

Predição de resultados de partidas em League of Legends baseados em acontecimentos ao decorrer dos jogos

Luis F. M. Pozzobon¹, Douglas W. L. de Oliveira¹

¹ Engenharia de Software – Universidade Tecnológica Federal do Paraná (UTFPR)

pozzobon@utfpr.edu.br, douglaso@alunos.utfpr.edu.br

Abstract. *This article deals with the development of an algorithm to try to predict the outcome of matches in League of Legends, based on the results of some objectives within each game, such as kills, destruction of towers, and neutral objectives (also known as monsters of the jungle) . Using machine learning, the algorithm has a database with data from 50 thousand matches that took place in 2019, it tries to predict the percentage of the team's chances of victory in real time using the events of the match.*

Resumo. *Este artigo trata do desenvolvimento de um algoritmo para tentar prever o resultado de partidas em League of Legends, baseando-se em resultados de alguns objetivos dentro de cada partida, como abates, destruição de torres, e objetivos neutros (também conhecidos como monstros da selva). Utilizando aprendizado de máquina, o algoritmo tem uma base de dados com dados de 50 mil partidas que ocorreram no ano de 2019, tenta prever a porcentagem de chances de vitória da equipe em tempo real utilizando os acontecimentos da partida.*

1. Introdução

League of Legends é um MOBA (Multiplayer Online Battle Arena) desenvolvido pela Riot Games (League of Legends, 2021), onde os jogadores se enfrentam com o objetivo de destruir o Nexus inimigo, o qual é protegido por outras estruturas como torres e inibidores, além dos próprios campeões que são controlados pelos jogadores.

Os campeões são personagens com habilidades únicas e com diferentes funções dentro da partida, as quais podem ser: tanque, assassino, mago, atirador, suporte, lutador. Utilizando esses campeões, o jogador ocupa uma atribuição dentro do jogo conhecidas como: topo, selva, meio, carregador e suporte.

A disputa acontece dentro do mapa de Summoner's Rift, que é repleto de vários outros pequenos objetivos, que dão recurso e experiência para os jogadores que faz com que o objetivo principal seja atingido. Ele possui três rotas principais, onde os jogadores se enfrentam com a ajuda das tropas geradas automaticamente pelo Nexus aliado. Essas rotas são caminhos em direção à base inimiga, que são protegidas pelas torres de defesa e que atacam qualquer alvo inimigo que se aproxima.

Entre as rotas existe uma área neutra conhecida como selva que abriga monstros neutros e grandes objetivos, como o barão, arauto e dragões. Esta área normalmente é explorada pelo caçador.

Com o grande crescimento do jogo, e em consequência do modo competitivo de League of Legends ao decorrer dos anos, um mercado voltado para apostas surgiu, onde os

telespectadores depositam dinheiro em diversas das plataformas existentes e apostam sobre os acontecimentos e resultados das partidas nos diversos campeonatos ao redor do mundo, o CBLOL (Campeonato Brasileiro de League Of Legends) é o mais conhecido no Brasil, por ser o campeonato oficial mantido pela própria Riot Games.

Essas apostas são diversas, podem variar desde a mais comum, como o time vencedor da partida, até apostas mais incomuns e arriscadas como qual será o primeiro dragão da partida, que time realizará o First Blood, quanto de ouro será ganho durante o jogo, etc.

O mercado se tornou ainda mais interessante quando as plataformas começaram a permitir as apostas em tempo real durante o jogo, utilizando de um sistema de aprendizado de máquina, para calcular a chance de acerto do objetivo disponibilizado para a aposta seja atingido, podendo assim alterar os valores dos possíveis resultados dos ganhadores da aposta, popularmente conhecida como ODD.

2. Trabalhos relacionados

O presente trabalho utiliza como *dataset* um arquivo disponibilizado através de um projeto chamado “League Of Legends Match Score Prediction” Demiryurek (2019), contendo cerca de 50 mil partidas que ocorreram no ano de 2020.

No trabalho de Demiryurek (2019), este dataset é utilizado em conjunto de um algoritmo de predição baseado nos modelos de Árvore de decisão e Busca em grade para realizar o treinamento e os testes utilizando estes dados, com intuito de retornar a precisão de acerto de qual time irá vencer.

