



Trabalho de Conclusão de Curso

**Análise Preditiva em Partidas de League of  
Legends: Aplicação de Regressão Logística para  
Prever a Vitória por Jogador**

Vinícius Farias Zaltron

24 de fevereiro de 2024

Vinícius Farias Zaltron

**Análise Preditiva em Partidas de League of Legends:  
Aplicação de Regressão Logística para Prever a Vitória por  
Jogador**

Trabalho de Conclusão apresentado à comissão de Graduação do Departamento de Estatística da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para obtenção do título de Bacharel em Estatística.

Orientador: Prof. Dr. João Henrique Ferreira Flores

Porto Alegre  
Fevereiro de 2024

Vinícius Farias Zaltron

**Análise Preditiva em Partidas de League of Legends:  
Aplicação de Regressão Logística para Prever a Vitória por  
Jogador**

Este Trabalho foi julgado adequado para obtenção dos créditos da disciplina Trabalho de Conclusão de Curso em Estatística e aprovado em sua forma final pela Orientador e pela Banca Examinadora.

Orientador: \_\_\_\_\_  
Prof. Dr. João Henrique Ferreira Flores, UFRGS  
Doutor pela Universidade Federal do Rio Grande do Sul - Porto Alegre, RS

Banca Examinadora:

Profa. Dra. Márcia Helena Barbian, UFRGS  
Doutora pela Universidade Federal de Minas Gerais - Belo Horizonte, MG

Profa. Dra. Stela Maris de Jesus Castro, UFRGS  
Doutora pela Universidade Federal do Rio Grande do Sul - Porto Alegre, RS

Porto Alegre  
Fevereiro de 2024

*“Já atingi meu limite inúmeras vezes. Mesmo assim, sigo em frente.”*  
(Atreus, League of Legends)

# Agradecimentos

Gostaria de agradecer a todos que me acompanharam nesta longa jornada do ensino superior. Refiro-me não apenas àqueles que participaram ativamente da minha vida acadêmica durante estes cinco anos de curso, mas também a todos que demonstraram apoio e companheirismo.

Em especial, quero agradecer aos meus pais Giovani e Luciane por sempre apoiarem minhas decisões e por me criarem com um amor incondicional que jamais poderia contestar. Agradeço também por sempre me guiarem e auxiliarem nas escolhas da vida.

Expresso minha gratidão aos meus "irmãos de outras mães" que estiveram sempre ao meu lado, nesta e em outras jornadas da vida. Obrigado Artur Meluci, Arthur Garin, Guilherme Gomes, Jean Ribeiro e Raphael Figueredo por todo o companheirismo, sinceridade e amizade. Jamais poderia pedir por amizades melhores.

Agradeço aos incríveis amigos que a UFRGS me presenteou: Francisco Pires, Guilherme Elias Doering, Lorenzo Copetti e Vitor Hugo Cunha. Sem vocês, a vida acadêmica não teria sido tão divertida.

Quero manifestar minha gratidão ao professor João Flores pela orientação e paciência no desenvolvimento deste trabalho. Muito obrigado por me guiar nesta fase final do curso.

Por fim, agradeço a todos os parentes, amigos, colegas e professores que me acompanharam não só na graduação, mas em todas etapas de minha vida. Vocês todos contribuíram para quem sou hoje. Muito obrigado.

# Resumo

League of Legends é, atualmente, um dos jogos eletrônicos mais populares em escala global. Portanto, possui grande relevância para a cultura e economia do cenário competitivo de jogos eletrônicos, os eSports. Este trabalho tem como objetivo aplicar regressão logística, bem como técnicas de seleção de variáveis e aprendizado de máquina, a fim de identificar variáveis relevantes que impactam diretamente na probabilidade de vitória. A fim de comparar modelos, para a seleção de variáveis foram utilizadas dois métodos distintos: regressão por *stepwise* e um método desenvolvido neste trabalho, nomeado de Método de  $R^2$  de Regressão Simples. Os modelos logísticos foram ajustados a dois bancos de dados com observações distintas, referentes a jogadores de níveis de ranque Desafiante e Ouro IV. A seleção destes ranques em específico se deve ao interesse de comparar, entre dois ranques com níveis de habilidade completamente distintos, as variáveis selecionadas pelos modelos ajustados. Estes conjuntos de dados possuem, respectivamente 5.600 e 6.000 observações. Adicionalmente, os bancos foram segmentados, cada um, em banco de treinamento e teste, com proporções de 70% e 30% do banco original, respectivamente. Visto que o primeiro consiste da elite dos jogadores, enquanto o segundo é o ponto de corte de 50% dos jogadores ativos em partidas ranqueadas. Adicionalmente, os modelos treinados em seus respectivos ranques, também foram testados para o outro ranque, a fim de testar sua generalização. Concluiu-se que, para regressão logística, nos dados estudados nesse trabalho, o método de seleção de variáveis via regressão por *stepwise*, possui um melhor desempenho se comparada à outra técnica que foi desenvolvida. Contudo, o processamento computacional do método de seleção de variáveis via  $R^2$  de Regressão Simples é muito mais rápido e atinge bons resultados. Por fim, obteve-se resultados preditivos melhores que os esperados, com modelos que atingem até 85% de acurácia no banco de teste do ranque Desafiante e 82,89% de acurácia no ranque Ouro IV.

**Palavras-Chave:** League of Legends, vitória, análise preditiva, regressão logística, seleção de variáveis, aprendizado de máquina.

# Abstract

League of Legends is, currently, one of the most popular electronic games on a global scale. Therefore, it holds significant relevance to the culture and economy of the competitive electronic gaming scene, the eSports. This paper aims to apply logistic regression, as well as variable selection techniques and machine learning techniques, in order to identify relevant variables that directly impact the probability of victory. In order to compare models, two different methods were used for variable selection: *stepwise* regression and a method developed in this work, named Simple Regression  $R^2$  Method. The logistic models were fitted to two databases with distinct observations, pertaining to players of Challenger and Gold IV rank levels. The selection of these specific ranks is due to the interest in comparing, between two ranks with completely different skill levels, the variables selected by the adjusted models. These datasets contain, respectively, 5,600 and 6,000 observations. Additionally, the databases were segmented, each into training and testing datasets, with proportions of 70% and 30% of the original dataset, respectively. Considering that the first consists of the top players, while the second is the cut-off point for 50% of the active players in ranked matches. Additionally, the models trained in their respective ranks were also tested for the other rank, in order to test their generalization. It was concluded that, for logistic regression, in the data studied in this work, the variable selection method by *stepwise* regression has a better performance compared to the other technique that was developed. However, the computational processing of the variable selection method by simple regression  $R^2$  is much faster and achieves good results. Finally, better than expected predictive results were obtained, with models that achieve up to 85% accuracy in the test dataset of the Challenger rank and 82.89% accuracy in the Gold IV rank.

**Keywords:** League of Legends, victory, predictive analysis, logistic regression, variable selection, machine learning.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>12</b>
1.1	Contexto, tema e delimitação	12
1.2	Problematização	13
1.3	Questões de pesquisa	13
1.4	Objetivo principal	13
1.5	Objetivos específicos	13
1.6	Hipótese de pesquisa	14
1.7	Metodologia de pesquisa	14
<b>2</b>	<b>League of Legends e banco de dados</b>	<b>16</b>
2.1	Comentários iniciais	16
2.2	Elementos do jogo	16
2.2.1	Mapa	16
2.2.2	Estruturas	18
2.2.3	Tropas	19
2.2.4	Objetivos neutros	20
2.2.5	Ouro	21
2.2.6	Atributos, experiência e itens	22
2.2.7	Campeões	23
2.2.8	Atualizações	23
2.3	Sistema de ranqueamento	23
2.3.1	Ranques e distribuição de jogadores	24
2.3.2	Pontos de liga e progressão de ranque	25
2.3.3	Temporadas ranqueadas	25
2.3.4	Cenário competitivo	26
2.4	Banco de dados	27
2.4.1	Fonte de dados	27
2.4.2	Obtenção dos dados	28
<b>3</b>	<b>Referencial teórico</b>	<b>30</b>
3.1	Regressão Linear e Modelagem	30
3.1.1	Tipos de Regressão	31
3.1.2	Regressão Logística	33
3.2	Seleção de Variáveis	34
3.2.1	Colinearidade	34
3.2.2	Regressão por <i>stepwise</i>	36
3.2.3	$R^2$ de Regressão Simples	36

3.2.4	Outros métodos . . . . .	37
<b>3.3</b>	<b>Aprendizado de Máquina . . . . .</b>	<b>38</b>
<b>4</b>	<b>Resultados . . . . .</b>	<b>41</b>
<b>4.1</b>	<b>Pré-processamento dos Dados . . . . .</b>	<b>41</b>
4.1.1	Filtragem dos Dados . . . . .	41
4.1.2	Segmentação dos Dados . . . . .	44
<b>4.2</b>	<b>Modelagem . . . . .</b>	<b>44</b>
4.2.1	Seleção de Variáveis . . . . .	44
4.2.2	Ajuste de Modelos . . . . .	45
4.2.3	Comparação de Modelos: Treinamento . . . . .	45
<b>4.3</b>	<b>Predição . . . . .</b>	<b>48</b>
<b>4.4</b>	<b>Comparação entre Modelos . . . . .</b>	<b>51</b>
<b>5</b>	<b>Conclusão . . . . .</b>	<b>53</b>
	<b>Referências Bibliográficas . . . . .</b>	<b>54</b>

# Lista de Figuras

Figura 2.1: Mapa de League of Legends com legenda. . . . .	17
Figura 2.2: Ondas de tropas da equipe azul. . . . .	19
Figura 2.3: Tipos de tropas. . . . .	20
Figura 2.4: Ranques de League of Legends. . . . .	24
Figura 2.5: Distribuição de jogadores por ranque na fila ranqueada Solo/duo em 2023. . . . .	24
Figura 2.6: SK Telecom T1 conquista seu tricampeonato mundial de LoL. . . . .	27
Figura 4.1: Histogramas de assistências ( <i>assists</i> ) para os ranques Desafiante (esquerda) e Ouro IV (direita). . . . .	43
Figura 4.2: Histogramas de mortes ( <i>deaths</i> ) para os ranques Desafiante (esquerda) e Ouro IV (direita). . . . .	44

## Lista de Tabelas

Tabela 3.1: Magnitude de efeitos de acordo com valores absolutos de coeficientes de correlação. . . . .	35
Tabela 3.2: Exemplo de matriz de correlação. . . . .	35
Tabela 3.3: Matriz de Confusão para Modelos de Classificação . . . . .	39
Tabela 4.1: Variáveis quantitativas do banco de dados do ranque Desafiante. . . . .	42
Tabela 4.2: Variáveis binárias do banco de dados do ranque Desafiante. . . . .	42
Tabela 4.3: Variáveis quantitativas do banco de dados do ranque Ouro IV. . . . .	43
Tabela 4.4: Variáveis binárias do banco de dados do ranque Ouro IV. . . . .	43
Tabela 4.5: Resultados do modelo logístico treinado no ranque Desafiante, com seleção de variáveis via $R^2$ de Regressão Simples. . . . .	45
Tabela 4.6: Resultados do modelo logístico treinado no ranque Ouro IV, com seleção de variáveis via $R^2$ de Regressão Simples. . . . .	46
Tabela 4.7: Resultados do modelo logístico treinado no ranque Desafiante, com seleção de variáveis via regressão por <i>stepwise</i> . . . . .	46
Tabela 4.8: Resultados do modelo logístico treinado no ranque Ouro IV, com seleção de variáveis via regressão por <i>stepwise</i> . . . . .	47
Tabela 4.9: Comparação entre os ajustes e acurácia dos quatro modelos. . . . .	47
Tabela 4.10: Resultados de predição do modelo treinado e testado no Desafiante, com seleção de variáveis via $R^2$ de Regressão Simples. . . . .	48
Tabela 4.11: Resultados de predição do modelo treinado e testado no Ouro IV, com seleção de variáveis via $R^2$ de Regressão Simples. . . . .	48
Tabela 4.12: Resultados de predição do modelo treinado no Desafiante e testado no Ouro IV, com seleção de variáveis via $R^2$ de Regressão Simples. . . . .	49
Tabela 4.13: Resultados de predição do modelo treinado no Ouro IV e testado no Desafiante, com seleção de variáveis via $R^2$ de Regressão Simples. . . . .	49
Tabela 4.14: Resultados de predição do modelo treinado e testado no Desafiante, com seleção de variáveis via regressão por <i>stepwise</i> . . . . .	50
Tabela 4.15: Resultados de predição do modelo treinado no Desafiante e testado no Ouro IV, com seleção de variáveis via regressão por <i>stepwise</i> . . . . .	50
Tabela 4.16: Resultados de predição do modelo treinado e testado no Ouro IV, com seleção de variáveis via regressão por <i>stepwise</i> . . . . .	50
Tabela 4.17: Resultados de predição do modelo treinado no Ouro IV e testado no Desafiante, com seleção de variáveis via regressão por <i>stepwise</i> . . . . .	51
Tabela 4.18: Comparação de desempenho preditivo dos quatro modelos após serem testados em ambos os bancos de dados. . . . .	51

# 1 Introdução

## 1.1 Contexto, tema e delimitação

O universo dos jogos competitivos *online*, ou *eSports*, como é comumente chamado, tem se expandido a uma velocidade surpreendente. Este ambiente em poucos anos se tornou um grande e lucrativo negócio, que, como analisado por (Mangeloja, 2019), gera milhões de dólares de receita todos os anos.

Neste contexto, existem jogadores que atingem níveis de habilidade muito acima da média e que são capazes de competir em alto nível, assim possuindo conquistas e oportunidades muito além dos demais, passando até mesmo a ter o jogo *online* como renda primária. Há também jogadores que encontraram a transmissão ao vivo (ou *streaming*) de suas partidas para espectadores como uma forma de renda. Um exemplo é o *streamer* norte-americano Tyler Steinkamp, mais conhecido por seu nome de usuário *loltyler1*. Que possui mais de 5 milhões de seguidores e mais de 9.000 inscritos pagos em seu canal na plataforma de transmissão de jogos Twitch (TwitchMetrics). Portanto, se faz necessária uma análise estatística detalhada para entender os fatores que levam ao alto nível nesse universo de natureza competitiva.

Neste estudo serão aplicados diversos procedimentos estatísticos, principalmente referentes a análises de regressão e predição, para detecção e testagem de fatores que possam influenciar a probabilidade de vitória de partidas. Estes procedimentos englobam modelagem por regressão logística, métodos de seleção de variáveis e análise de predição.

As análises serão realizadas em duas amostras do banco de dados de histórico de partidas de jogadores de League of Legends, que será obtido a partir de uma API, uma Interface de Programação de Aplicação gratuita, disponibilizada pela própria Riot Games, a empresa que possui os direitos do jogo, no site [Riot Developer Portal](#). Após um trabalho de concatenação dos bancos de dados e delineamento, isto é, organizar os dados de forma adequada para as análises, serão analisados dados de jogadores de alto e mediano níveis. Este procedimento é especialmente importante para a organização dos dados para este estudo, visto que possibilita o uso de funções específicas para análise em ambiente computacional.

Este trabalho pode auxiliar a montar estratégias para analisar dados do mundo dos eSports, cenário competitivo de jogos eletrônicos, visto a escassez de estudos estatísticos que levam em consideração variáveis do jogo. Também pode servir de base para estudos futuros para outros jogos competitivos *online*, o que se faz relevante, visto a crescente era dos eSports, que atraem grandes massas de público e geram expressiva receita para as organizações, bem como os organizadores de tais eventos.

## 1.2 Problematização

A proporção de jogadores de League of Legends de nível Desafiante, o ranque mais alto, no ano de 2023 flutuou em torno de 0,02% da população de jogadores ativos (League of Graphs, 2013). No cenário competitivo, com grandes premiações em dinheiro e patrocínios envolvidos, como campeonatos, a maioria dos jogadores presentes está nesse nível de habilidade.

Portanto, ao haver escassez de pesquisa científica neste âmbito, bem como valor expressivo de receita gerada pelo jogo, se faz necessário um estudo minucioso com técnicas estatísticas de análise de regressão e preditiva. Para que assim haja um maior entendimento sobre as variáveis dentro de partidas que possam influenciar a probabilidade de vitória dos jogadores.

Trabalhos similares são escassos. Todavia, existem estudos relacionados a jogos *online* que também utilizam de técnicas estatísticas para analisar diferenças entre jogadores, como no estudo realizado por Mora-Cantalops e Sicilia (2018), em que são observadas e analisadas diferenças comportamentais, mensuradas por *player experience* entre jogadores de League of Legends de acordo com seus ranques dentro do jogo. Outro estudo aplicado diretamente a League of Legends é (Kim et al., 2017). Entretanto este é mais focado em fatores extra partidas.

