

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Dissertação de Mestrado

Caracterização de Perfis de Comportamento de
Equipes em League of Legends

Fernando Felix do Nascimento Junior

Campina Grande, Paraíba, Brasil

Dezembro – 2017

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Caracterização de Perfis de Comportamento de Equipes em League of Legends

Fernando Felix do Nascimento Junior

Dissertação submetida à Coordenação do Curso de Pós-Graduação em
Ciência da Computação da Universidade Federal de Campina Grande -
Campus I como parte dos requisitos necessários para obtenção do grau
de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Mineração de Dados

Leandro Balby Marinho

(Orientador)

Campina Grande, Paraíba, Brasil

©Fernando Felix do Nascimento Junior, 15/12/2017

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

N244c	<p>Nascimento Junior, Fernando Felix do.</p> <p>Caracterização de perfis de comportamento de equipes em League of Legends / Fernando Felix do Nascimento Junior. ó Campina Grande, 2018. 62 f. : il. color.</p> <p>Dissertação (Mestrado em Ciência da Computação) ó Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2017.</p> <p>"Orientação: Prof. Dr. Leandro Balby Marinho".</p> <p>Referências.</p> <p>1. Mineração de Dados. 2. Análise de Jogos. 3. Jogos <i>Online</i>. 4. Jogos MOBA. 5. Desempenho de Equipes. I. Marinho, Leandro Balby. II. Título.</p> <p>CDU 004.65(043)</p>
-------	---

*Dedico essa dissertação a todas as pessoas que tiveram seus sonhos interrompidos.
Tenham foco, sigam em frente e não desistam dos seus sonhos.*

Agradecimentos

Agradeço à minha querida esposa, Taisa, e minha amável filha, Rebeca, por serem tão importantes na minha vida. Que por vezes ficaram bravas porque eu não queria parar para lhes dar atenção enquanto estudava. Obrigado por permanecerem comigo diante de tantas dificuldades de saúde que aconteceram comigo em paralelo ao mestrado. Essa vitória também é de vocês. Obrigado por tudo.

Agradeço ao meu pai. Por ser meu referencial como pessoa, pelo apoio incondicional e por sempre acreditar em mim.

Agradeço aos meus colegas de laboratório e de trabalho pelo companheirismo e amizade que tiveram comigo.

Agradeço ao Professor Leandro Balby Marinho pela excelente orientação e dedicação ao longo desses 2 anos e meio de mestrado. Obrigado pela paciência, confiança, conselhos e todo o conhecimento que me transferiu ao longo dos estudos.

Obrigado aos funcionários do PPGCC e a CAPES. Obrigado a todos que diretamente ou indiretamente contribuíram para o sucesso deste trabalho!

Resumo

Apesar da crescente popularidade dos esportes eletrônicos (eSports), ainda há uma escassez de trabalhos acadêmicos que exploram o comportamento de jogo das equipes. Compreender as características que ajudam discriminar entre equipes bem-sucedidas e malsucedidas poderia ajudar as equipes a melhorar suas estratégias, como determinar métricas de desempenho a serem alcançadas. Nesta dissertação, identificamos e caracterizamos padrões de comportamento de equipes com base nos dados de histórico de partidas de League of Legends, um *eSport* muito popular. Ao aplicar métodos de mineração de dados, como aprendizado da máquina e análise estatística, agrupamos o desempenho das equipes e investigamos para cada grupo como e em que medida essas características influenciam o sucesso e fracasso das equipes. Alguns grupos são mais propensos a terem equipes mais vencedoras do que outros, os resultados do nosso estudo ajudaram a descobrir as características que estão associadas a essa propensão e permitiram modelar métricas de desempenho de perfis de equipe bem-sucedidas e malsucedidas. Encontramos ao todo 7 perfis, que foram categorizados em quatro grandes níveis em termos de proporção de equipes vencedoras: Fraco, Mais ou Menos, Bom e Top.

Palavras-chave: Mineração de Dados, Agrupamento, Análise de Jogos, Jogos *Online*, Jogos MOBA, Desempenho da Equipes

Abstract

Despite the increasing popularity of electronic sports (eSports), there is still a scarcity of academic works exploring the playing behavior of teams. Understanding the features that help to discriminate between successful and unsuccessful teams would help teams improving their strategies, such as determine performance metrics to reach. In this dissertation, we identify and characterize team behavior patterns based on historical matches data from League of Legends, a very popular eSport. By applying methods from data mining, such as machine learning and statistical analysis, we clustered teams' performance and investigate for each cluster how and to what extent these features have an influence on teams' success and failure. Some clusters are more likely to have winning teams than others, the results of our study helped to discover the characteristics that are associated with this predisposition and allowed us to define performance metrics of successful and unsuccessful team profiles. At all, we found 7 profiles in which were categorized into four big levels in terms of winning team proportion: Weak, So-So, Good and Top.

Keywords: Data Mining, Clustering, Game Analytics, Online Games, MOBA Games, Team Performance

Conteúdo

1	Introdução	1
1.1	Objetivos	4
1.1.1	Objetivo Geral	4
1.1.2	Objetivo Específicos	4
1.2	Metodologia	4
1.3	Contribuições	6
1.4	Estrutura da Dissertação	6
2	Fundamentação Teórica	8
2.1	MOBA	8
2.1.1	Starcraft e o mapa Aeon of Strife	8
2.1.2	Warcraft III e o mapa Defense of the Ancients	9
2.1.3	League of Legends	10
2.1.4	Outros	11
2.2	Mineração de Dados	11
2.2.1	Dados Telemétricos em Jogos Digitais	13
2.2.2	Processo de Mineração de Dados	15
2.2.3	Preparação dos Dados	18
2.2.4	Principais Tarefas de Mineração de Dados	22
3	Trabalhos Relacionados	26
3.1	Predições de Partidas	26
3.2	Predições Diversas	27
3.3	Análise de <i>Outliers</i>	27

3.4	Análise de Toxicidade	27
3.5	Análise Social	28
3.6	Análises Diversas	28
3.7	Conclusão	29
4	Preparação dos Dados	30
4.1	Coleta de Dados	30
4.1.1	Estatísticas Básicas	31
4.2	Extração de Características	35
4.3	Limpeza de Dados	36
4.4	Remoção de Características com Baixa Variância	37
4.5	Remoção de <i>Outliers</i>	37
4.6	Transformação de Dados	39
4.6.1	Cálculo de métricas	39
4.6.2	Normalização de Dados	41
4.7	Análise de Redundância	41
5	Perfis de Comportamento	44
5.1	Agrupamento de Equipes	44
5.2	Caracterização de Perfis	45
5.2.1	Nível Fraco	51
5.2.2	Nível Mais ou Menos	51
5.2.3	Nível Bom	52
5.2.4	Nível Top	52
6	Conclusão	55
6.1	Trabalhos Futuros	56

Lista de Símbolos

API - *Application Programming Interface*

CRISP-DM - *Cross-Industry Standard Process for Data Mining*

IA - *Inteligência artificial*

LOL - *League of Legends*

MOBA - *Multiplayer Online Battle Arena*

PCA - *principal component analysis*

RTS - *Realtime Strategy*

SQE - *Soma dos Quadrados do Erro*

SQED - *Soma dos Quadrados do Erro Dentro dos Grupos*

SQEE - *Soma dos Quadrados do Erro Entre os Grupos*

SQET - *Soma dos Quadrados do Erro Total*

Lista de Figuras

1.1	Visão geral do mapa do jogo LoL. As linhas pontilhadas em azul e vermelho indicam as bases da equipe 1 e da equipe 2, respectivamente. Os círculos em azul e vermelho indicam os diferentes tipos de estruturas de defesa para cada equipe. Os quadrados em azul e vermelho indicam as lojas de item para cada equipe. As linhas pontilhadas em amarelo indicam as rotas principais que conectam as bases das equipes.	2
1.2	Metodologia usada para atingir o objetivo da dissertação.	5
2.1	Fases do processo CRISP-DM [5]: Compreensão do negócio (<i>Business understanding</i>), Compreensão dos dados (<i>Data understanding</i>), Preparação dos dados (<i>Data preparation</i>), Modelagem (<i>Modeling</i>), Avaliação (<i>Evaluation</i>), Implantação (<i>Deployment</i>)	16
4.1	Gráfico de <i>boxplot</i> para cada característica do conjunto de desempenho de equipes até o momento da etapa de remoção de <i>outliers</i>	40
4.2	Gráfico da correlação das características de desempenho das equipes. Cada célula ou valor no gráfico representa a correlação de um par de características do conjunto de dados X . A escala de cores indica a intensidade das correlações, quanto mais forte for uma cor, mais forte é a correlação das características. As cores em tom azul indicam correlações positivas. As cores em tom vermelho indicam correlações negativas. No gráfico, as correlações com $p\text{-value} > 0,05$ são consideradas insignificantes. Neste caso, os valores do coeficiente de correlação são indicados com um "X".	42

-
- 5.1 Curva de erro K-means. O eixo x indica o conjunto $K = \{1, \dots, 120\}$ de números possíveis de grupos. O eixo y indica os testes $SQEE(k)/SQET$ para $k \in K$ 46
- 5.2 Gráfico da proporção de equipes vencedoras (*win rate*) e equipes perdedoras (*loss rate*): Perfil 1 (quantidade de equipes vencedoras: 662, quantidade de equipes perdedoras: 5829), Perfil 2 (vencedoras: 5255, perdedoras: 4273), Perfil 3 (vencedoras: 6357, perdedoras: 1051), Perfil 4 (vencedoras: 502, perdedoras: 7676), Perfil 5 (vencedoras: 4776, perdedoras: 4233), Perfil 6 (vencedoras: 3650, perdedoras: 1785), Perfil 7 (vencedoras: 4485, perdedoras: 840). 47
- 5.3 Mapa de calor das relevâncias das métricas de desempenho (ou características) das equipes com base no cálculo do ganho de informação. O eixo x representa os perfis. O eixo y representa as características. Quanto mais forte for a cor de uma característica, mais intensa é a relevância dessa característica em relação as outras características para um determinado perfil. 49
- 5.4 Centroides dos perfis normalizados por métrica de desempenho das equipes. 50

Lista de Tabelas

4.1	Características resultantes da decomposição.	36
4.2	Características compostas.	36
4.3	Estatística descritiva das características do conjunto de desempenho de equipes.	38
5.1	Centroides das características (métricas de desempenho) para cada um dos 7 perfis encontrados. As últimas 5 colunas da tabela sumarizam as métricas de desempenho entre os perfis: valor mínimo (Min), média (Mean), variância (Var), desvio padrão (Sd), valor máximo (Max).	51
5.2	Pontuações de desempenho para o Perfil 1 e Perfil 4.	52
5.3	Pontuações de desempenho para o Perfil 2 e Perfil 5.	53
5.4	Pontuações de desempenho para o Perfil 6.	53
5.5	Pontuações de desempenho para o Perfil 2 e Perfil 5.	54

Capítulo 1

Introdução

A indústria de jogos digitais é um dos segmentos de entretenimento mais rentáveis do mundo atualmente, superando, por exemplo, a indústria do cinema e da música [1] [2] [3]. Conforme [1], o mercado de jogos digitais alcançou mais de 99,5 bilhões de dólares em 2016. Um fator importante para esse sucesso é a possibilidade de jogar *online* e se juntar com jogadores de todo o mundo.

Um segmento de jogo muito popular atualmente é o esporte eletrônico, mais conhecido como *eSport* [4]. Conforme [5], o mercado de *eSport* alcançou 493 milhões de dólares e uma audiência global de 191 milhões de entusiastas em 2016. O *eSport* mais popular e mais rentável do mundo hoje é o League of Legends (LoL) [6], desenvolvido pela Riot Games, com cerca de 67 milhões de jogadores ativos e um pico diário de mais de 7,5 milhões de jogadores *online* simultâneos [7].

LoL é um jogo de arena de batalha *online* para vários jogadores (*Multiplayer Online Battle Arena* - MOBA), um subgênero de jogos digitais de estratégia em tempo real (*Real-time Strategy* - RTS). Uma partida em um MOBA consiste em um cenário (mapa) em que duas equipes lutam entre si, a fim de destruir a base do oponente como objetivo principal, sem limite de tempo. Um mapa simples contém três estradas principais (rotas) que conectam a base de cada equipe (Figura 1.1). Em geral, uma equipe tem cinco jogadores e cada um seleciona e controla um personagem, conhecidos como campeões ou heróis, com atributos e habilidades distintas. Além disso, as equipes também contam com a assistência de estruturas de defesa e unidades controladas por Inteligência Artificial (IA), conhecidos como *minions* ou *creeps*, para vencer a partida. Ao longo de uma partida, os personagens ganham ouro -

que é usado para comprar itens que melhoraram seus atributos e habilidades - e pontos de experiência de várias maneiras, como matar unidades ou personagens e destruir estruturas da equipe inimiga [8].



Figura 1.1: Visão geral do mapa do jogo LoL. As linhas pontilhadas em azul e vermelho indicam as bases da equipe 1 e da equipe 2, respectivamente. Os círculos em azul e vermelho indicam os diferentes tipos de estruturas de defesa para cada equipe. Os quadrados em azul e vermelho indicam as lojas de item para cada equipe. As linhas pontilhadas em amarelo indicam as rotas principais que conectam as bases das equipes.

A grande diversidade e dinamicidade das ações dos personagens nas partidas [9], bem como seus desempenhos individuais (ou seja, ouro ganho, campeões abatidos ou derrotados, dano causado, cura recebida, etc.), tornam os MOBAs jogos muito competitivos. Em LoL, esta competitividade é ampliada devido à sua popularidade e torneios que fazem com que muitos jogadores se comportem como desportistas profissionais [10].

Dominar LoL é muito desafiador e requer um investimento substancial de tempo [9],

principalmente para equipes inexperientes que não sabem inicialmente como elaborar ou melhorar suas estratégias. Nesse sentido, uma alternativa que poderia ajudar essas equipes seria fornecer informações que levem a melhores decisões estratégicas no jogo. Um exemplo dessas informações poderia ser métricas de desempenho baseadas no comportamento de perfis de equipes bem-sucedidas ou malsucedidas. Com isso, as equipes poderiam monitorar seus desempenhos ao longo das partidas e verificar se estão no caminho correto ou não conforme algum referencial. Por exemplo, vamos supor que uma equipe não tenha se concentrado em melhorar uma determinada métrica de desempenho em suas partidas por algum motivo, como cura efetuada aos campeões aliados por minuto. No entanto, ao analisar o desempenho de equipes de sucesso em um conjunto de partidas, a equipe percebe que a pontuação dessa métrica em equipes de sucesso está em um nível alto e que, portanto, pode ter alguma relevância para melhorar o seu desempenho nas partidas.

Uma das abordagens para a criação de métricas de desempenho é o uso da mineração de dados [11], que permite o estudo da coleta, preparação, processamento, análise e obtenção de informações úteis a partir de dados para apoiar a tomada de decisões [12].

Uma das abordagens para a aplicação da mineração de dados é a utilização do aprendizado de máquina [12], uma subárea da IA que explora o estudo e construção de algoritmos que permitem ao computador aprender ou melhorar seu desempenho em alguma tarefa usando dados. Os algoritmos de aprendizado de máquina fazem uso do princípio de inferência chamado indução, com o qual é possível obter conclusões genéricas a partir de um conjunto de dados fornecido [13]. Esse aprendizado indutivo pode ser dividido em dois tipos principais: supervisionado e não supervisionado. No aprendizado supervisionado uma máquina é treinada com base em um conjunto de dados onde os rótulos das observações são previamente conhecidos para realizar tarefas de previsão, classificação e regressão. Já no aprendizado não supervisionado não existem observações rotuladas para que a máquina possa aprender sobre o domínio do problema, sendo seu objetivo descobrir padrões ou características nos dados [14] por meio de métodos de agrupamento, associação, estimação de densidade, dentre outros.

No contexto de análise de comportamento de jogadores, onde tipicamente não se conhece o que direciona os comportamentos e decisões dos jogadores, os algoritmos de aprendizado de máquina podem fornecer meios para descobrir as principais características nos dados e pa-

drões relacionados ao comportamento em função dessas características [11]. Esses padrões podem ser utilizados para caracterizar os perfis das equipes, ou seja, equipes que possuem comportamento similar, bem como classificar o perfil de uma equipe.

1.1 Objetivos

1.1.1 Objetivo Geral

Esta dissertação tem como objetivo encontrar e caracterizar perfis de comportamento de equipes bem-sucedidas e malsucedidas no contexto do jogo League of Legends usando um processo de mineração de dados baseado em dados sumarizados de desempenho de jogadores nas partidas.

1.1.2 Objetivo Específicos

- (i) Coletar dados de equipes em partidas de League of Legends;
- (ii) Extrair métricas de desempenho de equipes que sejam intuitivas para os jogadores e outros interessados a partir dos dados coletados;
- (iii) Encontrar padrões úteis de comportamento de equipes baseando-se nas métricas de desempenho das equipes;
- (iv) Caracterizar perfis de comportamento de equipes bem-sucedidas e malsucedidas usando esses padrões.

