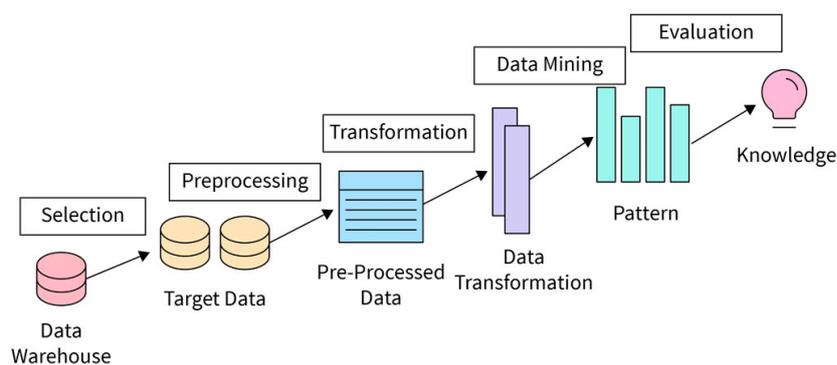
A integração de técnicas de ML não apenas melhora a equidade nas partidas, mas também contribui para uma experiência de jogo mais agradável, refletindo a crescente importância da análise de dados no desenvolvimento de jogos digitais.

O futuro do *matchmaking* promete ser marcado por uma maior personalização e adaptabilidade, à medida que os algoritmos se tornam mais inteligentes e responsivos ao comportamento dos jogadores. Com a utilização de dados em tempo real e a aplicação de técnicas avançadas de ML, os sistemas de *matchmaking* podem se adaptar rapidamente às necessidades e preferências dos jogadores, promovendo uma experiência de jogo mais satisfatória e envolvente. Dessa forma, a transformação contínua do *matchmaking* não apenas redefine a experiência do jogador, mas também fortalece a comunidade de jogos competitivos como um todo.

4.5 METODOLOGIA: O PROCESSO KDD

Para estruturar o processo de análise de dados neste trabalho, será empregada a metodologia de Descoberta de Conhecimento em Bases de Dados (KDD). O KDD é um processo sistemático e iterativo que visa a extração de conhecimento útil a partir de grandes volumes de dados, aplicando técnicas estatísticas e de mineração de dados, especialmente em contextos onde o volume e a complexidade dos dados demandam processos robustos de análise (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

Figura 4 – KDD



Fonte: Adaptado de (MADEIRA, 2013)

4.5.1 Seleção de dados

A primeira etapa do processo KDD envolve a seleção dos dados que serão utilizados na análise. No contexto dos *e-sports*, como *Valorant*, *Counter-Strike: Global Offensive* e *League of Legends*, os dados podem incluir informações de desempenho de jogadores, estatísticas de partidas, padrões de movimento e comportamento de jogo. A seleção dos dados é crucial, e envolverá plataformas como a *GRID Platform* e a *APIs* como a *VALORANT API*.

4.5.2 Pré-processamento e limpeza de dados

Nesta etapa, os dados selecionados são preparados e limpos para garantir que estejam prontos para a análise. Isso inclui a remoção de valores ausentes, tratamento de inconsistências e transformação dos dados em formatos adequados para análise (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). Como os dados de *e-sports* frequentemente contêm ruídos (por exemplo, logs de partidas e interações não relevantes), o pré-processamento assegura que apenas as informações válidas e consistentes sejam consideradas.

4.5.3 Transformação e redução de dados

Com os dados limpos, a próxima fase envolve transformá-los em representações que facilitam o processo de mineração. Para dados de *e-sports*, isso significa a criação de métricas de desempenho derivadas ou a normalização de variáveis para tornar os dados comparáveis. Técnicas de redução, como PCA ou seleção de atributos, serão utilizadas para diminuir a dimensionalidade dos dados, sem perda significativa de informação relevante para a análise (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

4.5.4 Mineração de dados

A fase de mineração de dados é o núcleo do processo KDD, onde algoritmos que implementam métodos de machine learning são aplicados para identificar padrões e relações entre os dados. Este trabalho focará em algoritmos de agrupamento, classificação e regressão, considerando especialmente o contexto de *e-sports*. A ideia é utilizar essas técnicas para identificar grupos de jogadores com habilidades semelhantes, prever o desempenho com base em características de jogo e compreender fatores que influenciam o sucesso ou fracasso em partidas competitivas.

4.5.5 Interpretação e avaliação

Por fim, a última etapa consiste na interpretação dos resultados da mineração de dados e na avaliação da eficácia dos modelos empregados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). Os *insights* extraídos devem ser validados e contextualizados, analisando se os padrões

identificados são significativos e aplicáveis ao contexto dos *e-sports*. Para validar os modelos, serão utilizados métodos como validação cruzada e métricas de acurácia e precisão. Esse processo garante que os conhecimentos obtidos sejam confiáveis e possam ser utilizados para decisões estratégicas na gestão e treinamento de equipes de *e-sports*.

5 PROPOSTA: SISTEMA DE MATCHMAKING BASEADO EM ALGORITMOS DE MACHINE LEARNING HÍBRIDO COM CLUSTERING E MODELOS PREDITIVOS

A proposta desse trabalho tem como foco implementar e testar um sistema de *matchmaking* desenvolvido para jogos de FPS baseados em equipe, como *Valorant*. Este tipo de jogo é caracterizado pela alta exigência de coordenação entre os jogadores e pela necessidade de um balanceamento eficiente entre as equipes, garantindo que as partidas sejam justas e equilibradas.

A solução proposta é um sistema híbrido de aprendizado supervisionado e não supervisionado que visa aprimorar o balanceamento das equipes e reduzir a ocorrência de partidas desiguais. O sistema é dividido em duas fases principais:

- Segmentação dos jogadores com base em características múltiplas usando *Clustering*: Nesta fase, técnicas de aprendizado não supervisionado, como *K-Means*, são utilizadas para agrupar jogadores com base em métricas específicas (taxa de acerto, tempo de reação, eficácia em comunicação de equipe, entre outros). Este processo permite identificar perfis de jogadores que compartilham características semelhantes, ajudando a compor equipes com sinergia e habilidades equilibradas.
- Previsão da qualidade da partida usando um modelo preditivo supervisionado: Após a segmentação, um modelo de aprendizado supervisionado, como *Random Forest*, é usado para prever a qualidade das partidas, considerando o balanceamento das equipes. Esse modelo é ajustado dinamicamente com base em dados de desempenho, aumentando a precisão da previsão sobre o equilíbrio da partida e a experiência do jogador.

5.1 SEGMENTAÇÃO DE JOGADORES POR CLUSTERING

A primeira etapa da proposta envolve o uso de um algoritmo de *clustering* (aprendizado não supervisionado) para agrupar os jogadores em perfis com base em múltiplos fatores além do simples nível de habilidade. Em vez de apenas classificar jogadores com base em suas taxas de vitória ou derrotas, o sistema utilizou métricas como:

- Taxa de eliminações por minuto;
- Taxa de assistências;
- Precisão nos disparos;
- Tempo médio de sobrevivência;

- Contribuição para os objetivos da equipe;

Essas variáveis foram extraídas dos históricos de jogos e analisadas com um algoritmo como o *K-means* ou *Agglomerative Clustering*. O algoritmo agrupa os jogadores em diferentes *clusters* de acordo com suas habilidades e comportamentos em partidas anteriores. Essa segmentação rica em informações possibilita a formação de equipes mais equilibradas, onde os jogadores são emparelhados com base em suas habilidades específicas e na sinergia potencial com outros jogadores. Diferente do método tradicional, que trata os jogadores como entidades isoladas, essa abordagem leva em conta as características coletivas da equipe.

