Dota 2 Prediction: Match Win Chance Prediction

Estela Miranda Batista[1], **Fabrício Aguiar Silva**[1]

[1]Instituto de Ciências Exatas e Tecnológicas
Universidade Federal de Viçosa (UFV)
Florestal - MG - Brasil

{estela.batista, fabricio.asilva}@ufv.br

Abstract. *Games of the Multiplayer Online Battle Arena (MOBA) genre has stood out in recent years due to their popularity and the creation of professional championships (eSports), and in conjunction with this, studies on match analysis. The objective of this work is the analysis of data from professional matches of the game Defense of The Ancients 2 (Dota 2) to predict the chance of winning at a given time using the smallest possible number of attributes, varying in 4 different configurations of the features. As a result, the best classification model developed, using the random forest algorithm, obtained an accuracy of 76%, using as features the character's identification number and the characters' victory rate. In addition, a web service was developed for use by gamers.*

Resumo. *Jogos do gênero Multiplayer Online Battle Arena (MOBA) tem se destacado nos últimos anos devido a sua popularidade e criação de campeonatos profissionais (eSports), e em conjunto a isso estudos sobre análise das partidas. O objetivo deste trabalho é a análise de dados de partidas profissionais do jogo Defense of The Ancients 2 (Dota 2) para predição da chance de vitória de um determinado time utilizando do menor número de atributos possível, variando em 4 diferentes configurações dos atributos. Como resultados o melhor modelo de classificação desenvolvido, usando do algoritmo random forest, obteve uma precisão de 76%, usando como atributos o número identificador do personagem e a taxa de vitória dos personagens. Além disso, foi feito o desenvolvimento de um web service para utilização das pessoas jogadoras.*

1. Introdução

A indústria dos video games se tornou nos últimos anos um dos setores mais importantes da economia, em que segundo a pesquisa anual newzoo[1] no ano de 2022 os rendimentos chegaram a mais de 174 bilhões de dólares. Dentre os gêneros de jogos, o gênero *Multiplayer Online Battle Arena* (MOBA) tem se destacado pela grande popularidade, em que foram desenvolvidos campeonatos profissionais, fazendo com que os mesmos se tornassem uma categoria de esportes, sendo chamado de esportes digitais ou *eSports*. Tais campeonatos tem disposto de altos valores de premiação, como no campeonato *The International* do jogo *Defense of The Ancients 2* (Dota 2) que teve sua premiação em 2022 de cerca de 8 milhões de dólares.

Jogos MOBA envolvem ação e estratégia, sendo executado por times formados por 5 pessoas jogadoras cada. As pessoas jogadoras de cada time lutam entre si em um

[1]https://newzoo.com/resources/blog/global-games-market-to-generate-175-8-billion-in-2021-despite-a-slight-decline-the-market-is-on-track-to-surpass-200-billion-in-2023

mapa simétrico, como apresentado na Figura 1[2], usando de personagens que contam com diversas habilidades e aptidões diferentes, tendo como principal objetivo avançar até a base inimiga e destruir o núcleo central, chamado de *ancient*. Levando em consideração especificamente o jogo Dota 2, desenvolvido pela Valve Software[3], o mapa conta com 3 torres em cada uma das 3 lanes, como apresentado na Figura 1, além de 2 barracas em cada lane. Além disso, atualmente cada um dos jogadores podem escolher entre 124 personagens disponíveis, que possuem no mínimo 2 habilidades e no máximo 13. Outro ponto importante sobre o jogo é o conceito das partidas ranqueadas, em que o jogador é avaliado com base em suas habilidades e a vitória ou derrota, em que o jogador pode ser classificado em 8 medalhas: Arauto, Guardião, Cruzado, Arconte, Lenda, Ancestral, Divino e, Imortal. Em cada uma dessas medalhas o jogador pode possuir entre 1 e 5 estrelas, além de poder movimentar entre as medalhas a medida que vai ganhando ou perdendo as partidas a cada temporada.