## 1.3 Questões de pesquisa

Como questões principais de pesquisa, temos: quais as variáveis intra-partidas, que impactam na probabilidade de vitória por jogador? Como seria o desempenho preditivo de modelos que lidem com essas variáveis? Estes modelos podem ser aplicados a outros ranques?

## 1.4 Objetivo principal

Este projeto tem como objetivo principal utilizar modelos e técnicas estatísticas, que possuam desempenho preditivo de pelo menos 70% de acurácia para a vitória de jogadores em partidas de League of Legends. Deste modo poderão ser analisadas as ações de jogadores dentro de partidas, medindo assim, suas contribuições para a vitória.

## 1.5 Objetivos específicos

(i) Realizar técnicas de regressão logística e métodos de seleção de variáveis com a ferramenta API de histórico de partidas de League of Legends.

(ii) Apresentar e explicar League of Legends e seu sistema de ranqueamento competitivo.

(iii) Implementar as técnicas de análise e uso da ferramenta API em ambiente computacional com a linguagem **R**.

(iv) Contribuir para o campo de estudo científico sobre jogos eletrônicos.

## 1.6 Hipótese de pesquisa

Suspeita-se que deve haver diferenças significativas em diversos fatores, como eliminações, assistências e dano a torres, entre partidas de jogadores de ranque Desafiante e partidas de jogadores de ranque Ouro IV. Assim identificando variáveis em que os jogadores devem atentar-se para aumentar suas probabilidades de vitória em partidas.

## 1.7 Metodologia de pesquisa

A pesquisa possui uma abordagem quantitativa, visto que as hipóteses de pesquisa se baseiam na diferença de resultados quando analisados diferentes fatores que supostamente contribuem para a probabilidade de vitória de cada jogador. Também se trata de uma pesquisa aplicada, dada a utilização de dados reais já existentes, com um objetivo de contribuir para análises estatísticas em jogos *online*. Faremos o uso de técnicas estatísticas bem estabelecidas na literatura e que foram descritas anteriormente.

Além do mais, o caráter da pesquisa é conclusivo, visto que o objetivo da pesquisa é identificar, analisar e quantificar os fatores que podem influenciar a vitória para diversos jogadores, não acompanhando seus dados ao longo do tempo, assim se fazendo também de uma pesquisa de caráter transversal. Quanto às etapas do estudo, temos:

### 1. Pesquisa referencial:

Após o levantamento de ideias e verificação de viabilidade do estudo, se faz necessária uma pesquisa acerca de artigos e publicações semelhantes a este estudo. Deste modo, pode-se verificar a originalidade do projeto, bem como sua importância para futuros trabalhos.

### 2. Coleta de dados:

Crucial para a realização do estudo, a coleta de dados deve ser realizada. Nesta etapa será utilizada a ferramenta Riot Games API. Após a obtenção dos links e download dos dados, se segue à concatenação dos bancos de dados e limpeza dos dados, para melhor uso das ferramentas estatísticas. Após a coleta e tratamento dos dados, é esperado que haja um maior embasamento sobre as hipóteses e questões de pesquisa.

### 3. Revisão bibliográfica:

A revisão bibliográfica é de suma importância para a validação e o embasamento teórico da pesquisa. Assim havendo uma revisão dos métodos estatísticos inicialmente propostos, e verificando como estes podem e devem ser aplicados no contexto do estudo.

### 4. Aplicação de métodos estatísticos:

Com um banco de dados devidamente concatenado, limpo e organizado, e com os métodos e modelos cogitados a priori devidamente revisados durante a etapa anterior, é possível aplicar as técnicas aos dados e determinar a adequação de cada método e modelos aplicados. Ao fim dos testes e aplicação dos modelos e métodos, se fazem necessárias ferramentas de visualização dos resultados, com o uso de gráficos, tabelas com estatísticas descritivas, como médias, desvios e quantis e resultados de testes de hipóteses, para que assim haja melhor compreensão dos resultados de pesquisa.

### 5. Redação do trabalho de conclusão:

Por fim, será finalizada a redação do trabalho, bem como as conclusões do estudo, que serão discutidas entre o autor e professor orientador após uma análise detalhada dos resultados obtidos durante a etapa anterior. Adicionalmente, também serão explicadas as limitações deste estudo. As conclusões, bem como os métodos utilizados para obtê-las servirão como base de defesa do projeto.

Tendo consciência da metodologia de pesquisa e etapas para conclusão do trabalho apresentadas neste capítulo, será apresentado no próximo capítulo o jogo League of Legends e as características importantes para o trabalho.

## 2 League of Legends e banco de dados

### 2.1 Comentários iniciais

League of Legends é um jogo multijogador *online* de arena de batalha (MOBA, do original em inglês *Multiplayer Online Battle Arena*), de caráter competitivo lançado em 27 de outubro de 2009. Desde então o jogo recebe constantes atualizações para que se mantenha renovado e diverso. É estimado que, em média, há cerca de 150 milhões de jogadores ativos mensalmente em 2023 ([Active Player, 2023](#)), divididos em 12 regiões globais.

O sucesso global de League of Legends pode ser explicado por alguns fatores, como sua disponibilidade gratuita para jogar, interação entre jogadores dentro de partidas, baixos requerimentos gráficos, podendo assim ser jogado em computadores de baixo desempenho e ausência de vantagens por micro-transações dentro do jogo, pois itens comercializáveis por dinheiro são apenas cosméticos, ou não dão vantagens dentro de partidas, mantendo o jogo equilibrado independentemente de recursos financeiros dos jogadores.

O jogo não é o único representante de jogos do gênero MOBA. Contudo, League of Legends é atualmente o jogo MOBA com o maior número de jogadores ativos mensalmente, sendo alguns dos seus principais concorrentes: Dota 2 ([Valve, 2013](#)), Smite ([Hi-rez Studios, 2015](#)) e Heroes of the Storm ([Activision Blizzard, 2015](#)). Os jogos desse gênero possuem algumas características que os definem, como: mapa do jogo sendo uma arena fechada, destruição de estruturas como objetivo principal, controle de um personagem específico por jogador, evolução de níveis do personagem, entre outras características.

### 2.2 Elementos do jogo

#### 2.2.1 Mapa

Há diversos modos de jogo e mapas em League of Legends, mas o primeiro e mais popular é Summoner's Rift, e foram providos de partidas situadas neste mapa os dados analisados neste trabalho. O modo de jogo original funciona por partidas, em que duas equipes de cinco jogadores disputam, em um mapa espelhado, com o objetivo de derrubar a estrutura conhecida como nexus do time adversário, ou que a equipe adversária se renda antes da destruição do nexus.

Summoner's Rift já foi atualizado diversas vezes, com inúmeras mudanças que vão desde alterações visuais, a alterações que afetam a jogabilidade. Em maio de

2012 o mapa passou por uma grande atualização, que o deixou mais aceitável para os padrões gráficos da época, bem como sua jogabilidade, que foi bastante alterada, pois, embora a base do mapa nunca tenha mudado, seu leiaute foi várias vezes reimaginado.

Nesse sentido, a atualização para a temporada 2020 foi especialmente significativa, visto que uma porção do leiaute de Summoner's Rift em cada partida passou a depender da presença de um objetivo específico no mapa.



Figura 2.1: Mapa de League of Legends com legenda.

Como pode ser observado na Figura (2.1), existem três rotas principais em Summoner's Rift: a rota superior, a rota do meio e a rota inferior. Estas rotas possuem grande importância no mapa, visto que a comunidade de jogadores, ao longo de anos de experiência, desenvolveu um sistema organizado para a disposição dos jogadores das equipes nestas rotas. Nas rotas superior e do meio, normalmente há apenas um jogador em cada uma delas por equipe, que enfrentarão o jogador da outra equipe. Contudo, na rota inferior há dois jogadores por equipe que se enfrentam. Além da selva, que é o espaço entre as rotas que dispõe dos objetivos neutros e "campos" de monstros da selva, que usualmente possui um jogador para cada equipe auxiliando os aliados que estão nas rotas e assim garantindo os objetivos neutros para sua equipe.

## 2.2.2 Estruturas

Estruturas são componentes principais dentro de partidas. Derrubá-las é o objetivo principal das equipes, sendo necessário derrubá-las sequencialmente, de acordo com a rota. No mapa Summoner's Rift há quatro tipos de estruturas que estão presentes para os dois times:

- **Fonte**

É nessa estrutura que os jogadores começam a partida, serve como um ponto de ressurgimento e recuperação de pontos de vida, além de ser o único local do mapa que possui uma loja, onde os participantes da partida podem comprar itens e consumíveis.

- **Torres**

Totalizando doze para cada time, as torres impedem o avanço de tropas inimigas e defendem os aliados caso sejam atacados por outros jogadores que estejam no alcance das torres. Todavia, caso hajam tropas inimigas no alcance da torre e um jogador inimigo não atacar um aliado dentro desse alcance, a torre terá como foco as tropas. Torres recebem dano reduzido caso jogadores do time adversário não possuam tropas no alcance destas.

Há uma torre especial para cada time, chamada de Obelisco do Nexus, é localizada dentro da fonte da equipe, e desfere muito dano contra oponentes que entrarem em seu alcance, que é exatamente o tamanho da fonte. Esta torre especial tem como objetivo prevenir que jogadores eliminem outros dentro de seu ponto de ressurgimento.

Além do mais, a primeira torre de cada rota possui cinco barricadas que reforçam sua defesa, cada uma representando 20% dos pontos de vida da torre. Essas barricadas, caso não sejam destruídas até 14 minutos de partida, desaparecem e deixam de reforçar a defesa da torre.

- **Inibidores**

Há três inibidores para cada equipe, todos esses sendo localizados dentro dos limites da base. Essas estruturas impedem que o time adversário possa destruir as últimas duas torres e o nexus, sendo necessário ter pelo menos um inibidor destruído para finalizar a partida.

Os inibidores tem a capacidade de regeneração total de pontos de vida, ou seja, caso seja destruído, um inibidor voltará totalmente renovado em 5 minutos. Contudo, durante o período em que estão destruídos, tropas mais poderosas, conhecidas como "super tropas", são geradas em cada onda de tropas do time adversário, sendo assim, a destruição de inibidores é um componente fundamental para a manutenção da pressão de mapa por uma equipe.

- **Nexus**

É a estrutura principal e destruir o nexus adversário é o maior objetivo de cada equipe em jogo. No nexus em que são liberadas as tropas, que permitem o avanço dos times pelo mapa e criam pressão nos adversários. Todavia, além dessa função, o nexus é a peça final da partida, visto que a equipe que destruir esta estrutura do adversário primeiro, vence a partida.

### 2.2.3 Tropas

As tropas são o que possibilita os jogadores a avançarem pelas rotas e derrubarem torres inimigas para se aproximarem do nexus adversário. A partir de exatamente um minuto e cinco segundos de partida, as tropas são liberadas em ondas, ou, *waves*, e saem em intervalos regulares de trinta segundos do nexus aliado.

As ondas de tropas possuem dois tipos. O primeiro consiste em seis tropas, sendo três tropas corpo-a-corpo e três tropas magas. O segundo tipo de onda mantém a mesma configuração, mas inclui uma tropa adicional chamada de invasora. Neste segundo tipo de onda, portanto, há sete tropas, três corpo-a-corpo, três magas e a tropa adicional invasora. As primeiras duas ondas de tropas a serem liberadas são do primeiro tipo. Já a terceira onda de tropas é do segundo tipo. Este padrão de liberação de ondas permanece durante a partida inteira, exceto quando o inibidor é destruído.

Quando a estrutura do inibidor é destruída pelo time adversário, o nexus do mesmo começará a liberar os mesmos tipos de ondas, porém com uma tropa adicional, chamada de super tropa. Esta tropa possui resistência e poder de ataque muito superiores às demais tropas. Caso os três inibidores adversários estejam simultaneamente destruídos, serão liberadas duas super tropas ao invés de uma.

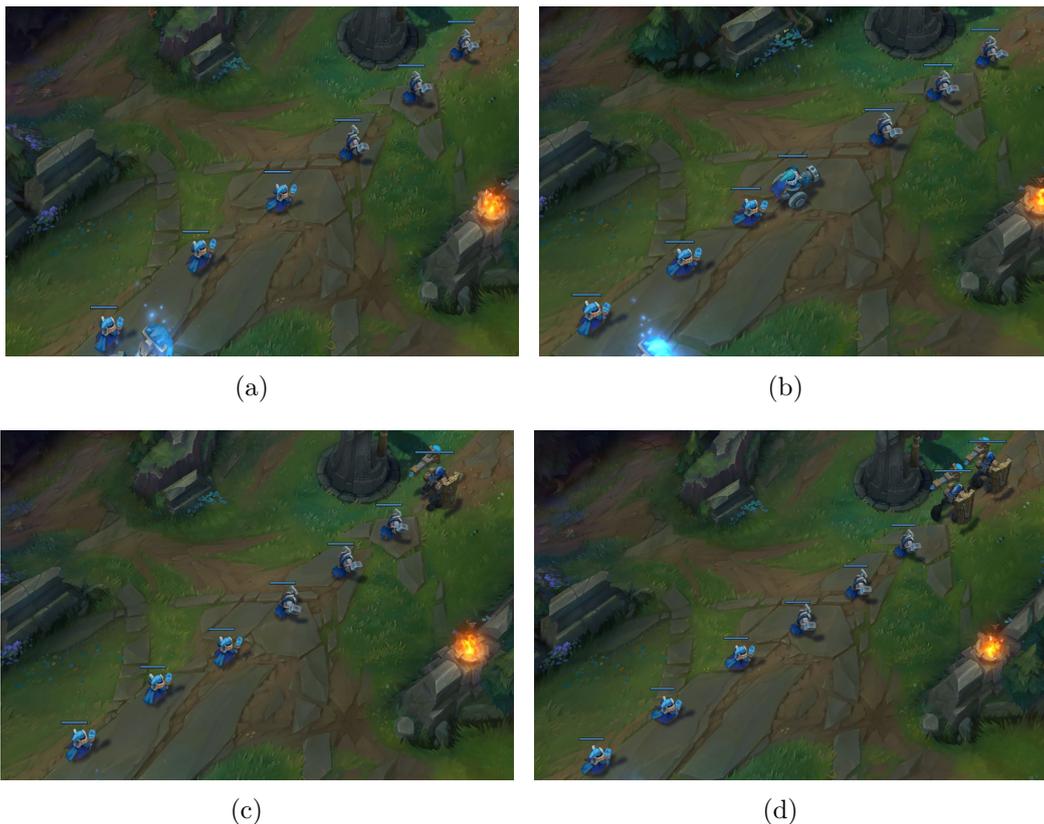


Figura 2.2: Ondas de tropas da equipe azul.

Pode-se observar na Figura (2.2) ondas do primeiro e segundo tipo com e sem super tropas. Na Figura (2.2(a)) podemos observar mais à direita três tropas corpo-a-corpo, seguidas de três tropas magas. Na sequência, a Figura (2.2(b)) mostra uma configuração similar, porém com a inclusão da tropa invasora no centro da formação. Na Figura (2.2(c)) a formação é iniciada por uma super tropa. Por fim,

na Figura (2.2(d)), temos a formação com duas super tropas liderando a onda. É importante mencionar que as tropas ficam mais fortes ao longo da partida. Além disso, ainda podem ficar mais fortes desde que haja uma diferença entre o nível médio dos campeões de cada equipe. Neste caso, o balanço de poder das tropas é maior para a equipe com maior nível médio dos campeões. Adicionalmente, tropas podem ser fortalecidas por itens que os jogadores comprem dentro das partidas e quando a equipe abate o objetivo neutro *Barão Nashor*.



Figura 2.3: Tipos de tropas.

Na Figura (2.3) temos os quatro tipos distintos de tropas. As diferenças entre tropas do mesmo tipo são puramente gráficas. Os elementos visuais seguem o padrão visto na Figura (2.1), tropas azuis para a equipe azul e tropas vermelhas para a equipe vermelha.

## 2.2.4 Objetivos neutros

Importantes para a vitória, objetivos neutros existem em League of Legends na forma de criaturas que se localizam no rio de Summoner's Rift, a parte central do mapa. Estas criaturas são chamadas de objetivos pois além de ouro, também fornecem algumas vantagens e aprimoramentos à equipe que os derrotarem.