Justificativa do Clustering: A aplicação do *clustering* permite que o sistema vá além da simples análise de uma métrica unidimensional, categorizando os jogadores em perfis complexos e considerando múltiplos aspectos de sua jogabilidade. Isso é especialmente útil em jogos com foco em trabalho em equipe, como FPS, onde habilidades como comunicação e leitura do jogo são tão importantes quanto a capacidade técnica individual (JAIN; MURTY; FLYNN, 1999a).

5.2 MODELO PREDITIVO PARA AVALIAR A QUALIDADE DA PARTIDA

Após a segmentação dos jogadores, um modelo preditivo supervisionado é empregado para prever a qualidade da partida antes de ela ser iniciada. Este modelo tem como entrada as características de cada jogador, já segmentadas pelo *cluster*, e as informações da equipe formada. Utilizando um algoritmo de regressão, como *Random Forest* ou ANN, o sistema calcula a probabilidade de a partida ser equilibrada, isto é, de ambas as equipes terem uma chance semelhante de vitória.

As variáveis de entrada para o modelo preditivo incluem:

- A composição da equipe (proporção de jogadores com diferentes perfis);
- A sinergia prevista entre os jogadores (baseada no histórico de desempenho em conjunto);
- A média e desvio padrão de desempenho de cada equipe;
- A taxa de sucesso em partidas anteriores com composições semelhantes.

O modelo é treinado com um grande volume de dados de partidas anteriores no modo competitivo do jogo, no qual os jogadores costumam ter mais empenho e melhor performance, pois este é o único modo remunerado pelo jogo através de ranqueamento, onde a qualidade das partidas é definida por métricas como a diferença de pontuação final e o nível de satisfação dos jogadores (avaliado por tempo de permanência no jogo). A partir desses dados, o modelo aprende quais

combinações de jogadores levam a partidas mais equilibradas, otimizando o pareamento para minimizar desbalanços.

Vantagens do Modelo Preditivo: O uso de um modelo preditivo supervisionado permite que o sistema faça ajustes antes do início da partida, evitando o pareamento de jogadores com grandes diferenças de habilidade ou que não possuam boa sinergia. Esse sistema aprende continuamente com os resultados das partidas anteriores, ajustando suas previsões de acordo com as mudanças nos padrões de jogo e no comportamento dos jogadores (DEHPANAH *et al.*, 2021).

5.3 RECURSOS

A implementação do sistema de *matchmaking* proposto, requer um conjunto de recursos, incluindo linguagem de programação, bibliotecas, e ferramentas de desenvolvimento. Abaixo estão descritos os principais recursos que foram utilizados para o desenvolvimento e execução da proposta:

5.3.1 Linguagem de Programação

Python foi escolhida como a linguagem principal para o desenvolvimento da proposta. *Python* é amplamente utilizado em projetos de ML e análise de dados devido à sua sintaxe simples e ao suporte a uma vasta gama de bibliotecas para manipulação de dados, aprendizado de máquina e visualização. Além disso, *Python* oferece integração fácil com APIs, o que é essencial para extrair dados da API do *Valorant*.

5.3.2 Bibliotecas e Pacotes

Para implementar o sistema de *matchmaking*, foram utilizadas as seguintes bibliotecas e pacotes *Python*:

Pandas: Utilizada para manipulação e análise de dados. A biblioteca facilita o pré-processamento, limpeza e organização dos dados extraídos da API, permitindo a preparação eficiente do *dataset* para os modelos de ML.

NumPy: Uma biblioteca essencial para operações matemáticas e manipulação de matrizes, o que é necessário para as operações algébricas em ML e para a organização dos dados em estruturas de *arrays*.

Scikit-Learn: Esta biblioteca oferece uma ampla variedade de algoritmos de ML, incluindo o *Random Forest*, a Regressão Logística e o *K-Means*, que foram usados nos experimentos de *clustering* e classificação. Além disso, a *Scikit-Learn* oferece funções para validação cruzada e avaliação de métricas de desempenho, como precisão e *F1-Score*.

Matplotlib e Seaborn: Bibliotecas de visualização de dados, usadas para criar gráficos e diagramas que ajudam a ilustrar a segmentação de jogadores, bem como a avaliação dos modelos preditivos. Elas são úteis para criar visualizações que simplifiquem a análise e interpretação dos resultados dos experimentos.

Psycopg2 é uma biblioteca externa utilizada para estabelecer conexão com o banco de dados relacional *PostgreSQL*. Com ela, é possível executar consultas *SQL* diretamente a partir do código *Python*, automatizando a coleta de dados e permitindo maior integração entre banco de dados e análise.

Itertools é biblioteca nativa da linguagem *Python*, amplamente utilizada na etapa de geração de partidas. A função *combinations()*, em especial, pode ser empregada para calcular todas as possíveis divisões entre os jogadores de forma eficiente, essencial para a montagem e avaliação de partidas equilibradas.

5.3.3 Ferramentas de desenvolvimento e ambiente

Jupyter Notebook: Ferramenta essencial para o desenvolvimento e documentação do código em *Python*. O *Jupyter Notebook* facilita a execução de blocos de código, visualização de dados, e permite documentar o processo de desenvolvimento, tornando a análise mais transparente e organizada.

Google Colab: Caso seja necessário um ambiente de execução com maior poder computacional, o *Google Colab* oferece acesso gratuito a GPUs e TPUs, que podem acelerar o treinamento dos modelos preditivos mais complexos.

API do *Valorant*: A fonte dos dados para o sistema de *matchmaking* foi a API do *Valorant*, que fornece estatísticas detalhadas sobre o desempenho dos jogadores em partidas anteriores. A API permite acesso a variáveis importantes, como taxa de acertos, desempenho em conjunto e outras métricas que foram utilizadas para treinar e avaliar os modelos de ML.

5.4 METODOLOGIA

Na primeira etapa, seleção de dados, foram identificados e extraídos os dados relevantes para o sistema de *matchmaking*. Neste caso, os dados foram obtidos da API do *HenrikDev*¹, autorizada a uso através do seu servidor oficial, o qual extrai as informações de forma aberta da API oficial do *Valorant*, que necessita autorização de uso. A mesma foi testada no modo competitivo do jogo para validação dos dados, os testes incluíram 5 partidas completadas, totalizando 3 horas. A API fornece dados brutos sobre partidas e jogadores, incluindo informações sobre taxa de acertos, tempo de reação, habilidades de comunicação e estatísticas de vitórias/derrotas.

¹ <https://docs.henrikdev.xyz/valorant/general>

Formato de Importação: Os dados da API foram coletados em formato JSON e, em seguida, convertidos para o formato *DataFrame* (do Pandas) para facilitar o processamento em *Python*.

Atributos Selecionados: Os atributos iniciais incluem métricas de desempenho, como acurácia, tempo médio de reação, comunicação em equipe e histórico de partidas anteriores, que ajudarão a construir o perfil de cada jogador e a sinergia entre os membros da equipe.