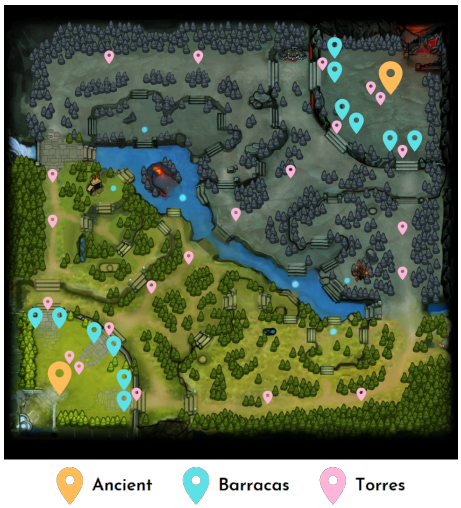


Figura 1. Mapa do Jogo Dota 2

O objetivo deste trabalho é apresentar a chance de vitória de um time em partidas do jogo MOBA Dota 2 utilizando do menor número de atributos possível visto a mecânica complexa presente no jogo. Para a classificação dos dados serão apresentadas 4 abordagens diferentes, que variam os atributos utilizados:

1. Identificador de cada personagem selecionado por cada um dos times;
2. Identificador de cada personagem selecionado por cada um dos times e a taxa de vitória do personagem escolhido;
3. Identificador de cada personagem selecionado por cada um dos times e a medalha de cada um dos jogadores;
4. Identificador de cada personagem selecionado por cada um dos times, a taxa de vitória do personagem escolhido e a medalha de cada um dos jogadores.

É importante ressaltar que o melhor modelo gerado baseado nas abordagens descritas geraram o desenvolvimento de um *web service* para que a chance de vitória em uma determinada possa ser utilizada por usuários finais: as pessoas jogadoras.

[2]Adaptado de: https://dota2.fandom.com/pt/wiki/Minimapa
[3]https://www.valvesoftware.com/pt-br/

Para realizar a previsão os dados foi utilizado de dados extraídos da API Open Dota[4] e com enriquecimento dos dados sobre a taxa de vitória de cada personagem usando da ferramenta DotaBuff[5]. Na API Open Dota são disponibilizados a distinção no tipo das partidas, normais ou profissionais, dessa forma foram utilizados de dados de partidas profissionais, a fim de se obter um melhor balanceamento, visto que em partidas normais os jogadores possuem diversos níveis de conhecimento, além de que alguns jogadores privam o compartilhamento dos dados de sua conta, fazendo com não seja possível por exemplo identificar o personagem selecionado na partida.

O restante deste trabalho está organizado contando com a apresentação de trabalhos relacionados na seção 2; os materiais e métodos utilizados durante a pesquisa na seção 3; os resultados obtidos na seção 4; e, por fim, as considerações finais na seção 5.

2. Trabalhos Relacionados

No que compete às pesquisas que abordam a predição de vitórias em partidas de jogos MOBA [Almeida et al. 2017] analisou 123.326 partidas do jogo Dota 2 utilizando dos algoritmos de classificação (executados na ferramenta WEKA) *Naive Bayes*, KNN e, Árvore de Decisão, alcançando uma precisão de aproximadamente 77% levando em consideração duas abordagens: a seleção dos personagens em cada time, e além da seleção dos personagens considerando também a duração da partida.

[Hodge et al. 2021] analisou partidas ao vivo e profissionais de Dota 2 utilizando de modelos padrão de aprendizado de máquina (usando do software WEKA treinaram um algoritmo *Logistic Regression* e um algoritmo *Random Forest*, além do algoritmo *Microsoft LigthGBM*), engenharia de recursos e otimização, atingindo uma precisão de cerca de no máximo 75% em todos os algoritmos.

[Ani et al. 2019] analisou partidas de outro conhecido jogo MOBA: League of Legends. Foram analisadas 1500 partidas de pessoas jogadoras profissionais, que possuíam em seu *dataset* 97 atributos, de forma que foram feitas análises e predições de vitória antes da partida e durante a partida. Na análise dos dados foi utilizado dos algoritmos random forest, adaboost, gradient boosting, extreme gradient boosting, que atingiram antes da partida uma precisão de 95%, durante a partida uma precisão de 98% e ao combinar ambas as abordagens uma precisão de 99.75%.

