**FACULDADE FIA DE ADMINISTRAÇÃO E NEGÓCIOS**

**Fundação Instituto de Administração**

Julian Philippe Monteiro Mussumeci

**TÉCNICAS DE DATA MINING PARA ANÁLISE DE PARTIDAS RANQUEADAS DE LEAGUE OF LEGENDS**

São Paulo

2017

Julian Philippe Monteiro Mussumeci

Pós Graduação em Análise de Dados e Data Mining

**TÉCNICAS DE DATA MINING PARA ANÁLISE DE PARTIDAS RANQUEADAS DE LEAGUE OF LEGENDS**

Trabalho de Conclusão de Curso apresentado à Fundação Instituto de Administração – FIA como requisito para obtenção do certificado de conclusão do curso de Pós-Graduação Latu Sensu – Análise de Dados e Data Mining.

Orientadores: Prof. Dra. Alessandra Montini

Prof. Dr. Adolpho W. P.Canton

Julian Philippe Monteiro Mussumeci

São Paulo

2017

**FOLHA DE APROVAÇÃO**

**Técnicas de Data Mining para análise de partidas ranqueadas de League Of Legends**

\_\_/\_\_/\_\_

Banca examinadora:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. (Nome do orientador)

Afiliações

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. (Nome do professor avaliador)

Afiliações

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. (Nome do professor avaliador)

Afiliações

RESUMO

Este trabalho faz o estudo de partidas ranqueadas do jogo League of Legends com o objetivo de utilizar ferramentas estatísticas e Data Mining para prever a chance de um time vencer ou perder uma partida de acordo com uma coleção pré-definida de variáveis. Partindo pela análise exploratória, foi possível identificar padrões e tendências nos dados analisados. Utilizando métodos de seleção de variáveis foi possível identificar quais variáveis respondem melhor o target definido e utilizando a técnica de regressão logística foi possível obter um resultado positivo, de acordo com as variáveis disponíveis. Para realizar as análises e aplicar o modelo de regressão logística, foram utilizados os softwares R e Excel.

**Palavras-chave:** Regressão Logística, League of Legends, Data Mining, R

Sumário

[1 INTRODUCAO 3](#_Toc490000418)

[2 OBJETIVO 5](#_Toc490000419)

[3 EXTRAÇÃO DE DADOS 6](#_Toc490000420)

[4 BASE DE DADOS E VARIÁVEIS 7](#_Toc490000421)

[4.1 Variáveis qualitativas 7](#_Toc490000422)

[4.2 Variáveis quantitativas 7](#_Toc490000423)

[4.3 Transformação de variáveis qualitativas 8](#_Toc490000424)

[5 ANÁLISE EXPLORATÓRIA DOS DADOS 10](#_Toc490000425)

[5.1 Análise exploratória de dados qualitativos 10](#_Toc490000426)

[5.2 Variável Duração 11](#_Toc490000427)

[5.3 Variáveis teamBlue, teamPurpple 14](#_Toc490000428)

[5.4 Variáveis firstBlood, firstTower, firstInhibitor, firstDragon e firstBaron 15](#_Toc490000429)

[5.5 Análise exploratória de dados quantitativos 16](#_Toc490000430)

[5.6 Variáveis towerKills, inhibitorKills, dragonKills, baronKills 17](#_Toc490000431)

[5.7 Variáveis kills, deaths e assists 18](#_Toc490000432)

[5.8 Variáveis minionsKilled, neutralMinionsKilled 19](#_Toc490000433)

[5.9 Variáveis neutralMinionsKilledTeamJungle e neutralMinionsKilledEnemyJungle 20](#_Toc490000434)

[5.10 Variáveis goldEarned e goldSpent 21](#_Toc490000435)

[5.11 Variáveis wardsPlaced, wardsKilled e visionWardsBoughtInGame 22](#_Toc490000436)

[5.12 Variáveis doubleKills, tripleKills, quadraKills, pentaKills e killingSprees 23](#_Toc490000437)

[6 MODELAGEM 24](#_Toc490000438)

[6.1 Seleção de variáveis 24](#_Toc490000439)

[6.2 Regressão Logística 25](#_Toc490000440)

[6.3 Matriz de confusão 27](#_Toc490000441)

[6.4 Teste do modelo 29](#_Toc490000442)

[7 CONCLUSÃO 30](#_Toc490000443)

[8 REFERÊNCIAS 31](#_Toc490000444)

# INTRODUCAO

O eSports é um termo associado a competições de jogos eletrônicos e vem crescendo muito nos últimos anos com o auxílio de novas tecnologias que permitem a exibição dos jogos profissionais e amadores via streaming, como por exemplo a plataforma Twicht.tv, onde são disponibilizado diversos jogos eletrônicos em tempo real.

Entre os jogos com maior popularidade no eSports estão jogos de estilo FPS (First Person Shooter) como Counter-Strike: Global Offensive, jogos de luta como Mortal Kombat XL e Street Fighter 5 e MOBAs (Multiplayer Online Battle Arena) como DOTA2 e League of Legends. Nesse trabalho será abordado partidadas ranqueadas de League of Legends (LOL).

O LOL possui diversos tipos de mapas e opções diferentes de jogos e objetivos, no entanto, o mais famoso e mais jogado é o mapa Summoner’s Rift (SR), sendo também esse o mapa onde jogadores profissionais batalham em campeonatos. Uma partida no mapa SR é composta por duas equipes de cinco jogadores e o objetivo do jogo é um time destruir a base do time adversário. Para isso é necessário destruir três torres que ficam dispostas em três distintas rotas (Bot Lane, Mid Lane e Top Lane). Após a destruição das torres há um inibidor para cada rota que deve ser destruído para ter acesso ao Nexus, que é protegido por mais duas torres. Não é necessário destruir todas as torres para ter acesso aos inibidores e nem destruir todos os inibidores para ter acesso ao Nexus, como por exemplo, é possível destruir as três torres da Mid Lane, destruir o Inibidor Mid Lane e partir para a destruição das torres do Nexus e do próprio Nexus.

Entre as rotas já especificadas fica a área conhecida como Jungle e essa área possui objetivos secundários como a destruição do Dragon e Baron. Esses objetivos ficam disponíveis em tempos determinados e concedem bônus para os jogadores, o que pode potencializar as batalhas ou facilitar a destruição das estruturas das rotas.

Cada jogador pode comprar itens dentro de sua base para aperfeiçoar habilidades de seus personagens e esses itens são pagos com o ouro ganho na partida. Existem diversas formas de ganhar ouro no jogo, seja cumprindo objetivos, matando personagens adversários ou destruindo Minions e monstros da Jungle. Minions e monstros da Jungle ficam disponíveis de tempos em tempos nas rotas e na Jungle e é a maior fonte de ouro do jogo.

A visão do mapa é um fator importante dentro do jogo, sendo que cada jogador possui visão limitada, ou seja, grande parte do mapa fica escura e cada jogador tem visão apenas no seu redor e ao redor de suas estruturas, como áreas próximas às torres. Dessa forma, os jogadores podem ser surpreendidos pelo adversário com ataques surpresa. A forma mais eficiente de ganhar mais visão no mapa é utilizando Wards que fazem o papel de iluminar uma área escura do mapa, potencializando a estratégia de defesa de um time, ou a estratégia de ataque ao encontrar um jogador inimigo mal posicionado. As Wards podem ser destruídas pelo time adversário, criando uma batalha por visão dentro da partida onde cada equipe busca limitar mais a visão do adversário e ganhar mais visão para o sua equipe.