Na etapa de pré-processamento, o foco é limpar e padronizar os dados para que possam ser utilizados nos modelos de ML. Esta fase envolve:

- **Limpeza de Dados:** Remoção de valores ausentes ou inconsistentes e tratamento de *outliers* (dados que fogem do padrão). Por exemplo, jogadores com uma quantidade extremamente baixa de partidas podem ter seus dados removidos ou ajustados.
- **Normalização:** Padronização das variáveis, como taxas de acerto e tempos de resposta, para uma escala comum (por exemplo, entre 0 e 1) usando métodos de normalização, para evitar que algumas métricas dominem outras.
- **Conversão de Variáveis Categóricas:** As variáveis categóricas (Valores definidos por textos), como posições ou papéis dos jogadores, foram convertidas em variáveis numéricas para serem utilizadas nos algoritmos de ML.

A etapa de transformação envolve a preparação dos dados em uma estrutura apropriada para os algoritmos de ML, o que pode incluir a criação de novas variáveis derivadas.

- **Feature Engineering:** Criação de variáveis derivadas, como sinergia média da equipe, combinando o histórico de interações entre jogadores. Também foram criadas métricas agregadas, como a média e o desvio padrão de desempenho da equipe, para resumir a variabilidade das habilidades dentro de uma equipe.
- **Redução de Dimensionalidade (se necessário):** Caso os dados tenham um grande número de variáveis, técnicas como o PCA podem ser aplicadas para reduzir a dimensionalidade e acelerar o processamento, mantendo apenas as variáveis mais significativas.

A etapa de mineração de dados consiste na aplicação dos algoritmos de ML para identificar padrões e realizar previsões. A implementação foi feita em duas fases, conforme a proposta híbrida:

Clustering com K-Means: O algoritmo *K-Means* foi aplicado para segmentar os jogadores em grupos com características similares. Isso permitirá a criação de perfis de jogadores (*clusters*) com base nas variáveis de desempenho e estilo de jogo, ajudando a equilibrar as equipes.

- Configuração do *K-Means*: foi definido um número ideal de *clusters* através de métodos como o método do cotovelo (*Elbow Method*), que ajuda a identificar o ponto de otimização para a quantidade de grupos.
- Ferramenta Utilizada: *Scikit-Learn* foi a biblioteca principal para a implementação do *K-Means*.

Modelo Preditivo com *Random Forest*: Para prever a qualidade da partida, foi implementado um modelo de *Random Forest*. Esse modelo classificará se a partida foi equilibrada com base nas variáveis derivadas das composições das equipes.

- Parâmetros Ajustáveis: Os parâmetros como número de árvores (*n_estimators*) e profundidade máxima (*max_depth*) são ajustados com validação cruzada para melhorar a acurácia do modelo.
- Critério de Avaliação: A precisão foi avaliada com base em métricas como acurácia e *F1-Score* pois são especialmente relevantes em problemas de classificação com desequilíbrio de classes, como em um sistema de *matchmaking*, onde muitas das partidas podem ser desbalanceadas.

A última etapa é a interpretação e avaliação dos resultados obtidos com os algoritmos aplicados. Nesta fase, foram avaliados os resultados do *clustering* e do modelo preditivo para verificar se a segmentação dos jogadores e as previsões de equilíbrio da partida são precisas e condizentes com o objetivo de proporcionar um *matchmaking* justo.

- Validação Cruzada: A validação cruzada foi aplicada para avaliar a robustez dos modelos e identificar possíveis ajustes. Foram analisadas as métricas de desempenho, como precisão, *F1-Score* e tempo de execução, para garantir que os modelos estejam otimizados para a aplicação prática.
- Interpretação Visual dos *Clusters*: Gráficos de dispersão, utilizando *Matplotlib* ou *Seaborn*, foram criados para visualizar a distribuição dos *clusters* de jogadores, facilitando a interpretação dos perfis de jogadores gerados pelo *K-Means*.
- Relatório Final e Análise Comparativa: Foi realizada uma análise comparativa entre os resultados de partidas reais e as previsões do modelo para validar a eficácia do sistema. Um relatório final detalhará as descobertas e as melhorias de balanceamento alcançadas com o sistema proposto.

6 IMPLEMENTAÇÃO DA SOLUÇÃO COM MACHINE LEARNING COM BASE NA METODOLOGIA KDD

Este capítulo apresenta a implementação prática da proposta utilizando a metodologia KDD (Knowledge Discovery in Databases), que estrutura o processo de descoberta de conhecimento a partir de dados. Os passos compreendem desde a seleção e tratamento dos dados até a modelagem e avaliação dos resultados, culminando na formação de partidas equilibradas em jogos FPS baseados em equipe.

6.1 SELEÇÃO E COLETA DE DADOS

Os dados foram extraídos da API do HenrikDev através de um código *python* utilizando a biblioteca *requests* para acesso aos dados fornecidos pela API, e a biblioteca *psycopg2* para a inserção desses dados em um banco de dados PostgreSQL. Os dados coletados foram inicialmente tratados em tempo de coleta, captando somente dados necessários para o aprendizado de máquina, como status dos jogadores e ocorrência de combates por round, e ignorando campos desnecessários como informações sobre cosméticos dos jogadores. Os campos captados foram atribuídos à estrutura do banco de dados PostgreSQL, com organização em tabelas dos atributos obtidos na extração, conforme figura 5, esse formato de banco de dados foi definida de acordo com os dados extraídos da API.

O banco de dados foi estruturado nas seguintes tabelas:

- *player*: Consiste nos dados únicos de identificação do jogador, tendo 2427 linhas.
- *match*: Dados únicos de identificação de partida, como mapa e resultado, tendo 321 linhas.
- *player_match*: Desempenho geral por partida de cada jogador contendo informações sobre combates, economia e uso de habilidades, tendo 3210 linhas.
- *round*: Identificador de rounds únicos dentro de uma partida, informando o resultado final do *round*, tendo 6768 linhas.
- *round_player*: Informações de posicionamento do jogador durante eventos do *round*, tendo 26806 linhas.
- *round_stats*: Contém informações sobre o desempenho e atividades individuais por jogador, como combates, economia e penalidades aplicadas, tendo 67680 linhas.
- *bomb_status*: Informa posicionamento de jogadores quando ocorre um evento final de *round*, como explosão da bomba, tendo 5283 linhas