Atualmente há três objetivos neutros presentes no jogo:

- **Dragão**

O dragão é um objetivo especialmente influente em partidas de maior duração. Isso se deve ao fato de que abater um dragão garante aprimoramentos permanentes para a equipe que o abateu, fortalecendo-a. Além disso, existe o efeito cumulativo destes aprimoramentos. Depois de derrotado o primeiro dragão, outro dragão surge a cada 5 minutos.

O primeiro dragão surge após 5 minutos de partida. Este dragão possui um elemento aleatoriamente selecionado. Os aprimoramentos obtidos após ser

abatido dependem do elemento do dragão. Os primeiros três dragões elementais da partida terão elementos diferentes uns dos outros. Contudo, todos os dragões elementais subsequentes possuirão o mesmo elemento do terceiro dragão. O terceiro dragão também modifica outras características do mapa, como terreno e leiaute. Estas modificações dependem do elemento sorteado.

Caso 4 dragões elementais sejam abatidos pela mesma equipe na partida, todos os integrantes serão recompensados com um poderoso aprimoramento que varia de acordo com o elemento do último dragão abatido, chamado de "alma do dragão". Após uma das equipes obter a "alma do dragão", os próximos dragões surgirão a cada 6 minutos, e todos serão de um tipo especial, chamados de dragões anciões, que, quando abatidos, recompensam os integrantes da equipe que o abateu com um aprimoramento especial temporário.

- **Arauto do vale**

O Arauto do Vale é uma criatura que surge no mapa a partir dos 8 minutos de partida. Depois de abatido ressurgirá a cada 6 minutos. Este objetivo só está disponível no mapa até os 20 minutos de partida, pois, após esse tempo, outro objetivo, chamado de Barão o substitui e começa a surgir no mesmo local.

Ao abater o arauto, um jogador da equipe poderá obter seu olho como um item consumível, que lhe concederá um aprimoramento que reduz o tempo de ida para a base pela metade. Quando o olho é consumido o arauto é invocado como um aliado e se dirige à torre inimiga mais próxima, causando uma quantia considerável de dano à estrutura.

- **Barão**

Considerado como o objetivo neutro mais importante, o Barão Nashor surge a partir dos 20 minutos e ressurgirá a cada 7 minutos quando derrotado. Seu aprimoramento é temporário e concedido a todos os integrantes da equipe que o abateu. Este aprimoramento garante redução de tempo para voltar à base pela metade e fortalecimento de atributos e tropas aliadas próximas. Esse aprimoramento é especialmente útil pois as tropas fortificadas causam muita pressão no mapa e dificuldade para a equipe adversária lidar com as ondas.

## 2.2.5 Ouro

O ouro é a principal fonte de vantagem dentro das partidas. Em cenários profissionais, por exemplo, a vantagem de uma equipe sobre a outra na partida é medida diretamente através da diferença de ouro entre as equipes. Afinal, o ouro é utilizado para comprar itens, que podem oferecer efeitos passivos, efeitos ativos e atributos base.

Portanto, tendo em vista que o ouro é o principal componente para obter vantagem sobre os adversários, há diversas formas de obtê-lo nas partidas. Todos os jogadores começam com 500 de ouro para a compra de seus itens iniciais, além de que há a geração passiva de aproximadamente 2 de ouro por segundo a partir de 1 minuto e 50 segundos de partida no mapa Summoner's Rift. Uma das principais maneiras de obter ouro é abatendo tropas, sendo necessário dar o golpe final para captar o ouro da tropa. Monstros e objetivos neutros da selva também são uma

fonte de ouro importante, visto que são adversários mais fortes e geram mais ouro que tropas ao serem abatidos.

A destruição de estruturas como torres é uma fonte muito expressiva de ouro, gerando ouro local (para quem estiver presente durante a destruição da torre) e global (para os demais aliados espalhados pelo mapa). Além das barricadas, que estão presentes até 14 minutos de jogo nas primeiras torres de cada rota. Representam 20% da vida dessas torres, e quando destruídas geram uma quantia expressiva de ouro para os jogadores que participaram de sua destruição. Por último temos os abates e assistências: quando um jogador abate ou participa do abate de um jogador adversário, recebe uma recompensa em ouro por este abate, que varia de acordo com o sistema de recompensas dentro do jogo. Para o abate, é necessário que um jogador dê o golpe que zera os pontos de vida do inimigo, ou dar o último golpe antes do inimigo morrer para outra fonte de dano que não seja de outro jogador. Caso o jogador ataque um inimigo que foi abatido por outro companheiro de equipe, ou tenha ajudado este companheiro com bônus, como cura e escudos, o mesmo receberá uma "assistência" em seu placar de pontos, que também recompensa em ouro o jogador, embora não tanto quanto um abate.

## 2.2.6 Atributos, experiência e itens

Por ser um MOBA, League of Legends se assemelha muito à jogos do gênero RPG (*Role-playing game*), em que o personagem evolui níveis para poder melhorar seus equipamentos e atributos. O sistema de atributos é uma maneira de mensurar o quão forte um jogador está dentro da partida. Alguns atributos básicos são poder de ataque, armadura, vida, mana, velocidade de movimento, poder de habilidade e resistência mágica. Valores altos destes atributos normalmente significam maior importância de um jogador dentro da partida.

Dentre as principais maneiras de aumentar atributos, estão a obtenção de pontos de experiência para evoluir níveis, visto que o seu personagem eleva alguns atributos sempre que sobe de nível e obtém pontos de habilidade, e a compra de itens, que se apresenta como a forma mais eficaz de aumentar seus atributos dentro da partida.

Pontos de experiência são obtidos de diversas maneiras na partida, como abatendo tropas, monstros da selva, participação em eliminações de adversários, entre outros. Além disso, todos os jogadores iniciam a partida no nível 1. Porém quando acumulados suficientes pontos de experiência, o jogador evolui um nível, e pode continuar evoluindo até o nível 18.

Os itens podem ser comprados nas lojas localizadas dentro das fontes de cada equipe, nessa loja, o jogador pode utilizar o seu ouro obtido para comprar itens que oferecem desde atributos brutos, à habilidades passivas e ativas que concedem vantagem na partida. É importante notar também que a seleção dos itens corretos para situações diferentes dentro de partidas é importante, pois há uma vasta gama de sinergias entre itens e campeões, além de itens adequados para lidar com determinados oponentes e composições de time inimigas. Um exemplo de itens especialmente importantes são as sentinelas da visão, que servem para revelar regiões do mapa em que a equipe aliada não está presente, tornando possível observar a movimentação da equipe adversária.

### 2.2.7 Campeões

O League of Legends atualmente conta com mais de 140 campeões disponíveis, todos únicos e com habilidades e estilos de jogo diferentes uns dos outros. Tendo em vista que os campeões são os principais responsáveis pela variedade no jogo a Riot Games lança ,anualmente, novos campeões para que a jogabilidade não fique repetitiva.

Cada campeão possui, por padrão, uma habilidade passiva e quatro habilidades distintas. Destas, três são habilidades básicas e uma é habilidade suprema, que pode ser desbloqueada a partir do nível 6 do campeão. Adicionalmente, o jogador pode usar pontos de habilidade para desbloqueá-las e evoluí-las (exceto a habilidade passiva), recebendo um ponto a cada nível. A utilidade e função de cada habilidade são únicos, bem como seus tempos de recarga. É interessante notar que, além disso, o papel dos campeões na partida são distintos, pois há aqueles que se destacam em diferentes competências dentro de partidas em comparação a outros.

Os jogadores selecionam seus campeões em uma tela pré-partida chamada de Seleção de Campeões. Nessa tela também é possível banir campeões para que outros jogadores não possam selecioná-los, inclusive de sua própria equipe. Durante a Seleção de Campeões, também é importante a escolha do personagem na situação correta, visto que há diversas sinergias entre campeões e diversos *picks*, ou escolhas de personagem, que afetam negativamente outros campeões presentes no time adversário.

### 2.2.8 Atualizações

Um dos atrativos principais de League of Legends são suas atualizações, ou *patches*, constantes, visto que estas são essenciais para manter o jogo balanceado, renovado, e mitigar a monotonia perceptível aos jogadores. Essas atualizações englobam mudanças em todo e qualquer aspecto do jogo.

Contudo, as atualizações não podem ser aplicadas diretamente ao jogo sem um comunicado adequado e sem serem testadas, e por isso existem as notas de atualização, que são lançadas semanas antes da atualização aos jogadores, disponíveis no próprio site de League of Legends ([Notas de Atualização - League of Legends, 2021](#)). Além da existência do servidor *PBE*, uma sigla para *Public Beta Environment*, onde jogadores cadastrados podem jogar gratuitamente partidas em uma versão de jogo em que as atualizações são aplicadas horas após serem comunicadas, assim sendo testadas as mudanças antes de serem definitivamente aplicadas aos demais servidores do jogo.

## 2.3 Sistema de ranqueamento

Dentro dos modos de jogo ou "filas"ranqueadas, os jogadores são atribuídos à ranques, também conhecidos como *elo*, que são basicamente níveis de habilidade, de acordo com o seu nível de habilidade. O ranque é atribuído a um jogador de acordo com seus Pontos de Liga, que são influenciados pelo sistema de MMR e ranque da temporada anterior.

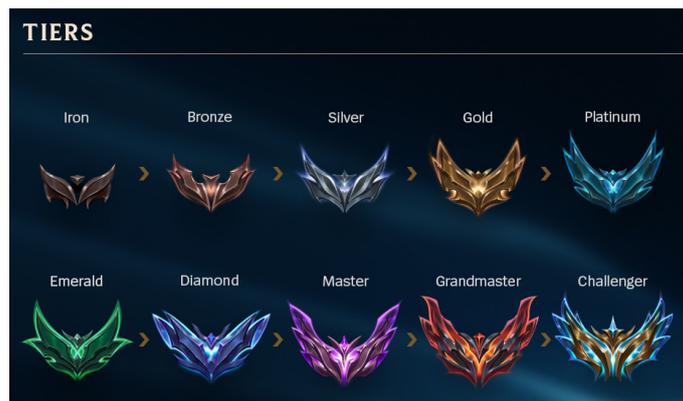


Figura 2.4: Ranques de League of Legends.  
Fonte: Riot Games, 2023.

### 2.3.1 Ranques e distribuição de jogadores

Na Figura (2.4) podem ser observados os atuais 10 ranques presentes no sistema de ranqueamento de League of Legends, que vão de Ferro a Desafiante. Estes ranques tem como objetivo manter o balanceamento dentro de partidas, ao designar jogadores de ranques iguais ou próximos às mesmas partidas. Dentro desses ranques, com exceção dos três mais altos (Mestre, Grão-Mestre e Desafiante), há quatro divisões, que vão de IV a I, em ordem ascendente. O jogador necessita subir de divisões ao obter pontos de liga para progredir seu ranque. Entretanto, é importante ressaltar que o rebaixamento de divisões e até mesmo elo pode ocorrer caso o jogador entre em uma sequência de derrotas no ponto mais baixo da divisão.

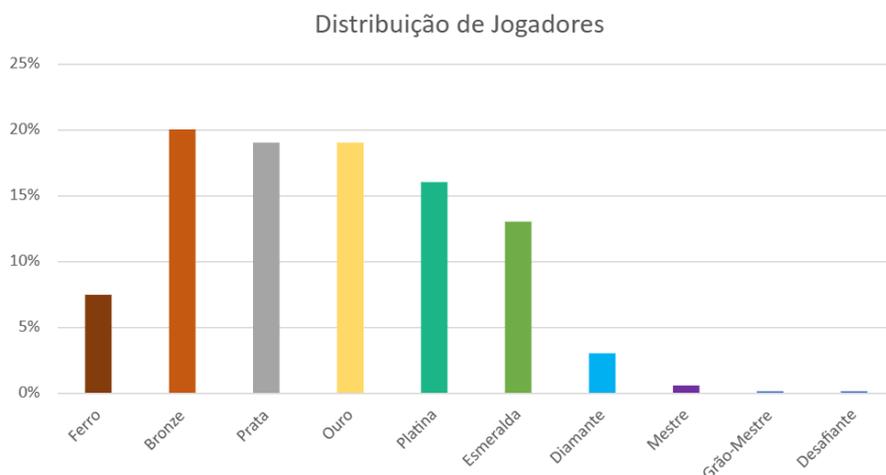


Figura 2.5: Distribuição de jogadores por ranque na fila ranqueada Solo/duo em 2023.

Fonte de dados: League of Graphs, 2023.

A Figura (2.5) representa a distribuição de ranques na fila ranqueada solo/duo, considerada como o principal modo de jogo ranqueado, para todas as regiões, com o ponto de corte de 50% dos jogadores ativos na fila ranqueada estando no ranque Ouro IV. Além de que os três maiores ranques, Mestre, Grão-Mestre e Desafiante representam apenas aproximadamente 0.6% da base de jogadores ativos na fila ranqueada. Os dois maiores ranques, Desafiante e Grão-Mestre, são casos especiais

dentre todos os outros ranques, visto que são limitados a 300 e 700 jogadores por região, respectivamente.

### 2.3.2 Pontos de liga e progressão de ranque

A menor unidade utilizada para definir os ranques de cada jogador são os pontos de liga, ou, *PdLs*. Estes PdLs são obtidos ao vencer cada partida, mas também são subtraídos do total quando o jogador perde uma partida. Uma divisão de ranque demanda 100 PdLs para que haja a promoção para a próxima divisão, e, quando o mesmo atinge 100 PdLs na divisão I de um determinado ranque, é necessário que este seja bem sucedido em uma série de promoção chamada de melhor de 5, ou, *md5*. Portanto, o jogador que se encontra em uma série de promoção, deve vencer 3 das 5 partidas presentes na série para assim ser promovido ao próximo ranque. Por exemplo, se um jogador vence sua série de promoção enquanto Prata I, é promovido ao próximo ranque na divisão mais baixa, ou seja, Ouro IV.

A exceção neste sistema de progressão de elo se encontra nos três elos mais altos, onde não há divisões e a colocação ranqueada é definida exclusivamente pelos PdLs. Jogadores podem atingir o ranque Mestre ao vencerem sua série md5 no Diamante I, e podem se tornar elegíveis para promoção para Grão-Mestre ao atingirem 200 PdLs enquanto no Mestre. Adicionalmente, o mesmo sistema se aplica ao maior ranque, Desafiante, onde os jogadores precisam atingir 500 PdLs enquanto no Grão-Mestre para serem promovidos. Entretanto, essas faixas de 200 e 500 PdLs são variáveis, visto que estes ranques possuem vagas limitadas. Portanto, além de ter que atingir esse valor mínimo de pontos, o jogador deve possuir uma quantia maior do que o jogador com menor posição ranqueada no Grão-Mestre e Desafiante.

Além do mais, outro fator importante que afeta diretamente a quantidade de pontos de liga recebidos ou perdidos após uma partida, é o sistema *MatchMaking Rating* (MMR), que é bastante semelhante a como os pontos de liga funcionam, aumentando ou diminuindo para cada jogador conforme o resultado da partida. Se trata de um sistema que apesar de confirmado pela Riot Games, possui poucas informações oficiais divulgadas. Esse sistema serve como uma espécie de mediador de partidas, ou seja, é o sistema que é responsável por juntar jogadores de níveis de habilidade semelhantes em partidas ranqueadas. Suponha dois jogadores no mesmo elo, um jogador que possui um MMR mais baixo e outro que possui um MMR mais alto, o jogador com o MMR mais baixo entre os dois encontrará mais dificuldades para subir seu elo, visto que receberá menos PdLs por vitória e terá maior penalidade de PdLs por derrota. O sistema funciona desse jeito por entender que o jogador não está no seu ranque adequado de acordo com seu nível de habilidade.

### 2.3.3 Temporadas ranqueadas

As temporadas ranqueadas são períodos que vigoram durante aproximadamente um ano. Estas temporadas podem ser definidas como o período em que um jogador pode alcançar um ranque, que é reiniciado após o fim deste período. Todas as temporadas possuem premiações para jogadores dependendo de seus elos, com jogadores que atingem o elo Ouro ou superior recebendo itens cosméticos que podem ser utilizados em partidas para suas contas, a depender da fila ranqueada no qual alcançaram esse elo.

Todavia, o destaque principal das temporadas são as atualizações. Visto que, apesar de constantes durante todo o período da temporada, para manter o balançamento dentro de partidas, há grandes atualizações no início de cada temporada, muitas vezes alterando sistemas e mecânicas essenciais dentro do jogo. Para que os jogadores não sejam totalmente surpreendidos com muitas mudanças ao trocar a temporada, existe um período entre temporadas chamado de pré-temporada, em que os jogadores podem obter um elo provisório e se acostumarem com as mudanças que serão implementadas quando de fato começar o período ranqueado.