Jogos ranqueados são jogos que garantem um ranque específico dentro do LOL e são divididos em Tiers: Bronze, Prata, Ouro, Platina, Diamante, Master e Desafiante. Jogadores profissionais são jogadores que ocupam os maiores Tiers (Master e Desafiante). Numa partida ranqueada é possível encontrar adversários com o mesmo Tier que o jogador, com um Tier superior ou um inferior. Para exemplificar, um jogador que possua o Tier Ouro poderá jogador contra jogadores dos Tiers Prata, Ouro ou Platina.

Diferente dos esportes habituais, um jogo eletrônico possui uma quantidade muito grande de dados que podem ser relevantes para diferentes tipos de análise. É possível armazenar um nível de detalhes de todos os aspectos das partidas, sendo assim, há uma grande dificuldade em ler e identificar padrões dentro de tantas possíveis variáveis existentes em cada partida.

# OBJETIVO

Esse trabalho tem como objetivo realizar a análise de variáveis de partidas ranqueadas de jogadores com o intuito de identificar quais variáveis são relevantes para se ganhar uma partida e a técnica estatística abordada será regressão logística.

Esse estudo é relevante para identificar possíveis falhas na forma de jogar, sendo que será possível identificar quais as variáveis são mais relevantes para responder qual time tem maior chance de vitória. Com essa informação os jogadores podem aprimorar seu estilo de jogo dando prioridade para objetivos relevantes, permitindo que certos objetivos sejam tratados de forma secundária em busca de um melhor rendimento no jogo enquanto os objetivos primários devem ser vistos como prioridade para obtenção da vitória.

# EXTRAÇÃO DE DADOS

Os dados das partidas que são objeto de análise deste trabalho são disponibilizados pela página de desenvolvimento da Riot Games (RG). A RG é a empresa criadora de LOL e ela disponibiliza uma API (Application Programming Interface) para a extração de dados de todos os aspectos do jogo, sendo possível encontrar informações sobre mapas, personagens, itens, partidas, entre outros.

Após realizar o cadastro no site “https://developer.riotgames.com/”, ou utilizar seu próprio cadastro do jogo, o usuário recebe uma chave que permite realizar a extração dos dados via API. Comumente as API são compostas por uma série de funções com o intuito de atingir um determinado objetivo e receber uma determinada resposta. No caso da API utilizada, é definida uma mensagem de requisição HTTP (Hypertext Transfer Protocol) e recebemos a resposta no formato JSON (JavaScript Object Notation).

Para realizar a coleta de dados, foi criado um conjunto de códigos na linguagem de programação Python para armazenar os parâmetros de entrada das requisições, realizar as requisições HTTP e receber e armazenar as informações JSON enviadas pela API para, após o processo, criar um arquivo no formato CSV (Comma-Separated Values) que pode ser lido facilmente por qualquer ferramenta ou linguagem de programação.

# BASE DE DADOS E VARIÁVEIS

A base criada contêm 2.748 observações extraídas de 1.374 partidas ranqueadas de times amadores aleatórios, sem considerar os *Tiers* de seus jogadores ou nível de habilidade no jogo entre eles. Existem 29 variáveis dentro da base de dados, sendo oito variáveis qualitativas e 21 variáveis quantitativas.

## Variáveis qualitativas

matchDuration: Tempo de duração da partida.

teamWinner: Define se o time é o vencedor da partida.

teamSide: Lado do time definido por cores.

firstBlood: Define se o time matou um jogador adversário primeiro.

firstTower: Define se o time destruiu uma torre primeiro.

firstInhibitor: Define se o time destruiu um inibidor primeiro.

firstDragon: Define se o time matou um dragão primeiro.

firstBaron: Define se o time matou um barão primeiro.

## Variáveis quantitativas

towerKills: Número de torres destruídas pelo time.

inhibitorKills: Número de inibidores destruídos pelo time.

dragonKills: Número de dragões destruídos pelo time.

baronKills: Número de barões destruídos pelo time.

kills: Número de abates conquistados pelo time.

deaths: Número de mortes do time.

assists: Número de assistências em abates do time.

minionsKilled: Número de Minions mortos pelo time.

neutralMinionsKilled: Número total de Minions mortos pelo time na *Jungle.*

neutralMinionsKilledTeamJungle: Número de Minions mortos pelo time na sua *Jungle.*

neutralMinionsKilledEnemyJungle: Número de Minions mortos pelo time na *Jungle* inimiga*.*

goldEarned: Ouro total ganho pela equipe.

goldSpent: Ouro total gasto pela equipe.

wardsPlaced: *Wards* posicionadas no mapa.

wardsKilled: *Wards* destruídas no mapa.

visionWardsBoughtInGame: *Wards* de controle posicionadas no mapa.

doubleKills: Quantidade total de dois abates sequenciais feitos pelo mesmo jogador do time.

tripleKills: Quantidade total de três abates sequenciais feitos pelo mesmo jogador do time.

quadraKills: Quantidade total de quatro abates sequenciais feitos pelo mesmo jogador do time.

pentaKills: Quantidade total de cinco abates sequenciais feitos pelo mesmo jogador do time.

killingSprees: Quantidade total de abates sequenciais sem o jogador ser abatido.

## Transformação de variáveis qualitativas

Para auxiliar a análise das variáveis qualitativas foram feitas transformações para que todas as variáveis se tornassem binárias. As variáveis teamWinner, firstBlood, firstTower, firstInhibitor, firstDragon e firstBaron foram codificadas para 1 como “Sim” e 0 como “Não”.

A variável teamSide informa se o time pertence ao lado do mapa azul ou roxo. O lado do time pode trazer alguma relevância no modelo pois a disposição dos objetivos da *Jungle*, pois o lado azul fica mais próximo do *Dragon* e o lado roxo fica mais próximo do *Baron* e isso pode trazer vantagens estratégicas para deter esses objetivos. Dessa forma, foram criadas duas variáveis novas em substituição da variável teamSide sendo codificado 1 para “Sim” e 0 para “Não”:

teamBlue: Define se o time pertence ao lado azul do mapa.

teamPurple: Define se o time pertence ao lado roxo do mapa.

Para a variável matchDuration foram criadas cinco novas variáveis indicando uma escala de tempo, sendo codificado 1 para “Sim” e 0 para “Não”:

duration\_0\_10: Partida de duração entre 0 e 10 minutos.

duration\_10\_20: Partida de duração entre 10 e 20 minutos.

duration\_20\_30: Partida de duração entre 20 e 30 minutos.

duration\_30\_40: Partida de duração entre 30 e 40 minutos.

duration\_40\_more: Partida de duração maior que 40 minutos.

# ANÁLISE EXPLORATÓRIA DOS DADOS

Com os dados disponíveis em formato CSV, foi realizada a análise exploratória utilizando a linguagem R, tendo em vista que a linguagem foi desenvolvida para análise estatística de dados e inclui uma grande quantidade de pacotes para manipulação, transformação e visualização de dados.