Por fim, considerando que jogos MOBA possuem uma mecânica muito complexa [Tyran and Chomatek 2021] analisou a presença de partidas atípicas em bases de dados, ou seja, partidas em que um determinado time perde embora tenha superado a equipe inimiga durante toda a partida. Um ponto importante tratado no trabalho é que tal situação ocorre em sua maioria em partidas em que os jogadores possuem diversos níveis de conhecimento, como em partidas no modo Turbo[6], dessa forma, como citado anteriormente foi feita a escolha de apenas partidas profissionais para um melhor treinamento dos dados.

[4]https://www.opendota.com/

[5]https://pt.dotabuff.com

[6]Neste modo de jogo é como se o tempo de jogo andasse de uma forma mais rápida, em que o *gold* por minuto é aumentado, tempo de dia/noite mais curtos, entre outras características.

3. Materiais e Métodos

A metodologia adotada neste trabalho levou em consideração as etapas apresentadas na Figura 2 que serão apresentados nas subseções a seguir, tendo como adição uma subseção de explicação sobre as tecnologias adotas para o desenvolvimento de um *web service*. É importante ressaltar que a análise dos dados desenvolvida utilizou da linguagem Python na ferramenta Jupyter Notebook[7].

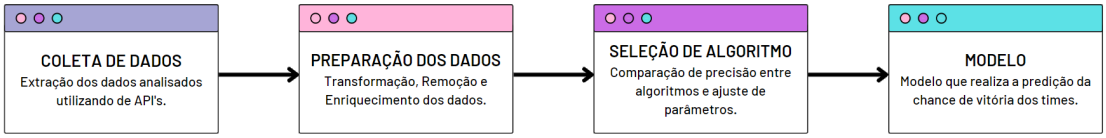
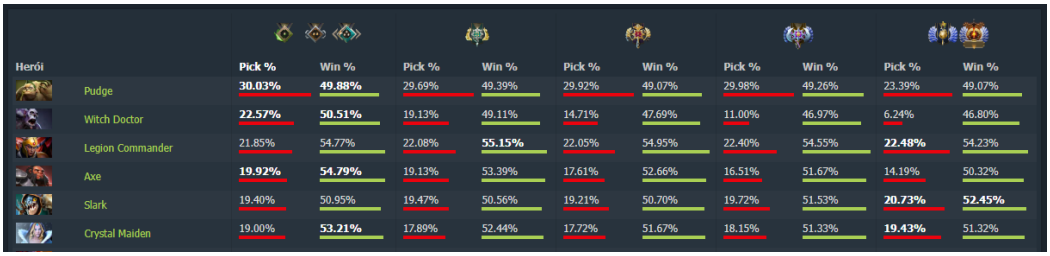


Figura 2. Etapas do Processo de Análise dos Dados

3.1. Coleta dos Dados

Durante a etapa de coleta dos dados, como citado anteriormente, na seção 1, foi utilizado da API Open Dota, em que inicialmente foi feita uma extração para salvar os valores de número identificador das partidas profissionais. De posse do valor identificador de cada partida foi realizado a extração dos dados de cada partida, em que além de informar os personagens selecionados também é possível verificar as medalhas em que cada jogador se encontra no momento da extração, dessa forma salvando essas informações em um arquivo CSV.

Como última etapa da coleta de dados, visto que a API citada anteriormente retorna apenas as últimas 100 partidas que um determinado personagem participou, indicando se naquela determinada partida o time do personagem perdeu ou venceu foi optado por salvar manualmente a taxa de vitórias de cada um dos personagens presentes no jogo usando da ferramenta DotaBuff, sendo importante ressaltar, que como apresentado na Figura 3 a taxa de vitória varia de acordo com a medalha em que se está jogando, dessa forma sendo anotado a taxa de vitória na medalha em que os jogadores profissionais em sua maioria estão: Divino e Imortal. Ao final da extração dos dados foram extraídas informações de 1387 partidas profissionais de Dota 2.