1.2 Metodologia

A Figura 1.2 apresenta a metodologia que usamos para atingir o objetivo. Primeiro, coletamos partidas do *site* oficial de histórico de partidas de LoL. Com esses dados, preparamos um conjunto de desempenho de equipes extraindo características a partir de estatísticas básicas de desempenho individual dos jogadores nas partidas, como quantidade de ouro ganho, quantidade de monstros abatidos, dano físico total recebido, dentre outros. Depois, para transformar esses dados em métricas úteis de desempenho (por exemplo, quantidade de ouro

ganho por minuto), realizamos outras tarefas relacionadas a preparação de dados, tais como limpeza de dados, remoção de características com baixa variância, remoção de *outliers*, cálculo de métricas e normalização de dados e análise de redundância de características. Depois disso, aplicamos uma análise de agrupamento com o método *K-means clustering* nos dados transformados e reduzidos para descobrir padrões de comportamento, ou seja, para descobrir um número ótimo de grupos e, assim, agrupar equipes similares. Finalmente, caracterizamos cada grupo por meio de uma análise exploratória dos dados. Alguns grupos são mais propensos a terem equipes mais vencedoras do que outros, os resultados do nosso estudo ajudaram a descobrir as características que estão associadas a essa propensão e nos permitiram definir perfis de equipes bem-sucedidas e malsucedidas baseando-se nas métricas de desempenho das equipes.

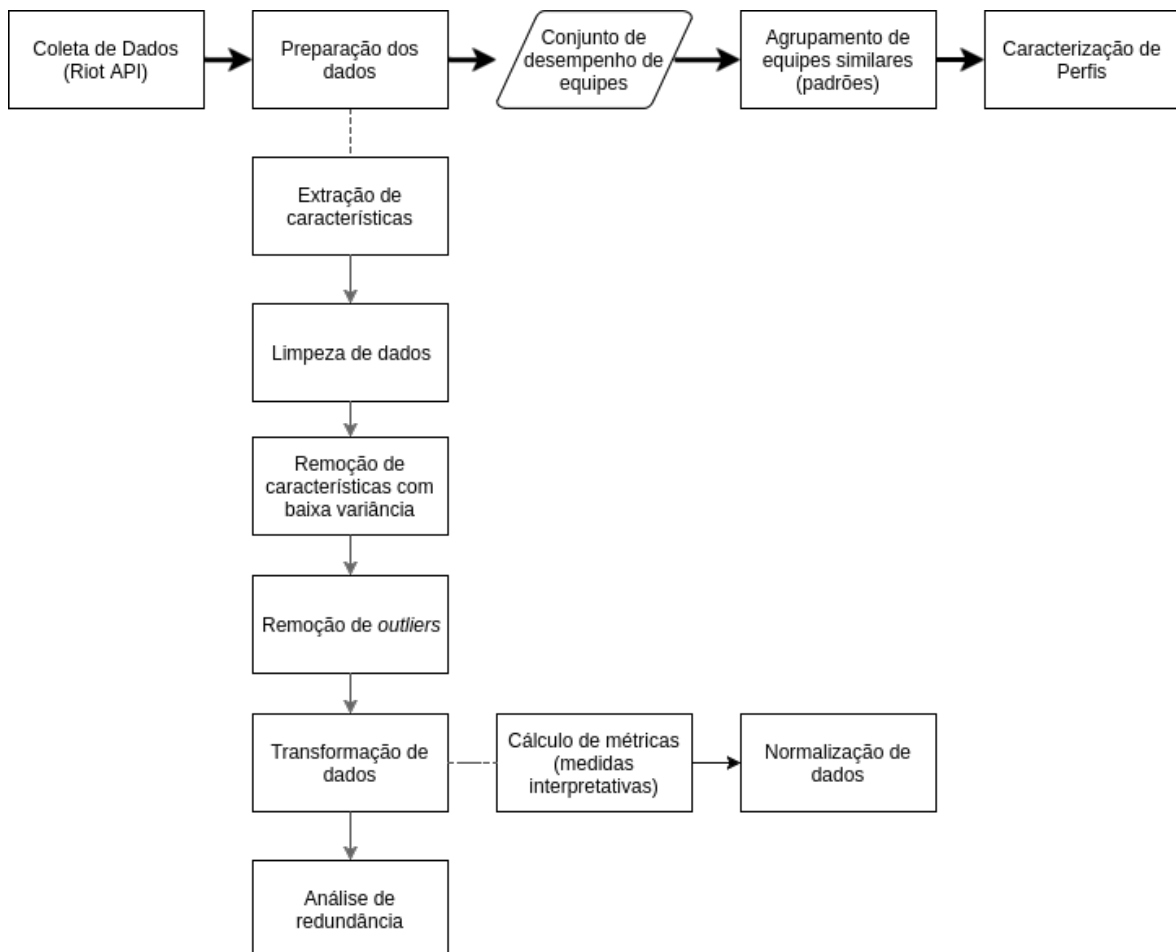


Figura 1.2: Metodologia usada para atingir o objetivo da dissertação.

1.3 Contribuições

Acreditamos que este trabalho resultou em uma importante contribuição ao estudar o comportamento de usuários no contexto dos jogos MOBA. Esses jogos possuem uma crescente popularidade e uma série de aspectos a serem estudados. Até onde sabemos, ainda existe uma escassez de trabalhos acadêmicos que os exploram [9] [15], especialmente em relação ao comportamento de jogadores e equipes em LoL.

Podemos identificar ao menos três classes principais de *stakeholder*¹ para as quais os resultados desta pesquisa são relevantes. Primeiro, a comunidade de mineração de dados, pois este trabalho representa um domínio novo e ainda pouco explorado. Em segundo lugar, a indústria de jogos digitais que pode se beneficiar dos modelos descritivos desenvolvidos para compreender melhor os jogadores e as equipes e, assim, fornecer ferramentas automáticas para melhorar a experiência de jogo. Finalmente, jogadores e equipes inexperientes que podem se beneficiar das informações obtidas a partir dos perfis de equipes bem-sucedidas e malsucedidas para apoiar suas decisões e, assim, melhorar suas habilidades, tornando o jogo ainda mais divertido.

Por fim, este trabalho também resultou na aceitação do artigo científico "*Profiling Successful Team Behaviors in League of Legends*" no XXIII Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia 2017), que possui classificação *qualis* B2. Nesta dissertação, foram abordados os temas e experimentos aqui tratados, sendo este trabalho uma extensão mais aprofundada do artigo.

1.4 Estrutura da Dissertação

Esta dissertação encontra-se dividida em 6 capítulos. O Capítulo 2 apresenta uma introdução sobre o histórico dos MOBAs e uma fundamentação teórica sobre mineração de dados. O Capítulo 3 apresenta os trabalhos relacionados, abordando os temas necessários ao desenvolvimento deste trabalho sobre MOBAs e aprendizado de máquina de uma maneira geral. O Capítulo 4 apresenta os resultados da metodologia adotada para preparar os dados de desempenho das equipes que foram usados nas análises. O Capítulo 5 apresenta os resultados

¹Uma pessoa ou organização com interesse ou preocupação em algo, especialmente um negócio.

da análise de agrupamento que foi usado para encontrar padrões ou grupos nos dados, bem como, a caracterização de perfis conforme os grupos ou padrões encontrados. O Capítulo 6 finaliza este trabalho com as conclusões e trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo aborda o histórico dos MOBAs e os principais conceitos relacionados a mineração de dados no contexto de jogos digitais.

2.1 MOBA

Os jogos MOBA têm suas origens nos jogos do gênero RTS. Suas características herdam traços que estão presentes em ambos os gêneros, sendo que os primeiros MOBAs foram criados utilizando as próprias ferramentas de criação de mapas personalizados RTS disponibilizados pelas desenvolvedoras. Assim, com a popularidade dos jogos de estratégia, houve grande interesse por parte dos jogadores em criar modos de jogo e mapas personalizados para estes, denominados de *mods*. Esses jogadores se tornaram, portanto, modificadores do conteúdo dos jogos ou *modders*. Os *modders* eram responsáveis pelo desenvolvimento de mapas alternativos, e muitos destes tendiam para o estilo de jogo de um MOBA. Desde o início do desenvolvimento do gênero, a jogabilidade consistia em controlar um personagem denominado "herói" ou "campeão", escolher entre três rotas para avançar suas tropas e conquistar a estrutura principal inimiga [16] [17] [18].

2.1.1 Starcraft e o mapa Aeon of Strife

Um dos primeiros MOBAs de sucesso teve origem no jogo RTS da Blizzard, o Starcraft, criado em 1998. Ainda em 1998, um *modder* conhecido como Aeon64 criou um mapa

alternativo para o Starcraft, o Aeon of Strife. Nesse mapa, os jogadores controlavam um único herói cada e lutavam em uma equipe contra outra controlada por IA em três rotas do mapa. Essas rotas conectam as bases das duas equipes. O objetivo era destruir a base da outra equipe.

Embora Aeon of Strife tenha estabelecido os fundamentos para o gênero MOBA, existiam algumas diferenças notáveis do que se espera de um MOBA atualmente. As equipes tinham quatro jogadores cada, em vez de cinco. Além disso, Aeon of Strife não era um jogo competitivo. Em vez disso, uma equipe de personagens controlados por jogadores combatia contra uma equipe de personagens controladas por IA. Os personagens também não subiam de nível enquanto o jogo progredia, e não existia uma selva no mapa com monstros e trilhas extras entre as três rotas.

2.1.2 Warcraft III e o mapa Defense of the Ancients

Em 2002, a Blizzard lançou outro jogo RTS que chamou a atenção da comunidade *modder*: Warcraft III. Este jogo era essencialmente parecido com Starcraft, porém tinha raças diferentes, gráficos tridimensionais melhores, uma série de cenários novos e ambiente de personalização mais poderoso com suporte a uma linguagem de programação.

Em 2003, um *modder* conhecido como Eul criou um mapa chamado de Defense of the Ancients (Dota) para o Warcraft III, do qual foi baseado no mapa Aeon of Strife do jogo Starcraft. Não muito tempo depois, outros jogadores criaram sua própria versão do Dota, cada um adicionando seus próprios heróis, itens e outras coisas.

Assim como Aeon of Strife, Dota permitiu que os jogadores controlassem uma poderosa unidade de heróis e combatessem uma equipe inimiga em três rotas que ligavam a base de cada equipe. No entanto, Dota fez com que duas equipes de personagens controlados por jogadores competissem uns contra os outros, cada equipe tinha cinco jogadores, os heróis subiam de nível à medida que ganhavam experiência e existia uma selva no mapa cheia de monstros entre as trilhas das rotas. Na prática, é exatamente o que se espera de um MOBA hoje.

Em 2004, um *modder* conhecido como Guinsoo combinou elementos de múltiplas variações do Dota e criou uma nova versão chamada de Dota Allstars (DotA). Esta versão rapidamente se tornou a mais popular do mapa. Em 2005, Guinsoo anunciou sua saída do

desenvolvimento do mapa, deixando o desenvolvimento nas mãos de outro *modder* conhecido como IceFrog.

Sob a direção de IceFrog, DotA continuou a crescer em popularidade, pois tornou as partidas mais equilibradas. As produtoras inclusive passaram a investir em campeonatos e competições específicas para o DotA. O sucesso foi tamanho que houve a criação de plataformas competitivas, como a Garena, que passaram a dar suporte específico para partidas de DotA.

2.1.3 League of Legends

Dota Allstars ainda era um *mod* para o Warcraft III. Isso significa que ele dependia de todos os recursos desse jogo, de modo que nunca poderia apresentar um personagem que usasse um modelo completamente novo, nem poderia dar a nenhum dos criadores que trabalhavam nisso um lucro monetário.

Devido ao sucesso de DotA, algumas desenvolvedoras resolveram apostar no estilo de jogo. Em 2009, a desenvolvedora Riot Games lançou o League of Legends. LoL é semelhante ao mapa de Warcraft III em estilo e *design*, mas sua estética é um pouco mais cartunesca e a mecânica um pouco mais fácil de entender. Este é o jogo mais jogado do mundo até os dias atuais, tendo superado o jogo World of Warcraft em meados de 2012. Até então o termo MOBA era inexistente e os jogos eram simplesmente denominados *Aeon like*, *DotA like* e *Action Real-Time Strategy*. A Riot Games foi considerada a responsável por cunhar o termo MOBA.

Um ponto importante a notar no LoL é a estrutura de preços. LoL é livre para jogar, o que significa que qualquer um pode fazer o *download* e experimentar o jogo imediatamente, usando uma seleção rotativa de personagens grátis semanal para jogar uma partida. No entanto, se algum jogador quiser um personagem específico para controlar a qualquer momento em uma partida, ele tem que comprar. Esse modelo de negócios foi um enorme sucesso quando o jogo foi lançado. As pessoas que nunca fariam o *download* de um *mod* para um antigo jogo de PC como Warcraft III estavam experimentando curiosamente o novo MOBA.

2.1.4 Outros

Além da Riot Games, a Valve resolveu apostar no mercado MOBA, lançando em 2013 uma nova versão do jogo Dota, o Dota2. Neste, os mesmos personagens e jogabilidades de DotA foram mantidos, porém recriados em uma plataforma dedicada e com gráficos melhorados. Outros jogos originados de DotA também ganharam forma, como o Heroes of Newerth (2010), da S2 Games, onde a jogabilidade de DotA foi reproduzida com personagens diferentes. Outros jogos de destaque na atualidade do gênero MOBA são: Smite (2014), da Hi-Rez Studios; Strife (2015), da S2 Games; e Heroes of the Storm (2015), da Blizzard.

2.2 Mineração de Dados

A mineração de dados é o estudo da coleta, preparação, processamento, análise e obtenção de informações úteis a partir de dados. Existe uma grande variação em relação ao domínio de problemas, aplicações, formulações e representações de dados. Portanto, mineração de dados é um termo abrangente que é usado para descrever diferentes aspectos do processamento de dados [12].

Atualmente, devido aos avanços tecnológicos e da informatização de todos os aspectos da vida moderna, praticamente todos os sistemas automatizados geram ou armazenam alguma forma de dados. Isso resultou em um crescimento excessivamente rápido do volume de dados, atingindo a ordem de petabytes ou exabytes [12].

Esse crescimento excessivo do volume de dados vem emergindo em diversas áreas da indústria e pesquisa, como bioinformática, análise de redes sociais, visão computacional e jogos digitais [11]. Portanto, é natural examinar se é possível extrair informações concisas e, possivelmente, acionáveis a partir dos dados disponíveis de modo a apoiar a tomada de decisões estratégicas. Há uma grande quantidade de informações escondidas em dados brutos (telemétricos) de jogos digitais, por exemplo, mas nem tudo está prontamente disponível, e alguns são difíceis de se descobrir sem o conhecimento especializado (ou mesmo com ele) [11]. Isso fez com que dados telemétricos de muitos jogos fossem rastreados, registrados e armazenados, mas não analisados e aplicados. O desafio enfrentado pela indústria de jogos para tirar proveito dos dados reflete o desafio geral de trabalhar com dados em grande escala. Apenas recuperar informações de bancos de dados não é suficiente para apoiar a tomada de

decisões. A revolução dos dados exige métodos e algoritmos de análise que dimensionem tamanhos de dados massivos e permitam análises efetivas e rápidas, bem como resultados que são intuitivamente acessíveis para não especialistas.

Para resolver este problema, os analistas usam um processo de mineração de dados, onde os dados brutos são coletados e preparados em um formato padronizado. Em seguida, os dados padronizados podem ser armazenados e, finalmente, processados para extrair informações com o uso de algoritmos analíticos. Esse *pipeline* de processamento é conceitualmente análogo ao de um processo de mineração real de um minério mineral para obter um produto final refinado.

Geralmente, a mineração de dados remete a noção de algoritmos analíticos, no entanto, a maior parte do processo está relacionada à preparação de dados. A fase de preparação de dados é altamente específica para cada aplicação porque os diferentes formatos de dados requerem algoritmos diferentes para serem aplicados. Essa fase pode incluir integração de dados, limpeza, extração de características e outras transformações. Em alguns casos, a seleção de características também pode ser usada para deixar os dados mais representativos.

Os dados brutos podem ser arbitrários, não estruturados ou mesmo em um formato que não é imediatamente apropriado para processamento automatizado. Por exemplo, os dados coletados manualmente podem ser extraídos de fontes variadas em diferentes formatos e, de alguma forma, precisam ser processados por uma aplicação automatizada para obter informações.

Existem também diferentes tipos de dados: quantitativo (por exemplo, idade), categórico (por exemplo, gênero), texto, espacial, temporal ou grafo. Embora a forma mais comum de dados seja multidimensional, uma proporção crescente pertence a tipos de dados mais complexos. Embora exista uma portabilidade conceitual de algoritmos entre muitos tipos de dados em um nível muito alto, esse não é o caso de uma perspectiva prática. A realidade é que o tipo de dados correto pode afetar significativamente o comportamento de um algoritmo particular.

Do ponto de vista analítico, a mineração de dados é desafiadora devido à grande diversidade nos problemas e tipos de dados encontrados. Mesmo considerando classes relacionadas de problemas, as diferenças são bastante significativas. Por exemplo, um problema de recomendação de produtos em uma base de dados multidimensional é muito diferente

de um problema de recomendação social devido às diferenças no tipo de dados subjacentes. No entanto, apesar dessas diferenças, as aplicações de mineração de dados são muitas vezes classificadas em um desses principais problemas ou tarefas de mineração de dados: caracterização, discriminação, associação, classificação, regressão, agrupamento, análise de *outliers*. Esses problemas são tão importantes porque eles são usados indiretamente como blocos de implementação em uma grande variedade de cenários de aplicações. O projeto final de uma solução para um problema específico de mineração de dados depende da habilidade do analista no mapeamento da aplicação para os diferentes blocos ou na utilização de novos algoritmos para uma aplicação específica.