## Análise exploratória de dados qualitativos

Os dados qualitativos foram analisados pela frequência absoluta e relativa de ocorrências zero ou um como pode ser visto na tabela abaixo.

**Tabela 1.** Tabela de Frequências Absolutas e Relativas

|  |  |  |  |
| --- | --- | --- | --- |
| **Variáveis** | **Ocorrência** | **Frequência Absoluta** | **Frequência Relativa** |
| teamWinner | 0 | 1374 | 0.50 |
| teamWinner | 1 | 1374 | 0.50 |
| teamBlue | 0 | 1374 | 0.50 |
| teamBlue | 1 | 1374 | 0.50 |
| teamPurple | 0 | 1374 | 0.50 |
| teamPurple | 1 | 1374 | 0.50 |
| duration\_0\_10 | 0 | 2650 | 0.96 |
| duration\_0\_10 | 1 | 98 | 0.04 |
| duration\_10\_20 | 0 | 2648 | 0.96 |
| duration\_10\_20 | 1 | 100 | 0.04 |
| duration\_20\_30 | 0 | 1962 | 0.71 |
| duration\_20\_30 | 1 | 786 | 0.29 |
| duration\_30\_40 | 0 | 1528 | 0.56 |
| duration\_30\_40 | 1 | 1220 | 0.44 |
| duration\_40\_more | 0 | 2204 | 0.80 |
| duration\_40\_more | 1 | 544 | 0.20 |
| firstBlood | 0 | 1395 | 0.51 |
| firstBlood | 1 | 1353 | 0.49 |
| firstTower | 0 | 1421 | 0.52 |
| firstTower | 1 | 1327 | 0.48 |
| firstInhibitor | 0 | 1489 | 0.54 |
| firstInhibitor | 1 | 1259 | 0.46 |
| firstDragon | 0 | 1431 | 0.52 |
| firstDragon | 1 | 1317 | 0.48 |
| firstBaron | 0 | 1965 | 0.72 |
| firstBaron | 1 | 783 | 0.28 |

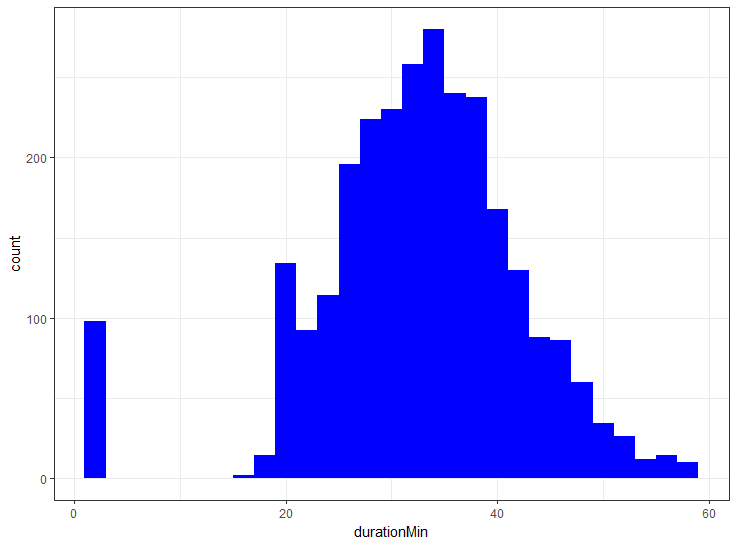
É possível identificar que metade dos jogos possuem vencedores e metade perdedores, cada metade está de um lado do mapa. Isso demonstra consistência da base, tendo em vista que a base é formada por 1.374 partidas ranqueadas e sempre que um time ganha uma partida o outro time perde.

Em relação à duração das partidas temos aproximadamente 73% de frequência em jogos entre 20 e 40 minutos, e uma grande minoria dos jogos dura menos de 20 minutos enquanto aproximadamente 20% dos jogos duram mais que 40 minutos.

Sobre os primeiros a conquistar objetivos, é possível visualizar que há uma grande disputa entre eles e os resultados ficam na maioria das vezes entre 40% e 60% de frequência, com exceção da variável firstBaron que apresentou uma diferença considerável de frequência.

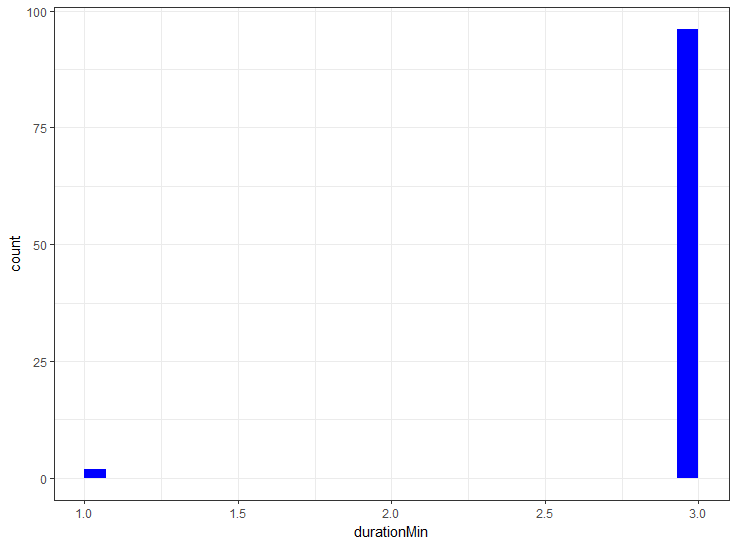
## Variável Duração

Analisando a variável duração podemos identificar a distribuição da variável conforme a figura abaixo.

**Gráfico 1.** Análise de distribuição da variável DuracaoMin

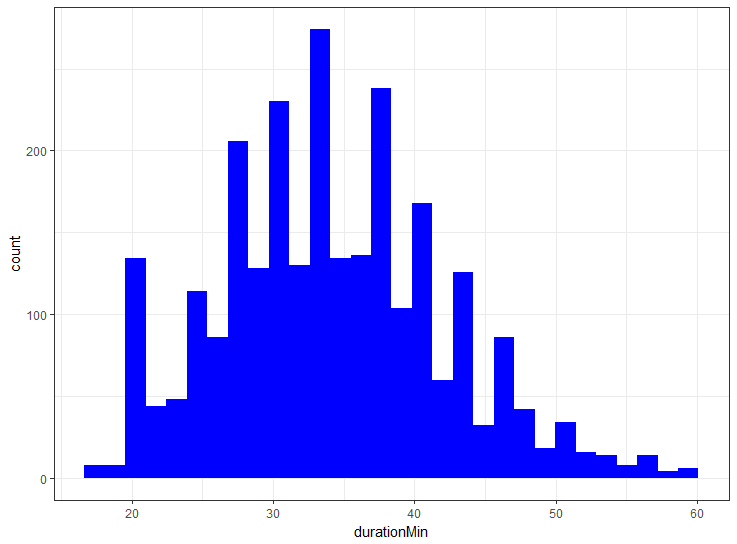
Podemos identificar que a distribuição possui quase 100 jogos com um tempo muito abaixo do normal, próximo dos 5 min. Fazendo a análise dos jogos mencionados foram encontradas 98 partidas que ocorreram em menos de 4 minutos.