Herói	Pick %	Win %	Pick %	Win %	Pick %	Win %	Pick %	Win %	Pick %	Win %
Pudge	**30.03%**	**49.88%**	29.69%	49.39%	29.92%	49.07%	29.98%	49.26%	23.39%	49.07%
Witch Doctor	**22.57%**	**50.51%**	19.13%	49.11%	14.71%	47.69%	11.00%	46.97%	6.24%	46.80%
Legion Commander	21.85%	54.77%	22.08%	**55.15%**	22.05%	54.95%	22.40%	54.55%	**22.48%**	54.23%
Axe	**19.92%**	**54.79%**	19.13%	53.39%	17.61%	52.66%	16.51%	51.67%	14.19%	50.32%
Slark	19.40%	50.95%	19.47%	50.56%	19.21%	50.70%	19.72%	51.53%	**20.73%**	**52.45%**
Crystal Maiden	19.00%	**53.21%**	17.89%	52.44%	17.72%	51.67%	18.15%	51.33%	**19.43%**	51.32%

Figura 3. Exemplo de Classificação de Taxa de Vitória de Personagens na Ferramenta Dota Buff

Ao final, o *dataset* geral, que apenas varia na remoção de algumas colunas dependendo da abordagem analisada, obteve um total de 32 colunas, constando o número

[7]https://jupyter.org/

identificador da partida (removido durante as análises de predição), a coluna que identifica o time vencedor (*dire* ou *radiant*), e uma coluna para um dos personagens selecionados (10 ao todo) e respectivamente a taxa de vitória do mesmo e a medalha do jogador que estava utilizando deste personagem.

3.2. Preparação dos Dados

Para realizar o treinamento dos modelos não foi necessária nenhuma preparação dos dados, ou seja, transformações e pré-processamento, isso porque todos os dados extraídos e enriquecidos na base de dados foram retornados em valores numéricos, dessa forma, sendo aceito em qualquer algoritmo de classificação.

Porém, deve-se ressaltar que ao receber dados de um usuário no *web service* desenvolvido, que será apresentado posteriormente, é realizada uma transformação e adição nos dados, de forma que o nome de cada personagem digitado é transformado em seu número identificador, salvo em um dicionário, e sua taxa de vitória de forma análoga é adicionado ao vetor de predição, que também foi salvo em um dicionário, para assim facilitar a inserção dos dados no vetor de predição.

3.3. Escolha do Modelo

Para a escolha do modelo foram selecionados previamente alguns algoritmos apresentados na literatura para a classificação de resultados em partidas de jogos MOBA. Os algoritmos selecionados foram Logistic Regression, Random Forest, Decision Tree, K-Nearest Neighbors (KNN) e Naive Bayes.

O algoritmo Logistic Regression é baseado no recurso de estimar a probabilidade associada à ocorrência de um determinado evento baseado em um conjunto de variáveis exploratórias, ou seja, ele estima a probabilidade de cada atributo ocorrer em uma determinada classe de classificação [Cramer 2003]. O algoritmo Decision Tree é baseada na divisão dos atributos em grupos suficientemente distintos entre si, de forma que ao gerar a árvore os nós filhos atribuídos devem possuir características diferentes [Quinlan 1996], como por exemplo uma sequência 0111, em que os dois primeiros valores são na cor rosa, os dois últimos na cor azul, os rosas passariam por uma segunda divisão, enquanto os azuis apenas verificariam a cor por serem iguais. O algoritmo Random Forest por sua vez utiliza de árvores de decisão em sua construção, de forma que são geradas diversas árvores de decisão, que possuem uma baixa correlação entre si, e durante a classificação aquelas que melhor se adequarem para os dados de entrada são usadas [Ho 1995]. O algoritmo K-Nearest Neighbors (KNN) utiliza de cálculos de distância (euclediana, manhattan, etc.) para verificar a distância de um determinado dado em relação aos demais, de forma que ele utiliza do valor dos atributos para determinar essa distância [Cover and Hart 1967], ou seja, valores de atributos similares tem uma distância menor. Por fim, o algoritmo Naive Bayes é um modelo probabilístico que utiliza como base o teorema de Bayes, que determina a probabilidade de um determinado evento A ocorrer em relação a um evento B que ocorreu, fazendo com que os atributos sejam independentes para se obter melhores resultados [Webb 2010].