2.2.1 Dados Telemétricos em Jogos Digitais

Jogos digitais modernos variam desde aplicativos simples a sistemas de informação incrivelmente sofisticados, mas comuns para todos eles é que precisam acompanhar as ações dos jogadores e calcular uma resposta para eles. Nos últimos anos, o rastreamento e o registro dessas informações, denominadas dados telemétricos em Ciência da Computação, têm-se generalizado na indústria de entretenimento digital, levando a uma riqueza de informações detalhadas sobre o comportamento dos jogadores [11].

A telemetria é um dos termos fundamentais da análise de jogos, descrevendo a coleta de dados à distância. A coleta, análise e geração de informações a partir de dados telemétricos de comportamento de usuários é a base para a análise atual no desenvolvimento de jogos.

A coleta e o uso da telemetria têm uma história que remonta ao século XIX, onde os primeiros circuitos de transmissão de dados foram desenvolvidos, mas hoje o termo abrange qualquer tecnologia que permita a medição à distância. Exemplos comuns incluem transmissão de ondas de rádio a partir de um sensor remoto ou transmissão e recepção de informações através de uma rede IP.

A telemetria de jogo é um termo da área de pesquisa e desenvolvimento de jogos que é usada para denotar qualquer fonte de dados obtida de uma certa distância. Existem muitos usos populares de telemetria em jogos, incluindo monitoramento remoto e análise de servidores, dispositivos móveis e comportamento de usuários. A fonte de telemetria mais importante no desenvolvimento de jogos atualmente é a telemetria do usuário, ou seja, dados sobre o comportamento dos usuários (jogadores), por exemplo, na interação com jogos, com-

portamento de compra, movimento físico ou interação com outros usuários ou aplicações.

Dados telemétricos de um jogo podem ser pensados como unidades brutas de dados que são derivadas remotamente de algum lugar, por exemplo, um cliente de um jogo instalado enviando dados sobre como um usuário interage com o jogo ou dados de transações de um sistema *online* de pagamento. Em relação aos dados de comportamento dos usuários em jogos *online* para vários jogadores, por exemplo, o código embutido no cliente de um jogo transmite dados para um servidor de coleta ou coleta dados dos servidores do jogo.

Os dados que estão sendo transmitidos seguem diferentes convenções de nomenclatura dependendo do campo de pesquisa ou domínio de aplicação ao qual as pessoas estão aplicando os dados. A essência é que a telemetria é medida em nível de atributo dos objetos. Por exemplo, a localização de um personagem do jogador enquanto navega em um ambiente virtual em determinado momento. Nesse caso, a localização é o atributo, o personagem do jogador é o objeto.

Para que se trabalhe com dados telemétricos, os dados de atributos precisam ser operacionalizados, o que significa ter que decidir uma maneira de expressar esses dados. Um exemplo seria decidir se os dados de registros de localização dos personagens (ou usuários de celular) podem ser estruturados como um número que representa a soma do deslocamento em metros. A operacionalização dos dados de atributos dessa forma os transforma em variáveis ou características. O termo varia de acordo com o campo científico, no entanto, o termo característica é frequentemente usado na mineração de dados.

Dados brutos telemétricos podem ser armazenados em vários formatos de base de dados e estruturados de forma a tornar possível a transformação dos dados em várias medidas interpretativas, como tempo médio de duração de jogo, receita por dia, número de usuários ativos por dia, e assim por diante. Essas medidas são chamadas de métricas de jogo. As métricas de jogo são, em essência, medidas interpretativas de algo e podem apresentar vantagens potenciais, como suporte à tomada de decisões estratégicas. Métricas podem ser características e vice-versa, ou agregados mais complexos ou valores calculados como a soma de múltiplas características. Para exemplificar: dados telemétricos de um atirador em um jogo de tiro podem incluir dados sobre a localização do personagem do jogador no ambiente virtual, as armas usadas e informações sobre se cada tiro atinge ou erra um alvo, dentre outros. Esses dados de atributos podem ser convertidos em características básicas, como número de acer-

tos ou número de erros em um domínio de 0:1000, sendo 1000 o maior número de acertos efetuados em um mapa específico. Por sua vez, essas características básicas podem ser usadas para realizar uma análise simples, como o cálculo da taxa de acerto/erro em cada mapa do jogo. Uma alternativa é usar características como identificação do jogador, duração da partida e pontos marcados para calcular a métrica de pontos marcados por minuto para cada jogador.

2.2.2 Processo de Mineração de Dados

Em 1996, foi desenvolvido um padrão interindústria que adotou uma abordagem neutra em termos de indústria, aplicação e ferramenta para mineração de dados, definindo como Processo Padrão Interindústria para Mineração de Dados (*Cross-Industry Standard Process for Data Mining* - CRISP-DM) ¹ [19]. O CRISP-DM representa uma abordagem fundamental para os processos de mineração de dados, e diversas variantes especializadas existem voltadas para indústrias ou problemas específicos. Embora possa parecer complicado aplicar o processo CRISP-DM completo para cada uma das questões que queremos responder usando telemetria de jogos, na prática, algumas fases podem ser muito rápidas de se passar, especialmente se a análise já foi realizada em um conjunto de dados de comportamento de usuários anterior. O CRISP-DM fornece um processo padrão não proprietário e livre disponível para adaptar a mineração de dados na estratégia geral de resolução de problemas de uma organização ou pesquisa. É um processo iterativo e sua sequência de fases é adaptativa, ou seja, a próxima fase, em uma sequência de seis fases definidas, depende da associação de resultados da fase anterior. Por exemplo, pode ser necessário retornar à fase de preparação de dados para refinar os dados antes de avançar para a fase de modelagem - é uma situação típica em que a solução para uma questão leva a mais questões ou quando se trabalha com comportamento de jogadores [11].

O CRISP-DM consiste em seis fases destinadas a um processo cíclico (Figura 2.1):

- **Compreensão do negócio:** A compreensão do negócio ou pesquisa inclui a determinação de objetivos, a avaliação da situação atual, a definição de metas de mineração de dados e o desenvolvimento de um plano de projeto.

¹www.crisp-dm.org

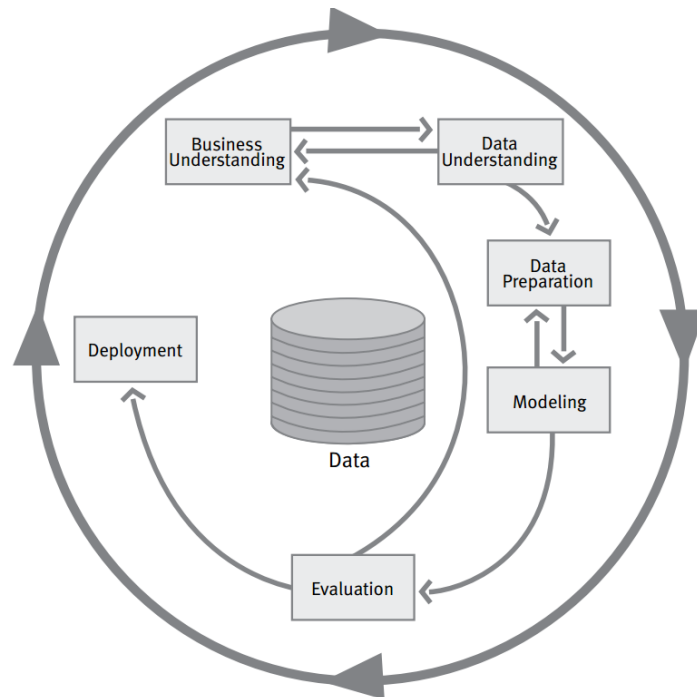


Figura 2.1: Fases do processo CRISP-DM [5]: Compreensão do negócio (*Business understanding*), Compreensão dos dados (*Data understanding*), Preparação dos dados (*Data preparation*), Modelagem (*Modeling*), Avaliação (*Evaluation*), Implantação (*Deployment*)

- **Compreensão dos dados:** Uma vez que os objetivos de negócios e o plano de projeto estão estabelecidos, a compreensão dos dados considera os requisitos dos dados. Esta fase pode incluir a coleta inicial de dados, descrição dos dados, exploração dos dados e verificação da qualidade dos dados. A exploração dos dados, como a visualização sumarizada de estatísticas (que inclui a exibição visual de características categóricas) pode ocorrer no final desta fase. Métodos como análise de agrupamento também podem ser aplicados durante esta fase, com o objetivo de identificar padrões nos dados.
- **Preparação dos dados:** Uma vez que os requisitos dos dados disponíveis estão identificados, eles precisam ser selecionados, limpos e estruturados na forma desejada. A limpeza de dados e a transformação de dados precisam ocorrer nessa fase. A exploração dos dados em maior profundidade pode ser aplicada durante essa fase, bem como outros métodos podem ser também utilizados, proporcionando novamente a oportunidade de identificar padrões baseados na compreensão do negócio ou pesquisa.
- **Modelagem:** Após a preparação dos dados, métodos de mineração de dados (asso-

ciação, agrupamento, classificação) podem ser selecionados e aplicados. Métodos diferentes podem ser usados para o mesmo problema. Se o método resultar em um modelo, as configurações do modelo podem ser calibradas para otimizar os resultados. Esse processo pode ser repetido com novas seleções e transformações de dados, refinando gradualmente o resultado ou integrando novas fontes de dados relevantes para obter resultados diferentes, mais apropriados e valiosos.

- **Avaliação:** Os resultados ou modelos devem ser avaliados conforme os objetivos de negócios definidos na fase de compreensão do negócio. Isso pode levar à identificação de outras necessidades, frequentemente retornando às fases anteriores. A obtenção de entendimento do negócio é um procedimento iterativo na mineração de dados, onde os resultados de várias ferramentas de visualização, estatística e IA mostram os novos relacionamentos que proporcionam uma compreensão mais profunda das operações organizacionais.
- **Implantação:** A mineração de dados pode ser usada para validar hipóteses anteriormente levantadas ou para descoberta de conhecimento, como identificação de relacionamentos inesperados e úteis. Por meio do conhecimento encontrado nas fases anteriores do processo, os resultados ou modelos obtidos podem então serem apresentados aos *stakeholders* e, então, aplicados para vários propósitos. As apresentações devem ser realizadas de forma a ajudar o *stakeholder* alvo a entender, interpretar e agir sobre os resultados da mineração de dados, isso pode ser feito usando uma representação do conhecimento, como uma visualização ou relatório. Por exemplo, um analista de dados telemétricos de jogos poderia desenvolver uma visualização de mapa de calor a partir dos registros de localização dos jogadores em um mapa de um jogo para apresentar ao líder de *designer* do jogo, que por sua vez, poderia ter algum *insight* inovador em relação ao mapa. Os resultados ou modelos precisam ser monitorados quanto às mudanças nas condições operacionais, porque o que pode ser verdade hoje pode não ser verdade no ano a partir desse instante. Se ocorrerem alterações significativas, o processo deve ser refeito. Também é aconselhável registrar os resultados dos projetos de mineração de dados, de modo que as evidências documentadas estejam disponíveis para estudos futuros.

O processo CRISP-DM em seis fases não é rígido, por meio de um procedimento exato. Geralmente, há uma grande quantidade de retrocesso. Além disso, analistas experientes podem não precisar aplicar todas as fases para um problema específico. Portanto, esse processo fornece um *framework* útil para a mineração de dados.

2.2.3 Preparação dos Dados

O formato bruto dos dados geralmente é amplamente variável. Muitos valores podem estar faltando, inconsistentes em diferentes fontes de dados e errôneos. Para o analista, isso leva a inúmeros desafios na utilização eficiente dos dados. Por exemplo, considere o caso de avaliar os interesses dos consumidores a partir de suas atividades em um site de redes sociais. O analista pode primeiro determinar os tipos de atividades que são valiosos para o processo de mineração. A atividade pode corresponder aos interesses inseridos pelo usuário, aos comentários inseridos pelo usuário e ao conjunto de amigos do usuário junto com seus interesses. Todas essas informações são diversas e precisam ser coletadas de diferentes bancos de dados no site de redes sociais. Além disso, algumas formas de dados, como *logs* brutos, muitas vezes não podem ser diretamente usados devido à sua natureza não estruturada. Em outras palavras, características importantes precisam ser extraídas dessas fontes de dados. Por isso existe a necessidade da fase de preparação dos dados.

A fase de preparação de dados é um processo de vários estágios que compreende várias etapas individuais, algumas ou todas podem ser usadas em uma determinada aplicação. Essas etapas são as seguintes:

- **Extração de Características:** Dados brutos geralmente estão em uma forma que não é adequada para o processamento. Os exemplos incluem *logs* brutos, documentos, dados semiestruturados e possivelmente outras formas de dados heterogêneos. Nesses casos, pode ser desejável obter características significativas a partir dos dados. Geralmente, as características com boa interpretação semântica são mais desejáveis porque simplificam a capacidade do analista de entender os resultados intermediários e geralmente têm melhor ligação aos objetivos da aplicação de mineração de dados. Em alguns casos onde os dados são obtidos de múltiplas fontes, os dados precisam ser integrados em uma única base de dados para serem processados. Além disso, alguns algoritmos

podem funcionar apenas com um tipo de dados específico, sendo que os dados podem conter tipos heterogêneos. Nesses casos, a portabilidade dos tipos de dados se torna importante quando os atributos de um tipo são transformados em outro. Isso resulta em um conjunto de dados mais homogêneo que pode ser processado pelos algoritmos existentes.

- **Limpeza de dados:** Na fase de limpeza de dados, os itens faltantes, errados e inconsistentes são removidas dos dados. Alguns itens faltantes também podem ser estimados para serem inseridos nos dados.
- **Redução e transformação de dados:** Nessa fase, o tamanho dos dados é reduzido através da seleção de subconjuntos de dados, seleção de subconjuntos de características ou transformação de dados. Os ganhos obtidos nesta fase são duplos. Primeiro, quando o tamanho dos dados é reduzido, os algoritmos geralmente são mais eficientes. Em segundo lugar, se características irrelevantes ou itens irrelevantes são removidos, a qualidade do processo de mineração de dados é melhorada. O primeiro objetivo pode ser alcançado por técnicas genéricas de amostragem ou redução de dimensionalidade. Para alcançar o segundo objetivo, uma abordagem altamente específica do problema em questão deve ser usada para seleção de características. Por exemplo, uma abordagem de seleção de características que funciona bem para agrupamento pode não funcionar bem para classificação.

Extração de Características

A primeira fase de extração de características é crucial, embora seja muito específica para a aplicação. Em alguns casos, a extração de características está intimamente relacionada ao conceito de portabilidade de tipo de dados, onde características de baixo nível de um tipo podem ser transformados em características de nível superior de outro tipo. A natureza da extração de características depende do domínio do qual os dados são coletados.

A extração de características às vezes é mais arte do que engenharia, ou seja, é altamente dependente da habilidade do analista em escolher as características e a representação que melhor se adequam ao problema. Embora este aspecto específico da análise de dados geralmente pertença ao especialista do domínio, talvez isso seja o mais importante. Se as

características corretas não forem extraídas, a análise só pode ser tão boa quanto os dados disponíveis.

Limpeza de Dados

O processo de limpeza de dados é importante por causa de erros associados ao processo de coleta de dados. Problemas de qualidade em dados brutos podem vir de várias formas, por exemplo, erros ortográficos durante a entrada de dados, informações faltantes ou presença de dados inválidos. Quando várias fontes de dados são integradas em uma base de dados ou uma análise é realizada usando múltiplas fontes de dados (por exemplo, telemetria de diferentes jogos), uma limpeza de dados mais cuidadosa deve ser realizada devido ao potencial de erro introduzido quando os conjuntos de dados são combinados [11].

Os problemas mencionados acima podem ser uma fonte significativa de imprecisão para aplicações de mineração de dados. São necessários métodos para remover ou corrigir itens faltantes e incorretos dos dados. Existem, portanto, diversos aspectos importantes da limpeza de dados:

- **Tratamento de entradas faltantes:** Muitos itens nos dados podem permanecer não especificados por causa de deficiências na coleta de dados ou na natureza inerente dos dados. Esses itens faltantes podem precisar ser estimados. O processo de estimativa de itens faltantes também é chamado de imputação.
- **Tratamento de entradas incorretas:** Nos casos em que um mesmo dado está disponível em várias fontes, podem ser detectadas inconsistências. Tais inconsistências podem ser removidas como parte do processo analítico. Outro método para detectar as entradas incorretas é usar o conhecimento específico do domínio sobre o que já se sabe sobre os dados. Por exemplo, se a altura de uma pessoa for listada como 6 metros, provavelmente está incorreta. De modo mais geral, os pontos de dados que são incompatíveis com a distribuição de dados restante são frequentemente ruidosos. Esses pontos de dados são referidos como *outliers*. No entanto, pode ser arriscado assumir que esses pontos de dados são sempre causados por erros.

Redução e Transformação de Dados

O objetivo da redução de dados é representar os dados de forma mais compacta. Quando o tamanho dos dados é menor, é mais fácil aplicar algoritmos sofisticados e computacionalmente custosos. A redução dos dados pode ser em termos do número de linhas (entradas, registros ou itens) ou em termos de número de colunas (dimensões). A redução de dados resulta em alguma perda de informação. O uso de um algoritmo mais sofisticado às vezes pode compensar a perda de informações resultante da redução de dados. Diferentes tipos de redução de dados são usados em várias aplicações:

- **Amostragem de dados:** Os registros de dados são amostrados para criar uma base de dados muito menor. A amostragem geralmente é mais difícil no cenário de fluxo de dados (*streaming*) onde a amostra precisa ser mantida dinamicamente.
- **Seleção de Características:** Somente um subconjunto de características dos dados é usado no processo analítico. Normalmente, esses subconjuntos são escolhidos de uma maneira específica de aplicação. Por exemplo, um método de seleção de características que funciona bem para análise de agrupamentos pode não funcionar bem para a classificação e vice-versa.
- **Redução de dados pela rotação de eixo:** As correlações dos dados são aprimoradas para representar os dados em um número menor de dimensões. Um exemplo desse método de redução de dados inclui a análise de componente principal (*Principal Component Analysis* - PCA).
- **Redução de dados pela transformação de tipo:** Esta forma de redução de dados está intimamente relacionada à portabilidade de tipo de dados. Por exemplo, séries temporais são convertidas em dados multidimensionais de tamanho ou complexidade menores usando transformações de discretização. Da mesma forma, grafos podem ser convertidos também em representações multidimensionais.