Gráfico 2. Jogos com menos de 5 min.



Esses casos poderiam apenas ocorrer caso houvesse a desconexão do jogo, pois o tempo de duração média das partidas é de 33 minutos. Sendo assim, considerando que as variáveis envolvendo esses jogos não apresentam representatividade, elas serão retiradas da base. O histograma com os dados com tempo maior que 5 minutos possui a seguinte distribuição de dados.

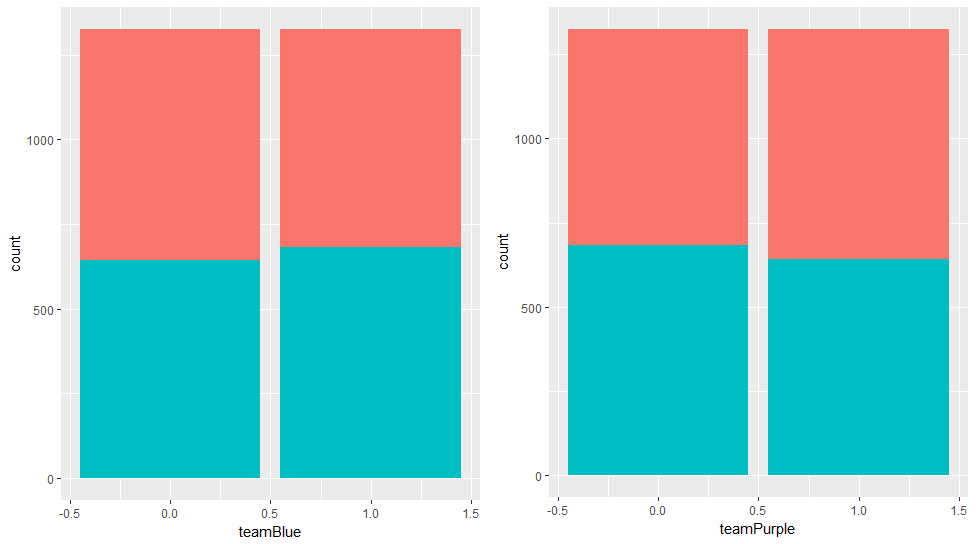
Gráfico 3. Análise de distribuição da variável duracaoMin com duração maior que 5 minutos



## Variáveis teamBlue, teamPurpple

Referente às variáveis que informam o em qual lado do mapa os times jogaram, podemos verificar se há alguma vantagem para a vitória em algum desses lados.

Grafico 4. Frequência das variáveis teamBlue, teamPurpple por time vencedor

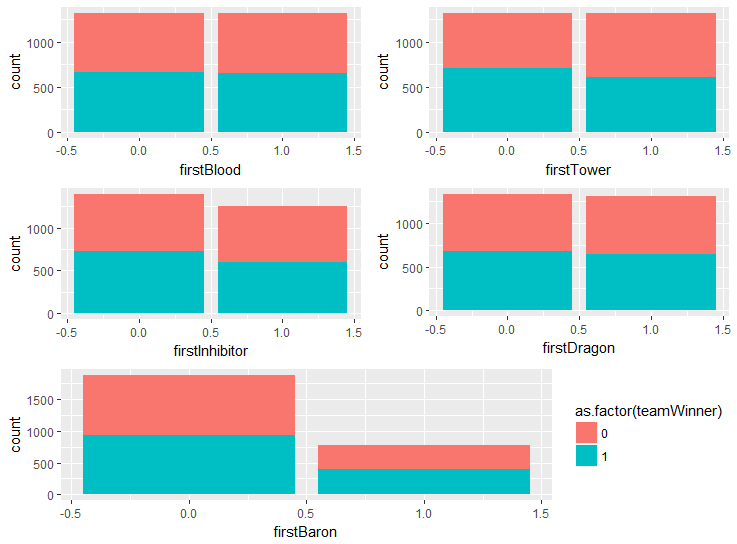


Como podemos ver, há pouca diferença em vitórias pelo lado do mapa em que os times jogaram, no entanto, os times que jogaram no lado azul do mapa (teamBlue) tiveram mais vitórias do que os times que jogaram no lado roxo (teamPurple).

## Variáveis firstBlood, firstTower, firstInhibitor, firstDragon e firstBaron

As variáveis categóricas foram analisadas de acordo com suas frequências e em comparação com o time vencedor, sendo 0 o time perdedor e 1 o time vencedor.

Gráfico 5. Frequência de variáveis firstBlood, firstTower, firstInhibitor, firstDragon e firstBaron por time vencedor



Como é possível visualizar a variável firstBlood não apresenta variação entre os times vencedores e perdedores, o que nos leva a crer que essa variável não será relevante no modelo. As variáveis firstTower, firstInhibitor e firstDragon apresentam apenas diferenças pequenas em comparação com a variável firstBaron, onde podemos observar que é um objetivo que ocorre menos vezes nos jogos, no entanto não parece influenciar muito a questão de vitória de um time conforme o comprimento do objetivo.

## Análise exploratória de dados quantitativos

Na análise exploratória das variáveis quantitativas foram utilizadas as métricas Média, Desvio Padrão, Mediana, Mínimo, Máximo, Quartil 1 e Quartil 3 conforme tabela abaixo.

**Tabela 2.** Tabela de estatísticas das variáveis quantitativas

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Variáveis** | **Média** | **Desvio Padrão** | **Mediana** | **Mínimo** | **Máximo** | **Q1** | **Q3** |
| towerKills | 5.96 | 3.88 | 7.0 | 0 | 11 | 2 | 10.0 |
| inhibitorKills | 1.19 | 1.44 | 1.0 | 0 | 10 | 0 | 2.0 |
| dragonKills | 1.60 | 1.35 | 2.0 | 0 | 6 | 0 | 3.0 |
| baronKills | 0.37 | 0.58 | 0.0 | 0 | 3 | 0 | 1.0 |
| kills | 27.11 | 13.56 | 27.0 | 0 | 71 | 17 | 37.0 |
| deaths | 33.80 | 13.36 | 35.0 | 0 | 81 | 26 | 43.0 |
| assists | 39.80 | 23.65 | 37.0 | 0 | 125 | 23 | 56.0 |
| minionsKilled | 590.34 | 206.97 | 591.0 | 13 | 1383 | 470 | 715.0 |
| neutralMinionsKilled | 83.63 | 39.39 | 80.0 | 0 | 255 | 56 | 108.0 |
| neutralMinionsKilledTeamJungle | 61.26 | 28.24 | 59.0 | 0 | 168 | 42 | 79.0 |
| neutralMinionsKilledEnemyJungle | 22.36 | 20.73 | 18.0 | 0 | 142 | 5 | 33.0 |
| goldEarned | 56026.61 | 19895.91 | 56173.5 | 3687 | 114187 | 43443 | 68986.0 |
| goldSpent | 51713.79 | 19004.52 | 52149.0 | 1500 | 120170 | 40045 | 64004.5 |
| wardsPlaced | 57.17 | 24.78 | 56.0 | 0 | 155 | 41 | 73.0 |
| wardsKilled | 7.20 | 7.87 | 5.0 | 0 | 90 | 2 | 10.0 |
| visionWardsBoughtInGame | 3.33 | 3.13 | 3.0 | 0 | 20 | 1 | 5.0 |
| doubleKills | 2.53 | 2.16 | 2.0 | 0 | 13 | 1 | 4.0 |
| tripleKills | 0.33 | 0.64 | 0.0 | 0 | 4 | 0 | 1.0 |
| quadraKills | 0.04 | 0.22 | 0.0 | 0 | 2 | 0 | 0.0 |
| pentaKills | 0.01 | 0.08 | 0.0 | 0 | 1 | 0 | 0.0 |
| killingSprees | 6.25 | 3.66 | 6.0 | 0 | 21 | 4 | 9.0 |

