

# IA Aplicada no LoL Worlds 2020

Bruno Pires Moreira Silva  
UNIFESP – Universidade Federal de São Paulo  
Mogi das Cruzes, Brasil  
bruno.pires22@gmail.com

*Através dos dados coletados no campeonato mundial de League of Legends, um jogo eletrônico de estratégia, três técnicas diferentes de Inteligência artificial foram aplicadas neles, a fim de poder concluir sobre as equipes participantes e sobre os resultados gerais do campeonato.*

**Palavras-chave** — *League of Legends, Composições, estratégias, campeonato, combinações.*

## I. INTRODUÇÃO

A análise de competições esportivas é algo muito comum e que sempre existiu, independentemente do uso de tecnologias para isso. Tentar prever qual time tem mais chance de ganhar um determinado jogo, calcular as chances de ser campeão com base nos rivais mais difíceis ou entender a forma que o time busca a vitória é algo historicamente muito discutido, independente do esporte, da região ou da competição em si. Um fato que facilita isso é a forma genérica que torneios como a Copa do Mundo, NBA, Champions League e diversos outros campeonatos utilizam para a disputa. Normalmente, ele é baseado em fases: as primeiras, no qual os times previamente classificados são divididos em grupos de mesmo tamanho, ou seja, com a mesma quantidade de times, e jogam uns contra os outros uma determinada quantidade de vezes, a fim de que alguns deles possam se sobressair, assim se classificando para a segunda fase. Essa fase, também é conhecida como “mata, mata”, pois os times jogam um contra os outros e perdedor desses embates é desclassificado. Isso é feito de forma recursiva, enquanto sobrar ao menos um time, sendo que alguns padrões podem ser observados.

O campeonato mundial de League of Legends, um jogo do tipo MOBA (Multiplayer Online Battle Arena, um gênero dos jogos onde dois times de cinco jogadores se distribuem em um mapa simétrico, a fim de destruir as torres e a base do time inimigo) que também utiliza esse formato para consagrar um time como campeão. Com isso, será analisada as estratégias desses times, utilizando o algoritmo “K-nearest neighbor” além das formas que uma equipe pode chegar ao título, explorando o maior número possível de possibilidades com heurísticas de busca. As relações entre os outros artigos não terão como foco semelhanças entre os algoritmos, mas sim, pelas semelhanças de como a relação entre matemática, computação e o esporte é feita.

## II. TRABALHOS RELACIONADOS

### A. Aplicação de Algoritmos de Machine Learning para Classificação Automática de Problemas Futebolístico

Esse artigo utiliza de diferentes algoritmos para tentar prever se dado um jogo de futebol da liga portuguesa, ambos os times marcam ou não um gol. Entre as técnicas utilizadas, kNN está presente, sendo inclusive, a de maior acurácia. Entretanto, como os grupos são apenas dois, e não cinco como será feito nesse caso, os dados podem facilmente se tornarem binários, aumentando a eficiência do kNN.

### B. Utilização de algoritmos do tipo machine learning supervisionado para a caracterização dos resultados da copa do mundo de futebol de 2018

Analisando os 64 jogos da Copa do Mundo realizada em 2018, foram levantadas 40 variáveis de fatores que se relacionavam com o resultado final da partida, e a partir delas, com o uso de diversas técnicas dentro de machine learning, os resultados dos jogos foram previsto com uma taxa de acerto de 57,18%.

### C. Proposta de ferramenta de predição utilizando machine learning

Com base em ligas nacionais de futebol de países pertencentes a Europa, diversos dados de competições completas foram coletados e usados para treinamento. Com isso, informações sobre o primeiro tempo de outros jogos que não foram usadas para o treinamento foram fornecidas, a fim de prever o vencedor da partida, ao fim do segundo tempo, que teve 62% de acurácia, mesmo com a imprevisibilidade das disputas.

### D. Previsão de resultados no futebol por meio de técnicas de aprendizado de máquina

Tendo como base jogos das 7 principais ligas nacionais de futebol da Europa e utilizando diversas variáveis para avaliação dos 3 setores de um time (meio de campo, defesa e ataque), além de relacionar os resultados de acordo com o mando de campo, a predição de outros jogos foi feita, obtendo 52% de acurácia, sendo que esse modelo se mostrou rentável se usado como base para apostas esportivas.