Na etapa de transformação de dados, os dados podem ser transformados ou consolidados em formas apropriadas para mineração. Estratégias para a transformação de dados incluem os seguintes:

- **Suavização:** A suavização é usada para remover o ruído dos dados. Alguns métodos incluem *binning*², regressão e agrupamento.
- **Agregação:** Métodos de sumarização ou agregação são aplicadas aos dados. Por exemplo, os dados diários de tiros de um jogador podem ser agregados de modo a calcular quantidades mensais e anuais totais. Esse passo geralmente é usado para construção características básicas a partir de atributos.
- **Normalização:** Os dados são dimensionadas de modo a ficar dentro de uma escala predefinida, como -1 a 1 ou 0 a 1 .

2.2.4 Principais Tarefas de Mineração de Dados

Em geral, tarefas de mineração de dados podem ser classificadas em duas categorias: mineração de dados descritiva e mineração de dados preditiva. O primeiro descreve um conjunto de dados de forma concisa e sumarizada e apresenta propriedades gerais interessantes dos dados. O segundo constrói um ou mais modelos para realizar inferências no conjunto de dados disponível e tenta prever o estado de novos conjuntos de dados.

Portanto, o modelo descritivo encontra padrões ou relacionamentos em dados e explora as propriedades dos dados examinados. Agrupamento, sumarização e análise de associação são exemplos de modelos descritivos. O agrupamento é semelhante à classificação, exceto que os grupos não estão predefinidos, mas são definidos apenas pelos dados. É um método de aprendizado não supervisionado. Corresponde a partição ou segmentação dos dados em grupos distintos de tuplas ou itens similares. Os grupos são caracterizados pelo estudo do comportamento dos dados por especialistas do domínio. O termo segmentação é usado em um contexto muito específico. A sumarização é a técnica de apresentar a informação resumida dos dados. A regra de associação encontra a associação entre os diferentes atributos. A mineração de regras de associação é um processo de dois passos: encontrar todos os conjuntos de itens frequentes, gerando regras de associação fortes a partir dos conjuntos de itens frequentes. A descoberta de sequência é um processo de encontrar os padrões de sequência nos dados. Esta sequência pode ser usada para entender uma tendência.

²*Binning* é uma maneira de agrupar uma série de valores mais ou menos contínuos em um número menor de "caixas".

O modelo preditivo faz previsões sobre valores de dados desconhecidos usando valores conhecidos. Classificação, previsão e regressão são exemplos de modelos preditivos. Muitas das aplicações de mineração de dados visam prever o estado futuro dos dados. Previsão é o processo de análise dos estados atuais e passados do atributo e previsão de seu futuro estado. A classificação é uma técnica de mapeamento dos dados de destino para grupos ou classes predefinidos, isso é um aprendizado supervisionado porque as classes são predefinidas antes dos dados de destino serem examinados. A regressão envolve o aprendizado de uma função que mapeia o item de dados para uma variável numérica de previsão de valor real.

Uma aplicação de mineração de dados pode realizar uma ou mais das seguintes tarefas de mineração de dados:

- **Caracterização:** É o resumo das características gerais dos objetos em uma classe alvo e produz o que é chamado de regras de característica. Os dados relevantes para uma classe específica de usuário são normalmente consultados na base de dados para serem executados através de um método de sumarização e, assim, extrair a essência dos dados em diferentes níveis de abstrações. Por exemplo, talvez seja desejável caracterizar os usuários de um jogo que regularmente jogue mais de 4 horas por dia.
- **Discriminação:** A discriminação de dados produz o que são chamados de regras de discriminação e é basicamente a comparação das características gerais dos objetos entre duas classes referidas como classe alvo e classe contrastante. Por exemplo, pode-se comparar as características gerais dos jogadores que mais venceram partidas no último ano com aqueles que perderam. As técnicas utilizadas para a discriminação de dados são semelhantes às técnicas utilizadas para a caracterização de dados com a exceção de que os resultados de discriminação de dados incluem medidas comparativas.
- **Análise de associação:** Análise de associação estuda a frequência de itens que ocorrem em base de dados transacionais, e com base em um limite chamado suporte, identifica os conjuntos de itens frequentes. Outro limite, a confiança, que é a probabilidade condicional do que um item, aparece em uma transação quando outro item aparece, é usado para identificar regras de associação. Isso é comumente usado para análise de cesta de mercado. Por exemplo, pode ser útil para o gerente saber quais filmes são

frequentemente alugados ou se existe uma relação entre alugar um certo tipo de filmes e comprar pipoca.

- **Classificação:** É a organização de dados em determinadas classes. A classificação usa rótulos de uma dada classe para ordenar os objetos na coleta de dados. As abordagens de classificação normalmente usam um conjunto de treinamento onde todos os objetos já estão associados a rótulos de classe conhecidos. O algoritmo de classificação aprende do conjunto de treinamento e gera um modelo. O modelo é usado para classificar novos objetos. Por exemplo, depois de iniciar uma política de crédito, o gerente de uma loja poderia analisar o comportamento dos clientes em relação ao seu crédito, e rotular, portanto, os clientes que receberam créditos com três possíveis rótulos "seguros", "arriscados" e "muito arriscado". A análise de classificação geraria um modelo que poderia ser usado para aceitar ou rejeitar solicitações de crédito no futuro.
- **Previsão e regressão:** A previsão tem atraído uma atenção considerável tendo em conta as implicações potenciais da previsão bem-sucedida em um contexto de negócios. Existem dois principais tipos de previsões: pode-se tentar prever alguns valores de dados indisponíveis ou tendências pendentes, ou prever um rótulo de classe para alguns dados. O último está vinculado à classificação. Uma vez que um modelo de classificação é construído com base em um conjunto de treinamento, o rótulo de classe de um objeto pode ser previsto com base nos valores de atributos do objeto e nos valores de atributo das classes. Regressão é frequentemente referida à previsão de valores numéricos ou tendências de crescimento ou não em dados relacionados ao tempo. A principal idéia é usar um grande número de valores passados para considerar valores futuros prováveis.
- **Análise de agrupamentos:** Semelhante à classificação, a análise de agrupamentos é a organização de dados em classes. No entanto, ao contrário da classificação, os rótulos das classes são desconhecidos e depende do algoritmo de agrupamento para descobrir classes aceitáveis. É chamado também de classificação não supervisionada, porque a classificação não é definida por determinados rótulos de classe. Existem muitas abordagens de grupo baseadas no princípio de maximizar a semelhança entre objetos em uma mesma classe (similaridade intraclasse) e minimizar a semelhança

entre objetos de diferentes classes (similaridade interclasse).

- **Análise de *Outliers*:** *Outliers* são elementos de dados que não podem ser agrupados em uma determinada classe ou grupo. Também conhecidos como exceções ou surpresas, muitas vezes é importante que sejam identificados. Enquanto os *outliers* podem ser considerados ruídos e descartáveis em algumas aplicações, eles podem revelar conhecimentos importantes em outros domínios e, portanto, podem ser muito significativos e suas análises valiosas.

Capítulo 3

Trabalhos Relacionados

Um conjunto de publicações tem investigado sobre MOBAs por meio de análise de dados e mineração de dados para realizar diversas tarefas.

3.1 Predições de Partidas

Riolut et al. [10] investigam o comportamento dos jogadores em Dota 2 - outro MOBA popular desenvolvido pela Valve Corporation - e sua relevância para prever o resultado das partidas a partir de dados posicionais dos jogadores nas partidas. Conley et al. [20] e Kalyanaraman [21] apresentam, cada, um preditor de resultados de partidas e um recomendador de heróis com base em dados de composição de heróis (personagens) de equipes de Dota 2, onde cada observação corresponde a um vetor que codifica a presença ou não de um herói escolhido por um jogador na equipe. Kinkade e Lim [22] apresentam dois preditores de resultados de partidas de Dota 2: um preditor usa dados sumarizados do estado final das partidas e outro usa dados de composição de heróis. Ong et al. [15] apresentam uma abordagem para agrupar diferentes comportamentos de desempenho individual dos jogadores e, assim, prever a equipe vencedora de uma partida com base na composição de comportamentos de jogadores das equipes. Johansson e Wikström [23] criam um modelo para prever a equipe vencedora de uma partida de Dota 2, usando dados parciais coletados à medida que uma partida avança. Schubert et al. [24] apresentam uma técnica para segmentar dados de jogadores de Dota 2 nas partidas de maneira espacial e temporal onde cada segmento é referenciado como encontro de combate e, assim, possibilitar análises de desempenho e prever resultados

de partidas com base nesses encontros.

3.2 Predições Diversas

Edge [25] cria um modelo para prever quando os jogadores irão abandonar uma partida de Dota 2 antes de terminar, modelando o estado motivacional dos jogadores na partida. Yang et al. [26] modelam interações entre os jogadores em partidas de Dota 2 como uma sequência de grafos para identificar padrões de combate bem-sucedidos e, assim, prever resultados de combates durante a partida. Eggert et al. [27] apresentam uma abordagem para classificar o papel de um jogador dentro de uma equipe de Dota 2 a partir de dados sumarizados de eventos de baixo nível na partida.

3.3 Análise de *Outliers*

Kim et al. [28] propõem um método baseado em dados multimodais de jogadores de LoL durante uma partida (teclado e mouse, tela do jogo, expressão facial, volume e movimento do jogador) para detectar automaticamente as vezes em que esses jogadores exibem um comportamento atípico específico, como excitação, concentração, imersão e surpresa. Cavadenti et al. [29] propõem um método que ajude os jogadores de Dota 2 a melhorar suas habilidades descobrindo padrões estratégicos atípicos bem-sucedidos a partir de traços comportamentais históricos, ou seja, dado um modelo que codifica uma maneira esperada de se jogar (a norma), eles investigam padrões que se desviam da norma que pode explicar o resultado de uma partida.

3.4 Análise de Toxicidade

Shim et al. [30] propõem um esquema de suporte à decisão de jogador automático com base em dados da antiga plataforma de denúncia do LoL (O Tribunal) para encontrar jogadores com mau comportamento (tóxico), como abusos e ataques maliciosos contra outros jogadores. Kwak e Blackburn [31] realizam uma série de análises linguísticas para caracterizar o comportamento linguístico de jogadores tóxicos em League of Legends usando dados de

contribuição colaborativa (*crowdsourcing*) de jogadores acusados como tóxicos. Blackburn e Kwak [32] exploram o uso do desempenho no jogo, relatórios de vítimas de comportamentos tóxicos e características linguísticas de mensagens de jogadores tóxicos em LoL para treinar classificadores para detectar a presença e gravidade de toxicidade. Shores et al. [33] examinam a natureza e os efeitos do comportamento tóxico em League of Legends para desenvolver uma métrica chamada índice de toxicidade e investigar os efeitos da interação de outros jogadores com jogadores tóxicos, incluindo taxa de retenção de jogadores.

3.5 Análise Social

Pobiedina et al. [34] analisam padrões de comportamento social de trabalho em equipe usando dados de comunidades virtuais de Dota 2. Park e Kim [35] analisam a rede social de jogadores de LoL para encontrar conhecimento relevante que ajude, por exemplo, a entender como os jogadores formam equipes. Buchan e Taylor [36] exploram o jogo em equipe analisando as experiências subjetivas dos participantes (tais como comunicação, papel, estado psicológico e nível de habilidade de jogo) ao jogar MOBAs e, assim, criar um modelo conceitual baseado em processos de grupo tradicionais (papéis de equipe, desenvolvimento de grupo e percepções e comportamento durante o estado de desvinculação).

3.6 Análises Diversas

Drachen et al. [9] apresentam um método para investigar como o comportamento das equipes em Dota 2 varia dependendo do nível de habilidade dos jogadores, analisando seus movimentos e a distância entre eles ao longo da partida. Claypool et al. [37] analisam dados quantitativos e qualitativos de partidas de LoL, como duração da partida e contentamento do jogador, para investigar se partidas criadas automaticamente (*matchmaking*) são equilibradas ou não. Neidhardt et al. [38] investigam os impactos de três categorias de fatores de equipe (habilidades dos jogadores, relações de colaboração e parcerias com jogadores em equipes anteriores) em relação ao desempenho das equipes e duração das partidas de Dota 2. Leavitt et al. [39] analisam jogadores em partidas de League of Legends para investigar o impacto do sistema de comunicação não-verbal (*ping*) no desempenho das equipes. Kim et al. [40]

examinam quais fatores individuais e de grupo estão associados à inteligência colaborativa em equipes de LoL e se ela é preditiva para o desempenho dessas equipes, com base em conjuntos de dados que compreendem métricas no jogo, resultados laboratoriais e questionários de equipes.

3.7 Conclusão

Embora estes estudos desempenhem um papel essencial na literatura, nenhum deles se propôs a modelar perfis de equipe em LoL com base no comportamento de desempenho das equipes nas partidas, bem como, fornecer métricas de desempenho bem-sucedidas e malsucedidas de equipes que possam apoiar a tomada de decisão.

Capítulo 4

Preparação dos Dados

Esta seção descreve a coleta de dados e também uma série de tarefas executadas para preparar o conjunto de dados de desempenho de equipes: extração de características; limpeza de dados; remoção de características com baixa variância; remoção de *outliers*; transformação de dados; normalização de dados; análise de redundância de características.

4.1 Coleta de Dados

A Riot Games, desenvolvedora do LoL, fornece uma Interface de Programação de Aplicação (*Application Programming Interface* - API) baseada na Web para acessar os históricos das partidas em formato JSON [7]. Um histórico de partida contém dados como modo de jogo ¹, tipo de fila ², duração da partida, time vencedor e perdedor, e número de identificação única. Um histórico contém também dados de cada jogador que participa da partida, como escolha de personagem e estatísticas sumarizadas de desempenho do jogador em toda a partida. Essas estatísticas indicam, por exemplo, dano total causado e recebido, ouro total recebido e gasto, dano total causado e sofrido, e cura total recebida na partida. A Seção 4.1.1 apresenta as descrições de todas as 38 estatísticas numéricas fornecidas pela API.

LoL tem um pico diário de mais de 7,5 milhões de jogadores [7] no mundo todo. Portanto, devido a enorme quantidade de partidas que existem, coletamos aleatoriamente apenas

¹O modo clássico é o mais escolhido pelos jogadores e uma partida desse modo deve ter 10 participantes duelando uns aos outros em duas equipes diferentes.