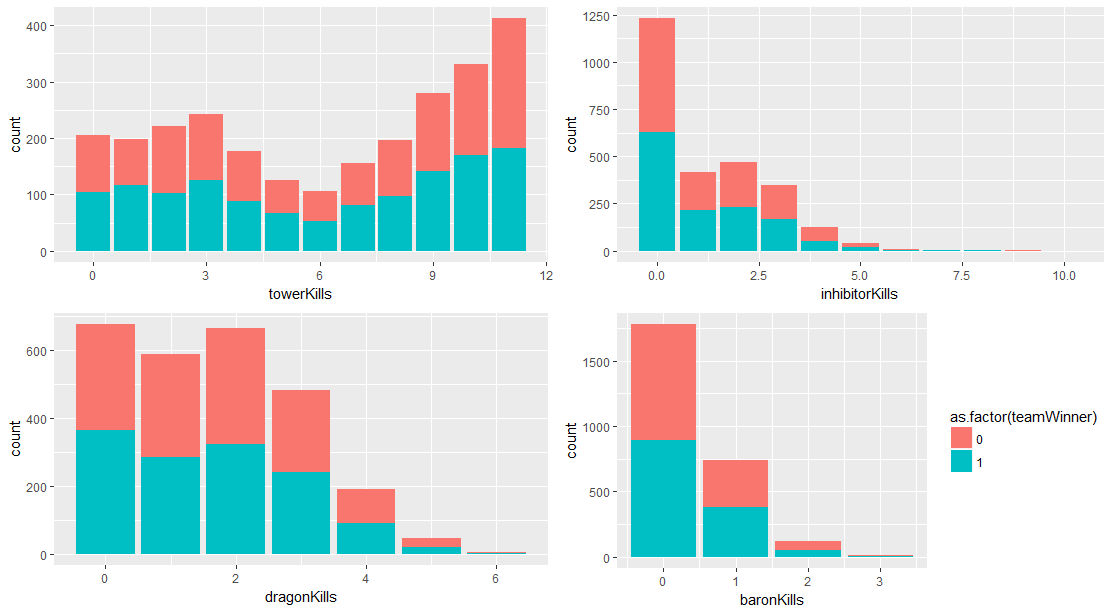
A tabela mostra informações relevantes quanto a diferença entre as variáveis voltadas para o objetivo do jogo e as informações sobre as estatísticas dos times. Pode-se observar que as dimensões das variáveis são grandes, como por exemplo a variável towerKills que contem um máximo de 11 e a variável minionsKilled que tem o máximo de 1383, demonstrando a importância de realizar a normalização das variáveis para que todas possuam uma escala igual.

É visível também que as variáveis goldPlaced e goldSpend possuem valores muito próximos entre as métricas, o que indica que essas variáveis são semelhantes a ponto de não ser necessário possuir ambas no modelo de dados.

## Variáveis towerKills, inhibitorKills, dragonKills, baronKills

As variáveis quantitativas que se referem às quantidades de vezes que os objetivos foram realizados podem nos dar informações sobre a relevância das variáveis em relação ao target.

**Gráfico 6.** Frequência de variáveis quantitativas por time vencedor

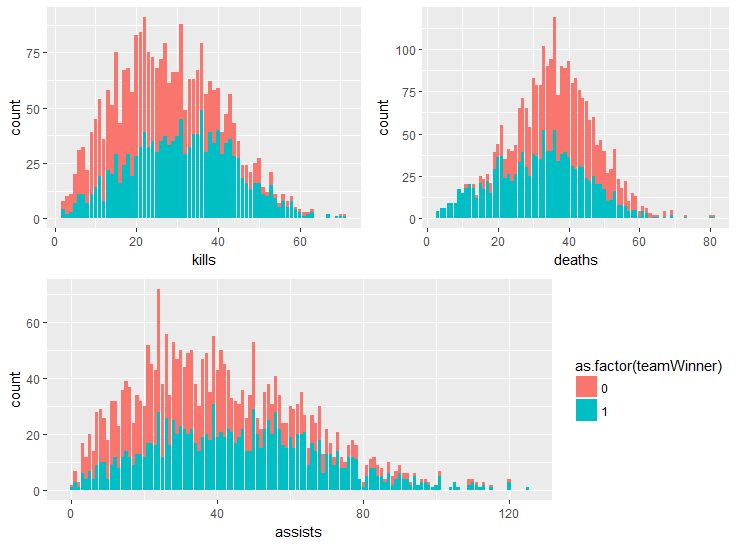


Como é possível verificar os times que destruíram mais torres e inibidores tendem a ser vencedores. Referente aos dragões e barões, não é possível fazer a mesma afirmação, pois os resultados são muito semelhantes, não levando a crer que essas variáveis são relevantes.

## Variáveis kills, deaths e assists

As variáveis que envolvem o desempenho dos jogadores dentro da partida mostram uma relevância maior para identificar quem ganha ou perde o jogo.

**Gráfico 7.** Frequência de variáveis kills, deaths e assists por time vencedor

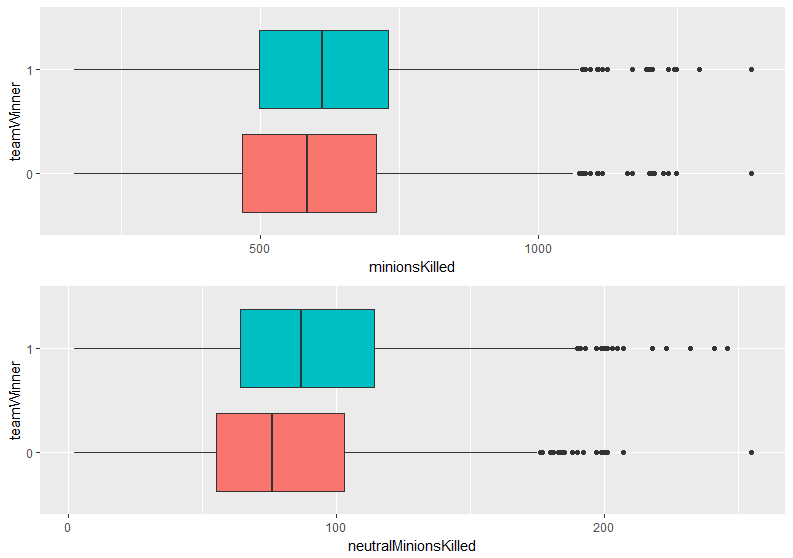


Como é possível verificar, times que mataram mais jogadores adversários (kills) tem maior propensão para a vitória. Da mesma forma, times que tiveram menos mortes (deaths) e obtiveram um maior número de assistências (assists) tem maior índice de vitórias, o que era esperado.