No início de cada temporada o ranque dos jogadores é reiniciado, e o jogador deve completar uma série de 10 partidas para obter seu ranque inicial. Contudo, a cada vitória o mesmo recebe PdLs em quantias elevadas que definirão um ranque provisório, que ao final da série, será seu elo inicial para a temporada vigente.

### 2.3.4 Cenário competitivo

Os esportes eletrônicos, ou eSports, emergiram como uma indústria global em constante expansão, com um dos títulos mais proeminentes sendo o League of Legends. O jogo, através de diversas campanhas de publicidade, feitas ao longo do tempo de vida do mesmo, conquistou uma enorme base de fãs e se estabeleceu como uma força dominante no cenário competitivo.

O cenário competitivo de League of Legends é caracterizado pela participação ativa de organizações de eSports. Essas organizações, associadas a equipes profissionais, desempenham um papel crucial na formação e gestão de talentos. As ligas profissionais são compostas por equipes que competem em temporadas regulares e buscam a qualificação para os campeonatos internacionais.

O calendário de competições de League of Legends é robusto, composto por várias ligas regionais, cada uma alimentando o cenário internacional. Além das LCS (League Championship Series) na América do Norte e LCK (League of Legends EMEA Championship) que compreende as regiões da Europa, Oriente Médio e África, há ligas em regiões como Coreia do Sul (LCK, sigla para League of Legends Champions Korea), e outras. Todas as competições oficiais possuem premiações em dinheiro para os participantes, que variam de acordo com a competição e classificação da equipe. As equipes buscam a excelência em suas ligas regionais classificatórias para garantir uma vaga nos prestigiados campeonatos internacionais.

A maior liga oficial no Brasil é o CBLOL (Campeonato Brasileiro de League of Legends), que compreende as maiores organizações nacionais e internacionais de eSports presentes no país, incluindo até mesmo equipes pertencentes a grandes empresas de outros ramos e clubes de futebol, como é o caso da KaBuM! Esports e Flamengo eSports, respectivamente. Estas empresas contratam como profissionais alguns dos melhores jogadores da região, assim criando um mercado do entretenimento saudável e competitivo em torno das competições. O CBLOL possui duas fases, ou *splits*, que possuem fase de pontos e eliminatórias cada. O vencedor do primeiro split se classifica para o Mid-Season Invitational, o MSI, que é o segundo campeonato internacional mais importante do cenário. Já o vencedor do segundo *split*, que também pode ser o campeão do primeiro *split*, se classifica diretamente para o campeonato mundial.

Atualmente os eSports de LoL possuem grande alcance e popularidade, passando a ter campeonatos como o CBLOL transmitidos em grandes canais de televisão

esportivos, como o SporTV, que em 2017 transmitiu o campeonato e foi palco de um momento simbólico, onde o narrador Galvão Bueno narrou um trecho da partida disputada entre as equipes KEYD e INTZ ([Globo Esporte, 2022](#)).

O maior torneio da temporada competitiva é o Campeonato Mundial de League of Legends, também conhecido como *Worlds*. Em que as equipes qualificadas de diferentes regiões se enfrentam em uma competição que envolve fase de grupos e eliminatórias para determinar a melhor equipe do mundo. O caminho para o Worlds inclui as performances nas ligas regionais, com os vencedores e os melhores colocados ganhando o direito de representar suas regiões no torneio global.



Figura 2.6: SK Telecom T1 conquista seu tricampeonato mundial de LoL.  
Fonte: Riot Games, 2023.

Na Figura (2.6) observa-se a equipe campeã mundial de 2023, a equipe sul-coreana SK Telecom T1, que conquistou seu tricampeonato mundial. Nota-se, ao fundo, a presença de grande plateia e o troféu da competição.

## 2.4 Banco de dados

### 2.4.1 Fonte de dados

Serão utilizados dados de histórico de partidas de jogadores anônimos e aleatoriamente selecionados, de acordo com o grupo de nível de habilidade desejado. Estes dados são obtidos a partir da ferramenta Riot Games API. Portanto, é possível pela API estabelecer uma conexão entre os servidores da Riot Games e o computador utilizado na pesquisa, para assim obter dados de histórico de partida.

Esta conexão com a API e o computador é estabelecida a partir do site Developer Portal da Riot Games ([Riot Developer Portal, 2021](#)), onde é requisitada a criação de uma conta para obter as chaves de acesso. Ao obter a chave de acesso, é possível utilizar as funções prontas de obtenção de dados oferecidas pela própria Riot Games em diversas linguagens de programação, bem como a criação de funções que atendam o objetivo específico do usuário. Neste trabalho optou-se por utilizar a linguagem de programação **R** como ferramenta de análise e organização dos dados. Além disso, o site oferece diferentes filtros para obtenção dos dados, referentes a características próprias do jogo League of Legends, como vitórias, entre outros.

## 2.4.2 Obtenção dos dados

Com o uso de funções criadas em **R** que estabelecem conexão com a API por meio de URLs (do inglês, *Uniform Resource Locator* ou "Localizador Uniforme de Recursos"), são obtidos os dados desejados de diversas partidas. Para conseguir os dados, foram realizadas algumas etapas com as funções disponíveis na API:

### 1. Obtenção da lista de jogadores:

Primeiramente é feita a obtenção da lista de jogadores de acordo com o filtro de ranque, conseguindo assim sua posição ranqueada e *id* de invocador, que seria a variável identificadora.

### 2. Obtenção de partidas recentes:

Pelo *id* de invocador, são obtidos os identificadores de suas partidas recentes, que serão analisadas no próximo passo. Importante deixar claro que o objeto de estudo são as partidas.

### 3. Obtenção de informações de partida:

Utilizando o *id* das partidas, é possível obter, por meio de uma função definida na API, bancos de dados referentes a cada partida, com diversas informações, como tempo de jogo, ranque da partida, modo de jogo, participantes, entre outros.

### 4. Concatenação de dados dos jogadores:

Após serem obtidas as informações da partida, são concatenados dados da partida para cada um dos dez participantes. Tendo em vista que uma partida de League of Legends necessita de dez jogadores, obtemos mais de 200 variáveis referentes a dados dentro de partida, como vitória, abates e assistências para dez jogadores.

### 5. Transformação do banco:

Os dados concatenados são transformados, removendo variáveis que sabemos a priori não serem relevantes para a análise e adicionando uma variável com o *id* da partida referente aos dados obtidos, a fim de remover partidas duplicadas.

Este processo pode ser replicado e repetido, portanto foi desenvolvida uma função que realiza todas essas etapas em loop, com parâmetros definidos, que são: banco de dados a ser lido, que se trata da lista de jogadores do ranque de interesse; número de jogadores que terão suas partidas recentes obtidas, que é uma sequência numérica discreta que vai de 1 ao valor selecionado; e número de partidas por jogador, uma variável discreta de 1 a 20.

Portanto, o cálculo do número de jogadores obtidos pela função pode ser definido por:

$$(N \cdot P \cdot 10) - R, \quad \text{em que } N, P, R \in \mathbb{N}. \quad (2.1)$$

$N$  é o número de jogadores que terão suas partidas recentes analisadas,  $P$  é o número de partidas recentes de cada jogador que serão analisadas e  $R$  é o número de partidas repetidas após a obtenção e agrupamento de todos os dados. A partir dessa função foram obtidos dois bancos de dados de jogadores do servidor europeu (EUW,

ou Europe West), ambos com  $N = 200$  e  $P = 3$ , porém um com a lista de jogadores referente ao ranque Desafiante e outro com o ranque Ouro IV. O servidor EUW foi selecionado pois além de ser um dos servidores mais antigos do jogo, também é um dos mais competitivos e balanceados nas filas ranqueadas, possuindo um equilíbrio de habilidades individuais e coletivas entre os jogadores.

O banco relativo ao ranque Desafiante possui 5.600 observações, ou jogadores, visto que foram obtidos 400 dados de partidas repetidas. Partidas repetidas no banco de dados dos Desafiantes eram esperadas, visto a escassez de partidas neste ranque. Contudo, o banco de dados do ranque Ouro IV possui exatas 6.000 observações. Essa ausência de partidas repetidas neste ranque pode ser explicada pelo alto volume e volatilidade de jogadores, pois jogadores são constantemente promovidos ou rebaixados. No próximo capítulo será apresentado o referencial teórico para a realização deste trabalho.

## 3 Referencial teórico

A base referencial teórica deste trabalho foi retirada de livros, sendo os principais (Montgomery, 2013) e (James et al., 2013). Adicionalmente, a ferramenta de pesquisa educacional e científica (Google Scholar) foi amplamente utilizada para auxiliar na obtenção do referencial teórico.

### 3.1 Regressão Linear e Modelagem

Análise de regressão é uma técnica estatística para investigar e modelar a relação entre as variáveis (Montgomery, 2013). A análise de regressão busca identificar padrões, tendências e prever valores desconhecidos com base nas informações disponíveis. A relação é expressa através de uma equação matemática que descreve como as variáveis independentes influenciam a variável dependente. Comumente, análises de regressão são acompanhadas de um gráfico de dispersão, o que evidencia a relação entre as variáveis ao traçar a reta formada pela equação de regressão no gráfico e compará-la com os dados dispersos, simbolizados por pontos.

Para avaliar a presença de uma relação linear, um teste de hipóteses é empregado, em que a hipótese nula considera que todos os coeficientes são iguais a zero, enquanto a hipótese alternativa afirma que pelo menos um dos coeficientes é diferente de zero. O teste é conduzido através da estatística F.

Uma métrica amplamente utilizada para medir o ajuste do modelo é o coeficiente de determinação  $R^2$ , que é o quadrado da correlação de Pearson entre duas variáveis quantitativas, podendo ser escrito como

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}. \quad (3.1)$$

Quando o  $R^2$  está próximo de 1, isso indica que o modelo explica uma grande parcela da variação na variável resposta. No entanto, à medida que novas variáveis são adicionadas ao modelo, o coeficiente tende a aumentar, mesmo que essas variáveis estejam apenas fracamente associadas à resposta (James et al., 2013).

Modelos de regressão geralmente são representados pela seguinte forma:

$$y = \beta_0 + \beta_1 x + \epsilon, \quad (3.2)$$

em que  $x$  é a variável independente, que é a responsável por atribuir a relação entre  $x$  e  $y$ ,  $y$  é a variável dependente ou fixa, que depende dos outros termos da equação,  $\beta_0$  é o intercepto da reta, é um parâmetro estimado por um modelo de

regressão, representando o valor de  $y$  quando  $x = 0$ ,  $\beta_1$  é um parâmetro estimado que representa o coeficiente angular, ou inclinação, da reta, e  $\epsilon$  o termo de erro, uma variável aleatória que representa a parte da variação em  $y$  que não é explicada pelo modelo.

Utilizando o exemplo de [Montgomery \(2013\)](#), para um maior entendimento sobre modelos de regressão, suponha que podemos fixar o valor de  $x$  e observar o valor correspondente de  $y$ . Agora que  $x$  está fixado, o componente aleatório  $\epsilon$  determina as propriedades de  $y$ . Suponha também, que a média e variância da variável aleatória  $\epsilon$  são 0 e  $\sigma^2$ , respectivamente. Então a resposta média para qualquer valor da variável regressora é

$$E(y|x) = \mu_{y|x} = E(\beta_0 + \beta_1x + \epsilon) = \beta_0 + \beta_1x. \quad (3.3)$$

A variância de  $y$  dado qualquer valor de  $x$  é

$$\text{Var}(y|x) = \sigma_{y|x}^2 = \text{Var}(\beta_0 + \beta_1x + \epsilon) = \sigma^2. \quad (3.4)$$

Então, o verdadeiro modelo de regressão  $\mu_{y|x} = \beta_0 + \beta_1x$  é uma reta de valores médios, ou seja, a altura da reta de regressão para qualquer valor de  $x$  é simplesmente o valor esperado de  $y$  para este  $x$ . A inclinação,  $\beta_1$  pode ser interpretada como a mudança na média de  $y$  por unidade em  $x$ . Além do mais, a variabilidade de  $y$  em algum valor particular de  $x$  é determinada pela variância do componente de erro do modelo,  $\sigma^2$  ([Montgomery, 2013](#)).

### 3.1.1 Tipos de Regressão

Existem diversos tipos de modelos de regressão disponíveis, sendo cada um adequado a diferentes contextos. Entre alguns dos tipos mais comuns de regressão estão:

- **Regressão Linear Simples:**

Como o nome sugere, é utilizada para modelos mais simples, em que há uma relação entre variável dependente e independente que pode ser modelada como uma linha reta. Neste tipo de regressão há apenas uma variável dependente e uma variável independente. O modelo pode ser escrito como:

$$y = \beta_0 + \beta_1x + \epsilon. \quad (3.5)$$

- **Regressão Linear Múltipla:**

É utilizada quando há várias variáveis independentes que podem influenciar  $y$ . Também possui  $k$  parâmetros  $\beta_j, j = 1, 2, \dots, k$  que podem influenciar  $y$ . Esta relação é modelada como um hiperplano em um espaço multidimensional. O modelo

$$y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \dots + \beta_kx_k + \epsilon \quad (3.6)$$

é chamado de modelo de regressão com  $k$  regressores. Os parâmetros  $\beta_j, j = 1, 2, \dots, k$ , são chamados de coeficientes de regressão. Este modelo descreve um hiperplano no espaço  $k$ -dimensional das variáveis regressoras  $x_j$  ([Montgomery, 2013](#)).

- **Regressão Logística:**

Trata-se de um modelo de regressão pertencente à classe de modelos lineares generalizados (Nelder e Wedderburn, 1972), modelos de regressão linear que abordam a violação das suposições de normalidade dos erros do modelo. Assim sendo mais adequados para dados que possuem outras distribuições.

Empregada quando a variável dependente é categórica (por exemplo, sim/não, sucesso/fracasso). Modela a probabilidade de pertencer a uma categoria. Suponha um modelo com a forma

$$y_i = \mathbf{x}'_i \boldsymbol{\beta} + \epsilon_i, \quad (3.7)$$

em que  $\mathbf{x}'_i = [1, x_{i1}, x_{i2}, \dots, x_{ik}]$ ,  $\boldsymbol{\beta}' = [\beta_0, \beta_1, \dots, \beta_k]$ , e a variável resposta  $y_i$  assume os valores 0 ou 1. Neste caso assumimos que a variável resposta  $y_i$  é uma variável aleatória Bernoulli, com a seguinte distribuição de probabilidade:

$y_i$	Probabilidade estimada de sucesso
1	$P(y_i = 1) = \pi_i$
0	$P(y_i = 0) = 1 - \pi_i$

Como  $E(\epsilon_i) = 0$ , o valor esperado da variável resposta é

$$E(y_i) = 1(\pi_i) + 0(1 - \pi_i) = \pi_i. \quad (3.8)$$

Isso implica que

$$E(y_i) = \mathbf{x}'_i \boldsymbol{\beta} = \pi_i. \quad (3.9)$$

Há alguns problemas claros com o modelo apresentado na Equação (3.7). Primeiramente, perceba que se a resposta é binária, os erros de termo  $\epsilon_i$  só podem receber dois valores, sendo esses,

$$\epsilon_i = 1 - \mathbf{x}'_i \boldsymbol{\beta} \text{ quando } y_i = 1; \quad (3.10)$$

$$\epsilon_i = -\mathbf{x}'_i \boldsymbol{\beta} \text{ quando } y_i = 0. \quad (3.11)$$

Consequentemente, os erros neste modelo não podem ser normais. Além disso, a variância do erro não é constante, visto que

$$\sigma_{y_i}^2 = E y_i - E(y_i)^2 = (1 - \pi_i)^2 \pi_i + (0 - \pi_i)^2 (1 - \pi_i) = \pi_i (1 - \pi_i). \quad (3.12)$$

Note que essa última expressão é apenas

$$\sigma_{y_i}^2 = E(y_i)[1 - E(y_i)], \quad (3.13)$$

dado que  $E(y_i) = \mathbf{x}'_i \boldsymbol{\beta} = \pi_i$ . Isto indica que a variância das observações é uma função da média. Logo, há uma limitação na função de resposta, já que

$$0 \leq E(y_i) = \pi_i \leq 1. \quad (3.14)$$

Esta restrição pode causar sérios problemas quanto à escolha da função de resposta linear, que inicialmente havia sido assumida pela Equação (3.7). Isto

significa que seria possível ajustar um modelo aos dados em que os valores preditos da resposta estariam fora do intervalo  $(0, 1)$  (Montgomery, 2013).