²Um jogador precisa entrar no sistema de filas para encontrar e participar de uma partida; existem vários tipos de filas em LoL

110.000 históricos de partidas da API a partir de fevereiro de 2016 até dezembro de 2016, de modo que essas partidas fossem representativas. Para cada mês, coletamos 10.000 históricos. Consideramos os seguintes filtros para a coleta:

- Região: Brasil ³
- Temporada: 2016
- Modo de jogo: Clássico
- Tipo de fila: Competitivo (*ranked*) *solo*;
- Quantidade de jogadores em cada equipe: 5
- Versão da partida: v6.x.y

4.1.1 Estatísticas Básicas

Descrições das 38 estatísticas básicas de desempenho dos jogadores nas partidas:

goldEarned - A quantidade de ouro que o campeão ganhou na partida. O ouro é a moeda de League of Legends. No início da partida, os campeões recebem o ouro inicial e podem ganhar mais ouro através de vários meios. Os campeões podem ganhar ouro passivamente, sem exigir ação direta do jogador, a cada 10 segundos. O ouro obtido através da ação direta do jogador é considerado um ganho de ouro ativo: pode ser obtido aumentado por itens, habilidades do campeão, matando *minions* inimigos e monstros neutros; matando ou ajudando a matar campeões inimigos; destruindo estruturas de defesa da equipe inimiga; vendendo itens na loja.

goldSpent - A Quantidade de ouro que o campeão gastou na partida. Ele é usado para comprar itens na loja que oferecem campeões com estatísticas e habilidades de bônus, o que, por sua vez, é uma das principais maneiras pelas quais os campeões aumentam seu poder ao longo de um jogo.

assists - A quantidade de assistências que o campeão fez. Uma assistência é a ação de ajudar um campeão aliado a matar um campeão inimigo. Um campeão pode marcar

³A API do LoL não permite coletar partidas de regiões de maneira aleatória, por isso, foram coletadas apenas partidas do Brasil.

uma assistência atacando (sem matar) o campeão inimigo nos últimos 10 segundos antes da morte dele ou contribuindo passivamente durante esse período de tempo. Quando o campeão inimigo é morto, o assassino recebe uma certa quantidade de ouro, enquanto 50% desse montante é dividido entre todos da equipe que ajudaram.

deaths - A morte ocorre quando um campeão tira danos suficientes para reduzir a saúde do oponente à zero. Quando um campeão morre, a tela do seu jogador fica cinza sombreado, uma janela com informações sobre a morte e um temporizador para que o campeão volte ao combate são exibidos. A morte dos cinco membros de uma equipe simultaneamente é referida como um *ace*; é possível que ambas as equipes façam um *ace* cada ao mesmo tempo e que nenhum campeão esteja presente no campo de combate.

kills - A quantidade de campeões inimigos que o campeão matou. O ato de matar é o evento de diminuir a saúde de um campeão inimigo para zero, enquanto o inimigo não tem habilidades ou itens para evitar a morte.

doubleKills - O ato de matar dois campeões dentro de 10 segundos.

tripleKills - O ato de matar três campeões dentro de 10 segundos um do outro.

quadraKills - O ato de matar quatro campeões dentro de 10 segundos um do outro.

pentaKills - O ato de matar cinco campeões dentro de 10 segundos um do outro.

largestMultiKill - A maior quantidade de *kills* seguidos do campeão dentro de 10 segundos um do outro.

killingSprees - A quantidade de *killing spree*s realizadas pelo campeão. *Killing spree*s são níveis especiais que se ganha quando uma certa quantidade de inimigos é morta, para alcançar os *killing spree*s é necessário que o campeão não morra enquanto soma os seus *kills*. Um campeão que esteja em *killing spree* e é morto garante ao campeão que o matou ouro extra.

largestKillingSpree - A maior quantidade de *killing spree*s do campeão.

largestCriticalStrike - O maior ataque crítico que o campeão efetuou. Um ataque crítico é um ataque básico ou uma habilidade que causa duas vezes o seu dano normal.

physicalDamageDealt - O dano físico total causado na partida. O dano físico é um dos tipos de dano na League of Legends. O dano físico é causado por todos os ataques básicos, por algumas habilidades de campeão e por alguns efeitos de itens. O dano físico total causado na partida corresponde a soma do dano físico total causado aos campeões inimigos e o dano

físico total causado aos *minions* inimigos e monstros neutros.

physicalDamageDealtToChampions - O dano físico total causado aos campeões inimigos.

physicalDamageTaken - O dano físico total que o campeão recebeu durante a partida.

magicDamageDealt - O dano mágico total causado. O dano mágico é um dos tipos de dano em *League of Legends*. O dano mágico é causado por alguns itens e a maioria dos feitiços dos campeões, embora alguns feitiços causem dano físico (*physicalDamageDealt*) ou dano verdadeiro (*trueDamageTaken*). O dano mágico total causado corresponde a soma do dano mágico total causado aos campeões inimigos e o dano mágico total causado aos *minions* inimigos monstros neutros.

magicDamageDealtToChampions - O dano mágico total causado aos campeões inimigos.

magicDamageTaken - O dano mágico total que o campeão recebeu durante a partida.

trueDamageDealt - O dano total verdadeiro causado aos campeões inimigos e monstros. O dano verdadeiro é um dos tipos de dano na *League of Legends*. O dano verdadeiro ignora a resistência das armaduras e mágica, bem como outras formas de redução de danos. O dano verdadeiro corresponde a soma do dano verdadeiro total causado aos campeões inimigos e o dano verdadeiro total causado aos *minions* inimigos e monstros neutros.

trueDamageDealtToChampions - O dano verdadeiro total causado aos campeões inimigos.

trueDamageTaken - Dano verdadeiro total sofrido de campeões e *minions* da equipe inimiga e monstros neutros.

totalDamageDealt - O dano total causado aos campeões inimigos e monstros. Corresponde a soma do dano físico total (*physicalDamageDealt*) e o dano mágico total (*magicDamageDealt*) causado aos campeões inimigos e monstros.

totalDamageDealtToChampions - Dano total causado aos campeões inimigos. Corresponde a soma do dano físico total (*physicalDamageDealtToChampions*) e o dano mágico total (*magicDamageDealtToChampions*) causado aos campeões inimigos.

totalDamageTaken - O dano total sofrido dos campeões inimigos e monstros. Corresponde a soma do dano físico total (*physicalDamageDealt*) e o dano mágico total (*magicDamageDealt*) sofrido dos campeões inimigos e monstros.

minionsKilled - O total de *minions* da equipe inimiga que foram matados na partida.

neutralMinionsKilled - O total de monstros neutros que foram matados na partida. Corresponde a soma do total de monstros neutros matados na selva do inimigo (*neutralMinionsKilledEnemyJungle*) e do total de monstros neutros matados na selva da equipe aliada (*neutralMinionsKilledTeamJungle*).

neutralMinionsKilledEnemyJungle - O total de monstros neutros matados na selva do inimigo.

neutralMinionsKilledTeamJungle - Total de monstros neutros matados na selva da equipe aliada.

sightWardsBoughtInGame - A quantidade de sentinelas de visão que o campeão comprou na partida. A visão é uma propriedade de unidades, estruturas e certos feitiços, itens e habilidades de invocação em League of Legends que indica que uma equipe possui visão de uma área no mapa. Uma sentinela de visão é um item que os campeões podem comprar e implantar para permitir que uma equipe possua visão de uma área alvo no mapa temporariamente.

visionWardsBoughtInGame - Quantidade de sentinelas detectoras que o campeão comprou durante a partida. Ao contrário do item sentinela de visão, o item sentinela detectora permite que o campeão e sua equipe veja campeões inimigos e objetos invisíveis no mapa.

wardsKilled - A quantidade de sentinelas inimigas que foram destruídas na partida.

wardsPlaced - A quantidade de sentinelas de visão e detectoras que foram implantadas no mapa.

totalHeal - A quantidade de cura total efetuada aos campeões e *minions* aliados. Curar é um feitiço que instantaneamente restaura certo nível de saúde do campeão e aliados próximos, além de conceder um bônus de velocidade de movimento temporário.

totalUnitsHealed - A quantidade de vezes que o campeão curou a saúde dos *minions* e campeões aliados.

totalTimeCrowdControlDealt - O tempo total em que o campeão controlou ou bloqueou as ações da equipe inimiga na partida. O *crowd control* é um termo geral usado em League of Legends para descrever habilidades ou feitiços que removem ou diminuem o controle que um ou mais campeões inimigos tem sobre aspectos de si mesmo - aspectos como ser capaz de lançar feitiços ou iniciar comandos de movimento.

towerKills - A quantidade de torres inimigas que o campeão destruiu. Uma torre é uma estrutura fortificada que bloqueia as ondas de *minion* e o caminho dos campeões da equipe inimiga para a base. Eles causam danos aos inimigos e proporcionam visão e controle geral do campo de batalha para a equipe aliada. Os campeões devem destruir torres inimigas para empurrar seu ataque no território dos inimigos. As torres causam grandes danos às unidades inimigas, mas apenas visam uma unidade por vez.

inhibitorKills - A quantidade de inibidores que o campeão destruiu na partida. Os inibidores são estruturas que bloqueiam a geração de *minions* inimigos em suas respectivas rotas. Uma vez que o inibidor da equipe inimiga é destruído, a base aliada começará a gerar *minions* mais fortes (super *minions*) em vez de *minions* comuns.

4.2 Extração de Características

Em seguida, extraímos características das partidas coletadas para modelar nosso conjunto de dados de desempenho de equipes X .

Seja M um conjunto de históricos de partidas, T um conjunto de equipes e P um conjunto dimensional de desempenho de jogadores. Um histórico de partida $m \in M$ consiste de duas equipes distintas $m = \{t_a, t_b | t \in T; 1 \geq a, b \leq |T|\}$, cada equipe $t \in T$ consiste do desempenho de 5 jogadores distintos $t = \{p_k | p_k \in P^t; 1 \geq k \leq 5; P^t \subset P\}$, e cada desempenho de jogador $p \in P$ é um vetor de estatísticas básicas de um jogador em uma partida, tais como *kills*, *minionsKilled* e *assists* apresentadas na Seção 4.1.1 (total de 38 estatísticas).

Nós coletamos $|M| = 110000$ históricos de partidas, o que corresponde a dados de $|T| = |M| * 2 = 220000$ equipes e $|P| = |T| * 5 = 1100000$ desempenho de jogadores. Para modelar o conjunto de desempenho de equipes X , selecionamos as $d = 38$ estatísticas da Seção 4.1.1 como características para o conjunto de desempenho de jogadores P e, então, resumizamos o desempenho dos jogadores por equipe de modo que $X = \{\sum P^t | t \in T\}$ e X seja uma matriz de dimensões 220000×38 .

Finalmente, para garantir a atomicidade no conjunto de dados X , decompomos algumas características por subtração para obter novas características: *physicalDamageDealtToMonsters* (dano físico total causado a monstros) e *magicDamageDealtToChampions* (dano mágico

Tabela 4.1: Características resultantes da decomposição.

Característica	Cálculo
physicalDamageDealtToMonsters	physicalDamageDealt - physicalDamageDealtToChampions
magicDamageDealtToMonsters	magicDamageDealt - magicDamageDealtToChampions

Tabela 4.2: Características compostas.

Característica	Cálculo
totalDamageDealt	physicalDamageDealt + magicDamageDealt
totalDamageDealtToChampions	physicalDamageDealtToChampions + magicDamageDealtToChampions
totalDamageTaken	physicalDamageTaken + magicDamageTaken
neutralMinionsKilled	neutralMinionsKilledEnemyJungle + neutralMinionsKilledTeamJungle
trueDamageDealt	trueDamageDealtToChampions + trueDamageDealtToMonsters
physicalDamageDealt	physicalDamageDealtToChampions + physicalDamageDealtToMonsters
magicDamageDealt	magicDamageDealtToChampions + magicDamageDealtToMonsters

total causado a monstros). A Tabela 4.1 demonstra os cálculos realizados para obter essas características.

Para evitar redundância nos dados, removemos também características compostas, ou seja, aquelas características que podem ser obtidas por meio da soma de outras características, tais como *totalDamageDealt*, *totalDamageDealtToChampions*, *totalDamageTaken*, *neutralMinionsKilled*, *trueDamageDealt*, *physicalDamageDealt* e *magicDamageDealt*. A Tabela 4.2 demonstra o cálculo de como se dá a composição de cada característica removida.

4.3 Limpeza de Dados

Para evitar conclusões falsas, realizamos a limpeza de dados para detectar e remover inconsistências no conjunto de desempenho das equipes. Em nossa análise, removemos partidas com as seguintes condições:

- Partidas rendidas: dependendo se uma partida está extremamente desequilibrada, geralmente a equipe que está perdendo pode solicitar a rendição à equipe inimiga a qualquer momento após 20 minutos do início da partida.
- Partidas que contêm jogadores que abandonam: Como as partidas de LoL são jogadas *online*, os jogadores podem sair das partidas a qualquer momento. Alguns motivos in-

cluem perda de conexão, desistência ou outro motivo desconhecido. Esses abandonos podem causar também um desequilíbrio nas partidas.

4.4 Remoção de Características com Baixa Variância

Às vezes, os valores de algumas características têm variância zero ou próxima de zero, o que implica que essas características têm nenhuma ou pouca informação e que, consequentemente, não têm significância suficiente para discriminar as observações de um conjunto de dados em várias condições [41]. Portanto, realizamos uma estatística descritiva no conjunto de desempenho de equipes (Tabela 4.3) para identificar e remover as características de variância zero ou próxima de zero de modo a selecionar as características que mais discriminam em relação à variância. Devido a análise da variância e seleção das características, a dimensionalidade do conjunto de desempenho de equipes ficou reduzido, permitindo que o tempo do agrupamento das equipes (abordado na seção 5.1) fosse aprimorado e uma melhor percepção da estrutura implícita relevante dos dados [42]. A Tabela 4.3 indica com (*) as características removidas do conjunto de dados: *doubleKills*, *inhibitorKills*, *largestMultiKill*, *pentaKills*, *quadraKills*, *sightWardsBoughtInGame* e *tripleKills*.

4.5 Remoção de *Outliers*

A fim de encontrar padrões, adotamos o *K-means clustering* para agrupar equipes similares conforme o desempenho das equipes nas partidas que foram coletadas (mais sobre o agrupamento no Capítulo 5). No entanto, o *K-means clustering* é muito sensível a dados discrepantes (*outliers*) [43], pois como o nome sugere, seu algoritmo é baseado no cálculo das médias entre os grupos para encontrar o grupo mais adequado para cada observação do conjunto de dados. Portanto, antes que fizéssemos o agrupamento dos dados, criamos gráficos *boxplots* a partir das características dos dados para analisar se existiam ou não *outliers* nos dados.

A Figura 4.1 ilustra os gráficos dos *boxplots* das características do conjunto de desempenho das equipes. Como podemos observar, existe uma quantidade muito considerável de *outliers* nos dados. Desse modo, como não temos ideia da causa desse comportamento, de-

Tabela 4.3: Estatística descritiva das características do conjunto de desempenho de equipes.

Característica	Mínimo	Máximo	Médio	Mediana	Variância	Desvio padrão
assists	0	166	46.41	45.0	501.83	22.40
deaths	0	96	31.86	32.0	157.25	12.54
doubleKills*	0	16	3.09	3.0	4.95	2.23
goldEarned	21241	167863	61624.94	61313.0	285885117.17	16908.14
goldSpent	17525	173490	56279.00	55783.0	254935731.04	15966.71
inhibitorKills*	0	14	0.97	0.0	1.48	1.22
killingSprees	0	27	7.34	7.0	11.81	3.44
kills	0	94	31.71	32.0	157.08	12.53
largestCriticalStrike	0	6832	1185.81	1051.0	538830.36	734.05
largestKillingSpree	0	49	14.11	14.0	38.85	6.23
largestMultiKill*	0	15	7.04	7.0	3.05	1.75
magicDamageDealtToChampions	1454	277258	45374.46	41341.0	559898084.93	23662.17
magicDamageDealtToMonsters	2664	1636239	185334.42	165063.0	11104812464.71	105379.37
magicDamageTaken	1455	280380	46026.27	41934.0	571260353.46	23901.05
minionsKilled	142	1988	620.06	606.0	29672.00	172.26
neutralMinionsKilledEnemyJungle	0	179	20.20	17.0	272.66	16.51
neutralMinionsKilledTeamJungle	0	236	74.58	72.0	759.39	27.56
pentaKills*	0	3	0.01	0.0	0.01	0.11
physicalDamageDealtToChampions	2516	209321	46657.02	43445.0	478524140.52	21875.19
physicalDamageDealtToMonsters	25356	1622946	307468.35	286914.5	19065412611.47	138077.56
physicalDamageTaken	12963	276723	75158.66	71446.5	722512844.62	26879.60
quadraKills*	0	5	0.06	0.0	0.07	0.26
sightWardsBoughtInGame*	0	0	0.00	0.0	0.00	0.00
totalHeal	1756	178665	25401.26	22394.0	203814822.76	14276.37
totalTimeCrowdControlDealt	40	44982	2556.19	1974.0	5118253.86	2262.36
totalUnitsHealed	3	149	12.52	10.0	80.57	8.98
towerKills	0	15	4.73	5.0	10.25	3.20
tripleKills*	0	7	0.44	0.0	0.51	0.72
trueDamageDealtToChampions	0	706412	4995.48	3654.0	38146615.15	6176.29
trueDamageTaken	0	717334	5141.91	3822.0	38482885.65	6203.46
visionWardsBoughtInGame	0	43	4.21	4.0	9.08	3.01
wardsKilled	0	173	10.87	8.0	120.48	10.98
wardsPlaced	7	621	62.39	60.0	449.11	21.19

cidimos remover apenas equipes consideradas *outliers* extremos, isto é, removemos aquelas equipes com algum *outlier* em suas características com base em um Intervalo Interquartil do Boxplot $IQR = 3, 5$.

Em seguida, realizamos uma estatística descritiva dos dados novamente e percebemos que algumas características estavam com variância próxima de zero. Após essa análise, as seguintes características com variância próxima de zero foram removidas: *totalUnitsHealed* e *visionWardsBoughtInGame*.

4.6 Transformação de Dados

Para transformar os dados, realizamos as seguintes tarefas: cálculo de métricas e normalização de dados.

4.6.1 Cálculo de métricas

O conjunto de dados X é apenas um reflexo do desempenho das equipes durante toda a partida, ou seja, estatísticas sumarizadas de desempenho, como total de ouro ganho e total de dano sofrido. Como a duração de uma partida não tem um limite específico, ou seja, uma partida somente termina quando uma equipe vencer, é necessário transformar os dados para tornar o desempenho das equipes comparáveis entre si independentemente da duração das partidas. Assim, para lidar com o problema de capturar o desempenho das equipes em diferentes durações de partidas, dividimos o desempenho de cada equipe $x \in X$ pela duração *duration* da partida m em que a equipe participou:

$$\frac{x}{m_{duration}}$$

Dessa forma, como foi discutido na seção 2.2.1, as características do nosso conjunto de dados deixaram de ser simples estatísticas sumarizadas para se tornarem métricas de desempenho mais interpretativas. Por exemplo, as características agora passam a ser número de assistências em abates por minuto (*assists*), número de mortes sofridas por minuto (*deaths*), número de abates duplos seguidos efetuados por minuto (*doubleKills*), e assim por diante.