## Variáveis minionsKilled, neutralMinionsKilled

As variáveis relacionadas aos minions possuem uma grande amplitude e para esta análise, foi utilizado o gráfico de boxplot para uma melhor visualização.

**Gráfico 8.** Boxplot das variáveis minionsKilled e neutalMinionsKilled

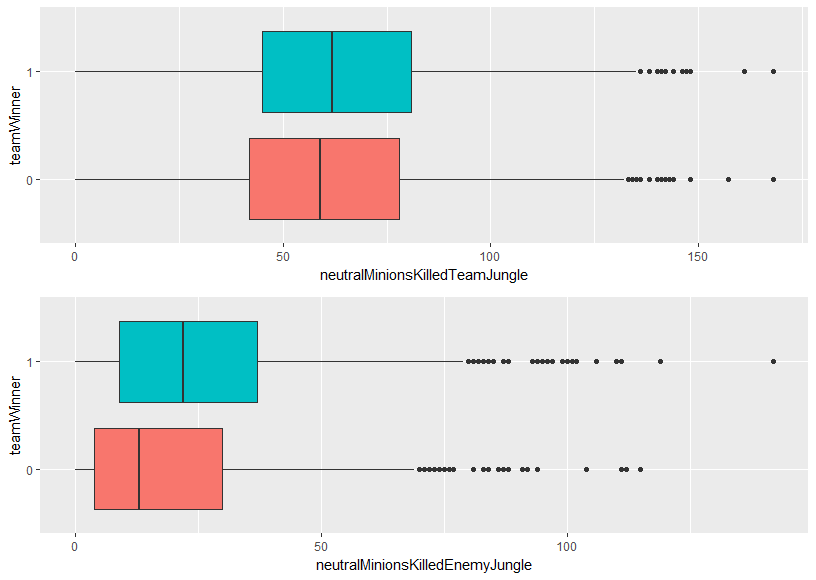


No gráfico acima é possível identificar que o time que mata mais minions tem maior possibilidade de vitória. Em relação aos minions da jungle o padrão se repete e é visível que em ambos os casos há outliers, ou seja, em determinados jogos houve resultados atípicos em relação a essas variáveis onde um número muito grande de minions ou minions da jungle foi morto.

## Variáveis neutralMinionsKilledTeamJungle e neutralMinionsKilledEnemyJungle

As variáveis que envolvem os minions da jungle demonstram o controle dos times entre o mapa. Teoricamente, times que ganham tem um maior controle do mapa e assim da sua jungle e da jungle inimiga.

**Gráfico 9.** Boxplot das variáveis neutralMinionsKilledTeamJungle e neutralMinionsKilledEnemyJungle

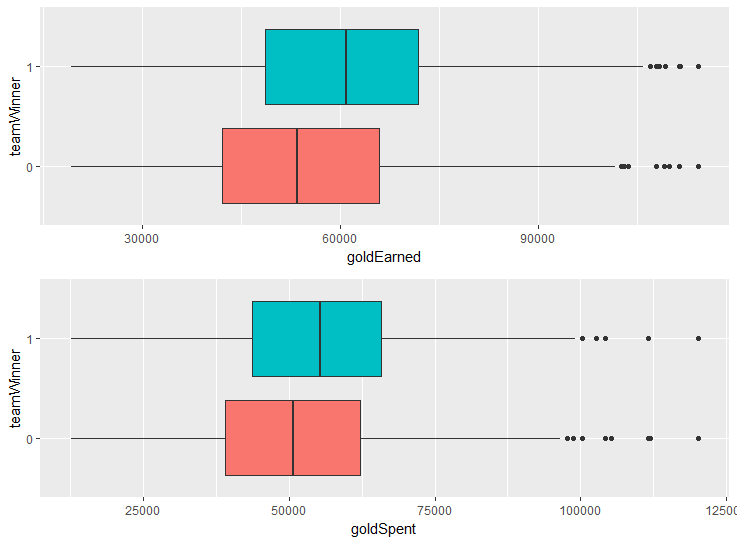


Como esperado, os times vencedores dominam tanto a sua própria jungle quanto a inimiga. Os boxplots também apresentam outiliers, mas é interessante verificar que a mediana do boxplot da variável neutralMinionsKilledEnemyJungle do time perdedor, mesmo com a presença de outiliers é menor que o normal.

## Variáveis goldEarned e goldSpent

As variáveis que envolvem o ouro ganho e gasto na partida mostram a tendência esperada de quem tem mais ouro e gasta mais, ganha a partida.

**Gráfico 10.** Boxplot das variáveis goldEarned e goldSpent

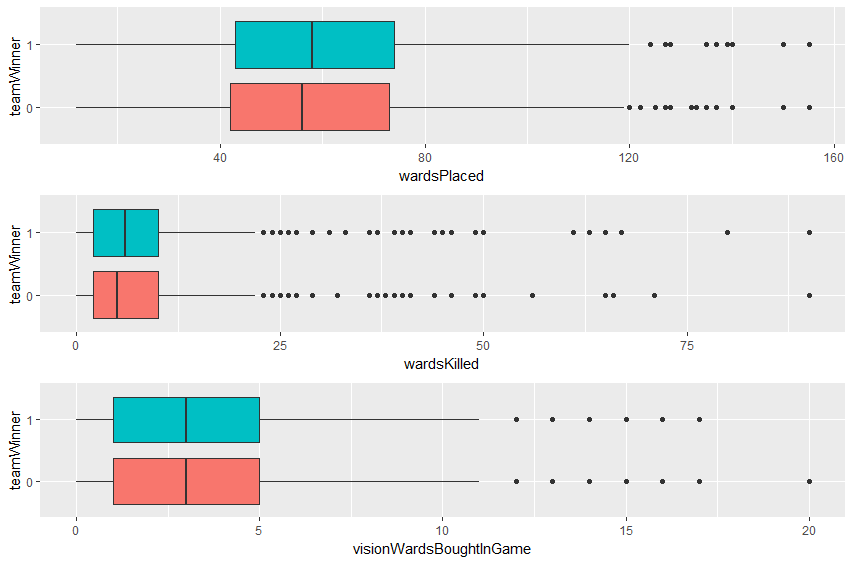


Nos gráficos é possível identificar que ambas as variáveis tem uma tendência semelhante e provavelmente não há a necessidade da utilização das duas variáveis simultaneamente no modelo, pois pode ocorrer multicolinearidade.

## Variáveis wardsPlaced, wardsKilled e visionWardsBoughtInGame

As variáveis que envolvem wards nos informam sobre a visão de cada time no mapa.