A partir deste arquivo de referência para partidas que já ocorreram, utilizamos um processo utilizado no artigo escrito por Arturo Rey no Medium (2021), onde ele descreve passo a passo de como é feito o seu algoritmo para predição de resultados a partir dos 10 minutos.

Neste artigo citado anteriormente, é utilizado um dataset de mais de 9800 partidas e utilizou alguns tipos de algoritmos de treinamento para o aprendizado, para ver qual equipe iria vencer o confronto. Porém isso ainda não satisfaz a nossa solução onde queremos encontrar uma porcentagem de vencer a partida.

3. Proposta

Visando atingir esse mercado de apostas, é proposto neste trabalho um sistema baseado em Aprendizado de Máquina, para realizar a predição de resultados das partidas em League of Legends, se baseando nos próprios acontecimentos ao decorrer dos jogos, como abates e outros objetivos como dragões, torres, inibidores e barões.

Após ser desenvolvido e treinado com base nos dados das mais diversas ligas ao redor do mundo, o sistema poderá por conta própria disponibilizar as probabilidades, estatísticas e possíveis resultados durante as partidas de League of Legends.

4. Experimento

4.1 Tecnologias

A linguagem utilizada para o desenvolvimento dos scripts foi o Python, devido a natureza da base de dados e a afinidade dos envolvidos com a linguagem.

Já os modelos utilizados para os treinamentos e testes de predição foram o *Support Vector Machine*, *Adaboost* e *Random Forest* do pacote *Sklearn*, para ter diferentes resultados a serem analisados.

4.2 Base de dados

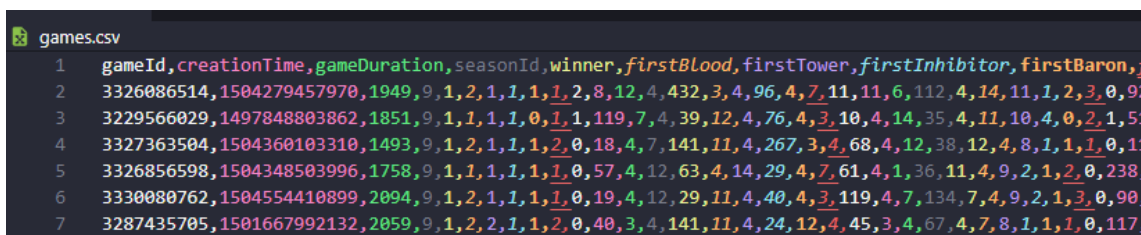
A base de dados utilizada foi um arquivo CSV com 51.490 partidas, registradas a partir de 61 colunas de acontecimentos e informações variadas que ocorreram durante os jogos.

Estas informações estão separadas por colunas, que possuem números que podem variar de acordo com o time vencedor ou a métrica que está sendo mostrada na coluna em questão.

Por exemplo, a coluna `firstBlood` refere-se apenas a uma informação, o time que realizou o primeiro abate em algum adversário, representado pelo número 1 caso o Time 1 tenha realizado o abate, 2 caso o Time 2 tenha o feito e 0 caso o jogo termine sem que o primeiro abate ocorra.

Já a coluna `t1_towerKills` representa a quantidade de torres que o Time 1 destruiu, o mesmo vale para `t2_towerKills`, representando o Time 2.

A imagem a seguir demonstra as colunas separadas por cores, de algumas informações utilizadas para o treinamento e testes.