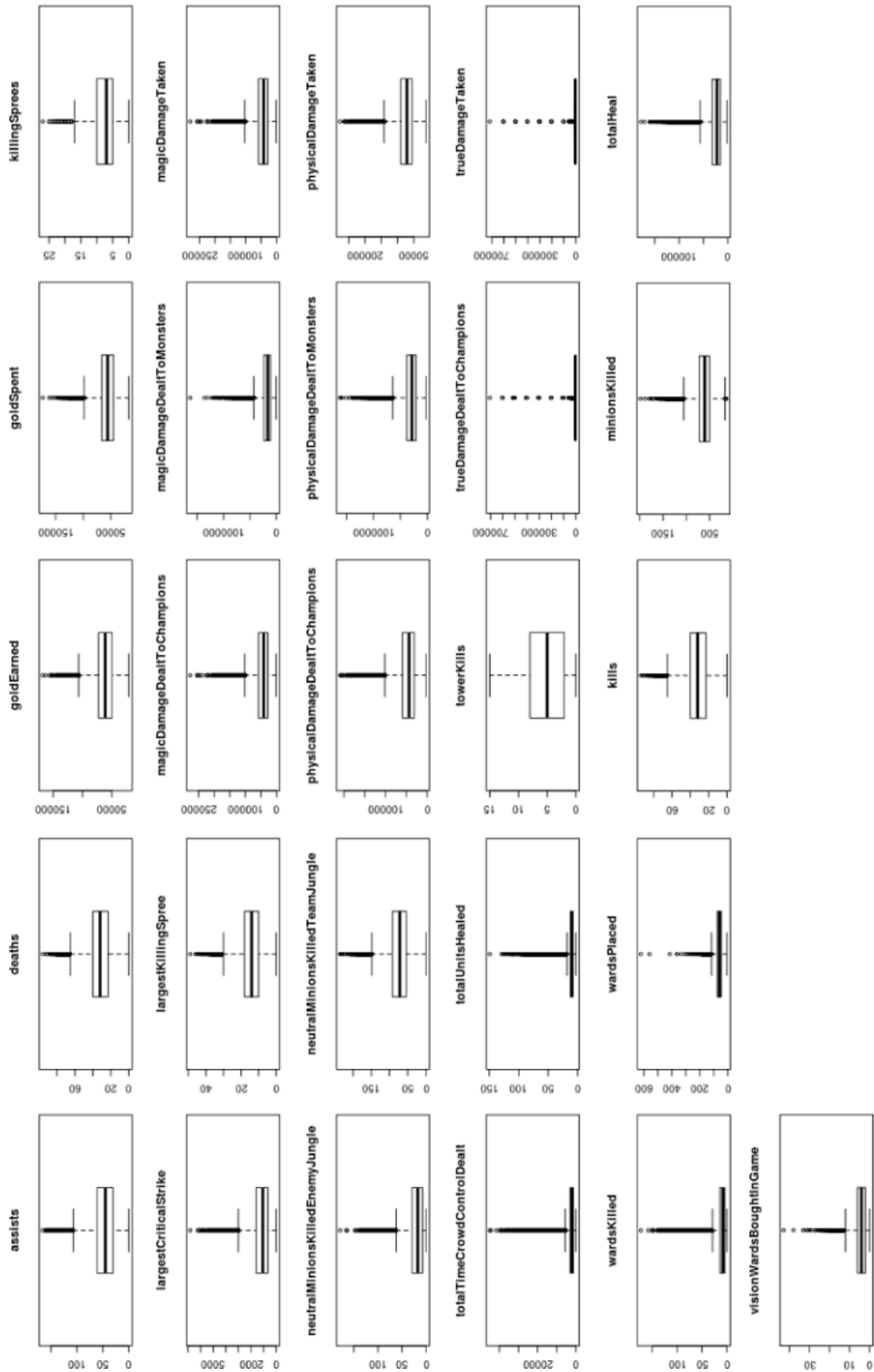


Figura 4.1: Gráfico de *boxplot* para cada característica do conjunto de desempenho de equipes até o momento da etapa de remoção de *outliers*.

4.6.2 Normalização de Dados

Conforme a Figure 4.1, podemos perceber também que o intervalo dos valores variam amplamente entre as características. Essa variação entre as características se dá também mesmo após os dados serem transformados na etapa anterior. Portanto, aplicamos a normalização de dados *min-max* [44] nas características no conjunto de desempenho X para que cada uma contribua proporcionalmente em uma mesma escala e forneça uma maneira fácil de comparar valores na realização da análise de redundância e agrupamento. Para cada característica $X_j \in X$, aplicamos a normalização baseada nos valores *mínimo* e *máximo* para escalar X_j no intervalo r :

$$x'_j = \frac{x_{ji} - \min(X_j)}{r_j} = \frac{x_{ji} - \min(X_j)}{\max(X_j) - \min(X_j)}$$

Após a normalização, as características do conjunto de dados X assumiram valores na faixa $[0, 1]$.

4.7 Análise de Redundância

De modo a evitar ruídos e aprimorar o tempo de agrupamento dos dados, bem como facilitar a análise dos dados pelos usuários finais, removemos as características mais redundantes do conjunto de dados por meio de uma análise de correlação. Com base no teste de correlação de Spearman [45], consideramos que um par de características de alta ou altíssima correlação são redundantes. A Figura 4.2 ilustra o gráfico de correlação entre as $d = 24$ características restantes do conjunto de dados X reduzido e transformado até esta etapa. De acordo com o gráfico, podemos observar que existem várias características redundantes.

Desse modo, dado que $X_j \in X$ é uma característica de X , para cada par de características (X_a, X_b) ; $a, b = \{1, \dots, d\}$; $a \neq b$ com alta ou altíssima correlação $r_{ab} \geq 0.65$, removemos de X a característica que tem a maior correlação no par, ou seja, a característica do par em que a soma de todas as suas correlações com as demais características de X tem o maior valor:

$$\max\left\{\sum_{i=1}^d r_{ai}, \sum_{i=1}^d r_{bi}\right\}; a \neq i; b \neq i$$

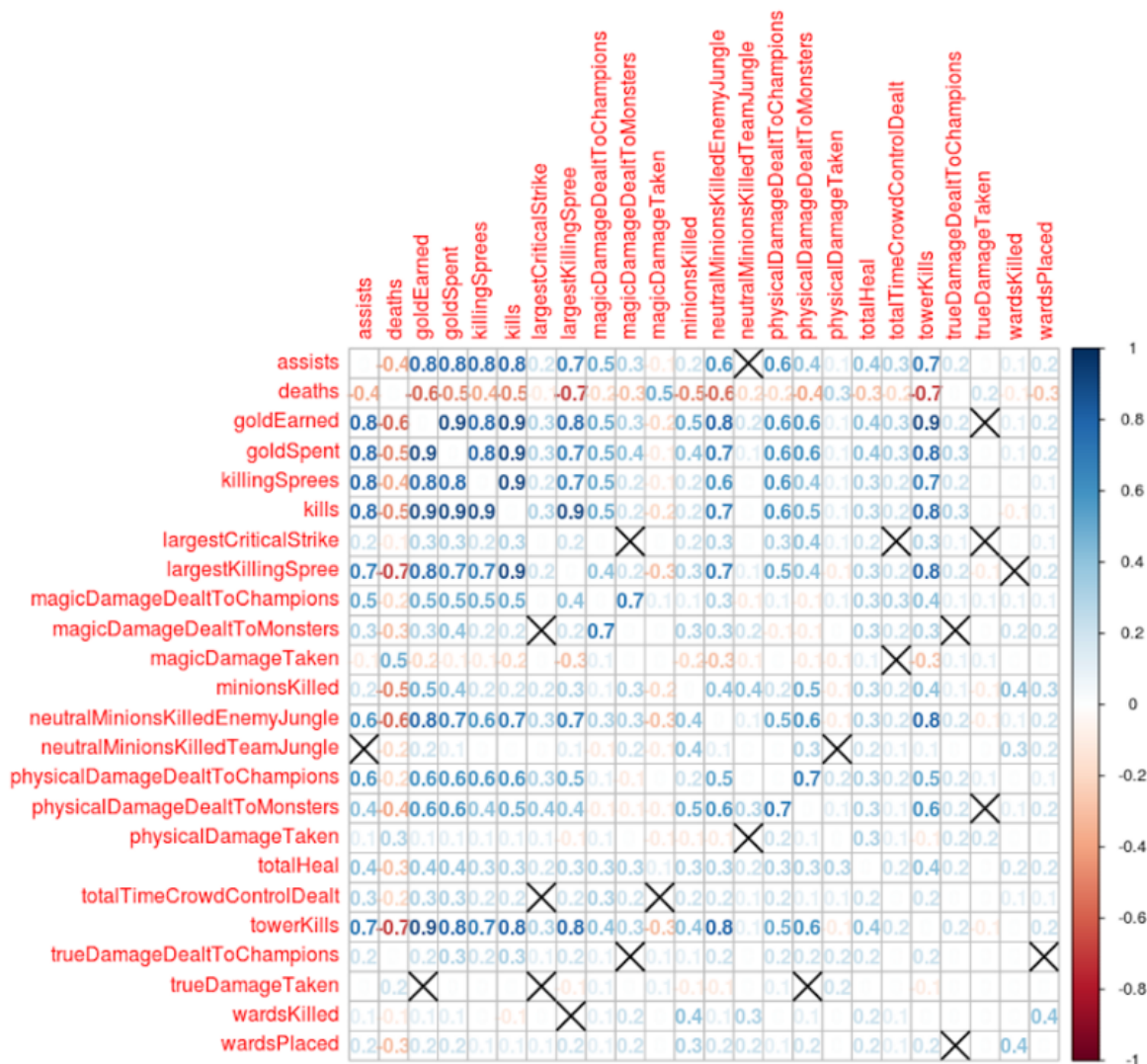


Figura 4.2: Gráfico da correlação das características de desempenho das equipes. Cada célula ou valor no gráfico representa a correlação de um par de características do conjunto de dados X . A escala de cores indica a intensidade das correlações, quanto mais forte for uma cor, mais forte é a correlação das características. As cores em tom azul indicam correlações positivas. As cores em tom vermelho indicam correlações negativas. No gráfico, as correlações com $p\text{-value} > 0,05$ são consideradas insignificantes. Neste caso, os valores do coeficiente de correlação são indicados com um "X".

Após essa análise, removemos as seguintes características: *assists*; *goldEarned*; *goldSpent*; *kills*; *largestKillingSpree*; *magicDamageDealtToChampions*; *physicalDamageDealtToChampions*; *towerKills*.

Capítulo 5

Perfis de Comportamento

O problema considerado equivalente ao problema de identificar grupos de desempenho de equipes de comportamento semelhante. Para abordar esse problema, usamos o método de mineração de dados *K-means clustering* [44] para agrupar equipes semelhantes conforme os seus desempenhos nas partidas. Em seguida, caracterizamos os grupos de desempenho de equipes de modo a colocar em perspectiva suas principais características ou métricas de desempenho e, assim, definir perfis de comportamento.

5.1 Agrupamento de Equipes

O problema de encontrar perfis de desempenho de equipes pode ser considerado equivalente ao problema de identificar grupos de desempenho de equipes de comportamento semelhante. Para abordar esse problema, usamos o método de mineração de dados *K-means clustering* [44]. Dado um conjunto dimensional de observações (x_1, \dots, x_n) , o *K-means clustering* visa particionar as n observações em $k(\leq n)$ grupos distintos, denotados por $C = \{c_1, \dots, c_k\}$, de modo a minimizar a Soma dos Quadrados do Erro Dentro dos Grupos (SQED); ou seja, seu objetivo é encontrar o minimizador C^* da função:

$$\arg \min_C \sum_{i=1}^k \sum_{x_j \in c_i} \|x_j - \mu_i\|^2; \arg \min_C \{SQE(C)\} = \arg \min_C \{SQED\}$$

Onde μ_j é a média das observações em c_i , e indica também o i -ésimo centroide de C .

Conforme Zaki et al. [44], o *K-means* usa uma abordagem iterativa gananciosa (*iterative greedy*) para encontrar um grupo que minimize a função objetivo *SQE* e, como tal, pode

convergir para um grupo ótimo local em vez de um grupo ótimo global.

Em nosso conjunto de dados X usamos o algoritmo de Lloyd [15], uma heurística que consiste em escolher aleatoriamente observações como centroides para os k grupos de C e atribuir iterativamente cada observação $x \in X$ ao centroide mais próximo e depois atualizar os centroides com a média de seus respectivos grupos.

Para encontrar um número ótimo k de grupos usamos um método que encontra o "cotovelo" ou "joelho" da curva de erro. Este método tenta descobrir um número apropriado de grupos analisando a curva de um gráfico gerado a partir de um teste realizado para cada número possível de grupos [46]. No nosso caso, o teste foi baseado na função objetivo $SQED$.

A Figura 5.1 ilustra o gráfico $SQED$ para cada número $k \in K$ possível de grupos, onde $K = \{1, \dots, 120\}$. Como podemos observar, o "joelho" do gráfico não deixa explícito o número ótimo k de grupos, de modo que o número adequado para agrupar o conjunto pode estar entre $k \geq 7$ e $k \leq 10$ grupos distintos.

Então, para escolher o número ótimo de k , analisamos o quanto a variabilidade dos dados diminuiu ao agrupar o desempenho das equipes em relação ao conjunto total dos dados X calculando a proporção da Soma dos Quadrados do Erro Entre os Grupos $SQEE$ e a Soma dos Quadrados do Erro Total $SQE(X)$ para cada $k \in K$ da seguinte forma:

$$SQEE(k) = SQE(X) - SQED(k); PSQE(k) = SQEE(k)/SQE(X)$$

O resultado das proporções para $K = \{1, \dots, 12\}$ foi de $PSQE(K) = \{0.0, 0.413, 0.579, 0.657, 0.716, 0.746, 0.777, 0.796, 0.813, 0.827, 0.836, 0.847\}$. Analisando as proporções, escolhemos $k = 7$ como sendo o número ótimo de grupos dentre os valores de k presentes no "joelho" do gráfico, pois (i) diminui consideravelmente a variabilidade dos dados em 78%, (ii) as proporções PSQE presentes no "joelho" são bem aproximadas, (iii) a variabilidade não diminui tão drasticamente à medida que k é incrementado ao agrupar os dados em $k > 7$ grupos, (iv) $k = 7$ é o menor valor presente no "joelho", o que diminui a complexidade da interpretação dos grupos.

5.2 Caracterização de Perfis

A fim de compreender como os grupos ou perfis encontrados diferem entre si, analisamos:

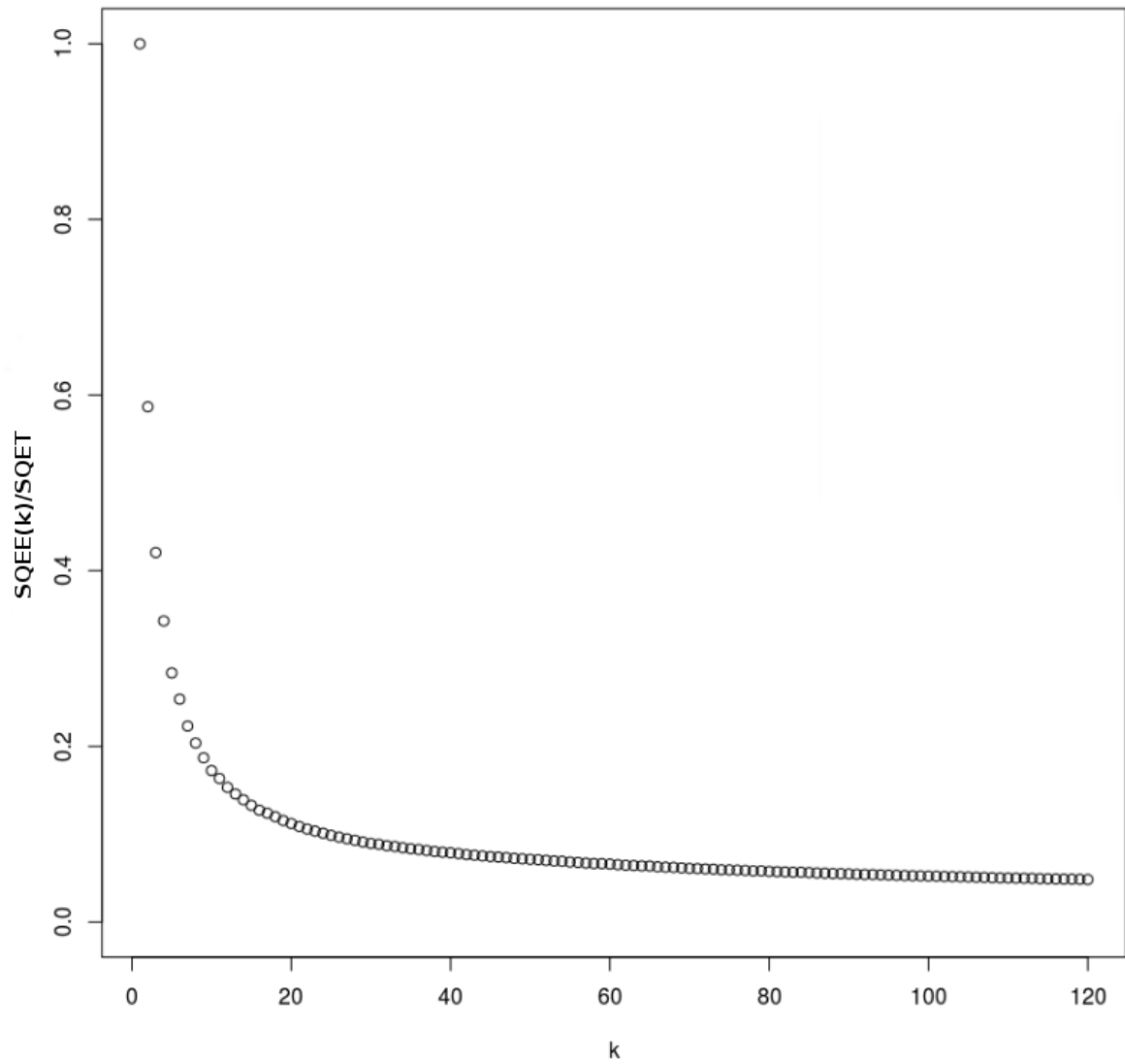


Figura 5.1: Curva de erro K-means. O eixo x indica o conjunto $K = \{1, \dots, 120\}$ de números possíveis de grupos. O eixo y indica os testes $SQEE(k)/SQET$ para $k \in K$

- (i) A quantidade de equipes e a proporção de equipes vencedoras (taxa de vitórias) e a proporção de equipes perdedoras (taxa de derrotas);
- (ii) Os centroides que resumizam as características dos perfis;
- (iii) Em que medida as características têm influência ou relevância nos perfis.

