

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data

FABRICIO GOMES SOARES DE CARVALHO

**MODELOS DE MACHINE LEARNING PARA PREVISÃO DO RESULTADO DE
PARTIDAS DA COPA DO MUNDO 2022**

Belo Horizonte
2022

FABRICIO GOMES SOARES DE CARVALHO

**MODELOS DE MACHINE LEARNING PARA PREVISÃO DO RESULTADO DE
PARTIDAS DA COPA DO MUNDO 2022**

Trabalho de Conclusão de Curso apresentado
ao Curso de Especialização em Ciência de
Dados e Big Data como requisito parcial à
obtenção do título de especialista.

Belo Horizonte

2022

SUMÁRIO

| | |
|--|-----------|
| 1. Introdução | 5 |
| 1.1. Contextualização | 5 |
| 1.2. O problema proposto | 5 |
| 2. Coleta de Dados | 6 |
| 2.1 Tabela Histórica de Partidas | 7 |
| 2.2 Tabelas Histórica do Ranking de Seleções da Fifa | 8 |
| 3. Processamento/Tratamento de Dados | 12 |
| 4. Análise e Exploração dos Dados | 23 |
| 5. Criação dos Modelos de Machine Learning | 30 |
| 5.1 Escolha dos Modelos | 31 |
| 5.1.1 Logistic Regression | 31 |
| 5.1.2 Ridge Classifier | 33 |
| 5.1.3 XGB Classifier | 34 |
| 5.1.4 KNeighbors Classifier | 35 |
| 5.1.5 Gradient Boosting Classifier | 36 |
| 5.1.6 Decision Tree Classifier | 37 |
| 5.2 Decisão | 38 |
| 5.2.1 O Problema do Empate | 38 |
| 5.2.2 O problema da Base de Dados | 39 |
| 6. Apresentação dos Resultados | 40 |
| 6.1 RidgeClassifier | 40 |
| 6.1.1 - Fase de Grupos | 41 |
| 6.1.2 - Oitavas de Final | 43 |
| 6.1.3 - Quartas de Final | 43 |
| 6.1.4 - Semifinal | 44 |
| 6.1.5 - Final | 44 |
| 6.2 LogisticRegression | 44 |

| | |
|--|-----------|
| 6.2.1 Fase de Grupos | 44 |
| 6.2.2 Oitavas de Final | 47 |
| 6.2.3 - Quartas de Final | 47 |
| 6.2.4 - Semifinal | 48 |
| 6.2.5 - Final | 48 |
| 7. Conclusões | 49 |
| 8. Atualizações Pós Eliminatórias | 49 |
| 8.1 Escócia / Ucrânia / País de Gales | 49 |
| 8.2 Emirados Árabes / Austrália / Peru | 49 |
| 8.3 Costa Rica / Nova Zelândia | 50 |
| 9. Conclusão | 50 |
| 10. Links | 51 |
| Downloader Ranking Fifa | 51 |
| Kaggle do Dataset Partidas | 51 |
| Github do Notebook | 51 |
| Youtube da apresentação | 51 |

1. Introdução

1.1. Contextualização

Na era digital, muitos recorrem a plataformas online para apostas, e assim tentar gerar algum lucro através de operações simples ou complexas de apostas online. Para tanto existem plataformas especializadas no ramo de Apostas Online. De jogos eletrônicos, a corrida de cavalos, as apostas online têm tido uma curva ascendente tanto na quantidade de apostas, quanto no volume de dinheiro movimentado.

Em 2022, ano de copa, as previsões para o cenário não são diferentes, pois é um período quase festivo para principalmente brasileiros, que são conhecidos pela sua paixão pelo tema “Futebol”, o que gera um movimento ainda maior de apostas nos canais digitais através de sites confiáveis e licenciados atuando no mercado brasileiro.

O Estudo apresentado busca realizar testes em modelos de Machine Learning com o objetivo de identificar qual modelo traz o melhor custo benefício, mostra uma maior diversidade de resultados, se adapta mais ao conjunto de dados proposto e se mostra mais coerente com a realidade esperada para o problema proposto.