### E. Caracterização de Perfis de Comportamento de Equipes em League of Legends

Baseado no histórico de partidas de League of Legends, métodos de data mining foram aplicados para analisar pontos que interferem nas vitórias e derrotas das equipes, a fim de descobrir características relacionadas a esses fatos, criar perfis em cima disso e separar esses perfis, que acabaram sendo 7, em quatro grupos, avaliados com base no desempenho das equipes.

## III. METODOLOGIA

### A. A predição da estratégia utilizada com base nos campeões utilizados

Como já foi dito anteriormente, League of Legends é um jogo de estratégia com o objetivo de destruir a base inimiga, com duas equipes de 5 componentes cada uma. Todos os jogadores de cada time escolhem um campeão (os personagens do jogo são denominados dessa maneira) dentre os 149 existentes, com cada um possuindo uma função específica, de acordo com o seu posicionamento no mapa, as chamadas rotas:



**Figura 1** – Mapa do League of Legends e suas rotas.

A rota superior (conhecida como “Top”) e a rota do meio (conhecida como “Mid”) são ocupadas por um jogador de cada time, assim como a Selva (ou “Jungle”), dividida em 4 partes e com um jogador de cada time atuando nela, por fim a rota inferior (conhecida como “Bot”), onde uma dupla jogadores de cada time, composta pelo atirador (ou “ADC”) e pelo Suporte (abreviado para “Sup”), iniciam a partida. Vale ressaltar que para cada uma dessas classes, Top, Jungle, Mid, ADC e Sup, existe um conjunto de campeões destinados a essas funções.

Ao decorrer do jogo, inevitavelmente ocorrem situações que esses dois times precisam lutar entre si, pois após abater os inimigos, a conquista de objetivos, como torres, monstros dos mapas que concedem aprimoramentos para todo o time (como mais vida ou dano, por exemplo), ou até mesmo a destruição da base, quando a partida já está mais próxima do seu fim. Existem algumas combinações entre os campeões de um time que deixa clara a forma que se pretende lutar, ou seja, uma estratégia de luta é adotada, pois um campeão costuma ser possuir uma função específica, de acordo com suas habilidades. Essas

estratégias serão as classes do algoritmo kNN, ([Item III.C](#)), e elas serão 5 diferentes:

- Estratégia 1 - Peel: Campeões com alta capacidade de “carregar” as lutas sozinho, ou seja, conseguem abater o time inimigo com facilidade entretanto podem ser facilmente abatidos, precisam de proteção. Campeões que dão uma grande quantidade de danos, normalmente a distância, e campeões que os protegem, recebem uma nota maior nessa categoria. Exemplos: Lulu (protetor), Twitch (fonte do dano), Taric (protetor), Azir (fonte de dano).
- Estratégia 2 - Dive: Campeões que possuem habilidades que encurtam a distância até o inimigo “pulando” neles. Exemplos: Sylas, Volibear, Renekton, Wukong.
- Estratégia 3 - Split: Campeões que jogam separados do time, os quais querem que a luta aconteça entre 4 jogadores enquanto eles duelam, normalmente nas rotas laterais (Top ou Bot). Exemplos: Camille, Jax, Ekko.
- Estratégia 4 - Poke: Campeões que causam dano a uma longa distância, sem se colocar em perigo e para que o inimigo comece a luta sem parte de sua vida. Exemplos: Jayce, Nidalee, Zoe, Orianna.
- Estratégia 5 - Wombo Combo: O termo tem origem dos jogos de luta, como Street Fighter, pois é dito quando uma sequência de golpes é encaixada no oponente, sem dar chance de reação. No LoL, quando um time possui campeões com capacidade de aplicar uma sequência de habilidades em área, atingindo vários inimigos, podemos rotular esse time como um time de Wombo Combo. Exemplos: Malphite, Orianna, Leona, Aphelios.

É importante ressaltar que, em determinadas situações, mesmo com uma análise humana, a estratégia utilizada pela equipe não fica muito evidente. Porém, em grande parcela dos jogos podemos apenas ver os 5 campeões escolhidos por um time e a estratégia adotada ficará extremamente clara. Assim, teríamos um comportamento padronizado, que poderia ser discretizado de tal forma que um algoritmo de aprendizado supervisionado pudesse definir a estratégia em questão. Analisando métodos para fazer isso, a melhor forma foi fazer duas análises:

1. Tendo como inspiração jogos de esporte, como o jogo de futebol “FIFA”, lançado anualmente pela produtora EA Sports onde jogadores reais são representados por cartas, com números para mensurar suas habilidades de jogo, as mesmas representações foram aplicadas aos 68 campeões de escolhidos na primeira fase do Campeonato Mundial de League of Legends, denominada PlayIn (mais informações sobre o formato do campeonato em [III.B](#)). A avaliação foi feita com a

atribuição manual de uma nota entre 0 e 5 para as mesmas estratégias apresentadas anteriormente, ou seja, 5 notas, uma para peel, dive, split, poke, e wombo combo.



