

League of Legends

Modelo de recomendação de escolha

x x x x x
x x x x x
x x x x x
x x x x x



Pautas

01

Introdução

O que é League of Legends?

02

Objetivos

O que queremos prever?

03

Procedimentos

Qual foi o raciocínio seguido

04

Resultados Próximos passos

O modelo final foi bom?
O que pode ser melhorado?

x x x x x
x x x x x
x x x x x
x x x x x
x x x x x

01

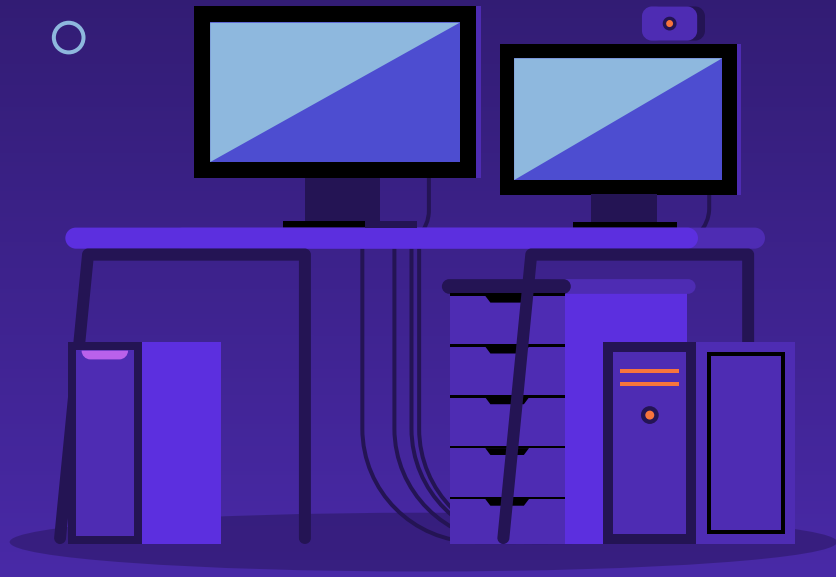
Introdução

× × × × ×
× × × × ×
× × × × ×
× × × × ×
× × × × ×



O que é League of legends?





02

Objetivos

Como aumentar a chance de vitória?



As escolhas acontecem por turnos e sempre haverá uma pessoa que escolherá por último.

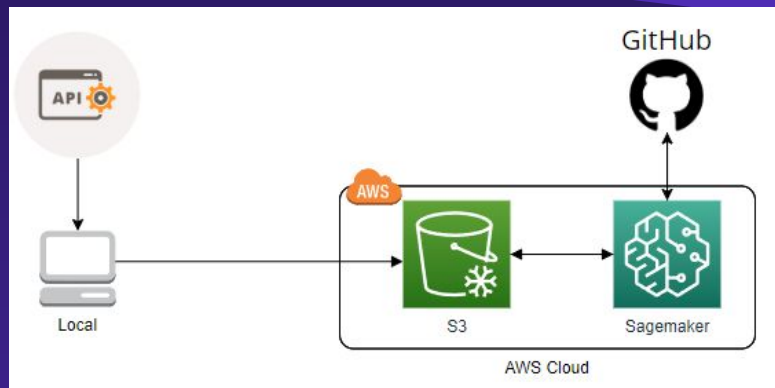
Nesse sentido, qual seria a melhor escolha para ela pensando em otimizar a chance do time ganhar?

The background is a solid dark purple. It features several abstract elements: a large, stylized, light purple shape on the left; a jagged, light purple shape on the top right; a dark purple jagged shape on the bottom left; a series of four small orange triangles pointing up and to the right on the middle left; a horizontal line with small circles at its ends and at intervals; a grid of small white 'x' marks in the bottom center; and a tall, dark purple rectangular object with a white top and a white oval on its side on the right. Scattered throughout are several small white circles.