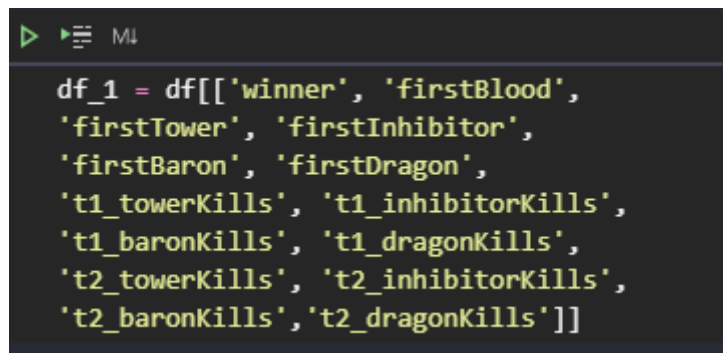
```
games.csv
1 gameId,creationTime,gameDuration,seasonId,winner,firstBlood,firstTower,firstInhibitor,firstBaron,
2 3326086514,1504279457970,1949,9,1,2,1,1,1,1,2,8,12,4,432,3,4,96,4,7,11,11,6,112,4,14,11,1,2,3,0,9,
3 3229566029,1497848803862,1851,9,1,1,1,1,1,0,1,1,119,7,4,39,12,4,76,4,3,10,4,14,35,4,11,10,4,0,2,1,5,
4 3327363504,1504360103310,1493,9,1,2,1,1,1,1,2,0,18,4,7,141,11,4,267,3,4,68,4,12,38,12,4,8,1,1,1,0,1,
5 3326856598,1504348503996,1758,9,1,1,1,1,1,1,1,0,57,4,12,63,4,14,29,4,7,61,4,1,36,11,4,9,2,1,2,0,238,
6 3330080762,1504554410899,2094,9,1,2,1,1,1,1,0,19,4,12,29,11,4,40,4,3,119,4,7,134,7,4,9,2,1,3,0,90,
7 3287435705,1501667992132,2059,9,1,2,2,1,1,2,0,40,3,4,141,11,4,24,12,4,45,3,4,67,4,7,8,1,1,1,0,117,
```

Imagem 1: Demonstrativo da base de dados utilizada para o treinamento e testes.

4.4 Desenvolvimento

Durante o desenvolvimento do algoritmo para Aprendizado de Máquina, fora percebido nos testes iniciais que ao utilizar a função *predict* dos modelos escolhidos, ele tenta realizar a predição do jogo em questão, baseando-se nos treinos que foram realizados para o aprendizado. Porém para o nosso projeto, era necessário que o algoritmo retornasse a probabilidade de vitória para cada equipe.

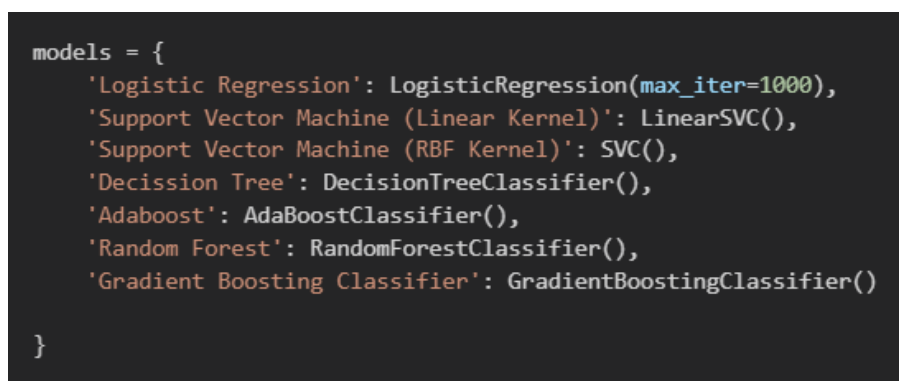
Para os testes, houve a separação das colunas específicas para realizar o treino, conforme dito anteriormente e informado abaixo:



```
df_1 = df[['winner', 'firstBlood',
'firstTower', 'firstInhibitor',
'firstBaron', 'firstDragon',
't1_towerKills', 't1_inhibitorKills',
't1_baronKills', 't1_dragonKills',
't2_towerKills', 't2_inhibitorKills',
't2_baronKills', 't2_dragonKills']]
```