Figura 2 – Cristiano Ronaldo mostrando a sua representação por meio de atributos numéricos no jogo “Fifa 18”.



Campeao	Peel	Dive	Split	Poke	Wombo Combo
Twitch	5	0	0	0	5

Figura 3 – Foto do campeão “Twitch” e seus atributos. Esse campeão se encaixa na função de atirador, sendo um carregador (peel=5), pois possui habilidades com grande capacidade de causar dano em área (Wombo Combo =5)

2. Analisando os times, com base em grupos de jogadores e ex-jogadores profissionais de League of Legends que se reúnem para discutir os jogos, e visando conclusões para prever diversos dados para realizar apostas, as estratégias para cada um dos times foi anotada, sem relação nenhuma com os atributos criados no item anterior.

Time	Top	Jungle	Mid	ADC	Sup	Vitoria	Classe
INTZ	Urgot	Gragas	Tristana	Ashe	Alistar	0	Dive
MAD Lions	Renekton	Evelynn	Lucian	Jhin	Pantheon	1	Split

Figura 4 – Combinações de campeões das equipes que fizeram a primeira partida do mundial. A equipe europeia MAD Lions foi a vencedora, utilizando como estratégia Split, em cima dos brasileiros da INTZ, que fracassaram ao tentar utilizar um time de Dive.

Com isso, foram construídos dois conjuntos de dados: um com os 5 campeões de cada um dos 76 jogos e outro com os já ditos 68 campeões escolhidos nesses jogos.

B. O Campeonato mundial de League of Legends de 2020, o LoL worlds.

O LoL Worlds é realizado anualmente, normalmente entre os meses de setembro e outubro, no país do ganhador do ano anterior (a China, no caso desse ano, devido a vitória da FunPlus Phoenix no LoL Worlds 2019). Ele é tradicionalmente feito em 3 fases, que nesse ano, funcionaram da seguinte forma:

- Primeira Fase – PlayIn: 10 times de regiões secundárias são dispostos em 2 grupos diferentes, de 5 equipes cada. Essas equipes jogam uma vez entre si, totalizando 4 jogos para cada equipe. Os primeiros colocados avançam para a próxima fase; o segundo, disputa a classificação para a próxima fase com o vencedor da partida entre o terceiro e o quarto time do outro grupo; e o último colocado, o quinto, é desclassificado. Ou seja, graficamente temos:

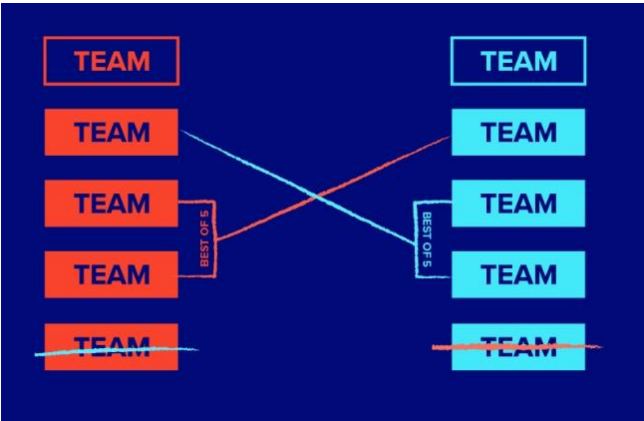


Figura 5 – Representação gráfica do PlayIn.

Os 10 times dispostos nos dois grupos deste ano, após os jogos, são:




Group A						
	Team				Games	
1	NA		Team Liquid		3 - 1	75%
2	OCE		Legacy Esports		3 - 1	75%
3	TR		SuperMassive Esports		2 - 2	50%
4	EU		MAD Lions		1 - 3	25%
5	BR		INTZ		1 - 3	25%

Figura 6 – Grupo A Play In


Group B						
	Team				Games	
1	PCS		PSG Talon		3 - 1	75%
2	CIS		Unicorns Of Love		3 - 1	75%
3	LAT		Rainbow7		2 - 2	50%
4	CN		LGD Gaming		1 - 3	25%
5	JP		V3 Esports		1 - 3	25%

Figura 7 – Grupo B Play In.