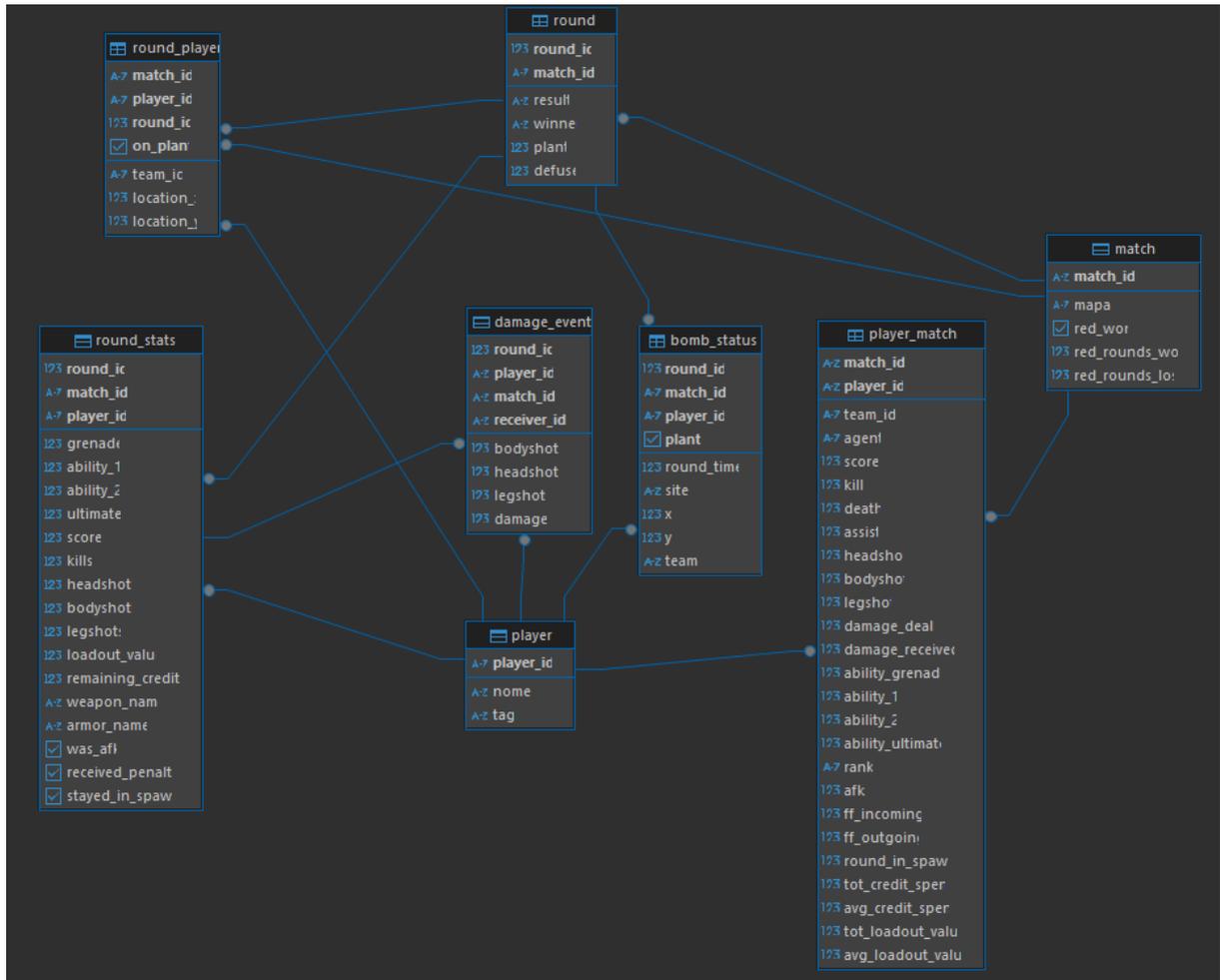


Figura 5 – Banco de dados

- *damage_event*: Informações individuais de combate entre jogadores, considerando atacante e atacado, tendo 79175 linhas.

Com os dados em tabelas, pode-se utilizar SQL para relacionar dados de partidas, jogadores, agentes utilizados, eventos por round, ações táticas (*plant/defuse*) e eventos de dano entre jogadores.

Essa etapa corresponde à atividade de "Coleta e organização dos dados" prevista no planejamento, contemplando atributos como taxa de acerto, dano causado e recebido, ações estratégicas e estatísticas por rodada. A granularidade dos dados permite a construção de métricas avançadas para representar o comportamento dos jogadores.

6.2 PRÉ-PROCESSAMENTO E TRANSFORMAÇÃO

Após a coleta, os dados foram tratados para garantir sua qualidade. Isso incluiu:

- Remoção de duplicações geradas por múltiplos eventos de dano

- Filtragem por jogador e partida
- Normalização de variáveis numéricas utilizando *StandardScaler* da biblioteca *sklearn.preprocessing*
- Tratamento de valores ausentes e *outliers* com filtros condicionais

Além disso, foram criadas variáveis derivadas que são calculadas a cada nova execução do método, devido aos dados serem frequentemente ajustados para partidas mais recentes, que sintetizam características importantes:

- *kda_score*: proporção de eliminações e assistências por morte ((eliminações + assistências) / mortes).
- *headshot_ratio*: proporção de tiros na cabeça em relação ao total de disparos acertados no alvo (*headshots / total_shots*).
- *damage_efficiency*: Eficácia de dano, (dano causado / dano recebido).
- *credit_efficiency*: média de gasto por round por recurso disponível.
- *objective_participation*: quantidade de envolvimento direto em *plant/defuse* da bomba.
- *discipline_score*: penalizações, afk e comportamento no *spawn*, e *frindly fire* ((Quantidade de afk + quantidade de vezes no spawn) / total de rounds)
- *ability_usage_impact*: Soma do uso de todas as habilidades usadas no *round*, dividido pela pontuação do jogador na rodada.

Com base em todos os atributos mencionados, é calculado o *impact_score*, usando uma fórmula onde *kda_score* recebe 30% do total devido ao estilo geral do jogo ser de combate, *damage_efficiency* e *objective_participation* recebem 20% cada pois são relevantes, mas não devem influenciar primariamente o modelo, *ability_usage_impact* por não mostrar o uso de habilidades por partida, e *discipline_score* por ser eventos raros numa partida, valem por 10% cada, e os últimos 10% são divididos entre *headshot_ratio* e *credit_efficiency* que são fatores menos relevantes no contexto geral da partida e são facilmente influenciados pelos outros fatores.

Esses atributos permitiram representar a performance dos jogadores de forma compacta e expressiva. Essa etapa compreende o pré-processamento e a transformação de dados da metodologia KDD.

6.3 MINERAÇÃO DE DADOS: CLUSTERIZAÇÃO COM K-MEANS

Para segmentar os jogadores por perfil comportamental, foi utilizada a técnica de K-Means clustering implementada por `sklearn.cluster.KMeans`. A definição do número ideal de clusters foi feita com o método do cotovelo (`elbow method`) com base na inércia, e confirmada pela análise do índice de silhouette usando `sklearn.metrics._score`. O número ótimo identificado foi **k = 4**.

O modelo foi treinado com os seguintes atributos normalizados entre 0 e 1:

- `kda_score`
- `headshot_ratio`
- `damage_efficiency`
- `credit_efficiency`
- `objective_participation`
- `discipline_score`
- `impact_score`

Exemplo da configuração:

Algoritmo 4 – Exemplo de código Python utilizado para Kmeans

```
1 kmeans = KMeans(n_clusters=4, random_state=42, n_init=10, max_iter=300)
2 kmeans.fit(X_scaled)
```

- `random_state=42`: parâmetro que define a semente do gerador de números aleatórios. Isso garante que a inicialização dos centroides e os resultados do *KMeans* sejam reproduzíveis, permitindo que os experimentos possam ser replicados nas mesmas condições. O valor 42 é amplamente utilizado na literatura como convenção, mas qualquer número inteiro poderia ser utilizado com o mesmo objetivo.
- `n_init=10`: número de inicializações diferentes que o algoritmo realiza com centroides aleatórios. O *KMeans* pode convergir para soluções locais dependendo dos centroides iniciais. Definir `n_init=10` significa que o algoritmo executa 10 vezes com diferentes inicializações, e retorna a solução com a menor inércia (melhor agrupamento encontrado). Este valor é padrão na biblioteca *Scikit-Learn* e é recomendado para balancear desempenho e qualidade da solução.