A partir dos algoritmos selecionados e das bases de dados foi realizada uma aplicação do algoritmo *cross validation* em todos os algoritmos para medir a precisão dos mesmos em relação a cada uma das bases de dados com diferentes atributos. É importante ressaltar que a aplicação do algoritmo utilizou de 30 iterações, em que as bases

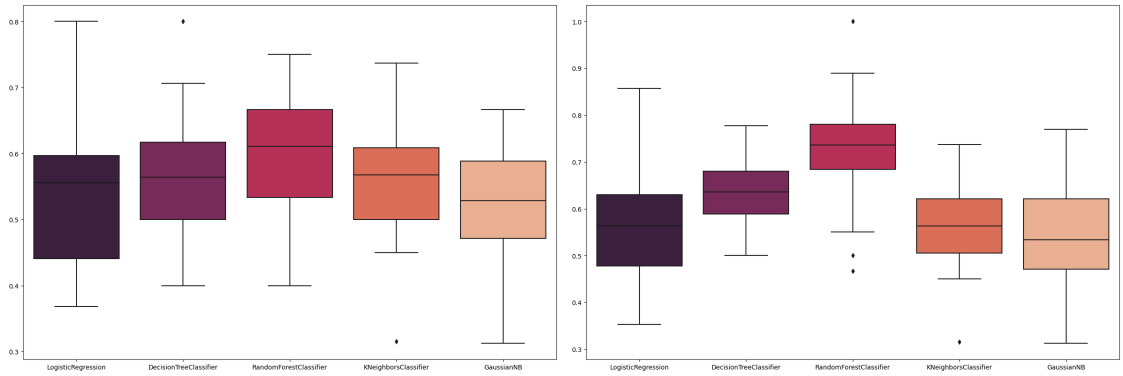
de dados foram separadas em treino e teste no formato 80% treino e 20% teste. Além disso, durante a validação usando do *cross validation* para a escolha do melhor algoritmo foi utilizado apenas da base de treino.

Na Tabela 1 podemos verificar os resultados encontrados de precisão após a execução do *cross validation* para cada um dos algoritmos em cada uma das abordagens, em que os algoritmos estavam com as configuração padrão da biblioteca *sklearn*[8] para cada um deles, sendo os valores demarcados em negrito o algoritmo selecionado para cada uma das abordagens.

Abordagem	Logistic Regression	Decision Tree	Random Forest	KNN	Naive Bayes
(1) Personagem Selecionado	54%	55.6%	**61.4%**	55.6%	53.5%
(2) Personagem Selecionado e Taxa de Vitória	56%	63.1%	**74.3%**	55.8%	54.3%
(3) Personagem Selecionado e Medalha da Pessoa Jogadora	53.5%	57.7%	**59.4%**	54.1%	49.7%
(4) Personagem Selecionado, Taxa de Vitória e Medalha da Pessoa Jogadora	56.1%	61.5%	**70.5%**	52.7%	52.8%

Tabela 1. Comparação de Algoritmos por Abordagem no Cross Validation

Na abordagem (1) o algoritmo random forest foi escolhido por possuir um valor de precisão superior aos dos demais algoritmos, além de possuir um intervalo de variação similar aos demais na execução do *cross validation*, podendo isso ser observado na Figura 4(a), em que cada *boxplot* apresentado segue a ordem apresentada na Tabela 1 dos algoritmos. Na abordagem (2) novamente o algoritmo random forest se sobressaiu aos demais possuindo uma precisão maior, além de uma variação menor na aplicação do *cross validation* apresentado na Figura 4(b).