Geralmente, quando a variável de resposta é binária, como é o caso deste estudo, há evidência empírica o suficiente indicando que a forma da função de resposta deve ser não linear. Normalmente a função aplicada a este caso é a *função resposta logística*, que possui a forma

$$E(y_i) = \frac{\exp(x'_i\beta)}{1 + \exp(x'_i\beta)} = \frac{1}{1 + \exp(-x'_i\beta)}. \quad (3.15)$$

Uma vantagem desta função é a capacidade de ser facilmente linearizada. Uma abordagem define a parte estrutural do modelo em termos de uma função da média da função de resposta (Montgomery, 2013). Seja

$$\eta = x'\beta \quad (3.16)$$

o preditor linear.  $\eta$  é definido pela transformação

$$\eta = \ln \frac{\pi}{1 - \pi}. \quad (3.17)$$

Esta transformação é comumente chamada de *transformação logit* da probabilidade  $\pi$ . A proporção  $\pi/(1 - \pi)$  na transformação é chamada de *odds*, ou chance. Também podemos chamar esta transformação de *log-odds*. Por exemplo, se um evento tem uma probabilidade  $\pi = 0,3$  então temos uma chance (*odds*)  $0,3/(1 - 0,3) = 0,429$  e uma *log-odds* de  $\ln(0,3/(1 - 0,3)) = -0,847$ .

A regressão logística é especialmente importante em análise preditiva quando a variável resposta é binária, visto sua capacidade de classificação quanto à probabilidade. Logo, tendo consciência de que este trabalho tem como objetivo principal a testagem de modelos quanto sua capacidade preditiva, a regressão logística é a principal técnica que será aplicada nos dados de estudo.

### 3.1.2 Regressão Logística

Considerando as informações apresentadas na seção anterior, informações adicionais acerca da regressão logística se fazem necessárias. Então, devemos entender melhor a estimação de parâmetros deste método.

A forma geral da regressão logística é

$$y_i = E(y_i) + \epsilon_i, \quad (3.18)$$

em que as observações de  $y_i$  são variáveis independentes com distribuição Bernoulli, com valores esperados

$$E(y_i) = \pi_i = \frac{\exp(x'_i\beta)}{1 + \exp(x'_i\beta)}. \quad (3.19)$$

É utilizado o método de máxima verossimilhança para estimar os parâmetros no preditor linear  $x'\beta$ .

Visto que cada observação segue a distribuição Bernoulli, esta função pode ser escrita por

$$f_i(y_i) = \pi_i^{y_i} (1 - \pi_i)^{1-y_i}, \quad i = 1, 2, \dots, n, \quad (3.20)$$

em que cada observação  $y_i$  possui o valor 0 ou 1. Visto que as observações são independentes, a função de verossimilhança é

$$L(y_1, y_2, \dots, y_n, \beta) = \prod_{i=1}^n f_i(y_i) = \prod_{i=1}^n \pi_i^{y_i} (1 - \pi_i)^{1-y_i}. \quad (3.21)$$

É mais fácil trabalhar com a *log-verossimilhança*, isto é, aplicar o logaritmo natural à função de verossimilhança, visto que há transformação de produtos em somas e facilitação do processo de derivação. Logo, após algumas transformações na função, se obtém uma *log-verossimilhança* definida por

$$\ln L(y, \beta) = \sum_{i=1}^n y_i \ln(\pi_i) + \sum_{i=1}^n (n_i - y_i) \ln(1 - \pi_i). \quad (3.22)$$

Ao maximizar a função obtida na Equação (3.22), o parâmetro máximo estimado  $\hat{\beta}$  é obtido (Montgomery, 2013). Esta maximização da função pode ser realizada via métodos numéricos iterativos ou Mínimos Quadrados Reponderados. Felizmente a linguagem de programação **R** realiza automaticamente este processo de maximização para encontrar a *estimativa por máxima verossimilhança* (EMV).

Caso um modelo assuma as suposições anteriores de independência, então, assintoticamente,

$$E(\hat{\beta}) = \beta \quad \text{e} \quad \text{Var}(\hat{\beta}) = (\mathbf{X}'\mathbf{V}\mathbf{X})^{-1}, \quad (3.23)$$

em que  $\mathbf{V}$  é uma matriz diagonal  $n \times n$  contendo a variância estimada de cada observação na diagonal principal. O valor estimado de um preditor linear é  $\hat{\eta}_i = \mathbf{x}'_i \hat{\beta}$ , e o valor ajustado do modelo logístico pode ser escrito como

$$\hat{y}_i = \hat{\pi}_i = \frac{\exp(\hat{\eta}_i)}{1 + \exp(\hat{\eta}_i)} + \frac{\exp(\mathbf{x}'_i \hat{\beta})}{1 + \exp(\mathbf{x}'_i \hat{\beta})} = \frac{1}{1 + \exp(-\mathbf{x}'_i \hat{\beta})}. \quad (3.24)$$

## 3.2 Seleção de Variáveis

### 3.2.1 Colinearidade

Colinearidade se refere a situação em que duas ou mais variáveis preditoras são bastante associadas uma a outra, e a presença deste conceito pode acarretar problemas aos modelos. Visto que pode haver dificuldade em separar efeitos individuais na resposta de variáveis colineares. Sabendo disso, uma maneira simples de detectar a colinearidade entre variáveis, embora limitada, é a observação da matriz de correlação (James et al., 2013).

Levando em conta que a observação da matriz de correlação foi o suficiente para lidar com os problemas de colinearidade nos dados de estudo, foram analisados os coeficientes de correlação de Pearson entre as variáveis dos bancos de dados. O coeficiente de correlação de Pearson, em uma amostra, pode ser escrito como

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}, \quad (3.25)$$

em que  $n$  é o tamanho amostral e  $x_i, y_i$  são os valores individuais da amostra. O valor do coeficiente de correlação está sempre entre 1 e -1. Valores elevados do

coeficiente de correlação indicam forte dependência linear entre as variáveis. Jacob Cohen (Cohen, 1988) sugere classificações quanto a magnitude de efeitos de acordo com o valor do coeficiente.

Na Tabela (3.1) são demonstrados a magnitude de efeitos por faixa de valor de coeficiente de correlação. Vale lembrar que os valores apresentados estão em módulo. Ou seja, o mesmo vale para estes valores negativos, porém com relação linear inversamente proporcional.

Coeficiente de Correlação	Descrição
0.0-0.1	trivial, minúsculo, praticamente zero
0.1-0.3	pequeno, baixo, inferior
0.3-0.5	moderado, mediano
0.5-0.7	largo, alto, superior
0.7-0.9	muito largo, muito alto, enorme
0.9-1	praticamente, quase ou próximo a perfeito

Tabela 3.1: Magnitude de efeitos de acordo com valores absolutos de coeficientes de correlação.

É possível obter uma matriz de correlação ao realizar a Equação (3.25) com forma

	Variável 1	Variável 2	Variável 3
Variável 1	1	0.32	0.25
Variável 2	-0.56	1	0.13
Variável 3	0.87	-0.44	1

Tabela 3.2: Exemplo de matriz de correlação.

A Tabela (3.2) apresenta um exemplo de matriz de correlação, com dados fictícios. Além do mais, (Landis e Koch, 1977) sugerem que um coeficiente de correlação de, em módulo, 0,8 ou superior, indica forte colinearidade entre variáveis. Tendo consciência sobre o conceito de colinearidade e como lidar com este, se pode, enfim, discorrer sobre métodos de seleção de variáveis.

Considerando que este trabalho envolve modelagem, é imprescindível a implementação de métodos de seleção de variáveis para uma modelagem precisa, concisa e confiável. Modelos preditivos podem ser desenvolvidas de muitas maneiras, entretanto, uma estratégia de seleção de variáveis apropriada deve ser seguida em qualquer caso (Chowdhury e Turin, 2020).

Existem outros métodos que são superiores para a detecção de colinearidade entre as variáveis de um modelo. Um destes métodos é o Fator de Inflação de Variância (em inglês, *Variance Inflation Factor*, ou VIF). O VIF é a razão da variância de  $\hat{\beta}_j$  do modelo completo ajustado dividido pela variância de  $\hat{\beta}_j$  caso ajustado sozinho. O menor valor possível para VIF é 1, que indica completa ausência de colinearidade (James et al., 2013). O VIF para cada variável pode ser calculado por

$$VIF(\hat{\beta}_j) = \frac{1}{1 - R_{X_j|X-j}^2}, \quad (3.26)$$

em que  $R_{X_j|X_{-j}}^2$  é o  $R^2$  de uma regressão de  $X_j$  em todos outros preditores. Caso  $R_{X_j|X_{-j}}^2$  seja próximo a 1, então a colinearidade está presente, e o VIF terá um valor alto (James et al., 2013).

Embora o VIF seja um método mais complexo e provavelmente mais adequado que a análise de correlação para a detecção de colinearidade, a segunda técnica foi a utilizada neste trabalho. O motivo se deve pelo fato da redundância das variáveis ser muito explícita no banco de dados, com relação direta entre algumas variáveis. Portanto, com o auxílio da análise de correlação pelo coeficiente de Pearson, foram facilmente detectadas as variáveis que possuem colinearidade. Por exemplo, a variável *nexusTakedowns*, que explica completamente a variável resposta *win*, visto que caso o nexus seja derrubado por sua equipe, o jogador vence a partida.

Neste estudo, foram utilizados dois métodos de seleção de variáveis, o primeiro, regressão por *stepwise* se trata de um método já consolidado que, neste estudo, seleciona variáveis por *Akaike information criteria* (AIC), um estimador de predição do modelo, bem como medidor de qualidade de ajuste (Akaike, 1998). O segundo,  $R^2$  de Regressão Simples, foi um método desenvolvido em conjunto pelo autor e professor orientador deste trabalho, utiliza a medida  $R^2$  como critério de seleção.

### 3.2.2 Regressão por *stepwise*

Este método de seleção de variáveis pode ter como critério de ajuste a minimização do AIC ou maximização da significância do ajuste. O AIC pode ser calculado por

$$\text{AIC} = 2k - 2 \ln(\hat{L}), \quad (3.27)$$

em que  $k$  é o número de parâmetros estimados do modelo e  $\hat{L}$  o valor maximizado da função de verossimilhança. O método *stepwise* possui três modos de operação: *backward elimination*, *forward selection* ou *bidirectional elimination* (ou simplesmente *stepwise regression*) (Montgomery, 2013). Uma seleção de variáveis via regressão por *stepwise* com eliminação *backward* começa o processo com um modelo ajustado por todas as variáveis do banco de dados, e vai retirando-as caso o AIC diminua com a retirada e não impacte expressivamente no ajuste do modelo. Essa eliminação continua até que não possam ser mais removidas variáveis sem afetar significativamente o ajuste do modelo. Ao contrário, pela direção *forward* inicia-se sem variáveis no modelo, e vão sendo adicionadas as variáveis com maior significância para o modelo, uma a uma. Por fim, a *bidirectional elimination*, ao combinar os dois primeiros modos, remove e adiciona variáveis a cada passo conforme a alteração de valor do critério AIC e ajuste do modelo.

Embora seja um método já consolidado e bastante eficiente em análise estatística, a regressão por *stepwise* pode ser computacionalmente exigente em um conjunto de dados que possua um número elevado de variáveis. Todavia, tendo em vista esta limitação, outro método de seleção de variáveis foi desenvolvido e testado nos bancos de dados deste estudo.

### 3.2.3 $R^2$ de Regressão Simples

Neste método, a seleção de variáveis possui como critério único o valor  $R^2$  da regressão. O  $R^2$ , anteriormente citado, é uma medida que é apropriada para medir o ajuste do modelo. É definido pela Equação (3.1). Portanto, o método de seleção

via  $R^2$  de regressão simples, seleciona as variáveis que, ao serem individualmente ajustadas por regressão simples para modelar a variável resposta, como na Equação (3.5), possuam valores iguais ou superiores a um limiar escolhido a priori.

Alguns autores propuseram métodos para utilizar o  $R^2$  para variáveis resposta categóricas. Um destes métodos é o pseudo- $R^2$  de Cox e Snell (Cox e Snell, 1989), que pode ser escrito por

$$R_{CS}^2 = 1 - \left[ \frac{L(0)}{L(\hat{\beta})} \right]^{\frac{2}{n}}, \quad (3.28)$$

em que  $L(0)$  é a verossimilhança logarítmica do modelo nulo,  $L(\hat{\beta})$  é a verossimilhança logarítmica do modelo ajustado, e  $n$  é o tamanho da amostra.

Adicionalmente, Nagelkerke (Nagelkerke, 1991) propôs uma correção para o  $R_{CS}^2$ , visto que o mesmo não pode chegar a um valor igual a 1. Sua correção pode ser definida por

$$R_{Nagelkerke}^2 = \frac{R_{CS}^2}{1 - (L(0))^{\frac{2}{n}}}. \quad (3.29)$$

Tendo em vista que a variável resposta deste estudo é categórica e dicotômica, foi utilizado o  $R_{Nagelkerke}^2$  como parâmetro deste método. Entretanto, caso a variável resposta seja contínua, é indicado utilizar um  $R^2$  como o calculado pela (3.1).

Utilizando este método, apenas as variáveis que explicam alguma parcela da variação na variável resposta, a um valor mínimo de  $R^2$  escolhido a priori, são selecionadas. Deste modo, trata-se de um método simples e limitado, entretanto, é muito rápido computacionalmente, e, para os dados deste estudo, atingiu resultados bastante satisfatórios.

### 3.2.4 Outros métodos

Outros métodos pertinentes para a seleção de variáveis e detecção de colinearidade são a regressão Ridge e Lasso. Embora não tenham sido utilizados neste trabalho, é importante descrevê-los, visto que poderiam ser aplicados a este estudo.

- **Regressão Ridge**

A regressão Ridge pode ser utilizada como guia para selecionar o melhor conjunto de variáveis e mitigar o problema de multicolinearidade em regressão linear. Isto é, regressão Ridge é uma alternativa à regressão por *stepwise* (Hilt e Seegrist, 1977).

Este método foi criado devido à imprecisão de estimativas pelo método dos mínimos quadrados quando há alta correlação entre variáveis, ou seja, colinearidade ou multicolinearidade. O estimador Ridge pode ser escrito por

$$\hat{\beta}^* = A(D + kI)^{-1} A'X'Y', \quad (3.30)$$

em que os termos estão em formato matricial, e  $X$  é a matrix  $n \times p$  de variáveis independentes,  $Y$  é o vetor  $n \times 1$  de observações,  $A$  é a matriz de autovetores de  $X'X$ ,  $D$  é a matriz diagonal de autovalores de  $X'X$ ,  $I$  é a matriz identidade e  $k$  é o parâmetro de regularização ou penalização do estimador (Hilt e Seegrist, 1977).

- **Lasso**

Lasso (sigla para *least absolute shrinkage and selection operator*), proposta por Tibshirani (1996), é uma técnica de seleção de variáveis e regularização de dados. A técnica Lasso pode ser escrita por

$$\hat{\beta}^{\text{lasso}} = \arg \min_{\beta} \left\{ \frac{1}{2n} \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij})^2 + \lambda \sum_{j=1}^p |\beta_j| \right\}, \quad (3.31)$$

em que  $\hat{\beta}^{\text{lasso}}$  é o vetor de coeficientes estimados pelo método Lasso,  $y_i$  é o valor da variável resposta na observação  $i$ ,  $\beta_0$  é o intercepto do modelo de regressão,  $x_{ij}$  é o valor da variável preditora  $j$  na observação  $i$ ,  $\beta_j$  são os coeficientes a serem estimados para cada variável preditora  $j$  e  $\lambda$  é o parâmetro de regularização, que controla a força da penalização (Tibshirani, 1996).

Esses métodos não foram utilizados em prol de testar  $R^2$  de Regressão Simples e regressão por *stepwise* neste trabalho e não utilizar técnicas computacionalmente complexas, para facilitar a replicação dos códigos computacionais escritos.

Agora, entendendo os conceitos de regressão logística, bem como colinearidade e seleção de variáveis, se faz necessário uma última seção para entender técnicas de aprendizado de máquina. Essa seção servirá como base para a análise preditiva, com conceitos de treinamento e teste, medidas preditivas, etc.

### 3.3 Aprendizado de Máquina

Aprendizado de máquina, ou, em inglês, *machine learning*, é uma subárea do campo de estudo de inteligência artificial. Esta tem como objetivo desenvolver algoritmos e modelos capazes de aprender e reconhecer padrões a partir de dados. Já o *statistical learning* se refere a um conjunto de ferramentas para modelagem e entendimento de bancos de dados complexos. É uma área em desenvolvimento na estatística e se mistura com desenvolvimentos nos campos de ciências da computação, e, em particular, *machine learning* (James et al., 2013).