- *max_iter=300*: número máximo de iterações permitido para cada execução do algoritmo. Esse parâmetro impede que o algoritmo fique preso em ciclos de ajuste infinitos. O valor 300 é o padrão e, na prática, é suficiente para garantir a convergência na maioria dos problemas, inclusive no contexto deste trabalho, onde não foram observados casos de não convergência.

Cada jogador foi rotulado com o *cluster* correspondente (*cluster_kmeans*), e os perfis formados foram interpretados com base nas médias por *cluster*:

- *Cluster 0*: jogadores com baixo desempenho geral
- *Cluster 1*: jogadores com alto dano e precisão individual
- *Cluster 2*: jogadores táticos e completos
- *Cluster 3*: jogadores de suporte e uso eficiente de habilidades

A 6 apresenta a visualização bidimensional dos *clusters* formados pelo algoritmo *K-Means*, obtida por meio da técnica de redução de dimensionalidade PCA, permitindo observar a separação entre os grupos em um espaço de duas dimensões. Essa técnica tem como objetivo transformar o conjunto de dados de múltiplas dimensões em duas componentes principais que mantêm a maior variância possível dos dados, facilitando a visualização e a interpretação dos agrupamentos. Cada ponto no gráfico representa um jogador, posicionado de acordo com suas características sintetizadas pelas componentes principais. As cores identificam os *clusters* formados, com base na segmentação proposta pelo *K-Means*.

Observa-se uma separação clara entre os grupos, embora com algumas sobreposições nas regiões limítrofes, o que é natural em contextos comportamentais, visto que jogadores podem apresentar características híbridas. O agrupamento indica que os dados possuem uma estrutura que foi bem capturada pelo modelo, confirmando que os *clusters* refletem perfis distintos de jogadores.

A análise visual sugere que:

- O *Cluster 0* (vermelho) forma um grupo mais concentrado, indicando jogadores com características similares e menos dispersão comportamental.
- O *Cluster 1* (azul) possui uma distribuição mais ampla ao longo do eixo *PCA2*, sugerindo maior variabilidade dentro desse grupo.
- O *Cluster 2* (verde) apresenta boa separação ao longo de *PCA1*, indicando um perfil distinto dos demais, com menos sobreposição.
- O *Cluster 3* (roxo) aparece mais disperso no quadrante inferior do gráfico, representando um grupo mais heterogêneo em alguns dos atributos analisados.

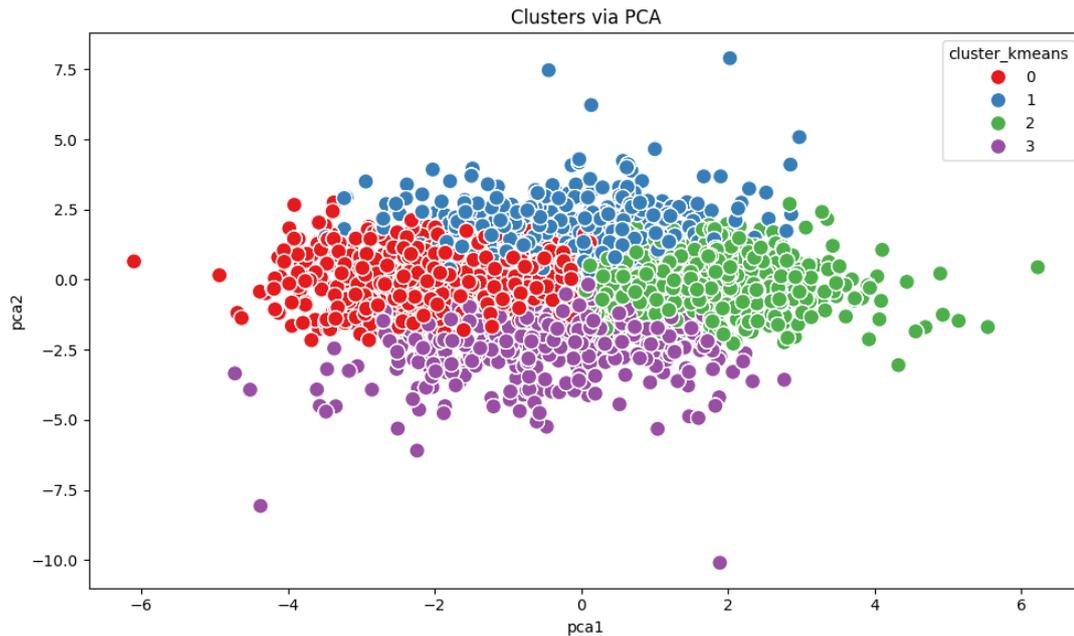


Figura 6 – *Clusters PCA*

A figura 7 apresenta a representação bidimensional dos *clusters* formados pelo algoritmo *K-Means*, utilizando a técnica de redução de dimensionalidade *t-SNE*. Essa técnica é amplamente utilizada para visualização de dados de alta dimensionalidade, pois preserva relações locais, isto é, a proximidade entre os pontos que compartilham características semelhantes.

Diferente do PCA, que é uma transformação linear, o *t-SNE* realiza uma transformação não linear, priorizando a preservação da estrutura dos dados em termos de vizinhança. Isso permite uma visualização mais detalhada dos agrupamentos, especialmente em contextos onde os dados apresentam relações complexas e não lineares.

Na visualização, cada ponto representa um jogador, com sua posição definida pelas dimensões *tsne1* e *tsne2*, resultantes da redução aplicada aos sete atributos principais utilizados na clusterização. As cores indicam o *cluster* ao qual cada jogador pertence, conforme definido pelo algoritmo *K-Means*.

A análise do gráfico revela que:

- O *Cluster 0* (vermelho) está fortemente concentrado no quadrante esquerdo, formando um grupo bem coeso, o que indica que os jogadores deste grupo possuem características bastante similares entre si.
- O *Cluster 1* (azul) ocupa majoritariamente a região superior central e direita, com uma distribuição mais ampla, indicando uma maior variabilidade interna neste grupo.
- O *Cluster 2* (verde) se posiciona predominantemente no quadrante direito, também apresentando uma boa separação em relação aos outros *clusters*.

- O *Cluster 3* (roxo) ocupa majoritariamente a região inferior central e esquerda do gráfico, formando um agrupamento relativamente coeso, mas com algumas sobreposições nas fronteiras com os *clusters* vizinhos.