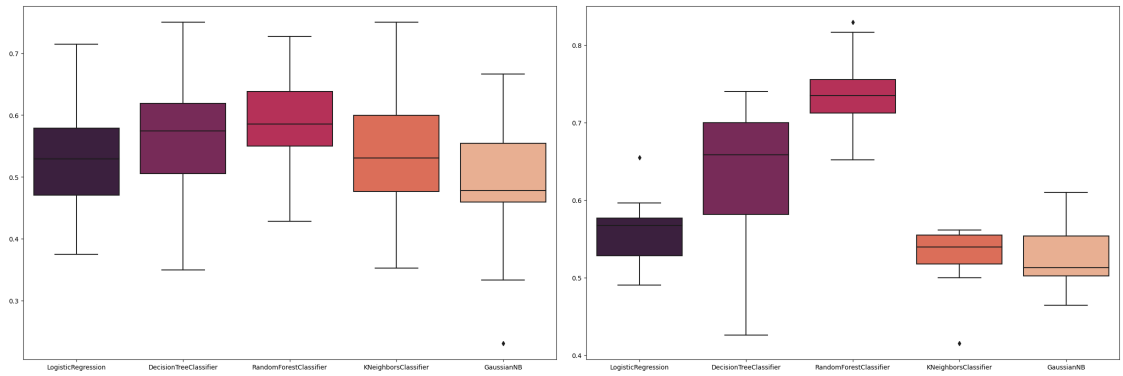


(a) Abordagem (1) (b) Abordagem (2)

Figura 4. Intervalo de variação de precisão no *cross validation*

Na abordagem (3) o algoritmo random forest teve como atribuído um valor de precisão similar aos demais algoritmos, porém ao olharmos para o seu intervalo de variação, na Figura 5(a), ele possui uma variação menor, sendo assim escolhido para essa abordagem. Na abordagem (4) o algoritmo random forest foi selecionado por possuir uma precisão maior em relação aos demais algoritmos, e uma variação menor variação na aplicação do *cross validation*, além de possuir um intervalo de maior precisão como apresentado na Figura 5(b).

[8]https://scikit-learn.org/stable/



(a) Abordagem (3) (b) Abordagem (4)

Figura 5. Intervalo de variação de precisão no *cross validation*

3.4. Criação do *Web Service*

4. Resultados

O algoritmo selecionado para cada uma das abordagens passou por uma análise do ajuste de parâmetros, para se determinar o melhor ajuste, em que foi utilizado do algoritmo *grid search*. O algoritmo do random forest foi variado em relação ao número de estimadores (*n_estimators*), assumindo os valores 50, 100 e 200; o critério (*criterion*), que pode assumir os valors *entropy* ou *gini*; a profundidade máxima das árvores (*max_depth*), assumindo os valores *None*, 50 ou 100; como divisão miníma (*min_samples_split*) variações de 2, 4 e 6; o número máximo de atributos por nó da árvore (*max_features*), podendo ser *auto*, *sqrt* ou *log2*; e, como forma de construção da árvore (*bootstrap*) podendo ser *True* para usar todo o conjunto de dados, e *False* caso contrário.

Na abordagem (1) os parâmetros selecionados para o algoritmo do random forest foram: *bootstrap* = *False*; *criterion* = *gini*; *max_depth* = *None*; *max_features* = *auto*; *min_samples_split* = 2; e, *n_estimators* = 50. Na abordagem (2) os parâmetros selecionados para o algoritmo random forest foram: *bootstrap* = *False*; *criterion* = *gini*; *max_depth* = *None*; *max_features* = *auto*; *min_samples_split* = 4; e, *n_estimators* = 200. Na abordagem (3) os parâmetros selecionados para o algoritmo random forest foram: *bootstrap* = *False*; *criterion* = *gini*; *max_depth* = *None*; *max_features* = *auto*; *min_samples_split* = 4; e, *n_estimators* = 100. Na abordagem (4) os parâmetro selecionados para o algoritmo random forest foram: *bootstrap* = *False*; *criterion* = *gini*; *max_depth* = *None*; *max_features* = *log2*; *min_samples_split* = 2; e, *n_estimators* = 100.