Modelos que utilizam de técnicas de aprendizado de máquina podem ser segmentados em dois tipos: modelos de aprendizado supervisionado e não supervisionado (Boehmke e Greenwell, 2019). O primeiro é definido por modelos em que o conjunto de dados é rotulado, e há a definição clara de entrada e saída desejadas no modelo. É com este tipo de aprendizado que os dados serão testados neste trabalho. Já no segundo tipo, não supervisionado, não são definidas a entrada e saída na modelagem. Desse modo, o objetivo do modelo treinado não supervisionado é explorar e identificar grupos dentro do conjunto de dados, técnica conhecida como *clustering*. Neste estudo, como explicado anteriormente, são desenvolvidos modelos de classificação supervisionados a partir de regressão logística.

Com a intenção de classificar uma variável resposta binária em positiva ou negativa, alguns indicadores são necessários para avaliar o desempenho preditivo de um modelo.

	Predição: Positivo	Predição: Negativo
Verdadeiro	VP	VN
Falso	FP	FN

Tabela 3.3: Matriz de Confusão para Modelos de Classificação

Considere a Tabela (3.3). Nesta tabela há duas categorias de classificação: positivo e negativo. As categorias marcadas como "referência" representam os dados reais observados. Já as marcadas como "predição" representam o valor predito das observações de acordo com os critérios de classificação do modelo. As classificações feitas podem ser separadas em quatro grupos: verdadeiro-positivo (VP), verdadeiro-negativo (VN), falso-positivo (FP) e falso-negativo (FN). Os grupos "verdadeiro" são os dados corretamente classificados de acordo com suas categorias, enquanto os grupos "falso" foram erroneamente classificados.

Tendo estes grupos em vista, existem métricas adequadas para medir o desempenho dos modelos preditivos. Dentre essas temos:

- **Acurácia ou Accuracy:** proporção de itens classificados corretamente pelo total de classificações feitas (Metz, 1978):

$$\text{Acurácia} = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{VN} + \text{FP} + \text{FN}}; \quad (3.32)$$

- **Precisão:** proporção de classificações corretas positivas em relação ao total de atribuições positivas (Olson e Delen, 2008):

$$\text{Precisão} = \frac{\text{VP}}{\text{VP} + \text{FP}}; \quad (3.33)$$

- **Revocação ou Recall:** razão entre verdadeiro-positivos e sua soma com as atribuições falso-negativas (Olson e Delen, 2008):

$$\text{Recall} = \frac{\text{VP}}{\text{VP} + \text{FN}}; \quad (3.34)$$

- **F1-Score:** relação entre os resultados de precisão e *recall* de um modelo, ou seja, relaciona todos os valores classificados positivos com todos os resultados que deveriam ser classificados como positivos (Taha e Hanbury, 2015):

$$\text{F1-Score} = 2 \times \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}}; \quad (3.35)$$

- **Kappa:** sugerido em 1960 por Jacob Cohen (Cohen, 1960), é uma métrica que avalia concordância ou confiabilidade de acordo com as classificações:

$$\kappa = \frac{P_o - P_e}{1 - P_e}, \quad (3.36)$$

em que os termos  $P_o, P_e$  são definidos por

$$P_e = \frac{(VP + FN) \cdot (VP + FP)}{n^2} + \frac{(VN + FP) \cdot (VN + FN)}{n^2}, \quad (3.37)$$

e

$$P_o = \frac{VP + VN}{n}, \quad (3.38)$$

em que  $n$  é o número total de observações.

Por fim, a segmentação de dados também é uma etapa importante em aprendizado de máquina. Consiste em divisão de um ou mais conjuntos de dados em bancos de treinamento e teste. Esta etapa é importante principalmente para a generalização do modelo. Algumas proporções para treinamento e teste são recomendadas, como 60% (treino)–40% (teste), 70%–30%, ou 80%–20%. A proporção escolhida depende do conjunto de dados. Em alguns bancos a proporção 80%–20% pode ser demasiada para treinamento, em outros, adequada ([Boehmke e Greenwell, 2019](#)).

Deste modo, foram apresentados todos os métodos, técnicas e métricas utilizados neste trabalho. O próximo capítulo tratará, exclusivamente, da apresentação de resultados obtidos com a aplicação dos métodos e técnicas explicitados neste capítulo aos conjuntos de dados.

## 4 Resultados

Neste capítulo serão apresentados os resultados das análises realizadas de acordo com as técnicas apresentadas no capítulo anterior. As seções deste capítulo serão divididas, de modo em que, a primeira seção discorrerá sobre o pré-processamento dos dados, incluindo o tema de filtragem de observações e variáveis dos bancos de dados utilizados. Na segunda seção, serão apresentados e comparados os modelos estatísticos utilizados nos bancos de dados de treinamento, bem como seus resultados e estimativas obtidas. Por fim, a última seção apresentará resultados relativos à predição de vitória, obtidos a partir da aplicação de técnicas de aprendizado de máquina aos modelos estatísticos.

### 4.1 Pré-processamento dos Dados

Os dois bancos de dados, referentes aos ranques Desafiante e Ouro IV, passaram por um pré-processamento, que envolve filtragem de variáveis que não poderiam ser incluídas nos modelos. O pré-processamento também envolve a segmentação dos dados em bancos de treinamento e teste.

#### 4.1.1 Filtragem dos Dados

Ambos os bancos de dados obtidos pela API possuíam inicialmente 295 variáveis, mas com a maioria destas sem relevância para este trabalho. Portanto, uma filtragem dos dados, necessária para que as técnicas e métodos sejam corretamente aplicadas, foi realizada seguindo as seguintes etapas em ambos os bancos de dados:

- **Remoção de variáveis esporádicas**

Algumas variáveis eram esporadicamente geradas de acordo com o jogador observado. Estas variáveis são, em sua maioria, categóricas, e relacionadas a fatores não relevantes ao trabalho. A diferença de dimensão das observações geradas acarretava conflitos durante o agrupamento dos dados. Portanto, remover estas variáveis esporádicas foi necessário.

- **Remoção de variáveis redundantes**

Variáveis que possuem relação de colinearidade com outras também foram removidas. Um exemplo destas variáveis removidas foi a variável *nexusTake-downs*, que explica completamente a variável resposta *win*, visto que caso o

nexus seja derrubado por sua equipe, o jogador vence a partida. Como referência, foi utilizado um coeficiente de correlação de Pearson ( $r_{xy}$ ), em módulo, maior ou igual a 0,8.

Após a remoção de variáveis esporádicas ou redundantes, os dois bancos de dados ficaram com 24 variáveis. De um total de 295 variáveis presentes no início do processo, 271 foram removidas.

Alguns exemplos de variáveis esporádicas, que eram geradas ocasionalmente pela ferramenta API, são as variáveis que iniciam com "*challenge*". Estas indicam desafios para os jogadores que não impactam na partida, e nem todos os jogadores possuem os mesmos desafios. Também há as variáveis que iniciam com "*perks*". Estas indicam as runas selecionadas antes da partida. Portanto, não são relevantes para este trabalho.

Removendo a variável de interesse *win*, há 23 possíveis variáveis para entrar nos modelos.

Variável	Média	Desvio Padrão	Mediana	Descrição
assists	7.95	5.89	7.0	Número de assistências do jogador
baronKills	0.11	0.36	< 0.01	Barões abatidos
damageDealtToObjectives	11931.15	14034.81	6639.0	Dano causado a objetivos neutros
damageDealtToTurrets	2656.55	3197.51	1559.5	Dano causado a torres
deaths	5.33	3.04	5.0	Número de vezes que o jogador foi eliminado
dragonKills	0.34	0.79	< 0.01	Número de vezes que o jogador abateu o dragão
kills	5.31	4.25	4.0	Número de eliminações
longestTimeSpentLiving	563.53	283.59	521.0	Maior tempo vivo antes de ser eliminado
neutralMinionsKilled	32.07	59.91	3.0	Total de monstros e objetivos neutros da selva abatidos
objectivesStolen	0.02	0.13	< 0.01	Roubo de objetivos neutros da equipe adversária
objectivesStolenAssists	0.01	0.08	< 0.01	Assistências em roubos de objetivos neutros da equipe adversária
timeCCingOthers	22.66	21.32	19.0	Tempo de controle de grupo em adversários
totalDamageDealtToChampions	17611.71	10601.20	15762.0	Total de dano causado a campeões
totalDamageShieldedOnTeammates	528.80	1732.13	< 0.01	Dano total mitigado aos aliados por escudos
totalDamageTaken	22720.03	1115.85	21216.0	Dano total recebido
totalHeal	7080.98	6803.69	4955.0	Cura total do jogador
totalHealsOnTeammates	686.03	2084.97	< 0.01	Cura total aos aliados
totalMinionsKilled	125.83	90.72	145.0	Total de tropas abatidas
visionScore	29.04	24.94	21.0	Placar de visão

Tabela 4.1: Variáveis quantitativas do banco de dados do ranque Desafiante.

A Tabela (4.1) apresenta as variáveis quantitativas que foram aprovadas no processo de filtragem. Os valores destas variáveis são referentes ao banco de dados do ranque Desafiante.

Variável	Falsos	Verdadeiros	Descrição
firstBloodAssist	4981	619	Jogador colaborou para a primeira eliminação da partida
firstBloodKill	5048	552	Jogador realizou a primeira eliminação da partida
firstTowerAssist	5328	272	Jogador colaborou para a primeira destruição de torre da partida
firstTowerKill	5186	414	Jogador realizou a primeira destruição de torre da partida

Tabela 4.2: Variáveis binárias do banco de dados do ranque Desafiante.

A Tabela (4.2) apresenta as variáveis categóricas binárias que foram aprovadas no processo de filtragem. A contagem de observações destas variáveis é referente ao banco de dados do ranque Desafiante.

Variável	Média	Desvio Padrão	Mediana	Descrição
assists	8.01	5.68	7.0	Número de assistências do jogador
baronKills	0.11	0.35	< 0.01	Barões abatidos
damageDealtToObjectives	12692.55	14059.94	7748.5	Dano causado a objetivos neutros
damageDealtToTurrets	2982.01	3477.17	1861.0	Dano causado a torres
deaths	6.57	3.52	6.0	Número de vezes que o jogador foi eliminado
dragonKills	0.40	0.88	< 0.01	Número de vezes que o jogador abateu o dragão
kills	6.55	5.05	6.0	Número de eliminações
longestTimeSpentLiving	591.26	288.87	548.0	Maior tempo vivo antes de ser eliminado
neutralMinionsKilled	32.38	58.13	4.0	Total de monstros e objetivos neutros da selva abatidos
objectivesStolen	0.03	0.17	< 0.01	Roubos de objetivos neutros da equipe adversária
objectivesStolenAssists	0.01	0.10	< 0.01	Assistências em roubos de objetivos neutros da equipe adversária
timeCCingOthers	26.45	26.65	21.0	Tempo de controle de grupo em adversários
totalDamageDealtToChampions	21406.45	13160.30	19201.5	Total de dano causado a campeões
totalDamageShieldedOnTeammates	383.10	1426.37	< 0.01	Dano total mitigado aos aliados por escudos
totalDamageTaken	27392.79	14315.30	25470.0	Dano total recebido
totalHeal	8138.45	8289.76	5494.0	Cura total do jogador
totalHealsOnTeammates	478.32	1989.54	< 0.01	Cura total aos aliados
totalMinionsKilled	124.35	85.70	135.0	Total de tropas abatidas
visionScore	26.50	20.16	21.0	Placar de visão

Tabela 4.3: Variáveis quantitativas do banco de dados do ranque Ouro IV.

A Tabela (4.3) apresenta as variáveis quantitativas que foram aprovadas no processo de filtragem. Os valores destas variáveis são referentes ao banco de dados do ranque Ouro IV.

Variável	Falsos	Verdadeiros	Descrição
firstBloodAssist	5455	545	Jogador colaborou para a primeira eliminação da partida
firstBloodKill	5417	583	Jogador realizou a primeira eliminação da partida
firstTowerAssist	5756	244	Jogador colaborou para a primeira destruição de torre da partida
firstTowerKill	5554	446	Jogador realizou a primeira destruição de torre da partida

Tabela 4.4: Variáveis binárias do banco de dados do ranque Ouro IV.

A Tabela (4.4) apresenta as variáveis categóricas binárias que foram aprovadas no processo de filtragem. A contagem de observações destas variáveis é referente ao banco de dados do ranque Ouro IV.

Tendo em vista a recorrência e significância das variáveis *assists* e *deaths*, faz-se imprescindível analisar o comportamento destas. Portanto, os seguintes histogramas foram realizados:

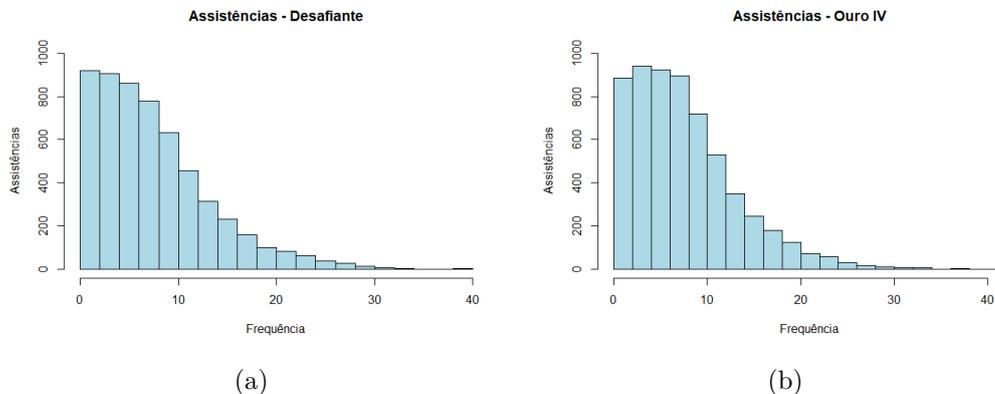


Figura 4.1: Histogramas de assistências (*assists*) para os ranques Desafiante (esquerda) e Ouro IV (direita).

Na Figura (4.1) são apresentados dois histogramas referentes à variável *assists*. A Figura (4.1(a)) apresenta os valores para o banco de dados do ranque Desafiante, enquanto a Figura (4.1(b)) apresenta os valores para o Ouro IV. Percebe-se que a moda do ranque Ouro IV, que fica entre duas e quatro assistências, é maior que a moda do ranque Desafiante, que se encontra entre zero e duas assistências.

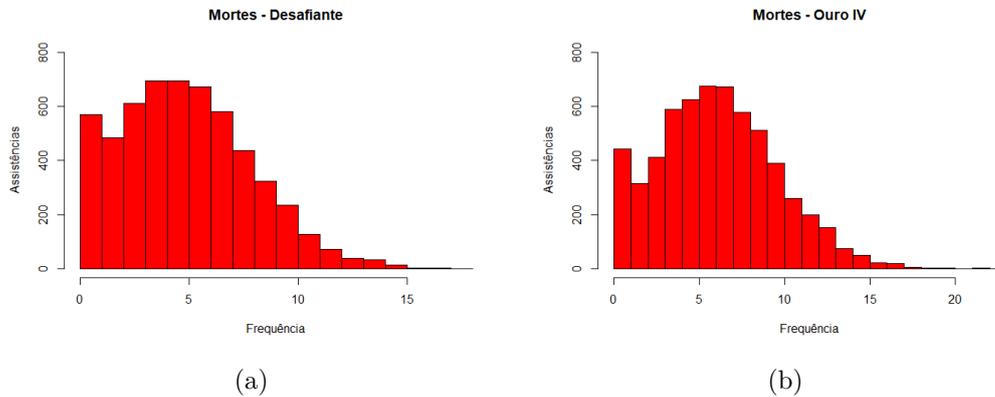


Figura 4.2: Histogramas de mortes (*deaths*) para os ranques Desafiante (esquerda) e Ouro IV (direita).

Na Figura (4.2) são apresentados os dois histogramas referentes à variável *deaths*. Com valores para o banco de dados Desafiante na Figura (4.2(a)) e valores para o banco de dados Ouro IV na Figura (4.2(b)). Observando os histogramas, nota-se que a quantidade de mortes no ranque Desafiante aparenta ser menor do que no ranque Ouro IV.

## 4.1.2 Segmentação dos Dados

Os bancos de dados foram segmentados, cada um, em conjuntos de treinamento e teste, a uma proporção de 70% e 30% do número de observações de cada banco, respectivamente. Os conjuntos de treino totalizaram 3.920 e 4.200 observações para os ranques Desafiante e Ouro IV, respectivamente. Já os conjuntos de teste totalizaram 1.680 e 1.800 observações.