Imagem 2: Modelos utilizado para realização dos primeiros testes

Então, após algumas pesquisas foi encontrado a função *predict_proba* que retorna a probabilidade que o algoritmo encontra como retorno após os treinos. Porém essa função não é utilizada em alguns modelos de treino utilizados no algoritmo, como o modelo de “Máquina de Suporte de Vetores Linear”, alguns outros modelos foram descartados por apresentarem inconstâncias nos resultados, como valores acima de 90% com apenas 1 parâmetro inicial, como o “Classificadores de Aumento de Gradiente” ou “Gradient Boosting Classifier”. Conforme as imagens 3 e 4, é possível ver as alterações dos modelos escolhidos para um retorno de porcentagens, assim como está proposto no projeto.



```
models = {
    'Logistic Regression': LogisticRegression(max_iter=1000),
    'Support Vector Machine (Linear Kernel)': LinearSVC(),
    'Support Vector Machine (RBF Kernel)': SVC(),
    'Decision Tree': DecisionTreeClassifier(),
    'Adaboost': AdaBoostClassifier(),
    'Random Forest': RandomForestClassifier(),
    'Gradient Boosting Classifier': GradientBoostingClassifier()
}
```

Imagem 3: Modelos utilizado para realização dos primeiros testes

```
models = {
    'Support Vector Machine (RBF Kernel)': SVC(probability=True),
    'Adaboost': AdaBoostClassifier(),
    'Random Forest': RandomForestClassifier()
}
```

Imagem 4: Modelos utilizado para realização dos testes finais

Além da mudança dos modelos, houve também uma alteração no modelo de Máquina de Suporte de Vetores que é o parâmetro *probability* para retornar essa probabilidade de qual lado irá vencer.

5. Resultados

O projeto obteve resultados significativos e podem ser usados para testes futuros em tempo real. Nos testes realizados com valores como se o jogo estivesse acabado de iniciar (com valores igual a 0), é possível ver que o jogo começa bem empatado e o algoritmo não consegue definir um possível vencedor com clareza, acredita-se que ele está levando em consideração o lado do mapa que mais vence.

```
Predict Support Vector Machine (RBF Kernel): [[0.5 0.5]]
Predict Adaboost: [[0.49792828 0.50207172]]
Predict Random Forest: [[0.42 0.58]]
```

Imagem 5: Exemplo de predição com o início da partida

Para fins de teste, utilizamos uma partida jogada por um jogador aleatório de um nível mais alto (Match History, 2021) nas filas ranqueadas do servidor brasileiro. Utilizando os dados dos 15 primeiros minutos e depois dos 20 para ver a diferença.

```
Predict Support Vector Machine (RBF Kernel): [[0.5 0.5]]
Predict Adaboost: [[0.50439113 0.49560887]]
Predict Random Forest: [[0.51 0.49]]
```

Imagem 6: Predição do algoritmo com dados dos 15 primeiros minutos do jogo.

```
Predict Support Vector Machine (RBF Kernel): [[0.5 0.5]]
Predict Adaboost: [[0.50114219 0.49885781]]
Predict Random Forest: [[0.5 0.5]]
```

Imagem 7: Predição do resultado com os dados dos 20 minutos do jogo.

O resultado final dessa partida analisada teve o time 1 como vencedor, E aos 20 minutos o algoritmo ainda não conseguia definir o time que iria vencer com os dados inseridos.

6. Considerações finais

Em relação ao processo de desenvolvimento, não houve muitas dificuldades, porém o tempo necessário para a execução do algoritmo é elevado mesmo com uma base de dados relativamente pequena, dada a quantidade de partidas jogadas nas ligas ao redor do mundo.

Já em relação ao resultado dos testes, houve pouco aproveitamento, pois os resultados estatísticos se mostraram inviáveis dada a porcentagem de chance de vitória parecida entre as duas equipes.

7. References

League of Legends (2021) “Como jogar - League of Legends”, <https://br.leagueoflegends.com/pt-br/how-to-play/>, April.

Demiryurek, G. (2019) “League Of Legends Match Score Prediction”, <https://www.kaggle.com/gulsahdemiryurek/let-s-predict-league-of-legends-match-score>, April.

Rey, A. (2021), “League of Legends Win Prediction”, <https://medium.com/swlh/league-of-legends-win-prediction-5f5516c4b1d7>, Abril.

Match History (2021) “Histórico de partidas”, <https://matchhistory.br.leagueoflegends.com/pt/#match-details/BR1/2268462568/201295570?tab=overview>, May.