Para fins de comparação, após os ajustes de parâmetro realizado em cada uma das abordagens foi calculado o valor de sua precisão usando do ajuste de parâmetros na base de treino e teste como um todo, sendo os resultados apresentados na Tabela 2.

Além do resultado da precisão dos algoritmos com base em seu ajuste de parâmetros foi realizado a implementação das abordagens apresentadas na seção 2 por [Almeida et al. 2017] utilizando da base de dados deste trabalho, sendo os resultados obtidos por cada algoritmo em cada abordagem apresentados na Tabela 3. Pode-se observar então que usando de atributos similares, organizados de outras formas, e usando de outros atributos é possível se obter um resultado similar ou com maior precisão, como apresentado nas abordagens (3) e (4) apresentadas neste trabalho.

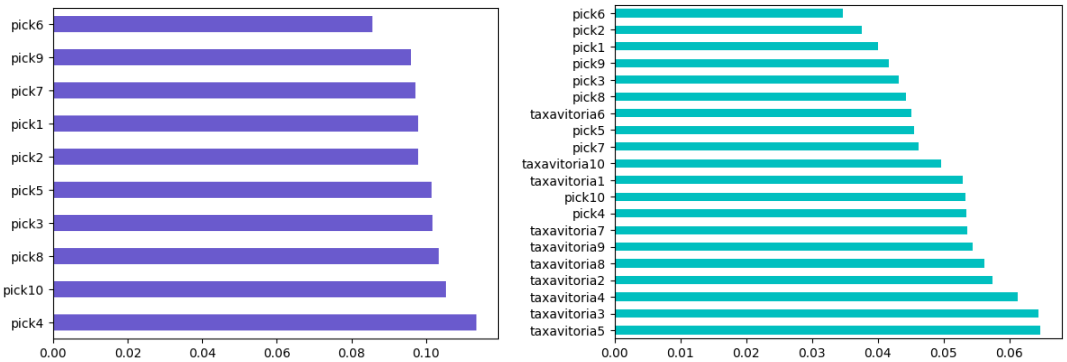
Abordagem	Precisão
(1) Personagem Selecionado	60.93%
(2) Personagem Selecionado e Taxa de Vitória	76.49%
(3) Personagem Selecionado e Medalha da Pessoa Jogadora	61.16%
(4) Personagem Selecionado, Taxa de Vitória e Medalha da Pessoa Jogadora	74.98%

Tabela 2. Comparação de Resultados por Abordagem

Abordagem	Naive Bayes	Decision Tree	KNN
(1) Vetor Binário de Personagem Selecionado [Almeida et al. 2017]	68.4%	56.8%	40%
(2) Vetor Binário de Personagem Selecionado e Tempo de Partida [Almeida et al. 2017]	60.4%	56.8%	51%

Tabela 3. Comparação de Algoritmos por Abordagem no Cross Validation

Outra análise importante feita ao final da execução de cada uma das abordagens foi o entendimento dos atributos mais importantes para cada um dos casos, trazendo assim explicabilidade dos dados. Na abordagem (1), como podemos observar na Figura 6(a) todos os atributos possuíram importância similar, visto que como se utilizou apenas do número identificador dos personagens selecionados todos devem ser analisados em conjunto para determinar o time vencedor da partida. Na abordagem (2) como pode se observar na Figura 6(b) mesmo com alguns atributos se sobressaindo a outros, todos os atributos possuem valores similares de importância, e como foi utilizado apenas do número identificador de cada personagem e da taxa de vitória dos mesmos essas são informações relevantes para determinar a vitória de um time, porque aliado ao apresentado na abordagem (1) a informação da taxa de vitória se torna relevante visto que personagens com altas taxas de vitória podem indicar habilidades mais poderosas usadas pelos jogadores que se tornam facilitadores.