**Gráfico 11.** Boxplot das variáveis wardsPlaced, wardsKilled e visionWardsBoughtInGame

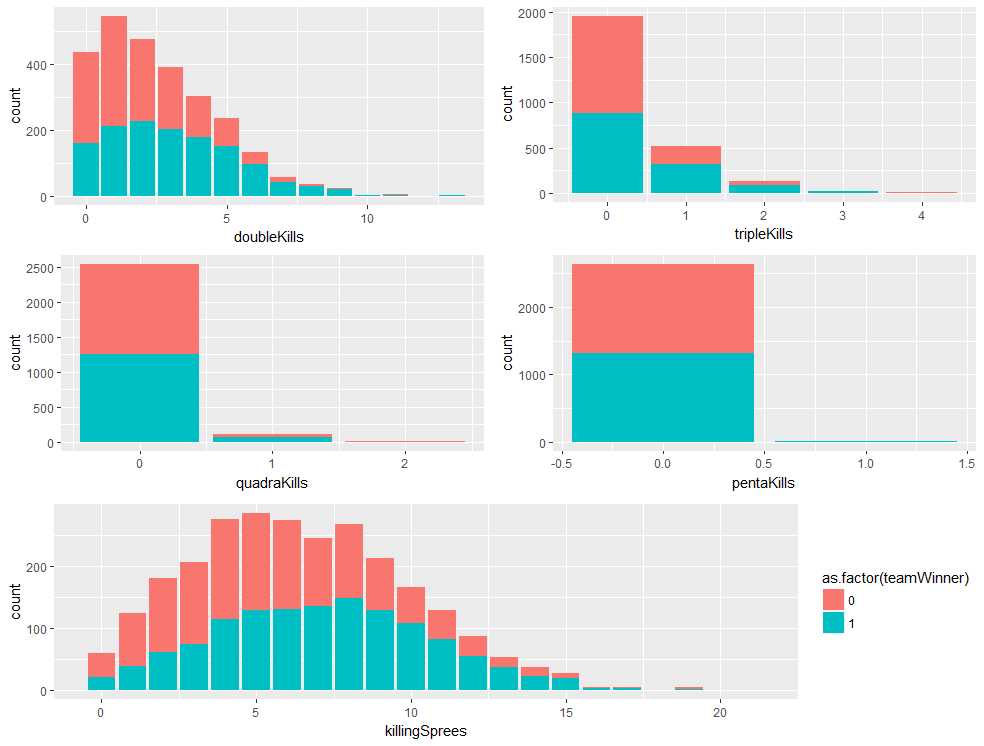


Os gráficos nos confirmam a teoria de que quem tem mais visão do mapa tem maior possibilidade de serem vencedores, mas é possível verificar a existência de outliers e uma diferença pequena entre o poder de visão dos times pela quantidade de wards colocadas no jogo, destruídas ou compradas.

## Variáveis doubleKills, tripleKills, quadraKills, pentaKills e killingSprees

As seguintes variáveis tem relação com a quantidade de vezes que foi feito um abate sequencial. Elas devem mostrar a relevância entre matar oponentes de forma acumulativa.

**Gráfico 12.** Frequência das variáveis doubleKills, tripleKills, quadraKills, pentaKills, killingSprees



Como esperado as variáveis demonstram que quanto mais abates sequenciais são feitos, maior a possibilidade de vitória dos times.

# Modelagem

Com o intuito de identificar a chance de uma equipe vencer ou perder uma partida com base nos dados descritos, uma opção de modelagem é a regressão logística, pois a variável resposta é uma variável binária. A variável teamWinner será a variável resposta e o evento de interesse será prever a vitória, sendo que na codificação da base de dados a opção 1 está relacionada ao time vencedor.

Nesta análise a base de dados final contém 2650 observações, sendo metade das variáveis resposta sendo 0 e metade sendo 1. Foi feita a divisão da base entre treinamento e teste, onde a base de treino representa 75% da base total e a base de testes possui 25% das observações finais. As bases foram distribuídas com target proporcional, ou seja, tanto na base teste, quanto na base de treinamento existem 50% de variáveis com target 0 e 1.

## Seleção de variáveis

Para identificar quais variáveis são relevantes no modelo foi utilizado o método Stepwise do software R e as variáveis com menor nível de significância são as variáveis que melhor explicam a variável target. O stepwise constrói iterativamente uma sequência de modelos de regressão pela adição ou remoção de variáveis em cada etapa. Para definir se a variável deve permanecer no modelo ela utiliza o teste F. As variáveis que possuem maior correlação com a variável resposta (target) permanecerão no modelo.

## Regressão Logística

A regressão logística é um algoritmo estatístico que possibilita saber se existe relação entre um conjunto de variáveis e uma variável resposta, ou seja, ela permite identificar qual a probabilidade de um evento ocorrer de acordo com as variáveis informadas. Sendo assim, a regressão logística binária estima a probabilidade de ocorrência do evento em estudo (FÁVERO, BELFIORE, 2017) e não estima os valores previstos da variável dependente.

A fórmula da regressão logística é:

Na regressão logística a probabilidade de um evento ocorrer sempre será entre 0 e 1. Sendo assim, após a execução do algoritmo será possível identificar qual a probabilidade de uma equipe vencer uma partida ranqueada conforme as variáveis indicadas. Após a execução do algoritmo é necessário realizar o resultado do P-Valor de cada variável para identificar se ela é relevante ou não no modelo.

O modelo estatístico foi processado na linguagem R, utilizando a função “gml”. Essa função possui diversos parâmetros de entrada e executa diversos tipos de algoritmos de acordo com a família selecionada. Para a análise de regressão logística a família a ser utilizada é “Binomial”. Nessa etapa do processo será utilizada a base de treinamento que contém 1988 observações com metade das observações sendo de times vencedores e metade de times perdedores.

Para aplicação do modelo foi utilizado o método de seleção Stepwise e o resultado pode ser apresentado pela tabela abaixo.

**Tabela 3.** Tabela com resultados da regressão logística com método de seleção Stepwise

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Coefficients:** | **Estimate** | **Std. Error** | **z value** | **Pr(>|z|)** | **Code** |
| (Intercept) | -234859532 | 0.67084440 | -3501 | 0.000464 | \*\*\* |
| duration\_10\_201 | 173.059.676 | 0.48239808 | 3.587 | 0.000334 | \*\*\* |
| duration\_20\_301 | 167119140 | 0.32347274 | 5.166 | 0.000000238639 | \*\*\* |
| duration\_30\_401 | 102812685 | 0.20972759 | 4.902 | 0.000000947686 | \*\*\* |
| teamBlue1 | 0.64153000 | 0.23749969 | 2.701 | 0.006909 | \*\* |
| firstTower1 | -0.26087622 | 0.12882948 | -2.025 | 0.042870 | \* |
| firstBaron1 | 0.67803669 | 0.25137044 | 2.697 | 0.006989 | \*\* |
| towerKills | -0.18776386 | 0.03822578 | -4.912 | 0.000000901661 | \*\*\* |
| dragonKills | -0.10872804 | 0.05944025 | -1.829 | 0.067370 | . |
| baronKills | -0.64329164 | 0.20361350 | -3.159 | 0.001581 | \*\* |
| kills | 0.07548039 | 0.01743994 | 4.328 | 0.000015045671 | \*\*\* |
| deaths | -0.10587019 | 0.00737444 | -14.356 | < 0.0000000000000002 | \*\*\* |
| minionsKilled | -0.00448724 | 0.00103633 | -4.330 | 0.000014914563 | \*\*\* |
| goldEarned | 0.00029880 | 0.00003068 | 9.738 | < 0.0000000000000002 | \*\*\* |
| goldSpent | -0.00015709 | 0.00002489 | -6.312 | 0.000000000276 | \*\*\* |
| wardsPlaced | -0.01651531 | 0.00436742 | -3.781 | 0.000156 | \*\*\* |
| wardsKilled | -0.01485457 | 0.00853875 | -1.740 | 0.081918 | . |
| killingSprees | -0.15510623 | 0.04069232 | -3812 | 0.000138 | \*\*\* |
| Signif. codes: | 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1 | | | |  |