Assim, as equipes PSG Talon e Team Liquid já se classificaram para a fase 2. Os outros classificados são os vencedores das seguintes partidas:









Elimination Round				Qualification Round			
				Qualification Match			
TR		SUP	3	CIS		UOL	3
EU		MAD	2	TR		SUP	0
				Qualification Match			
LAT		R7	0	OCE		LGC	0
CN		LGD	3	CN		LGD	3

Figura 8 – Disputa das outras 2 vagas ao fim do PlayIn

Ou seja, a LGD e a UOL também se classificaram para a segunda fase. Vale ressaltar que os dados coletados no item anterior, [III.A](#), foram provenientes de todos os jogos até aqui, o fim do PlayIn.

- Segunda Fase: Com os 4 grupos montados, 3 times previamente classificados e o time classificado do Play In, é disputada a fase de grupos, com 4 grupos de 4 equipes, com todas essas equipes jogando contra si, sendo que os dois primeiros são os classificados para

a terceira e última fase. Os grupos, após as partidas tiveram os seguintes resultados:

Group A						
	Team				Games	
1	CN		Suning		4 - 2	67%
2	EU		G2 Esports		4 - 2	67%
3	NA		Team Liquid		3 - 3	50%
4	PCS		Machi Esports		1 - 5	17%

Figura 9 – Grupo A Fase de grupos

Group B						
	Team				Games	
1	KR		DAMWON Gaming		5 - 1	83%
2	CN		JD Gaming		4 - 2	67%
3	PCS		PSG Talon		2 - 4	33%
4	EU		Rogue		1 - 5	17%

Figura 10 – Grupo B Fase de grupos.





Group C						
	Team				Games	
1	KR		Gen.G		5 - 1	83%
2	EU		Fnatic		4 - 2	67%
3	CN		LGD Gaming		3 - 3	50%
4	NA		Team SoloMid		0 - 6	0%

Figura 11 – Grupo C Fase de grupos


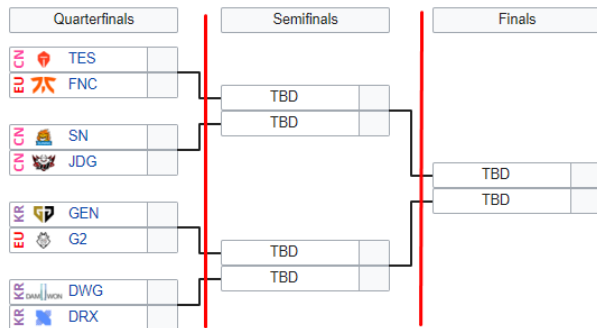
Group D						
	Team				Games	
1	CN		Top Esports		5 - 1	83%
2	KR		DRX		4 - 2	67%
3	NA		FlyQuest		3 - 3	50%
4	CIS		Unicorns Of Love		0 - 6	0%

Figura 12 – Grupo D Fase de grupos.



Os times destacados em verde foram para a terceira fase, conhecida popularmente como “mata-mata”.

- Terceira fase – Mata-mata: Organizadas em três partes:
  - Quartas de final, onde os primeiros colocados da fase de grupos jogam contra os segundos colocados de outros grupos, ou seja, é admitido a reincidência de algum confronto que já ocorreu na fase de grupos.
  - Semifinal: Ganhadores das quartas de final.
  - Final: Ganhadores da semifinal.



**Figura 13** – Mata-mata.

Até a data atual, somente a fase de grupos foi jogada, ou seja, ainda não temos nenhum semifinalista, apenas o sorteio de uma das quartas de final.

O ganhador da final é o campeão do LoL Worlds 2020.

### C. Algoritmo kNN

O algoritmo kNN é um algoritmo de aprendizado supervisionado muito utilizado, por ser simples e eficiente para determinadas tarefas de classificação. Ele funciona da seguinte maneira:

- Uma quantidade de vetores, com n elementos (n-uplas), são utilizados para fornecer informações numéricas sobre algum dado, sendo que cada um desses vetores possui uma classe. Essa parte é chamada de treinamento
- Ao inserir um novo vetor, com mesmas n-uplas de diferentes valores, é calculado a distância entre esse novo vetor e todos os outros que foram obtidos no treinamento. Podemos calcular essa distância com diferentes fórmulas:

- Distância Euclidiana:

$$E(x, y) = \sqrt{\sum_{i=0}^n (x_i - y_i)^2}$$

- Distância Minkowski

$$d = \left( \sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}$$

- Distância de Manhattan

$$d = \sum_{n=1}^{n-1} |x_n - x_{n+1}| + |y_n - y_{n+1}|$$

Após a escolha de um método para calcular as distâncias, podemos começar a pensar graficamente nos dados. Após o treinamento, os vetores de mesma classe gerarão pontos que estarão dispostos próximos uns dos outros. Assim, existirá a aglomeração desses grupos de pontos com a mesma classe, e na inserção, o novo vetor ficará mais próximo de um número k de pontos, pertencentes a grupos preexistentes, ou seja, terá menor distância. Assim, será classificado conforme o grupo predominante desses k vizinhos mais próximos, sendo k um número ímpar definido manualmente.

A aplicação do algoritmo é feita para que, dados o somatório de atributos dos 5 campeões de um time, o algoritmo classifique o time em uma determinada partida do PlayIn. As informações utilizadas para isso, primeiramente foram as notas individuais dos campeões (IV.A.2) e os times (IV.A.1). Foram criados também um vetor e uma lista, uma com os valores dos somatórios e a outra com as classes, respectivamente:

atributos

```
array([[ 8., 19.,  8.,  6., 19.,  0.],
       [ 6., 16., 17.,  6., 14.,  0.],
       [11., 21., 12.,  5., 15.,  0.],
       [ 9., 16.,  8., 10., 12.,  0.],
       [11.,  9.,  7.,  6., 21.,  0.],
       [14., 17.,  2.,  9., 23.,  0.],
       [13., 16.,  5.,  0., 19.,  0.],
       [ 8., 13., 10.,  8., 19.,  0.],
       [12., 20., 13.,  4., 17.,  0.],
       [11., 12.,  6.,  8., 19.,  0.],
       [16., 17.,  6.,  5., 18.,  0.],
       [13., 17.,  8.,  8., 14.,  0.],
       [11., 21.,  8.,  9., 15.,  0.],
       [22., 17.,  5.,  4., 16.,  0.],
       [14., 11.,  7.,  9., 17.,  0.],
       [12., 20., 10., 10., 17.,  0.],
       [12., 19.,  5.,  0., 18.,  0.]])
```

**Figura 14** – Início do vetor do somatório de atributos.

## Classe

```
' Dive',
' Wombo Combo',
' Wombo Combo',
' Wombo Combo',
' Dive',
' Peel',
' Dive',
' Wombo Combo',
' Split',
' Peel',
' Wombo Combo',
```

**Figura 15** – Início da lista das classes

A distância utilizada é a euclidiana, e o k selecionado é 5, para ser igual ao número de classes. Os resultados serão apresentados em (IV.D)

### D. Análise da árvore de possibilidades de confrontos

No início de um campeonato, dado os grupos montados, é feita uma rápida análise tendo como parâmetros o elenco, o momento do time, os prováveis adversários até o título, entre outros fatores, a fim de buscar as equipes mais forte de cada grupo mas sempre levando em consideração as chances que a equipe do torcedor teria de ser campeã. Com base nisso, outro programa foi feito, o qual busca simular uma fase de grupos, tendo em vista valores preestabelecidos ou notas para cada time, a sua chance de classificação e os adversários que ela pode pegar no mata-mata.

Aplicando isso a segunda fase do LoL Worlds, teríamos os 4 grupos com 4 times, que precisariam jogar essa fase para definir os primeiros e segundos colocados. Logo, como os times jogam duas vezes, cada time teria seis jogos para cumprir nessa fase. Para simular isso, foi desenvolvida uma fórmula que calcula a chance de vitória em porcentagem, visto que cada time possui uma nota atribuída a si:

- Nota do time 1: x
- Nota do time 2: y

Chance de vitória time 1 =  $50 + ((x-y)*10)$ .

Chance de vitória time 2 =  $50 + ((y-x)*10)$ .

Tendo as notas definidas com base no histórico do time antes do LoL Worlds, teríamos um conjunto de dados (IV.A.3) para a definição dessa fórmula, utilizando um exemplo dele, temos para o primeiro confronto da fase de grupos, do grupo A, G2 x Sunning:

Nota G2: 4.6, 56% de chance de vitória

Nota Sunning: 4.0 44% de chance de vitória

São sorteados 10 números aleatórios, os números de 0 a 56 favorecem a G2, de 56 a 100, favorecem a Sunning.