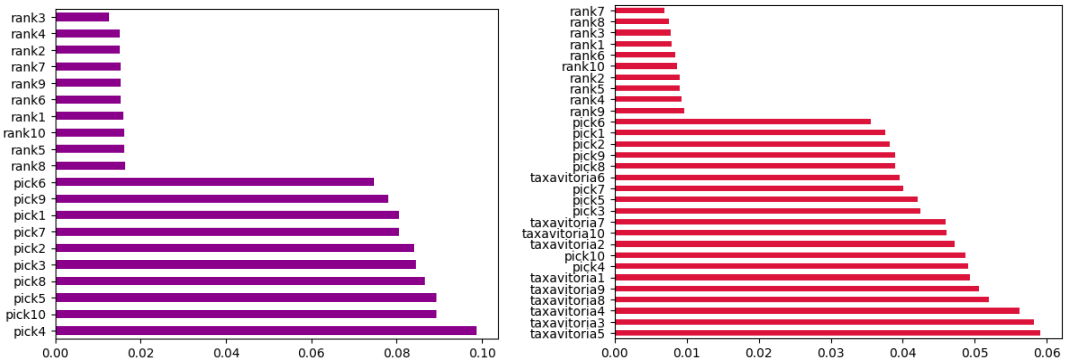


(a) Abordagem (1) (b) Abordagem (2)

Figura 6. Atributos Importantes

Na abordagem (3) como pode ser observado na Figura 7(a) os atributos que diziam respeito a medalha de cada uma das pessoas jogadoras possuíram uma baixa importância, e ao se analisar os dados é possível verificar que isso possa ter ocorrido neste modelo em

específico pelo fator de que os dados estão relacionados a pessoas jogadoras profissionais, e como citado anteriormente possuindo como valor atribuído a sua medalha valores entre 70 e 80. Por fim, na abordagem (4) foi possível observar uma relação com a abordagem (2), em que a informação da taxa de vitória e da escolha do personagem possuem importância similar, enquanto como apresentado na abordagem (3) a informação sobre a medalha da pessoa jogadora não se tornou tão relevante, porém ao olharmos para a precisão gerada é possível que tenha contribuído para um melhoramento no modelo, sendo isso apresentado na Figura 7(b).



(a) Abordagem (3) (b) Abordagem (4)

Figura 7. Atributos Importantes

5. Conclusão

Como trabalhos futuros sugere-se a ampliação da base de dados para se verificar o comportamento da precisão dos modelos em base de dados maiores. Outro trabalho futuro importante é a aplicação dos modelos em base de dados que possuam partidas normais, de forma a se variar as medalhas que as pessoas jogadoras possuam e entender se tal atributo continua a não possuir uma importância significativa para o modelo.

Referências

Almeida, C. E. M., Correia, R. C. M., Eler, D. M., Olivete-Jr, C., Garci, R. E., Scabora, L. C., and Spadon, G. (2017). Prediction of winners in moba games. In *2017 12th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6.

Ani, R., Harikumar, V., Devan, A. K., and Deepa, O. (2019). Victory prediction in league of legends using feature selection and ensemble methods. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pages 74–77.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27.

Cramer, J. (2003). The Origins of Logistic Regression. *SSRN Electronic Journal*.

Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1.

Hodge, V. J., Devlin, S., Sephton, N., Block, F., Cowling, P. I., and Drachen, A. (2021). Win prediction in multiplayer esports: Live professional match prediction. *IEEE Transactions on Games*, 13(4):368–379.

Quinlan, J. R. (1996). Learning decision tree classifiers. *ACM Computing Surveys*, 28(1):71–72.

Tyran, J. and Chomatek, L. (2021). Influence of outliers in moba games winner prediction. *Procedia Computer Science*, 192:1973–1981. Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 25th International Conference KES2021.

Webb, G. I. (2010). *Naïve Bayes*, pages 713–714. Springer US, Boston, MA.