A Figura 5.2 ilustra o gráfico de barras com a proporção de equipes vencedoras (taxa de vitórias) e a proporção de equipes perdedoras (taxa de derrotas) para cada perfil encontrado. Podemos observar pelo gráfico que existe uma diferença bem significativa de taxa de vitórias entre alguns perfis, no entanto, outros apresentam uma taxa de vitórias bem semelhante, como o Perfil 1 (11%) e o Perfil 4 (6%), o Perfil 2 (55%) e o Perfil 5 (53%), o Perfil 3 (86%) e o Perfil 7 (84%).

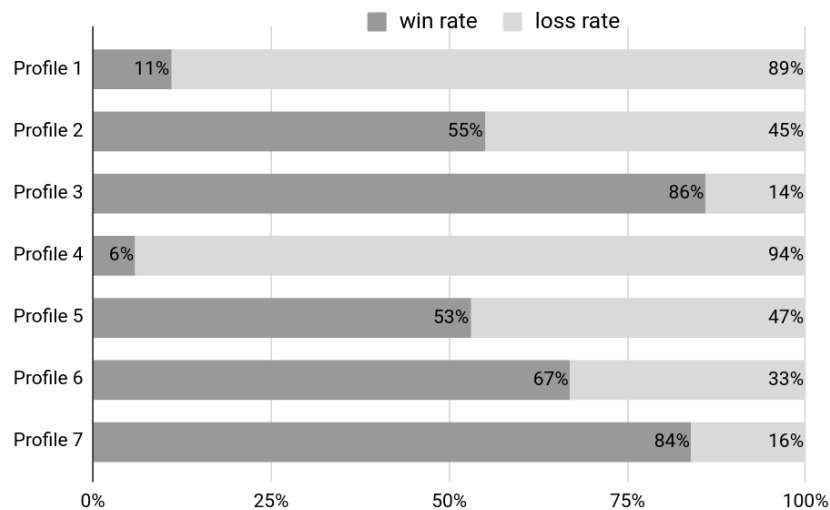


Figura 5.2: Gráfico da proporção de equipes vencedoras (*win rate*) e equipes perdedoras (*loss rate*): Perfil 1 (quantidade de equipes vencedoras: 662, quantidade de equipes perdedoras: 5829), Perfil 2 (vencedoras: 5255, perdedoras: 4273), Perfil 3 (vencedoras: 6357, perdedoras: 1051), Perfil 4 (vencedoras: 502, perdedoras: 7676), Perfil 5 (vencedoras: 4776, perdedoras: 4233), Perfil 6 (vencedoras: 3650, perdedoras: 1785), Perfil 7 (vencedoras: 4485, perdedoras: 840).

A Figura 5.3 ilustra o mapa de calor do ganho de informação aplicado as métricas de desempenho das equipes M (em unidades/minuto, calculadas na Seção 4.6.1 antes da normalização) para indicar como essas métricas são relevantes para cada perfil. Conforme a

figura, podemos observar que, no geral, as características mais importantes são: *deaths*, *killingSprees* e *neutralMinionsKilledEnemyJungle*. Assim como foi observado na análise das taxas de vitórias, alguns perfis também possuem características semelhantes considerando o ganho de informação.

Seja A a matriz de centroides das métricas de desempenho das equipes M por perfil, onde cada linha $a \in A$ indica os centroides das métricas de um perfil, e cada coluna $A_j \in A$ indica os centroides das métricas entre os 7 perfis encontrados. A Tabela 5.1 apresenta a matrix *transposta* dos centroides $t(A)$ e a sumarização desses centroides entre os perfis de modo a denotar como os perfis se diferem em termos de desempenho. Se multiplicarmos as métricas de desempenho M pela duração média das partidas ($mean(M_{duration}) \approx 29$ minutos) podemos ter uma estimativa do desempenho total dos perfis no final de uma partida. Se compararmos as principais métricas de desempenho do pior perfil (Perfil 4) e do melhor perfil (Perfil 3) em relação a taxa de vitórias, podemos verificar que o melhor perfil tem um desempenho de *deaths* 40% inferior, *killingSprees* 54% superior e *neutralMinionsKilledEnemyJungle* 78% superior. O nível elevado de *neutralMinionsKilledEnemyJungle* pode indicar que o Perfil 3 possui um excelente domínio do campo da equipe inimiga e dos *minions* que existem por lá, fazendo com que as equipes desse perfil tenham uma quantidade maior de ouro as custas dos recursos da equipe inimiga, e, assim, comprem melhores itens de defesa e ataque para garantir uma quantidade menor de *deaths* e maior de *killingSprees* por minuto, respectivamente.

A Figura 5.4 ilustra os gráficos de radar dos perfis, onde cada gráfico de radar representa as métricas de desempenho de um perfil, cada eixo representa uma métrica de desempenho e o comprimento do eixo indica qual a pontuação do perfil em uma métrica de desempenho específica. As métricas de desempenho usadas para modelar os gráficos de radar são baseadas nos centroides dos perfis normalizados por característica $normalization(A_j)$, de modo que o comprimento de um eixo seja proporcional em todos os centroides dos perfis e assuma um valor entre $[0, 1]$.

Ao observar os resultados, categorizamos os sete perfis em quatro grandes níveis em termos de taxa de vitória: Fraco, Mais ou Menos, Bom e *Top*. Nos parágrafos seguintes, elaboramos esses resultados ao caracterizar os perfis de desempenho das equipes para cada categoria.

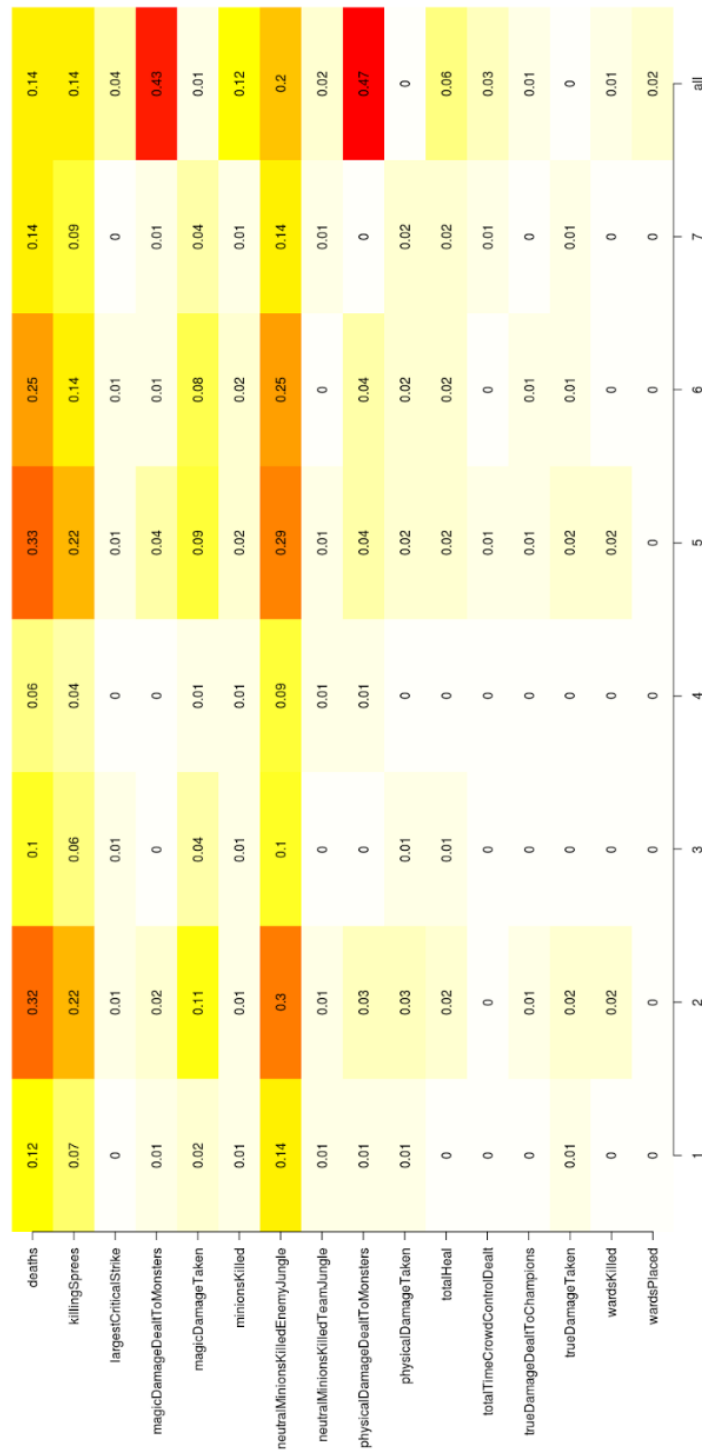


Figura 5.3: Mapa de calor das relevâncias das métricas de desempenho (ou características) das equipes com base no cálculo do ganho de informação. O eixo x representa os perfis. O eixo y representa as características. Quanto mais forte for a cor de uma característica, mais intensa é a relevância dessa característica em relação as outras características para um determinado perfil.

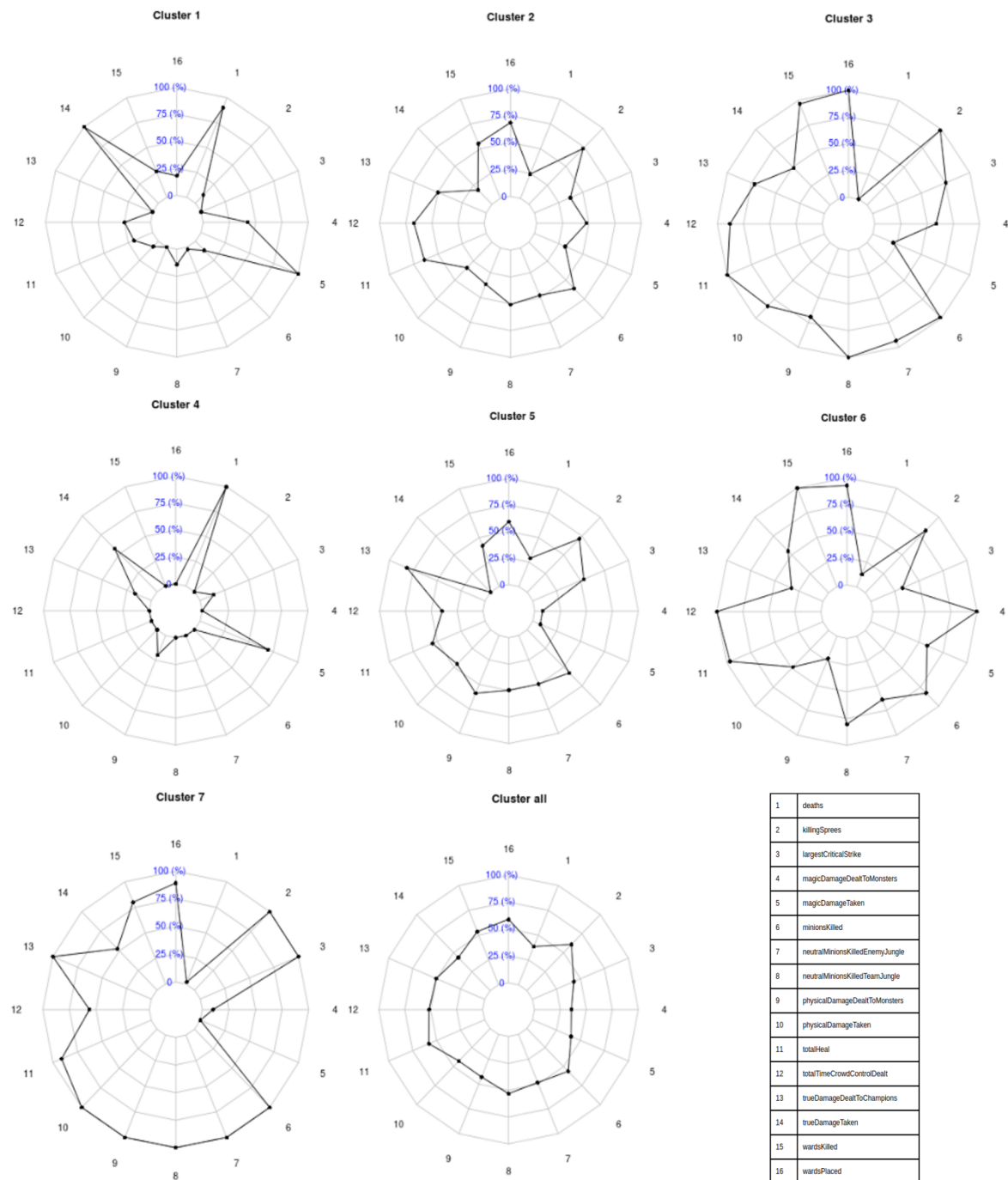


Figura 5.4: Centroides dos perfis normalizados por métrica de desempenho das equipes.

Tabela 5.1: Centroides das características (métricas de desempenho) para cada um dos 7 perfis encontrados. As últimas 5 colunas da tabela resumam as métricas de desempenho entre os perfis: valor mínimo (Min), média (Mean), variância (Var), desvio padrão (Sd), valor máximo (Max).

Característica	Perfil 1	Perfil 2	Perfil 3	Perfil 4	Perfil 5	Perfil 6	Perfil 7	Min	Mean	Var	Sd	Max
deaths	1.17	0.85	0.73	1.22	0.87	0.79	0.74	0.73	0.91	0.04	0.20	1.22
killingSprees	0.14	0.22	0.26	0.12	0.22	0.23	0.26	0.12	0.21	0.00	0.06	0.26
largestCriticalStrike	22.85	31.63	40.80	26.11	35.52	30.65	46.64	22.85	33.46	68.28	8.26	46.64
magicDamageDealtToMonsters	4805.31	5077.61	5621.34	2662.17	3043.93	7716.26	3174.08	2662.17	4585.81	3205475.58	1790.38	7716.26
magicDamageTaken	1187.75	1118.98	1108.49	1157.09	1095.07	1145.94	1087.15	1087.15	1128.64	1325.06	36.40	1187.75
minionsKilled	15.55	17.73	19.39	15.03	17.57	18.69	19.45	15.03	17.63	3.11	1.76	19.45
neutralMinionsKilledEnemyJungle	0.20	0.48	0.77	0.18	0.49	0.58	0.81	0.18	0.50	0.06	0.25	0.81
neutralMinionsKilledTeamJungle	1.93	2.10	2.33	1.87	2.10	2.24	2.33	1.87	2.13	0.03	0.18	2.33
physicalDamageDealtToMonsters	4661.69	7414.61	9875.30	6138.25	9103.80	6361.48	12198.87	4661.69	7964.86	6665807.44	2581.82	12198.87
physicalDamageTaken	1986.43	2014.44	2066.65	1979.37	2026.98	2029.57	2083.44	1979.37	2026.70	1472.64	38.38	2083.44
totalHeal	489.48	603.20	694.18	440.94	580.17	685.86	671.94	440.94	595.11	9861.34	99.30	694.18
totalTimeCrowdControlDealt	51.15	58.48	62.06	46.86	53.69	64.16	56.57	46.86	56.14	37.01	6.08	64.16
trueDamageDealtToChampions	85.38	100.79	107.55	90.51	110.61	95.45	116.25	85.38	100.93	125.44	11.20	116.25
trueDamageTaken	109.38	103.26	105.52	106.13	101.85	105.99	105.82	101.85	105.42	5.68	2.38	109.38
wardsKilled	0.20	0.23	0.27	0.17	0.21	0.27	0.25	0.17	0.23	0.00	0.04	0.27
wardsPlaced	1.62	1.76	1.85	1.56	1.73	1.83	1.82	1.56	1.74	0.01	0.11	1.85

5.2.1 Nível Fraco

O *Nível Fraco* consiste do Perfil 1 e do Perfil 4. Esse nível é distinguido por uma taxa de vitórias muito baixa, isto é $w_r = 10\%$ (Figura 5.2). Existem 10 características relevantes para o Perfil 1 e 7 características relevantes para o Perfil 4 (Figura 5.3). Os mais relevantes ordenados por importância são: *neutralMinionsKilledEnemyJungle*, *deaths*, *killingSprees*. A Tabela 5.2 resume as semelhanças e diferenças entre as pontuações das métricas de desempenho para os perfis desse nível (Figura 5.4). As pontuações que mais se diferenciam entre os perfis são: *magicDamageDealtToMonsters* (moderado para o Perfil 1 e muito baixo para o Perfil 4) e *trueDamageTaken* (muito alto para o Perfil 1 e o moderado para o Perfil 4).