Exemplo de sorteio: 38 40 10 89 64 38 37 86 69 49

Valores que favorecem a G2: 6

Valores que favorecem a Sunning: 4

G2 ganhou, é somado mais uma vitória para esse time.

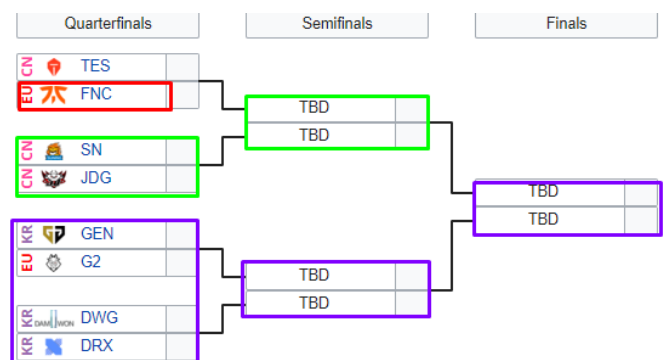
Assim é feita a simulação de todos os jogos da fase de grupos, para os 16 times, contabilizando as suas vitórias e gerando dois times classificados, um em primeiro e outro em segundo lugar. Assim, teremos uma árvore para percorrer e analisar os possíveis confrontos de cada time até o título.

Na última etapa, o mata-mata, começando pelas quartas de final, existem duas regras a serem seguidas:

- Dois times de um mesmo grupo não podem jogar novamente entre si nas quartas de final.
- Os primeiros colocados jogam contra os segundos colocados.

Duas heurísticas para percorrer a árvore são utilizadas, a primeira é a busca gulosa, que expande sempre os nós de menor custo, ou seja, de menor nota, buscando sempre os times mais fáceis. A outra, é a busca de aprofundamento iterativo, que expande sempre todos os nós e assim, podemos ter todas as possibilidades. Um ponto a se destacar é a forma que tudo foi implementado: a árvore como é conhecida na computação não foi realmente feita, com os conceitos de nós, filhos e relacionados, mas tudo foi feito com um vetor desses times, sendo que esses métodos de busca são mais uma padronização e inspiração para a manipulação desses vetores, que através de laços, geram o mesmo tipo de resultado da árvore.

Como as quartas de final são definidas por sorteio, teremos que levar mais possibilidades em conta, pois a parte da árvore em que o time está gera adversários específicos.



**Figura 16** – Projeção para a Fnatic, destacada em vermelha, chegar ao título. Note que, ao ganhar da TES, ela enfrentará a SN ou a JDG, e se passar na final, ela poderá enfrentar qualquer um dos times das quartas de final roxa. Entretanto, caso ela estivesse na parte roxa do chaveamento, a sequência de adversários que ela enfrentaria seria outra, totalmente diferente.

No exemplo da figura 15, a busca gulosa agiria da seguinte forma, analisando os adversários da Fnatic:

- Essa seria uma das simulações entre as 3 possíveis das quartas de final, pois como a Fnatic passou em segundo do grupo C, ela poderia enfrentar outros 2 primeiros colocados, a Sunning e a DAMWON, menos a Gen.G, que era do seu grupo na fase anterior
- Dado que a Fnatic ganhe da TES, a princípio não se saberia em qual lado do chaveamento das semifinais ela está, ou seja, na parte verde, podendo enfrentar a Sunning ou JDG, ou na roxa, podendo enfrentar Gen.G ou G2, caso seja a chave roxa inferior, ou a DAMWON ou DragonX, caso seja a chave roxa superior. Como estamos simulando o caminho mais fácil, procuraremos o adversário de menor nota dentre esses 6.
- Na final, a Fnatic já teria eliminado 2 times, sobrando apenas 5 para esse possível confronto. Também é buscado o adversário com menor nota dentre esses 5.

Para a busca de aprofundamento iterativo, as notas foram modificadas, tendo em mente que é extremamente comum um time melhorar sua performance por conta do sucesso em fases anteriores. Assim, para as quartas de final, as notas foram atualizadas e somadas com o número de vitórias.

Como só temos 3 jogos de cada time até a final, o nível de profundidade da busca seria 2 (tendo em vista que as quartas de final sejam o nível 0). As simulações são feitas independente das notas dos times, simulando todos os confrontos, ou seja, expandindo todos os nós e somando as suas notas para dizer o custo desse caminho. Essa parte do código tem custo computacional relativamente alto, devidos aos laços dentro de laços, gerando um maior custo de tempo, mas não de memória. A complexidade desse código, com o auxílio de uma rápida análise assintótica, seria da ordem de  $O(n^x)$ .