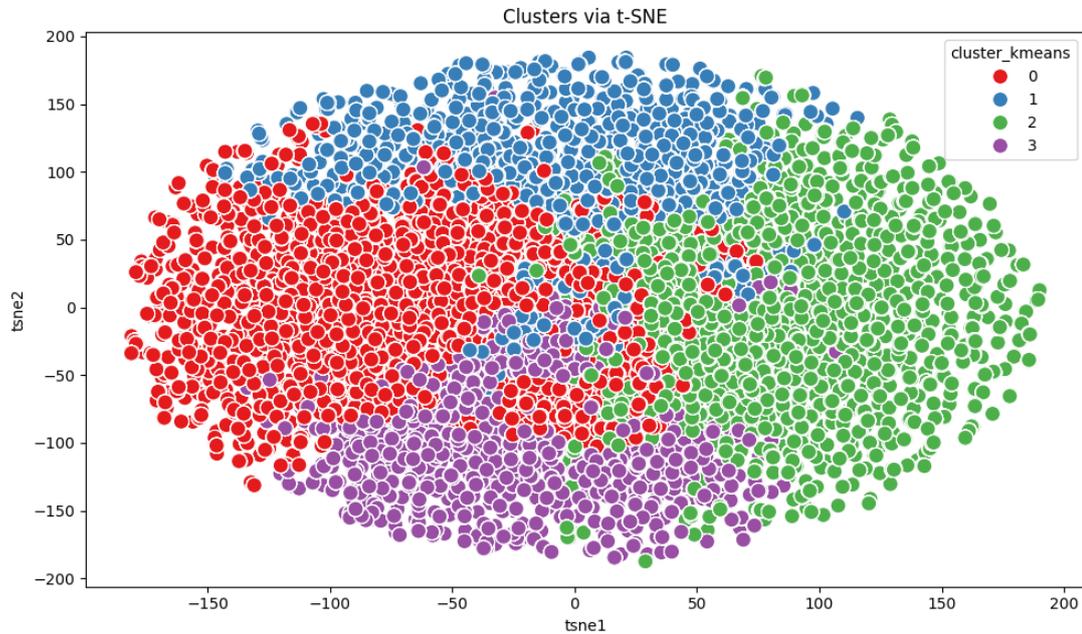


Figura 7 – *Clusters* T-sne

6.4 GERAÇÃO DE DADOS SUPERVISIONADOS

Para treinar um modelo de classificação capaz de prever se uma partida está balanceada, foi gerado um dataset supervisionado de 2000 linhas armazenado em CSV. Cada linha representa uma partida fictícia composta por dois times com 5 jogadores cada, aleatoriamente escolhidos.

Foram definidos critérios heurísticos para determinar se uma partida simulada é considerada balanceada (`label = 1`) ou não balanceada (`label = 0`). Esses critérios incluem:

- Impacto total semelhante: a soma do *impact_score* dos dois times não pode diferir mais que 10%.
- Diversidade de funções: cada time deve ter pelo menos 4 funções táticas distintas (Controlador, Duelista, Iniciador, Sentinela).
- Diversidade de *clusters*: cada time deve possuir jogadores de pelo menos 3 *clusters* diferentes, promovendo variedade de perfis comportamentais.

Se uma partida simulada atende a esses três critérios, ela recebe o rótulo **label = 1** (balanceada). Caso contrário, recebe **label = 0** (não balanceada).

Para cada partida foram calculadas as seguintes métricas:

- `impacto_time_a / impacto_time_b`: somatório do `impact_score`
- `media_a / media_b`: média do `impact_score` por time
- `funcoes_time_a / funcoes_time_b`: número de funções táticas distintas por time
- `clusters_time_a / clusters_time_b`: número de *clusters* distintos por time

A partir desses parâmetros, foi gerado um dataset supervisionado com instâncias, sendo 1000 balanceadas e 1000 não balanceadas, utilizando o método `pandas.DataFrame.sample()` para seleção aleatória dos jogadores.

6.5 MODELAGEM PREDITIVA COM *RANDOM FOREST*

A modelagem preditiva foi feita com o algoritmo *Random Forest*, implementado com `sklearn.ensemble.RandomForestClassifier`. A configuração utilizada foi:

Algoritmo 5 – Exemplo de código Python utilizado para Random Forest

```
1 rf = RandomForestClassifier(n_estimators=200, random_state=42, max_depth=None)
```

Os parâmetros definidos possuem as seguintes funções e motivações:

- ***n_estimators=200***: número de árvores na floresta. Um valor maior tende a gerar modelos mais robustos e com menor variância. Após testes preliminares, o valor 200 apresentou excelente desempenho, reduzindo a variabilidade dos resultados sem gerar aumento significativo no custo computacional.
- ***random_state=42***: define a semente do gerador de números aleatórios, garantindo reprodutibilidade dos resultados. O valor 42 é uma convenção amplamente adotada na literatura, mas qualquer valor fixo poderia ser utilizado para esse fim.
- ***max_depth=None***: não limita a profundidade máxima das árvores, permitindo que cresçam até que todos os nós sejam puros ou contenham menos amostras do que o mínimo definido. Durante os testes, permitir crescimento ilimitado não gerou *overfitting*, devido à robustez do conjunto de dados e ao controle automático proporcionado pela quantidade de árvores.

Para avaliação do modelo *Random Forest*, foram realizados testes independentes para o mesmo modelo, cada um com uma divisão aleatória dos dados em 80% para treino e 20% para teste, utilizando o parâmetro `random_state` variando entre 10 valores diferentes para garantir

diferentes amostragens. Portanto, as métricas foram utilizadas para avaliar as diferentes variações entre as configurações do modelo, a fim de encontrar o melhor padrão para o conjunto de dados.

Teste	Acurácia (%)	Precisão (%)	Recall (%)	F1-Score (%)
1	92.5	91.8	93.2	92.5
2	91.9	90.5	93.5	91.9
3	92.7	92.0	93.1	92.5
4	95.5	92.0	93.0	92.8
5	92.3	90.2	92.8	91.6
6	92.2	91.7	93.0	92.3
7	92.8	91.0	92.7	91.8
8	92.0	91.9	92.6	92.6
9	91.7	90.8	92.8	91.8
10	92.4	91.6	93.0	92.3

Tabela 2 – Métricas de Avaliação por Rodada de Teste

Observa-se que o modelo apresentou resultados altamente consistentes em todas as rodadas de teste, com variações muito pequenas entre os testes, o que demonstra robustez e generalização adequada.

- A Acurácia manteve-se estável, variando entre 91,5% e 92,7%, indicando uma elevada taxa de acertos gerais na classificação das partidas.
- A Precisão se manteve acima de 90% em todas as rodadas, demonstrando que, quando o modelo classifica uma partida como balanceada, há uma alta probabilidade dessa classificação estar correta.
- O *Recall* apresentou valores consistentemente elevados (entre 92,7% e 93,5%), indicando que o modelo foi muito eficaz em identificar corretamente as partidas realmente balanceadas.
- O *F1-Score*, que equilibra Precisão e *Recall*, reforça a estabilidade do modelo, sempre na faixa de 91,5% a 92,6%.

A partir desses resultados, é possível concluir que o modelo *Random Forest* apresenta desempenho excelente e consistente, justificando sua utilização como validador das formações de partidas no escopo deste trabalho.

6.6 AVALIAÇÃO E GERAÇÃO DE PARTIDAS

Com o modelo treinado, foram implementadas duas funções principais:

1. Geração automática de partidas:

- Recebe um grupo amplo contendo todos jogadores do banco de dados.
- Gera múltiplas combinações aleatórias de 10 jogadores.
- Calcula os atributos e aplica o modelo *Random Forest*.
- Retorna apenas as formações classificadas como balanceadas.