## 4.2 Modelagem

Para a obtenção de resultados preditivos quanto à vitória em partidas de League of Legends, modelos de regressão logística foram ajustados aos conjuntos de observações em partidas do ranque Desafiante e Ouro IV. Entretanto, para a obtenção de resultados confiáveis e provenientes de modelos corretamente adequados aos dados, se faz também necessária uma seleção de variáveis por métodos específicos.

### 4.2.1 Seleção de Variáveis

Foram utilizados dois métodos de seleção de variáveis distintos: regressão por *stepwise*, com direção *backward*, e seleção de variáveis por  $R^2$  de Regressão Simples com parâmetro  $R^2 \geq 0,05$ . Neste trabalho foi utilizado o  $R^2$  de Nagelkerke.

## 4.2.2 Ajuste de Modelos

Foram ajustados, ao total, quatro modelos por regressão logística com ponto de corte padrão, ou seja, 50% de probabilidade. As variáveis utilizadas no modelo são significativas ao nível de 5%, pelo menos.

Adicionalmente, todos os modelos foram testados nos dois bancos, a fim de verificar suas capacidades preditivas e generalização. Isto é, um modelo treinado no banco de dados referente ao ranque Desafiante, por exemplo, também foi testado ao banco de dados referente ao ranque Ouro IV.

## 4.2.3 Comparação de Modelos: Treinamento

Treinando modelos logísticos nos ranques Ouro IV e Desafiante, com o uso dos métodos de regressão por *stepwise* e  $R^2$  de Regressão Simples, foram obtidos, no total, quatro modelos logísticos distintos. Estes, foram comparados entre si por AIC a fim de observar os que possuem melhor ajuste aos dados treinados. Adicionalmente, os modelos também passaram por uma análise preditiva para comparar seus desempenhos, principalmente quanto à acurácia.

Valor Estimado	Erro Padrão	P-valor	Variável	Descrição
-0.7003	0.1093	< 0.05	(Intercept)	Intercepto do modelo
0.2038	0.0088	< 0.05	assists	Número de assistências do jogador
< 0.0001	< 0.0001	< 0.05	damageDealtToObjectives	Dano causado a objetivos neutros
0.0002	< 0.0001	< 0.05	damageDealtToTurrets	Dano causado a torres
-0.4426	0.0179	< 0.05	deaths	Número de vezes que o jogador foi eliminado
0.1001	0.0126	< 0.05	kills	Número de eliminações
<b>AIC:</b>	3438.5	<b>Precisão:</b>	0.8107	<b>IC95% Acurácia =</b> (0.7921, 0.8172)
<b>Acurácia:</b>	0.8048	<b>Recall:</b>	0.7954	
<b>Kappa:</b>	0.6097	<b>F1-Score:</b>	0.803	

Tabela 4.5: Resultados do modelo logístico treinado no ranque Desafiante, com seleção de variáveis via  $R^2$  de Regressão Simples.

Na Tabela (4.5), podemos observar os resultados do modelo treinado no banco de dados do ranque Desafiante, com método de seleção de variáveis por  $R^2$  de Regressão Simples. A variável mais impactante negativamente, por unidade, é *deaths*, em que cada unidade acrescentada equivale a uma redução estimada de 0,4426 *log-odds* para a vitória. Enquanto a variável mais impactante positivamente, por unidade, é *assists*, que aumenta, estimadamente, as *log-odds* para a vitória em 0,2038 por assistência que o jogador efetua. Adicionalmente, nota-se um baixo erro padrão para todas as variáveis observadas.

Valor Estimado	Erro Padrão	P-valor	Variável	Descrição
-0.5453	0.1016	< 0.05	(Intercept)	Intercepto do modelo
0.1728	0.0082	< 0.05	assists	Número de assistências do jogador
< 0.0001	< 0.0001	< 0.05	damageDealtToObjectives	Dano causado a objetivos neutros
0.0002	< 0.0001	< 0.05	damageDealtToTurrets	Dano causado a torres
-0.3782	0.0149	< 0.05	deaths	Número de vezes que o jogador foi eliminado
0.0552	0.0099	< 0.05	kills	Número de eliminações
<b>AIC:</b>	3804.6	<b>Precisão:</b>	0.8087	<b>IC95% Acurácia</b> = (0.7856, 0.8101)
<b>Acurácia:</b>	0.7981	<b>Recall:</b>	0.7810	
<b>Kappa:</b>	0.5962	<b>F1-Score:</b>	0.7946	

Tabela 4.6: Resultados do modelo logístico treinado no ranque Ouro IV, com seleção de variáveis via  $R^2$  de Regressão Simples.

A Tabela (4.6) apresenta resultados das mesmas variáveis selecionadas pelo modelo observado na Tabela (4.5), porém treinado no banco de dados do ranque Ouro IV. A métrica AIC possui maior valor neste modelo se comparada ao modelo treinado nos dados do ranque Desafiante. Este resultado sugere que, por método de seleção de variáveis via  $R^2$  de Regressão Simples, o modelo treinado no ranque Desafiante está melhor ajustado que o treinado no Ouro IV.

Observa-se que ao utilizarmos o método de seleção de variáveis via  $R^2$  de Regressão Simples, obteve-se as mesmas variáveis para os dois bancos de treinamento. Portanto, esta seleção idêntica nos dois bancos de dados, pode indicar consistência do método de seleção utilizado, ou similaridade entre os conjuntos de dados.

Analisando as variáveis, temos valores similares quanto às estimativas. Torna-se claro que a maximização de assistências e dano a torres e objetivos, combinados com uma minimização de mortes, ampliam as chances de vitória. Também é interessante analisar assistências e mortes como um conjunto. Isto é, segundo este modelo logístico, em média, o jogador necessita de cerca de duas assistências para compensar uma morte na partida.

Valor Estimado	Erro Padrão	P-valor	Variável	Descrição
0.8768	0.1786	< 0.05	(Intercept)	Intercepto do modelo
0.2923	0.0138	< 0.05	assists	Número de assistências do jogador
1.1378	0.1969	< 0.05	baronKills	Barões abatidos
< 0.0001	< 0.0001	< 0.05	damageDealtToObjectives	Dano causado a objetivos neutros
0.0003	< 0.0001	< 0.05	damageDealtToTurrets	Dano causado a torres
-0.5125	0.0226	< 0.05	deaths	Número de vezes que o jogador foi eliminado
0.2686	0.1113	< 0.05	dragonKills	Número de vezes que o jogador abateu o dragão
0.2162	0.0194	< 0.05	kills	Número de eliminações
-0.0026	0.0002	< 0.05	longestTimeSpentLiving	Maior tempo vivo antes de ser eliminado
-0.0101	0.0020	< 0.05	neutralMinionsKilled	Total de monstros e objetivos neutros da selva abatidos
1.1148	0.6228	> 0.05	objectivesStolenAssists	Assistências em roubos de objetivos neutros da equipe adversária
-0.0032	0.0023	> 0.05	timeCCingOthers	Tempo de controle de grupo em adversários
-0.0001	< 0.0001	< 0.05	totalDamageDealtToChampions	Total de dano causado a campeões
-0.0016	< 0.0001	< 0.05	totalDamageShieldedOnTeammates	Dano total mitigado aos aliados por escudos
-0.0001	< 0.0001	> 0.05	totalHealsOnTeammates	Cura total aos aliados
0.0019	0.0009	< 0.05	totalMinionsKilled	Total de tropas abatidas
<b>AIC:</b>	3039.1	<b>Precisão:</b>	0.8483	<b>IC95% Acurácia</b> = (0.8243, 0.8477)
<b>Acurácia:</b>	0.8362	<b>Recall:</b>	0.8189	
<b>Kappa:</b>	0.6724	<b>F1-Score:</b>	0.8333	

Tabela 4.7: Resultados do modelo logístico treinado no ranque Desafiante, com seleção de variáveis via regressão por *stepwise*.

Na Tabela (4.7), são apresentados os resultados de um modelo treinado no banco de dados Desafiante, com seleção de variáveis realizada via regressão por *stepwise*. Nota-se pela tabela que há muito mais variáveis significativas neste modelo, ao primeiro, que utiliza seleção por  $R^2$  de Regressão Simples. É interessante a adição de variáveis de abates de objetivos, como *baronKills* e *dragonKills*, que possuem valores

positivos estimados de *log-odds* bastante elevados, em comparação à maioria das variáveis significativas. Estes valores sugerem que, em partidas no ranque Desafiante, o abate desses objetivos neutros é bastante impactante para a vitória. Entretanto, o erro padrão de *dragonKills* é bastante acentuado, enquanto o erro de *baronKills* é muito baixo, sugerindo a consistência do impacto na vitória da primeira variável, ao contrário da segunda. Por fim, a relação entre as variáveis de eliminações de jogadores neste modelo é bastante similar aos dois anteriores.

Valor Estimado	Erro Padrão	P-valor	Variável	Descrição
0.4357	0.1378	< 0.05	(Intercept)	Intercepto do modelo
0.2228	0.0121	< 0.05	assists	Número de assistências do jogador
0.6738	0.1664	< 0.05	baronKills	Barões abatidos
< 0.0001	< 0.0001	< 0.05	damageDealtToObjectives	Dano causado a objetivos neutros
0.0003	< 0.0001	< 0.05	damageDealtToTurrets	Dano causado a torres
-0.3901	0.0199	< 0.05	deaths	Número de vezes que o jogador foi eliminado
0.3787	0.0959	< 0.05	dragonKills	Número de vezes que o jogador abateu o dragão
0.1457	0.0157	< 0.05	kills	Número de eliminações
-0.0018	0.0002	< 0.05	longestTimeSpentLiving	Maior tempo vivo antes de ser eliminado
-0.0075	0.0017	< 0.05	neutralMinionsKilled	Total de monstros e objetivos neutros da selva abatidos
-1.1598	0.2678	> 0.05	objectivesStolen	Roubos de objetivos neutros da equipe adversária
-0.5933	0.4013	> 0.05	objectivesStolenAssists	Assistências em roubos de objetivos neutros da equipe adversária
-0.0031	0.0020	> 0.05	timeCCingOthers	Tempo de controle de grupo em adversários
-0.0001	< 0.0001	< 0.05	totalDamageDealtToChampions	Total de dano causado a campeões
-0.0001	< 0.0001	< 0.05	totalDamageShieldedOnTeammates	Dano total mitigado aos aliados por escudos
-0.0001	< 0.0001	< 0.05	totalDamageTaken	Dano total recebido
-0.0001	< 0.0001	> 0.05	totalHealsOnTeammates	Cura total aos aliados
0.0094	0.0028	< 0.05	visionScore	Placar de visão
<b>AIC:</b>	3508.4	<b>Precisão:</b>	0.8222	<b>IC95% Acurácia</b> = (0.8090, 0.8324)
<b>Acurácia:</b>	0.8210	<b>Recall:</b>	0.8190	
<b>Kappa:</b>	0.6419	<b>F1-Score:</b>	0.8206	

Tabela 4.8: Resultados do modelo logístico treinado no ranque Ouro IV, com seleção de variáveis via regressão por *stepwise*.

Observa-se na Tabela (4.8) os resultados das variáveis quando o modelo treinado no Ouro IV tem suas variáveis selecionadas via método de regressão por *stepwise*. Nota-se a adição de uma variável significativa interessante, *visionScore*. Esta variável mede o controle de visão do mapa, seja colocando ou destruindo sentinelas de visão pelo mapa. Estas sentinelas de visão são itens que revelam a área ao redor delas sem a necessidade de um jogador estar presente na região.

Os valores estimados das demais variáveis continuam muito similares aos apresentado na Tabela (4.7). Adicionalmente, a medida AIC foi muito reduzida. Portanto, é correto afirmar que, apesar deste modelo ser treinado no Ouro IV, se comparado ao modelo treinado no ranque Desafiante, não é muito superior em ajuste ao conjunto de dados do ranque Ouro IV.

Banco de Treinamento	Método de Seleção	AIC	Acurácia	Número de Variáveis
Desafiante	Regressão por <i>stepwise</i>	3039.1	0.8362	16
Desafiante	R <sup>2</sup> de Regressão Simples	3438.5	0.8048	6
Ouro IV	Regressão por <i>stepwise</i>	3508.4	0.8210	18
Ouro IV	R <sup>2</sup> de Regressão Simples	3804.6	0.7981	6

Tabela 4.9: Comparação entre os ajustes e acurácia dos quatro modelos.

Na Tabela (4.9), é apresentado os resultado da métrica AIC por modelo explorado. Os dados estão ordenados por valor AIC, do menor para o maior valor. Portanto, considerando o AIC como medida principal para comparação de modelos, é indicado que o modelo que melhor se ajusta aos seus dados é Desafiante, utilizando o método de regressão por *stepwise*.

Apesar de existirem mais variáveis nos modelos que utilizam de regressão por *stepwise*, os valores AIC são menores quando comparados aos modelos que foram treinados nos mesmos bancos de dados, mas que utilizam do método de  $R^2$  de Regressão Simples para selecionar variáveis. Isto indica que, estes modelos que selecionam variáveis por *stepwise*, são melhores ajustados aos dados em relação aos modelos que fazem uso de seleção de variáveis por  $R^2$  de Regressão Simples.

Também é evidente que os menores valores de AIC se encontram em modelos treinados no banco de dados referente ao ranque Desafiante. Na próxima seção serão apresentados resultados de análises preditivas que utilizaram dos modelos comentados nesta seção.

### 4.3 Predição

Nesta seção serão apresentadas as análises preditivas obtidas ao utilizar os modelos já apresentados na seção anterior. Estes também serão comparados de acordo com seus desempenhos preditivos. Os resultados serão apresentados em formas de matrizes de confusão.

	Referência: Derrota	Referência: Vitória
Predição: Derrota	702	167
Predição: Vitória	137	674
Acurácia = 0.8190	Recall = 0.8014	
Kappa = 0.6381	F1-Score = 0.8160	
Precisão = 0.8311	IC95% Acurácia = (0.7998, 0.8372)	

Tabela 4.10: Resultados de predição do modelo treinado e testado no Desafiante, com seleção de variáveis via  $R^2$  de Regressão Simples.

A Tabela (4.10) apresenta a matriz de confusão para um modelo treinado e testado no ranque Desafiante. No geral, os valores das métricas apresentadas estão bons de um ponto de vista preditivo. Com 81,90% de acurácia, é possível afirmar que este modelo, apesar de simples, prediz bem a vitória. Adicionalmente, o modelo possui um valor *kappa* de 0,6381, o que indica um concordância considerável entre os critérios de avaliação. Um *F1-Score* de 0,8160 também indica um bom equilíbrio entre *precisão* e *recall*, assim sugerindo que o modelo possui uma boa capacidade de corretamente categorizar "vitória" e evitar falsos positivos.

	Referência: Derrota	Referência: Vitória
Predição: Derrota	742	194
Predição: Vitória	158	706
Acurácia = 0.8044	Recall = 0.7844	
Kappa = 0.6089	F1-Score = 0.8004	
Precisão = 0.8171	IC95% Acurácia = (0.7854, 0.8225)	

Tabela 4.11: Resultados de predição do modelo treinado e testado no Ouro IV, com seleção de variáveis via  $R^2$  de Regressão Simples.

Na Tabela (4.11) temos resultados parecidos com a Tabela (4.10). Entretanto, todos os valores estão um pouco abaixo dos obtidos pelo modelo anterior. Todavia,

ainda são valores satisfatórios, com uma acurácia de 80,44%. Ainda comparando aos resultados da tabela anterior, percebe-se que o valor de *kappa* foi o mais afetado negativamente, com uma diminuição de 0,0292. Isto indica uma redução na concordância entre os critérios de avaliação.

No geral, é possível afirmar que os modelos treinados, com utilização de método de seleção de variáveis via  $R^2$  de Regressão Simples, têm uma boa capacidade preditiva. Visto os resultados preditivos suficientemente satisfatórios apesar da simplicidade dos modelos.

	Referência: Derrota	Referência: Vitória
Predição: Derrota	742	194
Predição: Vitória	158	706
Acurácia = 0.8044	Recall = 0.7844	
Kappa = 0.6089	F1-Score = 0.8004	
Precisão = 0.8171	IC95% Acurácia = (0.7854, 0.8225)	

Tabela 4.12: Resultados de predição do modelo treinado no Desafiante e testado no Ouro IV, com seleção de variáveis via  $R^2$  de Regressão Simples.

Pela Tabela (4.12), percebe-se que ao ser testado no ranque Ouro IV, o modelo treinado no Desafiante, que utiliza do método de seleção  $R^2$  de Regressão Simples, retorna os mesmos resultados observados que o modelo treinado e testado no Ouro IV. Isso se deve ao fato de que foram selecionadas as mesmas variáveis, pelo método  $R^2$  de Regressão Simples, para ambos os bancos de dados.