5.2.2 Nível Mais ou Menos

O *Nível Mais ou Menos* consiste do Perfil 2 e do Perfil 5. Esse nível é distinguido por uma taxa de vitórias moderada, isto é $w_r = 55\%$ (Figura 5.2). Existem 14 características relevantes para o Perfil 2 e 15 características relevantes para o Perfil 5 (Figura 5.3). Os mais relevantes ordenados por importância são: *deaths*, *neutralMinionsKilledEnemyJungle*, *killingSprees*, *magicDamageTaken*. A Tabela 5.3 resume as semelhanças e diferenças entre as

Tabela 5.2: Pontuações de desempenho para o Perfil 1 e Perfil 4.

Pontuação	Perfil 1 e Perfil 4	Perfil 1	Perfil 4
Very low	killingsprees (2), largestCriticalStrike (3), minionsKilled (6), neutralMinionsKilledEnemyJungle (7), neutralMinionsKilledTeamJungle (8), physicalDamageTaken (10)	physicalDamageDealtToMonsters (9), trueDamageDealtToChampions (13)	magicDamageDealtToMonsters (4), totalHeal (11), totalTimeCrowdControlDealt (12), wardsKilled (15), wardsPlaced (16)
Low		totalHeal (11), totalTimeCrowdControlDealt (12), wardsKilled (15), wardsPlaced (16)	physicalDamageDealtToMonsters (9), trueDamageDealtToChampions (13)
Middle		magicDamageDealtToMonsters (4)	trueDamageTaken (14)
High			magicDamageTaken (5)
Very High	deaths (1)	magicDamageTaken (5), trueDamageTaken (14)	

pontuações das métricas de desempenho para os perfis desse nível (Figura 5.4). A pontuação que mais se diferencia entre os perfis é: *magicDamageDealtToMonsters* (moderado para o Perfil 2 e muito baixo para o Perfil 5).

5.2.3 Nível Bom

O *Nível Bom* consiste apenas do Perfil 6. Esse nível é distinguido por uma taxa de vitórias alta, isto é $w_r = 67\%$ (Figura 5.2). Existem 12 características relevantes para o Perfil 6 (Figura 5.3). Os mais relevantes ordenados por importância são: *deaths*, *neutralMinionsKilledEnemyJungle*, *killingsprees*, *magicDamageTaken*. A Tabela 5.4 resume as pontuações das métricas de desempenho do perfil desse nível.

5.2.4 Nível Top

O *Nível Top* consiste do Perfil 3 e Perfil 7. Esse nível é distinguido por uma taxa de vitórias muito alta, isto é $w_r = 85\%$ (Figura 5.2). Existem 8 características relevantes para o Perfil 3 e 11 características relevantes para o Perfil 7 (Figura 5.3). Os mais relevantes ordenados por importância são: *deaths*, *neutralMinionsKilledEnemyJungle*, *killingsprees*. A Tabela 5.5 resume as semelhanças e diferenças entre as pontuações das métricas de desempenho para os perfis desse nível (Figura 5.4). As pontuações mais diferenciadas entre os perfis são:

Tabela 5.3: Pontuações de desempenho para o Perfil 2 e Perfil 5.

Pontuação	Perfil 2 e Perfil 5	Perfil 2	Perfil 5
Very low			magicDamageDealtToMonsters (4), magicDamageTaken (5) and trueDamageTaken (14)
Low	deaths (1)	magicDamageTaken (5), physicalDamageTaken and trueDamageTaken (14)	
Middle	largestCriticalStrike (3), minionsKilled (6), neutralMinionsKilledEnemyJungle (7), neutralMinionsKilledTeamJungle (8), physicalDamageDealtToMonsters (9), wardsKilled (15)	magicDamageDealtToMonsters (4) and trueDamageDealtToChampions (13)	physicalDamageTaken (10), totalHeal (11) and totalTimeCrowdControlDealt (12)
High	killingsPrees (2) and wardsPlaced (16)	totalHeal (11) and totalTimeCrowdControlDealt (12)	trueDamageDealtToChampions (13)
Very High			

Tabela 5.4: Pontuações de desempenho para o Perfil 6.

Pontuação	Perfil 6
Very low	
Low	deaths (1), largestCriticalStrike (3), physicalDamageDealtToMonsters (9), trueDamageDealtToChampions (13)
Middle	magicDamageTaken (5), physicalDamageTaken, trueDamageTaken (14)
High	killingsPrees (2), minionsKilled (6), neutralMinionsKilledEnemyJungle (7), neutralMinionsKilledTeamJungle (8)
Very High	totalHeal (11), totalTimeCrowdControlDealt (12), wardsKilled (15), wardsPlaced (16)

Tabela 5.5: Pontuações de desempenho para o Perfil 2 e Perfil 5.

Nível de pontuação	Perfil 3 e Perfil 7	Perfil 3	Perfil 7
Very low	deaths (1)		magicDamageDealtToMonsters (4), magicDamageTaken (5)
Low		magicDamageTaken (5)	
Middle	trueDamageTaken (14)	magicDamageDealtToMonsters (4)	totalTimeCrowdControlDealt (12)
High		largestCriticalStrike (3), physicalDamageDealtToMonsters (9), physicalDamageTaken (10), trueDamageDealtToChampions (13)	wardsKilled (15)
Very High	killingsPrees (2), minionsKilled (6), neutralMinionsKilledEnemyJungle (7), neutralMinionsKilledTeamJungle (8), totalHeal (11), wardsPlaced (16)	totalTimeCrowdControlDealt (12), wardsKilled (15)	largestCriticalStrike (3), physicalDamageDealtToMonsters (9), physicalDamageTaken, trueDamageDealtToChampions (13)

magicDamageDealtToMonsters (moderado para o Perfil 3 e muito baixo para o Perfil 7) e *totalTimeCrowdControlDealt* (muito alto para o Perfil 3 e o moderado para o Perfil 7).

Capítulo 6

Conclusão

Neste estudo, buscamos coletar informações de equipes em partidas de LoL para encontrar padrões úteis de comportamento de equipes baseando-se nas métricas de desempenho dessas equipes e caracterizar perfis de comportamento de equipes bem-sucedidas e malsucedidas usando esses padrões.

Portanto, propomos uma abordagem que passa por várias tarefas de preparação de dados para computar métricas de desempenho de equipes a partir de partidas de LoL, usa o método de agrupamento de dados *K-means Clustering* para encontrar grupos (padrões) nas métricas de desempenho e, por fim, analisa diferentes aspectos dos grupos de desempenho de equipes para caracterizar perfis de comportamento de sucesso e fracasso.

Nossos resultados demonstram que as equipes das partidas coletadas compartilham diversas semelhanças e diferenças. Identificamos sete perfis de equipes que colocam em perspectiva tais semelhanças e diferenças de acordo com um conjunto de estatísticas sumarizadas e transformadas de desempenho de jogadores, tais como, quantidade de mortes, total de dano físico causado, total de cura efetuada, dentre outras. Calculamos, também, para cada perfil, a taxa de vitórias e a taxa de derrotas conforme a proporção de equipes vencedoras e derrotadas. Assim, categorizamos esses sete perfis em quatro grandes níveis em termos de taxa de vitórias da seguinte forma: Fraco, Mais ou Menos, Bom e Top. Em relação à proporção da quantidade de equipes nos perfis, 28% das equipes são do nível Fraco, 36% do nível Mais ou Menos, 11% do nível Bom e 25% do nível Top. Quanto à relevância das características, as que parecem ter mais influência no comportamento dos perfis são *deaths*, *killSprees* e *neutralMinionsKilledEnemyJungle*.

Na categoria de *Nível Bom*, encontramos equipes que normalmente apresentam níveis altos de *killSprees*, *totalHeal* e *wardsPlaced*. Com base nos dados, podemos conceber que essas equipes usam uma estratégia baseada em maximizar a quantidade de minions abatidos por minuto (*minionsKilled*, *neutralMinionsKilled*), obtendo vantagem sobre o ouro - que se converte em itens - e pontos de experiência e, assim, aumentando a probabilidade de *kills* ao combater heróis inimigos. A categoria de *Nível Fraco* pode ser interpretado exatamente o oposto do *Nível Bom*; equipes encontradas nesta categoria apresentam um nível muito baixo de *minionsKilled*, sendo incapaz de obter itens e tornam-se vulnerável ao ataque dos oponentes. Equipes do Nível Mais ou Menos apresentam baixos níveis de *magicDamageTaken* e *trueDamageTaken*, o que ocasiona, assim, um nível muito baixo de *deaths*; isso pode indicar que as equipes dessa categoria se concentram muito em estratégias de defesa. Equipes do Nível Bom apresentam um nível muito alto de *magicDamageDeltToMonsters* e *totalTimeCrowdControlDelt*, o que pode indicar que a principal estratégia dessas equipes são ataques mágicos a distância após o uso de habilidades de controle em área sobre as equipes inimigas, o que garante um nível muito alto de *killingSprees*; no entanto, essas equipes possuem um nível muito baixo de *physicalDamageDealtToMonsters*, impedindo uma taxa maior de vitória.

As descobertas do nosso estudo sugerem alguns elementos muito concretos que podem ser usados para enriquecer as estratégias de equipes inexperientes de LoL. Ao analisar o desempenho dos perfis de sucesso, uma equipe inexperiente poderia avaliar suas métricas de desempenho nas partidas para apoiar suas decisões conforme o perfil em que mais se adequa.

6.1 Trabalhos Futuros

Para o futuro, poderíamos ampliar nosso trabalho nas seguintes direções:

- (i) Investigar mais correlações de características extraindo métricas de desempenho de equipes com base em outros tipos de sumarização além da soma que foi utilizada a partir do desempenho dos jogadores; em estatísticas parciais que dependam da duração de cada partida; em estatísticas que considerem a dependência existente no desempenho das equipes durante cada partida; em dados categóricos;

- (ii) Construir e comparar modelos preditivos que possam classificar o perfil dos jogadores ou equipes e prever o resultado das partidas com base no perfil dos jogadores ou equipes;
- (iii) Entender as semelhanças entre o comportamento das equipes em esportes do mundo real e virtual.

Bibliografia

- [1] NewZoo. 2016 global games market report, 2016.
- [2] IFPI. Global music report 2017, 2017.
- [3] MPAA. 2016 theatrical market statistics report, 2017.
- [4] Forbes. New report details how esports is an effective engagement and marketing tool, 2017.
- [5] NewZoo. 2017 global esports market report, 2017.
- [6] Superdata. Worldwide digital games market: 2015 total, 2016.
- [7] Riot Games. Our games, 2017.
- [8] League of Legends. Game info, 2017.
- [9] Anders Drachen, Matthew Yancey, John Maguire, Derrek Chu, Iris Yuhui Wang, Tobias Mahlmann, Matthias Schubert, and Diego Klabajan. Skill-based differences in spatio-temporal team behaviour in defence of the ancients 2 (dota 2). In *Games media entertainment (GEM), 2014 IEEE*. IEEE, 2014.
- [10] François Rioult, Jean-Philippe Métivier, Boris Helleu, Nicolas Scelles, and Christophe Durand. Mining tracks of competitive video games. *AASRI Procedia*, 8:82–87, 2014.
- [11] Magy Seif El-Nasr, Anders Drachen, and Alessandro Canossa. *Game analytics*. Springer, 2016.
- [12] Charu C Aggarwal. *Data mining: the textbook*. Springer, 2015.

-
- [13] Ana Carolina Lorena and André CPLF de Carvalho. Uma introdução às support vector machines. *Revista de Informática Teórica e Aplicada*, 14(2):43–67, 2007.
- [14] R Sathya and Annamma Abraham. Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*, 2(2):34–38, 2013.
- [15] Hao Yi Ong, Sunil Deolalikar, and Mark Peng. Player behavior and optimal team composition for online multiplayer games. *arXiv preprint arXiv:1503.02230*, 2015.
- [16] Victor do Nascimento Silva. Desenvolvimento de agentes inteligentes para jogos moba. 2016.
- [17] Simon Ferrari. From generative to conventional play: Moba and league of legends. In *DiGRA Conference*, 2013.
- [18] M Minotti. The history of mobas: from mod to sensation.
- [19] Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and Rudiger Wirth. Crisp-dm 1.0 step-by-step data mining guide. 2000.
- [20] Kevin Conley and Daniel Perry. How does he saw me? a recommendation engine for picking heroes in dota 2. Technical report, 2013.
- [21] K Kalyanaraman. To win or not to win? a prediction model to determine the outcome of a dota2 match. Technical report, Technical report, University of California San Diego, 2014.
- [22] Nicholas Kinkade, L Jolla, and K Lim. Dota 2 win prediction. Technical report, tech. rep., University of California San Diego, 2015.
- [23] Filip Johansson and Jesper Wikström. Result prediction by mining replays in dota 2, 2015.
- [24] Matthias Schubert, Anders Drachen, and Tobias Mahlmann. Esports analytics through encounter detection. In *Proceedings of the MIT Sloan Sports Analytics Conference*, 2016.

- [25] Robert Edge. Predicting player churn in multiplayer games using goal-weighted empowerment. Technical report, Technical Report 13-024, 2013.
- [26] Pu Yang, Brent E Harrison, and David L Roberts. Identifying patterns in combat that are predictive of success in moba games. In *FDG*, 2014.
- [27] Christoph Eggert, Marc Herrlich, Jan Smeddinck, and Rainer Malaka. Classification of player roles in the team-based multi-player game dota 2. In *International Conference on Entertainment Computing*, pages 112–125. Springer, 2015.
- [28] Young Bin Kim, Shin Jin Kang, Sang Hyeok Lee, Jang Young Jung, Hyeong Ryeol Kam, Jung Lee, Young Sun Kim, Joonsoo Lee, and Chang Hun Kim. Efficiently detecting outlying behavior in video-game players. *PeerJ*, 3:e1502, 2015.
- [29] Olivier Cavadenti, Victor Codocedo, Jean-François Boulicaut, and Mehdi Kaytoue. What did i do wrong in my moba game? mining patterns discriminating deviant behaviours. In *Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on*, pages 662–671. IEEE, 2016.
- [30] Jae Youn Shim, Tae Hyun Kim, and Seong Whan Kim. Decision support of bad player identification in moba games using pagerank based evidence accumulation and normal distribution based confidence interval. *International Journal of Multimedia & Ubiquitous Engineering*, 9(8):13–16, 2014.
- [31] Haewoon Kwak and Jeremy Blackburn. Linguistic analysis of toxic behavior in an online video game. In *International Conference on Social Informatics*, pages 209–217. Springer, 2014.
- [32] Jeremy Blackburn and Haewoon Kwak. Stfu noob!: Predicting crowdsourced decisions on toxic behavior in online games. In *Proceedings of the 23rd International Conference on World Wide Web, WWW '14*, pages 877–888. ACM, 2014.
- [33] Kenneth B Shores, Yilin He, Kristina L Swanenburg, Robert Kraut, and John Riedl. The identification of deviance and its impact on retention in a multiplayer game. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 1356–1365. ACM, 2014.

- [34] Nataliia Pobiedina, Julia Neidhardt, Maria del Carmen Calatrava Moreno, and Hannes Werthner. Ranking factors of team success. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1185–1194. ACM, 2013.
- [35] Hyunsoo Park and Kyung-Joong Kim. Social network analysis of high-level players in multiplayer online battle arena game. In *International Conference on Social Informatics*, pages 223–226. Springer, 2014.
- [36] Alexandra Buchan and Jacqui Taylor. A qualitative exploration of factors affecting group cohesion and team play in multiplayer online battle arenas (mobas). *The Computer Games Journal*, 5(1-2):65–89, 2016.
- [37] Mark Claypool, Jonathan Decelle, Gabriel Hall, and Lindsay O’Donnell. Surrender at 20? matchmaking in league of legends. In *Games Entertainment Media Conference (GEM), 2015 IEEE*, pages 1–4. IEEE, 2015.
- [38] Julia Neidhardt, Yun Huang, and Noshir Contractor. Team vs. team: Success factors in a multiplayer online battle arena game. In *Academy of Management Proceedings*, volume 2015, page 18725. Academy of Management, 2015.
- [39] Alex Leavitt, Brian C Keegan, and Joshua Clark. Ping to win?: Non-verbal communication and team performance in competitive online multiplayer games. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4337–4350. ACM, 2016.
- [40] Young Ji Kim, David Engel, Anita Williams Woolley, Jeffrey Yu-Ting Lin, Naomi McArthur, and Thomas W Malone. What makes a strong team?: Using collective intelligence to predict team performance in league of legends. In *CSCW*, pages 2316–2329, 2017.
- [41] Zhi Yan Wang Qing Qiang Wu, Yi Dong Wang and Kun Hong Liu. Unsupervised feature selection algorithm based on spectral clustering and analysis variance. In *Future Information Engineering and Manufacturing Science: Proceedings of the 2014 International Conference on Future Information Engineering and Manufacturing Science*. CRC Press, 2015.

-
- [42] Christos Boutsidis, Petros Drineas, and Michael W Mahoney. Unsupervised feature selection for the k -means clustering problem. In *Advances in Neural Information Processing Systems*, pages 153–161, 2009.
 - [43] Guojun Gan and Michael Kwok-Po Ng. k -means clustering with outlier removal. *Pattern Recognition Letters*, 90:8–14, 2017.
 - [44] Mohammed J Zaki, Wagner Meira Jr, and Wagner Meira. *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press, 2014.
 - [45] Chengwei Xiao, Jiaqi Ye, Rui Máximo Esteves, and Chunming Rong. Using spearman’s correlation coefficients for exploratory data analysis on big dataset. *Concurrency and Computation: Practice and Experience*, 2015.
 - [46] Trupti M Kodinariya and Prashant R Makwana. Review on determining number of cluster in k -means clustering. *International Journal*, 1(6):90–95, 2013.