Verificando os resultados é possível ver que o método de seleção de variáveis reduziu a quantidade de variáveis para 17. Algumas das variáveis que foram analisadas na fase de análise exploratória de dados e que foram apontadas como variáveis que não discriminam a vitória de um time foram retiradas do modelo, o que reforça a análise já realizada. De acordo com os resultados ainda é possível identificar duas variáveis que possuem um P-valor maior que 0.05, sendo elas as variáveis dragonKills e wardsKilled. Utilizando os resultados obtidos pela regressão logística como o Stepwise na base de treinamento, será feito uma matriz de confusão para demonstrar a efetividade do modelo apresentado. Nesse trabalho, será necessário estabelecer um *cutoff* para definirquando será considerado que o time venceu ou perdeu.

Segundo Fávero (2017), “O *cutoff,* que nada mais é do que um ponto de corte que o pesquisador escolhe, é definido para que sejam classificadas as observações em função das suas probabilidades calculadas, e, desta forma, é utilizado quando há o intuito de se elaborarem previsões de ocorrência do evento para observações não presentes na amostra com base nas probabilidades das observações presentes na amostra.”.

Na modelagem em questão será definido um *cutoff* conforme abaixo:

Com o *cutoff* definido, foi feito a previsão dos dados na tabela de treinamento para avaliar a acurácia do modelo.

## Matriz de confusão

Matriz de confusão é um tipo de tabela que permite a visualização do desempenho de um algoritmo de aprendizado, tipicamente em um algoritmo supervisionado. Cada coluna da matriz representa as instâncias de uma classe prevista, enquanto as linhas representam os casos de uma classe real. Com essa tabela podemos avaliar a eficiência global do modelo, a sensitividade e a especificidade.

**Tabela 4.** Matriz de confusão base de treinamento com *cutoff* de 0,5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Predito** | | | | | |
| **Observado** | | | 0 | 1 |  |
| 0 | | | 718 | 251 | 969 |
| 1 | | | 276 | 743 | 1.019 |
|  | | | 994 | 994 | 1.988 |
|  |  |

A eficiência global do modelo ou acurácia é medida pelo percentual de acerto de classificação para o *cutoff* definido. A sensitividade está relacionada com o percentual de eventos positivos, no caso, seriam considerados os times que venceram e a especificidade refere-se ao percentual de acerto dos times que perderam. Sendo assim, a tabela abaixo exibe essas 3 análises.

**Tabela 5.** Análise da tabela de confusão com *cutoff* 0,5

|  |  |
| --- | --- |
| **Eficiência do Modelo** | |
| **Acurácia:** | 0,734909 |
| **Sensitividade:** | 0,747485 |
| **Especificidade:** | 0,722334 |

O modelo teve acurácia de 73,5% de acerto na base treinamento, considerando as variáveis analisadas pela regressão logística. O modelo acerta mais quem ganhará os jogos e erra uma pouco menos para quem perde. Fazendo a alteração do *cutoff* para 0,60 é possível observar que a acurácia e a sensibilidade são reduzidos enquanto a especificidade tem um grande aumento.

**Tabela 6.** Matriz de confusão base de treinamento com *cutoff* de 0,6

|  |  |  |  |
| --- | --- | --- | --- |
| **Predito** | | | |
| **Observado** | 0 | 1 |  |
| 0 | 811 | 361 | 1.172 |
| 1 | 183 | 633 | 816 |
|  | 994 | 994 | 1.988 |
|  |  |  |  |

**Tabela 7.** Análise da tabela de confusão com *cutoff* 0,6

|  |  |
| --- | --- |
| **Eficiência do Modelo** | |
| **Acurácia:** | 0,726358 |
| **Sensitividade:** | 0,636821 |
| **Especificidade:** | 0,815895 |

## Teste do modelo

Após a análise dos resultados da modelagem feita na base de treinamento, foi realizada a aplicação da regressão logística utilizando as variáveis selecionadas pelo método de seleção na base de testes, que neste trabalho será a base final. A base de testes contém 662 jogos e foram obtidos os seguintes resultados:

**Tabela 8.** Matriz de confusão base de teste com *cutoff* de 0,6

|  |  |  |  |
| --- | --- | --- | --- |
| **Predito** | | | |
| **Observado** | 0 | 1 |  |
| 0 | 270 | 127 | 397 |
| 1 | 61 | 204 | 265 |
|  | 331 | 331 | 662 |
|  |  |  |  |

**Tabela 7.** Análise da tabela de confusão com *cutoff* 0,6 da base teste

|  |  |
| --- | --- |
| **Eficiência do Modelo** | |
| **Acurácia:** | 0,716012 |
| **Sensitividade:** | 0,616314 |
| **Especificidade:** | 0,81571 |

O resultado apresentado na base de testes foi muito satisfatório, pois, não houve uma grande perda de desempenho entre a base de treinamento e de testes, de acordo com as tabelas de confusão, o que indica que esse modelo é relevante para identificar a chance de um determinado evento ocorrer. No caso exposto, é visível que o modelo tem boa capacidade para identificar a chance de um time perder uma partida.

# CONCLUSÃO

As análises realizadas durante esse trabalho apresentaram um resultado interessante e positivo, demonstrando que é possível prever a chance de um time ganhar ou perder utilizando a técnica de regressão logística binária. É importante também lembrar que os jogos eletrônicos como League of Legends possuem uma quantidade de variáveis muito grande, e com isso, a utilização de métodos de seleção de variáveis é imprescindível para se obter um resultado mais assertivo e menos moroso.

O ajuste do *cutoff* faz uma grande diferença nos resultados, com ele foi possível identificar um grande aumento de especificidade, o que mostra que é possível prever a chance de um time perder de forma mais eficaz.

Levando em consideração o modelo apresentado, o pesquisador acredita que é possível realizar novas análises similares a essa com o mesmo objetivo, no entanto, fazendo o incremento de variáveis específicas de cada jogador, podendo assim tentar localizar variáveis individuais que expliquem ainda mais o modelo proposto.

# Referências

FÁVERO, L. P.; BELFIORE, P.; **Manual de Análise de dados, Estatística e Modelagem Multivariada com Excel, SPSS e Stata**, 1. ed. Rio de Janeiro: Elsevier, 2017.

WICKHAM, H.;GROLEMUND, G.; **R for Data Science**, 2017; Disponível em http://r4ds.had.co.nz/

PINHEIRO, C. A. R.; Inteligência Analítica, Mineração de Dados e Descoberta de Conhecimento, 1. ed. Editora Ciência Moderna, 2008.