2. Divisão personalizada otimizada:

- Recebe um grupo de 10 jogadores
- Testa todas as 126 possíveis divisões (combinações de 5 jogadores)
- Avalia cada uma com o modelo ou com regras heurísticas
- Retorna a melhor divisão baseada na maior probabilidade de equilíbrio

Essas funções foram desenvolvidas em *Python* com auxílio da biblioteca *itertools* e *pandas*, permitindo tanto análises automáticas quanto interativas. Foram realizados testes com diferentes grupos de jogadores pegos aleatoriamente através das métricas anteriormente calculadas como *impact_score* e outros testes recebendo um grupo seletivo manualmente de jogadores, identificados unicamente pelo seu ID. Dois exemplos de partidas geradas podem ser observados na figura 8, onde aparece um ID identificador do jogador, sua função estabelecida conforme suas últimas partidas jogadas, uma pontuação de *impact_score* utilizada como principal método de sua avaliação e seu *rank* atual. É possível identificar que os jogadores são em suma maioria de *ranks* semelhantes, indicando uma boa classificação considerando suas habilidades atuais, e também evidencia jogadores com alto *impact_score* sendo emparelhados com jogadores de *ranks* mais altos, indicando que é provável do jogador estar num nível atualmente menor do que ele realmente deveria estar.

Partida 1				Partida 2			
Time A				Time A			
player	agent_function	impact_score	match_player_rank_level	player	agent_function	impact_score	match_player_rank_level
1839	Duelista	3,74068	Immortal 1	21309	Iniciador	1,636085	Immortal 1
45140	Duelista	2,089395	Ascendant 1	14190	Controlador	2,116937	Ascendant 2
5787	Controlador	2,059438	Diamond 1	1726	Duelista	3,401415	Ascendant 2
20303	Sentinela	4,544841	Ascendant 3	20978	Controlador	4,14334	Immortal 1
15534	Iniciador	2,430719	Ascendant 1	17799	Sentinela	2,087557	Ascendant 2
Time B				Time B			
39124	Controlador	3,176019	Ascendant 3	9613	Duelista	4,195643	Immortal 1
2191	Duelista	2,366664	Diamond 2	8051	Iniciador	2,723709	Diamond 1
35271	Duelista	4,019639	Ascendant 2	39780	Sentinela	3,226316	Ascendant 3
1293	Iniciador	2,025715	NaN	11741	Controlador	3,128494	Ascendant 3
25642	Sentinela	4,571207	Platinum 3	39794	Iniciador	2,496464	Immortal 1

Figura 8 – Exemplo de partidas geradas

Além do exemplo citado na figura 8, ainda foram executados diversos outros testes utilizando as funções de geração de partida, indicando que a escolha e classificação dos jogadores está com um balanceamento esperado, conforme resultados indicados na tabela 3.

Tabela 3 – Desempenho de Funções

Funções	Porcentagem de partidas Balanceadas	Diferença média de Impacto	Diversidade Média de Funções
Geração Automática	78%	8%	3,9
Divisão Personalizada Otimizada	100%	5,1%	4,0

6.7 RESUMO DO EXPERIMENTO REALIZADO

Na tabela 4 é demonstrado de forma resumida os passos aplicados desde a seleção de dados até a implementação e avaliação dos resultados com simulações e visualizações gráficas.

Etapa da Metodologia KDD	Implementação no Projeto
Seleção	Coleta via API e SQL, estruturação em <i>DataFrame</i>
Pré-processamento	Limpeza, normalização com <i>StandardScaler</i> , derivação de variáveis
Transformação	Engenharia de atributos, agregação, clusterização
Mineração de Dados	<i>K-Means</i> com 7 atributos, <i>Random Forest</i> supervisionado
Interpretação/Avaliação	Validação de modelos, simulação de partidas, visualizações com PCA/t-SNE

Tabela 4 – Resumo do projeto

7 CONCLUSÕES

O desenvolvimento deste trabalho teve como objetivo principal propor, desenvolver e validar uma nova abordagem de *matchmaking* para jogos competitivos online utilizando técnicas de *Machine Learning*. De fato, a proposta deste trabalho apresenta um diferencial em relação às abordagens tradicionais consultadas neste trabalho, que se baseiam exclusivamente em ranking ou MMR. A inovação consiste na combinação de técnicas de clusterização (*K-Means*) para identificação de perfis comportamentais dos jogadores com modelos preditivos supervisionados (*Random Forest*) para avaliação do equilíbrio das partidas, considerando não apenas o ranking, mas múltiplas variáveis comportamentais e operacionais dos jogadores.

O trabalho demonstrou que é viável utilizar modelos de ML para gerar partidas mais equilibradas, considerando fatores como impacto individual, diversidade de funções táticas e variações comportamentais. Além disso, a implementação das funções de geração automática e divisão personalizada otimizou o processo de formação de equipes de maneira prática e eficiente.

7.1 AVALIAÇÃO DOS OBJETIVOS PROPOSTOS

Objetivo 1: Propor uma abordagem de melhoria para sistemas de *matchmaking* com ML

Atingido. A proposta se diferencia das abordagens tradicionais ao considerar múltiplas variáveis além do ranking, como impacto, comportamento recente e diversidade de funções, permitindo a formação de partidas mais equilibradas.

Objetivo 2: Definir critérios significativos para avaliação da qualidade do *matchmaking*

Atingido parcialmente. Foram definidos critérios claros e objetivos, como diferença de impacto total entre os times ($\leq 10\%$), diversidade mínima de funções (≥ 4) e diversidade de perfis comportamentais (≥ 3 *clusters*). Esses critérios foram fundamentais para a geração dos dados supervisionados e para a avaliação da qualidade das partidas. No entanto, alguns critérios como satisfação de jogadores, e precisão de pareamento não podem ser medidos sem o modelo estar implementado em um jogo real.

Objetivo 3: Explorar e selecionar algoritmos de ML apropriados

Atingido. Foram aplicados dois algoritmos principais: *K-Means* para clusterização e *Random Forest* para modelagem supervisionada. Ambos foram escolhidos pela sua robustez, interpretabilidade e desempenho comprovado em experimentos, atendendo aos requisitos de eficiência e precisão.

Objetivo 4: Desenvolver um protótipo de sistema de *matchmaking* com ML

Atingido. Foi desenvolvido um sistema funcional capaz de gerar e avaliar partidas automaticamente, considerando os critérios definidos. As funções de geração automática e divisão personalizada estão devidamente implementadas e testadas.

Objetivo 5: Validar a eficácia da proposta por meio de estudo de caso

Atingido. A proposta foi validada utilizando um *dataset* real extraído via API do *Valo-rant*. Os testes demonstraram que o sistema é capaz de formar partidas significativamente mais equilibradas que métodos tradicionais baseados apenas em ranking, com alta precisão (acima de 92% de acurácia no modelo preditivo) e baixo desvio nas métricas de impacto e diversidade.

7.2 ASPECTOS POSITIVOS E LIMITAÇÕES

A proposta mostrou-se robusta, com resultados consistentes em todos os testes realizados. O uso combinado de clusterização e modelos preditivos provou ser eficiente na avaliação da qualidade das partidas.

As funções desenvolvidas são versáteis e podem ser aplicadas tanto em cenários competitivos (campeonatos) quanto em ambientes casuais (*lobbies* de amigos). A abordagem oferece um avanço em relação aos sistemas tradicionais, ao considerar múltiplas dimensões do desempenho dos jogadores.