03

Procedimientos

Etapas



ETL

A coleta de dados se deu através da API disponibilizada em <https://developer.riotgames.com/>



Machine Learning

A exploração de dados e testes do modelo ocorreram via AWS SageMaker



Recomendação

A recomendação acontece quando otimizamos a chance de vitória do modelo

ETL

Já no ETL temos uma gama enorme de variáveis possíveis a serem coletadas, mas a partir de um estudo do jogo foram selecionadas as consideradas mais importantes para a vitória descritas a seguir, além é claro de qual time foi vencedor.

Posições

- Top
- Jungle
- Mid
- ADC
- Sup



Times

- Azul
- Vermelho



Features

- Campeão
- Kills
- Mortes
- Assistências
- Dano
- Ouro
- Controle de visão



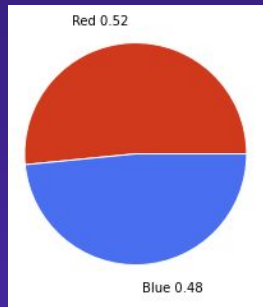
70 Colunas

Análise inicial

Para entender a natureza dos dados, precisamos entender como estão distribuídos nossa target (vitória ou derrota do time azul), e quais são as features que tem maior correlação com ela, além de definir o tratamento de cada um dos dados

Features mais relevantes

Pipeline



48% de vitória do time azul

```
[ 'assist_Sup_blue_side',  
'gold_per_minute_Jungle_blue_side' ,  
'gold_per_minute_Adc_blue_side' ,  
'kill_Jungle_blue_side' ,  
'gold_per_minute_Mid_blue_side' ,  
'gold_per_minute_Top_blue_side' ,  
'kill_Adc_blue_side' ,  
'kill_Mid_blue_side' ,  
'assist_Jungle_blue_side' ,  
'assist_Adc_blue_side' ,  
'assist_Sup_red_side' ,  
'gold_per_minute_Adc_red_side' ,  
'gold_per_minute_Jungle_red_side' ,  
'kill_Adc_red_side' ,  
'kill_Jungle_red_side' ,  
'gold_per_minute_Mid_red_side' ,  
'kill_Mid_red_side' ,  
'gold_per_minute_Top_red_side' ,  
'assist_Jungle_red_side' ,  
'kill_Top_red_side' ]
```

```
num_pipeline = Pipeline([  
    ('imputer', SimpleImputer(strategy='median')),  
    ('robust_scaler', RobustScaler())  
)  
  
cat_pipeline = Pipeline([  
    ('imputer', SimpleImputer(strategy='most_frequent')), # as  
    ('one-hot-encoding', OneHotEncoder(handle_unknown='ignore'))  
)  
  
# (name, transformer, columns)  
preprocessed_pipeline = ColumnTransformer([  
    ('numerical', num_pipeline, num_attributes),  
    ('categorical', cat_pipeline, categorical)  
)
```

Teste inicial de modelos

Com as features selecionadas começamos a testar os modelos de classificação com cross_validation medindo f1 score

- KNN
- SVC
- DecisionTreeClassifier
- RandomForestClassifier
- AdaBoostClassifier
- GradientBoostingClassifier
- LogisticRegression

Scores: [0.96992481 0.97110904 0.96060038 0.9637883 0.97185741 0.96545285
0.96030246 0.95732839 0.95586854 0.9544186 0.95497186 0.94924812
0.95586854 0.96240602 0.96135721 0.95530726 0.95813953 0.96091516
0.96435272 0.95692884]

Mean: 0.9605073020579141
Standard deviation: 0.005820406744395805

Scores: [0.95175644 0.94845361 0.94387278 0.94642857 0.93408135 0.93752936
0.9462771 0.94211765 0.9465721 0.94657919]

Mean: 0.9443668143341302
Standard deviation: 0.004968006740419588

Scores: [0.96567936 0.96123307 0.96946923 0.96411709 0.95765472 0.95028143
0.96244131 0.96219282 0.95750708 0.96275605]