IV. ANALISE EXPERIMENTAL

A. Conjunto de dados

1) Times

	Time	Top	Jungle	Mid	ADC	Sup	Vitoria	Classe
0	INTZ	Urgot	Gragas	Tristana	Ashe	Alistar	0	Dive
1	MAD Lions	Renekton	Evelynn	Lucian	Jhin	Pantheon	1	Split
2	Legacy	Camille	Nidalee	Sett	Tristana	Nautlius	1	Split
3	INTZ	Vladimir	Vollbear	Zoe	Ezreal	Bardo	0	Poke
4	LGD	Vollbear	Lilla	Lucian	Ashe	Braum	0	Wombo Combo
...	...	...	...	...	...	...	...	...
72	LGD	Renekton	Graves	Twisted Fate	Miss Fortune	Leona	1	Split
73	Legacy	Poppy	Lilla	Twisted Fate	Draven	Braum	0	Wombo Combo
74	LGD	Camille	Nidalee	Kassadin	Ezreal	Leona	1	Split
75	Legacy	Camille	Lilla	Gallo	Twitch	Taric	0	Peel
76	LGD	Vollbear	Graves	Twisted Fate	Ezreal	Alistar	1	Wombo Combo

Figura 17 – Jogos dos times participantes do PlayIn, com seus campeões escolhidos e a classificação gerada manualmente.

2) Campeões

	Campeao	Peel	Dive	Split	Poke	Wombo	Combo
0	Nidalee	3	3	0	5		0
1	Lilla	0	2	0	3		5
2	Orianna	3	3	0	4		5
3	Ornn	4	3	1	0		4
4	Graves	1	3	1	1		5
...	...	...	...	...	...		...
64	Kog'Maw	5	0	0	5		2
65	Skarner	1	5	0	0		0
66	Urgot	0	5	5	0		2
67	Vladimir	0	5	4	0		5
68	Xayah	4	1	0	0		5

Figura 18 – Campeões utilizados no PlayIn e seus atributos

3) Notas fase de grupos

	Time	Nota	Grupo
0	G2	4.6	A
1	Sunning	4	A
2	Machi	3.1	A
3	Liquid	3.5	A
4	DAMWON	5	B
5	JDG	4.95	B
6	Rogue	3.6	B
7	PSG	3.05	B
8	Fnatic	4.4	C
9	Gen.G	4.2	C
10	LGD	3.75	C
11	TSM	3.15	C
12	TES	4.8	D
13	DragonX	4.3	D
14	FlyQuest	3.25	D
15	UOL	3.15	D

Figura 19 – Times, avaliações e grupos.

### B. Configuração do algoritmo e ambiente computacional

Hardware: Intel Core i3-6100, com 8GB de memória RAM e 500GB de HD.

Software: Sistema operacional: Windows 10 Education.

Compiladores: kNN – Google Colab, serviço que executa python em nuvem, na versão 3.7.

Árvores – CodeBlocks v17.12 – Compilador para códigos escritos na linguagem C, como nesse caso, ou C++.

### C. Critérios de análise:

kNN – Acurácia, relatada em resultados (IV.D).

Árvores – Inexistência de erros na expansão dos confrontos

### D. Resultados e discussão

Acurácia do kNN nos primeiros testes – 60%, que decaiu para 50% mudando os arquivos de teste e treino

```
accuracy_score(Classe[50:70],neigh.predict(atributos[50:70]))  
0.6
```

**Figura 20** – Acurácia nos primeiros testes

```
accuracy_score(Classe[50:70],neigh.predict(atributos[50:70]))  
0.5
```

**Figura 21** – Acurácia após variar k e a fórmula de distância

Árvores: A análise manual foi feita sem que erros na expansão das árvores fossem encontrados.

```
G2 x Machi  
chances: G2 65.00% Machi 35.00%  
Simulacao: 88 18 65 84 40 62 10 19 67 56  
G2 7 x Machi 3  
G2 ganhou
```

**Figura 22** – Simulação de um jogo dos jogos da fase de grupos.

```
vitorias:  
G2 6  
Sunning 4  
Machi 1  
Liquid 1
```