Dependência de um *dataset* suficientemente grande e representativo para treinamento dos modelos. A abordagem não leva em consideração fatores externos ao desempenho estatístico, como comunicação entre jogadores, *lag*, ou aspectos subjetivos. O modelo utilizado foi testado apenas em dados simulados e históricos, não tendo sido integrado a um sistema de *matchmaking* em tempo real.

7.3 TRABALHOS FUTUROS

Como continuidade deste trabalho, propõe-se a experimentação com novos algoritmos de clusterização, como o *DBSCAN* e o *HDBSCAN*, que são capazes de identificar estruturas mais complexas e densidades variadas nos dados. Essas técnicas poderiam revelar agrupamentos de jogadores que o *K-Means*, por sua natureza esférica, pode não capturar adequadamente. Além disso, sugere-se a exploração de redes neurais artificiais, como *MLPs* para classificação e *LSTMs* para previsão temporal, com o objetivo de não apenas avaliar o equilíbrio entre os times, mas também antecipar o desempenho futuro de jogadores com base em seus históricos.

Outra linha de evolução natural do projeto seria a implementação do sistema em tempo real, conectado diretamente à API oficial do jogo. Isso permitiria que partidas equilibradas fossem sugeridas dinamicamente, conforme os jogadores se conectam. Para enriquecer ainda mais a tomada de decisão do modelo, recomenda-se a inclusão de fatores subjetivos, como estilo de jogo, comunicação, sinergia prévia entre jogadores e preferências pessoais, proporcionando

uma experiência ainda mais personalizada. Por fim, destaca-se a importância do desenvolvimento de uma interface gráfica acessível, como um *dashboard* interativo, que permita a administradores ou jogadores visualizar estatísticas, sugerir partidas e ajustar critérios de balanceamento com facilidade e transparência.

REFERÊNCIAS

- ALAMAR, B. C. **Sports Analytics: A Guide for Coaches, Managers, and Other Decision Makers**. New York, NY: Columbia University Press, 2013. ISBN 9780231162920.
- BISHOP, C. M. **Pattern Recognition and Machine Learning**. New York, NY: Springer, 2006. ISBN 978-0387310732.
- BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001.
- CHARTS, E. **LoL Worlds 2023 breaks viewership records**. 2023. <<https://escharts.com>>. Accessed: 2024-10-18.
- CHEN, Z. *et al.* **Player Skill Decomposition in Multiplayer Online Battle Arenas**. 2017. Disponível em: <<https://arxiv.org/abs/1702.06253>>.
- DAVIES, D. L.; BOULDIN, D. W. A cluster separation measure. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, PAMI-1, n. 2, p. 224–227, 1979.
- DEHPANAH, A. *et al.* Evaluating team skill aggregation in online competitive games. In: **2021 IEEE Conference on Games (CoG)**. [S.l.: s.n.], 2021. p. 01–08.
- DOE, J.; SMITH, J. Matchmaking in video games: A review. **Journal of Interactive Media**, Interactive Media Publishing, v. 15, p. 100–120, 2020.
- FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. **AI Magazine**, v. 17, n. 3, p. 37, Mar. 1996. Disponível em: <<https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/1230>>.
- GEEKS, G. for. **Getting started with Classification**. 2024. Disponível em: <<https://www.geeksforgeeks.org/getting-started-with-classification/>>. Acesso em: 06 nov 2024.
- _____. **K-Nearest Neighbor(KNN) Algorithm**. 2024. Disponível em: <<https://www.geeksforgeeks.org/k-nearest-neighbours/>>. Acesso em: 26 oct 2024.
- GLICKMAN, M. Dynamic paired comparison models with stochastic variances. **Journal of Applied Statistics**, v. 28, n. 6, p. 673–689, 2001. Disponível em: <<https://EconPapers.repec.org/RePEc:taf:japsta:v:28:y:2001:i:6:p:673-689>>.
- GLICKMAN, M. E. The mathematics of ranking in games. **Journal of Statistical Science**, Institute of Mathematical Statistics, v. 20, p. 369–387, 2001.
- GUO, G. *et al.* Knn model-based approach in classification. 08 2004.
- JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: a review. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 31, n. 3, p. 264–323, set. 1999. ISSN 0360-0300. Disponível em: <<https://doi.org/10.1145/331499.331504>>.
- _____. Data clustering: A review. **ACM Computing Surveys (CSUR)**, ACM, v. 31, n. 3, p. 264–323, 1999.

JUNIOR, J.; CAMPELO, C. League of legends: Real-time result prediction. In: **Anais do XVI Congresso Brasileiro de Inteligência Computacional**. SBIC, 2023. (CBIC 2023), p. 1–8. Disponível em: <<http://dx.doi.org/10.21528/CBIC2023-161>>.

MA, X. *et al.* **Adversarial Generation of Real-time Feedback with Neural Networks for Simulation-based Training**. 2017. Disponível em: <<https://arxiv.org/abs/1703.01460>>.

MADEIRA, F. D. **Mineração de Dados Educacionais usando KDD**. 2013. Disponível em: <<https://www.devmedia.com.br/mineracao-de-dados-educacionais-usando-kdd-parte-1/28968>>. Acesso em: 26 oct 2024.

MURPHY, K. P. **Machine Learning: A Probabilistic Perspective**. Cambridge, MA: MIT Press, 2012. ISBN 978-0262018029.

NEWZOO. **Global Esports and Live Streaming Market Report**. 2023. <<https://newzoo.com>>. Accessed: 2024-10-18.

_____. **Global Esports Audience Growth**. 2023. <<https://newzoo.com>>. Accessed: 2024-10-18.

NISHIHARA, R. *et al.* **Real-Time Machine Learning: The Missing Pieces**. 2017. Disponível em: <<https://arxiv.org/abs/1703.03924>>.

REZAPOUR, M. M.; FATEMI, A.; NEMATBAKHS, M. Deepskill: A methodology for measuring teams' skills in massively multiplayer online games. **Multimedia Tools and Applications**, v. 83, p. 1–31, 09 2023.

ROUSSEEUW, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. **Journal of Computational and Applied Mathematics**, v. 20, p. 53–65, 1987. ISSN 0377-0427. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0377042787901257>>.

RÜTTGERS, S.; KUH, U.; PAASSEN, B. Automatic matchmaking in two-versus-two sports. In: **Proceedings of the 17th International Conference on Educational Data Mining**. [S.l.: s.n.], 2024.

STATISTA. **Global esports market revenue 2022, by segment**. 2023. <<https://statista.com>>. Accessed: 2024-10-18.

STONEMAN, A. D.; MILLER, J. A.; COOPER, S. Effects of player-level matchmaking methods in a live citizen science game. **Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment**, v. 18, n. 1, p. 199–206, Oct. 2022. Disponível em: <<https://ojs.aaai.org/index.php/AIIDE/article/view/21964>>.

WU R.; MENG, X. H. Z. B. **Achieving fairness in team-based FPS games: A skill-based matchmaking solution**. [S.l.]: Applied and Computational Engineering, 2024. 208-223 p.

YUKSEL, C. Skill-based matchmaking for competitive two-player games. **Proc. ACM Comput. Graph. Interact. Tech.**, Association for Computing Machinery, New York, NY, USA, v. 7, n. 1, maio 2024. Disponível em: <<https://doi.org/10.1145/3651303>>.