	Referência: Derrota	Referência: Vitória
Predição: Derrota	702	167
Predição: Vitória	137	674
Acurácia = 0.8190	Recall = 0.8014	
Kappa = 0.6381	F1-Score = 0.8160	
Precisão = 0.8311	IC95% Acurácia = (0.7998, 0.8372)	

Tabela 4.13: Resultados de predição do modelo treinado no Ouro IV e testado no Desafiante, com seleção de variáveis via  $R^2$  de Regressão Simples.

Evidencia-se, pela Tabela (4.13), a mesma ocorrência da Tabela (4.12). Portanto, mesmo que um cruzamento de banco de dados tenha sido realizado para treinamento e teste, o fato de as variáveis selecionadas via  $R^2$  de Regressão Simples terem sido as mesmas é uma das razões pela qual as matrizes de confusão apresentam os mesmos valores quanto as suas predições.

	Referência: Derrota	Referência: Vitória
<b>Predição: Derrota</b>	719	132
<b>Predição: Vitória</b>	120	709
<b>Acurácia</b> = 0.8500	<b>Recall</b> = 0.8430	
<b>Kappa</b> = 0.7000	<b>F1-Score</b> = 0.8491	
<b>Precisão</b> = 0.8552	<b>IC95% Acurácia</b> = (0.832, 0.8667)	

Tabela 4.14: Resultados de predição do modelo treinado e testado no Desafiante, com seleção de variáveis via regressão por *stepwise*.

A partir da Tabela (4.14), se começa a ver os resultados dos modelos que utilizam do método de seleção de variáveis via regressão por *stepwise*. Com as medidas preditivas possuindo valores superiores às dos modelos anteriores, se faz perceptível que este primeiro modelo que faz uso de *stepwise* é superior aos que utilizam de R<sup>2</sup> de Regressão Simples como método de seleção de variáveis. É possível evidenciar essa superioridade, por exemplo, ao comparar os valores de acurácia deste modelo com o apresentado na Tabela (4.10). Visto que há uma diferença de 0,031 pontos percentuais entre os valores.

	Referência: Derrota	Referência: Vitória
<b>Predição: Derrota</b>	764	172
<b>Predição: Vitória</b>	136	728
<b>Acurácia</b> = 0.8289	<b>Recall</b> = 0.8089	
<b>Kappa</b> = 0.6578	<b>F1-Score</b> = 0.8254	
<b>Precisão</b> = 0.8426	<b>IC95% Acurácia</b> = (0.8107, 0.846)	

Tabela 4.15: Resultados de predição do modelo treinado no Desafiante e testado no Ouro IV, com seleção de variáveis via regressão por *stepwise*.

Observando os resultados da Tabela (4.15), se torna evidente a superioridade na capacidade preditiva dos modelos que utilizam o *stepwise* como método de seleção. Entretanto, apesar do bom desempenho deste modelo testado no ranque Ouro IV, com uma acurácia de 82,89% e *recall* de 0,8089, ainda se demonstra inferior em capacidade preditiva quando comparado ao mesmo modelo testado no Desafiante.

	Referência: Derrota	Referência: Vitória
<b>Predição: Derrota</b>	754	167
<b>Predição: Vitória</b>	146	733
<b>Acurácia</b> = 0.8261	<b>Recall</b> = 0.8144	
<b>Kappa</b> = 0.6522	<b>F1-Score</b> = 0.8241	
<b>Precisão</b> = 0.8339	<b>IC95% Acurácia</b> = (0.8078, 0.8434)	

Tabela 4.16: Resultados de predição do modelo treinado e testado no Ouro IV, com seleção de variáveis via regressão por *stepwise*.

O modelo observado na Tabela (4.16) apresenta-se inferior em acurácia ao modelo, também testado no conjunto de dados do Ouro IV, apresentado na Tabela (4.15). Todavia, trata-se de uma diferença de 0,28% de acurácia, ou cinco classificações erradas. Entretanto, este modelo possui um *recall* maior do que o anterior.

Isso significa que, nestes dados, o modelo é superior para identificar e classificar corretamente casos na categoria "vitória".

	Referência: Derrota	Referência: Vitória
Predição: Derrota	718	133
Predição: Vitória	121	708
Acurácia = 0.8488	Recall = 0.8419	
Kappa = 0.6976	F1-Score = 0.8479	
Precisão = 0.8540	IC95% Acurácia = (0.8308, 0.8656)	

Tabela 4.17: Resultados de predição do modelo treinado no Ouro IV e testado no Desafiante, com seleção de variáveis via regressão por *stepwise*.

Considerando os resultados apresentados na Tabela (4.17), é possível afirmar que, apesar de treinado no Ouro IV, este modelo possui uma capacidade preditiva surpreendentemente satisfatória quando comparado ao modelo da Tabela (4.14), com apenas dois erros de classificação a mais que o modelo que possui as mesmas condições, porém treinado no próprio ranque Desafiante.

## 4.4 Comparação entre Modelos

Com a apresentação dos resultados individuais da análise preditiva para quatro modelos, que realizam cruzamento de dados de treinamento e teste, faz-se necessária uma comparação coletiva entre todos os resultados obtidos. Portanto, a fim de concluir sobre os desempenhos destes modelos, quando comparados entre si, serão analisadas as diferenças entre seus resultados.

Banco de Treinamento	Banco de Teste	Método de Seleção	Acurácia	Kappa	Precisão	Recall	F1-Score
Desafiante	Desafiante	Regressão por <i>stepwise</i>	0.8500	0.7000	0.8552	0.8430	0.8491
Ouro IV	Desafiante	Regressão por <i>stepwise</i>	0.8488	0.6976	0.8540	0.8419	0.8479
Desafiante	Ouro IV	Regressão por <i>stepwise</i>	0.8289	0.6578	0.8426	0.8089	0.8254
Ouro IV	Ouro IV	Regressão por <i>stepwise</i>	0.8261	0.6522	0.8339	0.8144	0.8241
Desafiante	Desafiante	R <sup>2</sup> de Regressão Simples	0.8190	0.6381	0.8311	0.8014	0.8160
Ouro IV	Desafiante	R <sup>2</sup> de Regressão Simples	0.8190	0.6381	0.8311	0.8014	0.8160
Ouro IV	Ouro IV	R <sup>2</sup> de Regressão Simples	0.8044	0.6089	0.8171	0.7844	0.8004
Desafiante	Ouro IV	R <sup>2</sup> de Regressão Simples	0.8044	0.6089	0.8171	0.7844	0.8004

Tabela 4.18: Comparação de desempenho preditivo dos quatro modelos após serem testados em ambos os bancos de dados.

Analisando os resultados das medidas de predição na Tabela (4.18), ordenada de forma crescente pela acurácia, é seguro afirmar que, nos dados deste estudo, os modelos logísticos com variáveis selecionadas via regressão por *stepwise* foram os que apresentaram melhor desempenho em relação aos que possuem variáveis selecionadas via R<sup>2</sup> de Regressão Simples. Os modelos com uso de *stepwise* apresentaram melhor desempenho em todas as medidas preditivas estudadas, independentemente do banco de dados em que foram testados.

O modelo treinado no banco de dados referente ao ranque Desafiante, com utilização do método de regressão por *stepwise*, representado pela primeira e terceira linhas da Tabela (4.18), foi o que apresentou melhor desempenho entre todos os quatro modelos comparados. Fato este evidenciado pelas medidas preditivas, que são todas superiores aos outros modelos.

A fim de comparação, a métrica que mais variou entre os modelos foi o *kappa*, com diferença de 0,0911 entre o maior e menor valor. O modelo com maior *kappa*, que também apresenta a maior acurácia, apresentou um valor de 0,7. Enquanto o com menor *kappa*, que, do mesmo modo possui a menor acurácia, apresentou um valor de 0,6089. Observa-se ainda que os modelos testados no ranque Desafiante, na regressão por *stepwise*, obtiveram *kappa* de, respectivamente, 0,7 para o modelo treinado no banco desafiante e 0,6976 para o modelo treinado no ranque Ouro IV. Por outro lado, ambos os modelos, ao serem testados no ranque Ouro IV, obtiveram *kappa* inferior. Isso apresenta uma maior concordância das predições dos modelos quando testados no ranque Desafiante. No geral, é correto afirmar que os modelos desenvolvidos neste trabalho possuem um bom desempenho. Visto que, mesmo o modelo com os piores resultados, notadamente o modelo que utilizou  $R^2$  de Regressão Simples e foi testado no ranque Ouro IV, possui uma acurácia superior a 80%.

## 5 Conclusão

A partir dos resultados observados no capítulo anterior, conclui-se que as técnicas testadas para modelagem utilizadas neste trabalho, aplicadas ao conjunto de dados de partidas dos ranques Ouro IV e Desafiante, obtidos a partir da API da Riot Games, foram superiores às expectativas para prever a vitória de partidas por jogador em League of Legends. Nesses bancos de dados, os modelos logísticos treinados que utilizaram da técnica de regressão por *stepwise* para selecionar variáveis foram amplamente superiores aos que utilizaram do método de  $R^2$  de Regressão Simples.

Entretanto, o método de seleção de variáveis via  $R^2$  de Regressão Simples, que foi desenvolvido e testado durante a realização deste estudo, se apresentou como uma alternativa viável nos bancos de dados estudados, às técnicas já consolidadas na área de análise de dados. Visto que, embora seja matematicamente simples, e, não apresente desempenho superior à já conhecida regressão por *stepwise*, é superior em termos de processamento computacional, podendo ser fácil e rapidamente implementado a outros bancos de dados. Todavia, mais testes com esse método em diferentes contextos e conjuntos de dados necessitam ser realizados para entender o alcance de eficiência deste método.

Quanto aos resultados dos modelos na análise preditiva, pode-se afirmar que são suficientemente satisfatórios. Utilizando os resultados dos modelos com variáveis selecionadas via regressão por *stepwise*, observa-se que há sim diferenças entre as variáveis significativas para a vitória nos ranques Ouro IV e Desafiante. Os modelos sugerem para os dois ranques a minimização de mortes por partida e maximização de assistências, eliminações, abates de objetivos neutros e dano a estruturas. Adicionalmente, os modelos também sugerem que os jogadores presentes no ranque Ouro IV atentem-se ao placar de visão, enquanto os jogadores no Desafiante devem considerar o abate expressivo de tropas.

Algumas limitações de estudo são evidentes, e é sugerido que sejam consideradas em trabalhos futuros similares. Este trabalho leva em consideração apenas variáveis individuais por jogadores, que se limitam apenas a fatores dentro de partidas. Portanto, o desenvolvimento de variáveis que analisem desempenho coletivo e estudo do jogador, pode resultar em valores mais precisos. Também não foram analisadas muitas interações do jogo, como sinergias e interações entre campeões e itens. Apesar de que, existem variáveis obtidas pela própria API que retornam informações sobre os campeões e itens, mas não existem variáveis de interação entre os mesmos. Além do mais, os dados utilizados neste trabalho são provenientes de contas de jogadores pertencentes ao servidor EUW. Tendo em vista que é possível que existam estilos de jogo diferentes entre servidores, as variáveis significativas para alcançar a vitória,

analisadas neste estudo, possivelmente são diferentes em outros servidores.

No geral, apesar de um dos modelos apresentar superioridade aos demais quanto às medidas de predição, não se pode entendê-lo como absoluto. Isso se deve ao fato de que apenas dois ranques foram avaliados neste trabalho. Portanto, generalizar os resultados deste modelo para todos os ranques e jogadores seria errôneo. Além do mais, a diferença entre o melhor e pior modelos que utilizaram da seleção de variáveis por *stepwise*, é de apenas 2,39% de acurácia. Suponha então, que um jogador ativo jogue 100 partidas em determinado período. Neste caso, a diferença de 2,39% de acurácia para este jogador, representaria três partidas de erro de classificação.

Por fim, conclui-se que modelos logísticos com o uso de técnicas adequadas de seleção de variáveis foram adequados para este tipo de análise em League of Legends. Tendo em vista os resultados satisfatórios de análise preditiva, com o modelo com menor desempenho apresentando uma acurácia de 80,44% no ranque Ouro IV. Vale mencionar que os modelos testados no ranque Desafiante e, com variáveis selecionadas por método de  $R^2$  de Regressão Simples, obtiveram um limite superior da acurácia de 83,72%. Enquanto que o melhor modelo *stepwise*, testado no mesmo banco de dados, obteve um limite inferior da acurácia de 83,2%.

## Referências Bibliográficas

- Active Player (2023). League of Legends Live Player Count. <https://activeplayer.io/league-of-legends/>. [Acesso em 20 dez. 2023].
- Activision Blizzard (2023). Heroes of the Storm. <https://heroesofthestorm.blizzard.com/>. [Acesso em 19 dez. 2023].
- Akaike, H. (1998). *Information Theory and an Extension of the Maximum Likelihood Principle*, pages 199–213. Springer New York, New York, NY.
- Boehmke, B. e Greenwell, B. (2019). *Hands-On Machine Learning with R*. Chapman and Hall/CRC eBooks.
- Chowdhury, M. e Turin, T. C. (2020). Variable selection strategies and its importance in clinical prediction modelling. *Family Medicine and Community Health*, 8(1):e000262.
- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1).
- Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences (second edition)*. Lawrence Erlbaum Associates.
- Cox, D. R. e Snell, E. J. (1989). *Analysis of Binary Data*. Chapman and Hall.
- Google (2023). Google Scholar. <https://scholar.google.com/>. [Acesso em 21 ago. 2023].
- Hilt, D. E. e Seegrist, D. W. (1977). Ridge: a computer program for calculating ridge regression estimates.
- James, G., Witten, D., Hastie, T., e Tibshirani, R. (2013). *An Introduction to Statistical Learning*, volume 112. Springer.
- Kim, Y. J., Engel, D., Woolley, A. W., Lin, J. Y.-T., McArthur, N., e Malone, T. W. (2017). What makes a strong team? *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*.
- Landis, J. R. e Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159.
- League of Graphs (2023). Rank distribution - League of Legends. <https://leagueofgraphs.com/rankings/rank-distribution>. [Acesso em 16 ago. 2023].

- Mangeloja, E. (2019). Economics of esports. In Takala, T., Auvinen, T., Vesa, M., Tienari, J., Sajasalo, P., Heikkinen, S., Mills, J. H., e Kallinen-Kuisma, M., editors, *Electronic Journal of Business Ethics and Organization Studies*, volume 24, pages 34–42. Jyväskylän yliopisto, Business and Organization Ethics Network (BON).
- Metz, C. E. (1978). Basic principles of roc analysis. *Seminars in Nuclear Medicine*, 8(4):283–298.
- Montgomery, D. C. (2013). *Introduction to Linear Regression Analysis, Fifth Edition Set*. Wiley.
- Mora-Cantalops, M. e Sicilia, M.-A. (2018). Exploring player experience in ranked league of legends. *Behaviour & Information Technology*, 37(12):1224–1236.
- Nagelkerke, N. J. (1991). A note on a general definition of the coefficient of determination. *Biometrika*, 78(3).
- Nelder, J. A. e Wedderburn, R. W. M. (1972). Generalized Linear Models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3).
- Olson, D. L. e Delen, D. (2008). *Advanced Data Mining Techniques*. Springer.
- Redação do Globo Esporte (2022). Galvão narrou lances de LoL e brincou com Tetra Kill: relembre. <https://ge.globo.com/esports/noticia/2022/12/18/galvao-narrou-lances-de-lol-e-brincou-com-tetra-kill-relembre.ghtml>. [Acesso em 10 jan. 2024].
- Riot Games (2021). Notas de Atualização - League of Legends. <https://www.leagueoflegends.com/pt-br/news/tags/patch-notes/>. [Acesso em 20 dez. 2023].
- Riot Games (2023). Riot Developer Portal. <https://developer.riotgames.com/apis>. [Acesso em 30 jul. 2023].
- Steam (2023a). Dota 2 on Steam. [https://store.steampowered.com/app/570/Dota\\_2/](https://store.steampowered.com/app/570/Dota_2/). [Acesso em 19 dez. 2023].
- Steam (2023b). SMITE® on Steam. <https://store.steampowered.com/app/386360/SMITE/>. [Acesso em 19 dez. 2023].
- Taha, A. A. e Hanbury, A. (2015). Metrics for evaluating 3d medical image segmentation: Analysis, selection, and tool. *BMC Medical Imaging*, 15(1).
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.
- TwitchMetrics (2024). Métricas de loltyler1. <https://www.twitchmetrics.net/c/51496027-loltyler1>. [Acesso em 30 jan. 2024].