**Figura 23** – Exemplo de grupo após as simulações de todos os jogos.

```
Caminhos ate a final:  
Time : G2  
Busca Gulosa:  
Primeiros Adjacentes equivalentes as quartas de final  
Possivel Adversario: JDG (4.9)  
Possivel Adversario: Gen.G (4.2)  
Possivel Adversario: DragonX (4.3)  
Adversario mais facil das quartas: Gen.G (4.2) [ELIMINADO]  
  
Possivel adversario na semi: Sunning (4.0)  
Possivel adversario na semi: DAMWON (5.0)  
Possivel adversario na semi: JDG (4.9)  
Possivel adversario na semi: Fnatic (4.4)  
Possivel adversario na semi: TES (4.8)  
Possivel adversario na semi: DragonX (4.3)  
Adversario mais facil da semi: Sunning (4.0) [ELIMINADO]  
  
Possivel adversario na final: DAMWON (5.0)  
Possivel adversario na final: JDG (4.9)  
Possivel adversario na final: Fnatic (4.4)  
Possivel adversario na final: TES (4.8)  
Possivel adversario na final: DragonX (4.3)
```

**Figura 24** – Simulação conforme a Busca Gulosa

```
Time G2 Posicao 1  
  
Adversario das quartas: JDG (9.9) [ELIMINADO]  
Dificuldade do caminho: 9.9  
Adversario da semi: Sunning (8.0) [ELIMINADO]  
Custo desse caminho (quartas G2 x JDG e semi G2 x Sunning): 17.9  
Possivel adversario na final: DAMWON (custo total: 27.9)  
Possivel adversario na final: JDG (custo total: 27.9)  
Possivel adversario na final: Fnatic (custo total: 26.3)  
Possivel adversario na final: Gen.G (custo total: 26.1)  
Possivel adversario na final: TES (custo total: 27.7)
```

**Figura 25** – Exemplo de expansão de uma das combinações de quartas de final e semifinal possíveis, analisando apenas uma dessas possibilidades.

Como os conjuntos de dados foram montados de maneira muito subjetiva, com base na análise humana que tem caráter pessoal, podemos afirmar que o resultado da acurácia do kNN é no mínimo surpreendente, pois mesmos com dados totalmente irregulares, sujeitos a variações e outros pontos que o algoritmo dificilmente conseguiria prever, ele teve uma boa taxa de acerto. No caso das árvores, era uma questão mais de análise combinacional para prever todos os caminhos, mas enquadrados nessa heurística, também foi possível fazer todas essas simulações.

## V . CONCLUSÕES

Mesmo com os dados irregulares, a execução dos algoritmos foi no mínimo interessante. Entretanto, foi difícil aplicar esses dados em outros modelos além do kNN, no primeiro caso, como era desejado. O resultado não foi extremamente comprometido com isso, porém, a comparação dos métodos não foi possível. No caso das árvores, os métodos mais intuitivos foram aplicados e isso facilitaria a compreensão de indivíduos leigos quanto a esses tipos de algoritmo.



## REFERÊNCIAS

Site com os informações e do Worlds 2020, além das figuras 6, 7, 8, 9, 10, 11, 12 e 15, disponível em <[https://lol.gamepedia.com/2020\\_Season\\_World\\_Championship](https://lol.gamepedia.com/2020_Season_World_Championship)>, acessado em 11/10/2020.

Worlds 2020 Update: Format Changes, disponível em <<https://lolesports.com/article/worlds-2020-update-format-changes/blt844a57584da327bf>>, acessado em 11/10/2020

Figura 2, disponível em <<https://www.easports.com/fifa/fifa-18-cristiano-ronaldo>>, Acessado em 11/10/2020.

Figura 3, disponível em <<https://runes.lol/ranked/platinum/plus/champion/pick/Twitch/ADC/>>, acessado em 11/10/2020.

Canal oficial de transmissão do LoL Worlds no site de streaming Twitch, gerenciado pela empresa desenvolvedora do jogo (Riot Games), disponível em <<https://www.twitch.tv/riotgamesbrazil>>, Acessado entre os dias 25/09/2020 e 11/10/2020.

Canal de streaming de ex-jogadores e jogadores profissionais de League of Legends, disponível em <<https://www.twitch.tv/baiano>>, acessado entre os dias 25/09/2020 e 11/10/2020.

Figura 1, disponível em <<https://www.gameblast.com.br/2019/12/League-Of-Legends-guia-girias.html>>, acesso em 10/10/2020.