Mean: 0.9613332169741347
Standard deviation: 0.004971395114296107

Otimização de modelos

Com os modelos selecionados é a hora de otimizar os parâmetros utilizando **grid_search**, além disso a partir das probabilidades geradas foram testados alguns thresholds que melhorassem os resultados dos modelos

- SVC, THR = 0.44
- GradientBoostingClassifier THR = 0.42
- LogisticRegression THR = 0.44

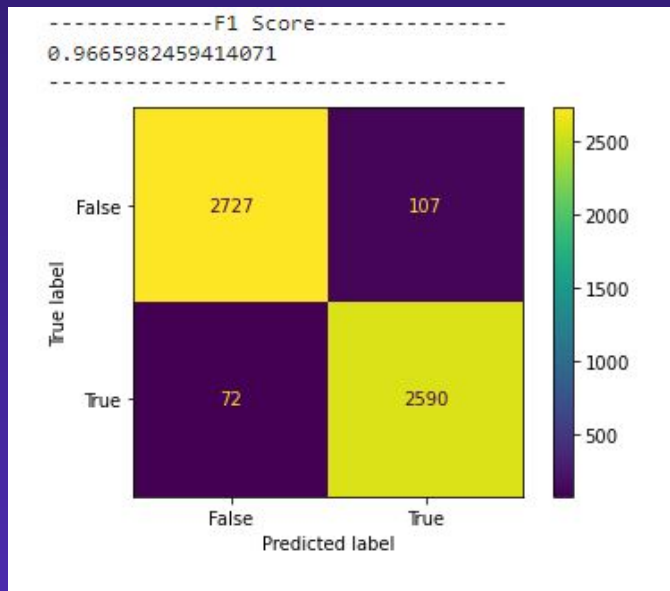
```
LogisticRegression()
```

```
SVC(C= 2, decision_function_shape = 'ovo', degree = 1, probability=True)
```

```
GradientBoostingClassifier(criterion= 'friedman_mse', learning_rate = 0.5, loss= 'exponential', max_depth= 3, n_estimators = 300)
```

Modelo Final

O modelo final dependia da previsão dos 3 modelos treinados onde pela decisão da maioria era decidido vitória ou derrota do time azul



Modelo de recomendação

Definição das escolhas e posição a ser recomendada

```
escolhas = {  
    'champion_Top_red_side' : "Tryndamere",  
    'champion_Top_blue_side' : "Illaoi",  
    'champion_Jungle_red_side' : "Amumu",  
    'champion_Jungle_blue_side' : "Rammus",  
    'champion_Mid_red_side' : None,  
    'champion_Mid_blue_side' : "Lux",  
    'champion_Adc_red_side' : "Xayah",  
    'champion_Adc_blue_side' : "Vayne",  
    'champion_Sup_red_side' : "Singed",  
    'champion_Sup_blue_side' : "Renata"  
}
```



Preenchimento das features a partir das médias da base de dados para cada campeão

Teste com todos os campeões e output dos 5 melhores desempenhos

campeao	
74	Urgot
39	Jayce
115	KogMaw
47	Nilah
0	Rengar



04

Resultados e
próximos passos

x x x x x
x x x x x
x x x x x
x x x x x

Considerações finais e Próximos passos



Resultados:

- Modelo preditivo de jogos passados funcional (mas sem muita utilidade)
- Modelo de recomendação falho (baseado em conhecimentos do jogo)

Próximos passos:

- Procurar melhorar modelo preditivo
 - Adicionar modelos à etapa de voto majoritário
 - Buscar melhorar os hiperparâmetros com mais testes
 - Olhar outros thresholds
 - Testar mais dados disponíveis na API
- Estudar modelos de recomendação
 - Qual a melhor estratégia para recomendar
 - Quais os melhores dados a serem utilizados?