Para apresentação dos dados, foi utilizado a plataforma Google Colab, mas para distribuição será via projeto no github, com os links sendo encontrados na seção 7.

1.2. O problema proposto

O problema proposto visa buscar o modelo de Machine Learning mais adequado para prever resultados de partidas de futebol entre Seleções, utilizando como base uma tabela histórica de jogos entre todas as seleções do mundo alimentada com registros que contém o resultado de partidas oficializadas pela Fifa,

e para agregação dos registros, serão utilizados os próprios dados de pontuação da Fifa no ranking de Seleções disponíveis em seu site na internet.

A proposta é que o resultado dos algoritmos ajude os analistas a tomar decisões sobre odds disponibilizadas em seus sites licenciados, ou até, futuramente, tornar o processo automático, analisando em tempo real outros inputs que podem surgir nos dias que antecedem a competição da Copa do Mundo 2022.

(Why?) Baseado no auxílio do modelo, a casa de apostas pode criar, validar, e automatizar o processo de geração de odds, para partidas entre seleções no período da Copa do Mundo 2022.

(Who?) Casas de apostas, analistas futebolísticos e apostadores que desejem basear suas apostas em resultados históricos.

(What?) Testar modelos capazes de trazer resultados significativos e coerentes com a realidade.

(Where?) Todos os dados foram coletados da internet através da plataforma Kaggle e uma base de dados própria adquirida através de scrapping no site da Fifa para coletar dados do ranking de seleções masculinas reconhecidas pela instituição.

(When?) Para análise serão utilizadas bases históricas que contém tanto os resultados de partidas de futebol entre seleções, quanto a movimentação de pontos utilizado pela Fifa para ranquear as seleções.

2. Coleta de Dados

Os dados coletados foram adquiridos de duas fontes, a primeira delas é uma tabela histórica com o resultado de partidas entre seleções que compreende o período entre 30/11/1872 até 30/03/2022, e segunda é o ranking oficial da Fifa, que para o objetivo do estudo foi baixado em sua totalidade, desde o ano de 31/12/1992 até o presente momento, tendo a última atualização sido realizada no dia 31/03/2022, pela própria instituição.

2.1 Tabela Histórica de Partidas

A tabela foi adquirida através da plataforma Kaggle, plataforma que reúne diversas bases de dados tratadas e preparadas para a utilização disponibilizada em arquivo no formato “.csv”. Por se tratar de uma base já preparada para a utilização, os registros já estão com os devidos tratamentos em relação a campos nulos, strings mal formatadas ou de alguma forma que necessite de um tratamento específico em cima de cada campo.

A base no entanto utiliza nomenclatura para o nome de algumas seleções de forma diferente à base de dados de rankings da Fifa, sendo necessário, a esse nível, intervenção para equiparação dos nomes e facilitar a agregação das bases, sendo este assunto a ser discutido melhor na Seção 3 deste documento, que trata do processamento e tratamento aplicados aos dados.

O arquivo é único e, como citado anteriormente, está formatado como arquivo “.csv” utilizando o caractere “,” (vírgula) como separador dos campos. A tabela abaixo mostra a uma listagem dos campos disponíveis, a coluna descrição apresenta um entendimento semântico do campo, bem como o tipo do campo no arquivo.

| Nome do campo | Descrição | Tipo |
|---------------|--|---------|
| Date | Data da partida no formato ISO “ano-mês-dia” | String |
| Home_team | Nome da seleção mandante | String |
| Away_team | Nome da seleção visitante | String |
| Home_score | Quantidade de gols marcados pelo mandante | Integer |
| Away_score | Quantidade de gols marcados pelo visitante | Integer |
| Tournament | Torneio pelo qual a partida foi realizada | String |
| City | Cidade na qual partida foi realizada | String |
| Country | País da cidade na qual a partida foi realizada | String |
| Neutral | Se a partida foi disputada em cidade Neutra | String |

O link para download do arquivo encontra-se na seção 7 deste documento.

Segue um exemplo dos dados encontrados no dataset:

| | date | home_team | away_team | home_score | away_score | tournament | city | country | neutral |
|-------|------------|-----------------|---------------|------------|------------|------------------------------|-------------|------------|---------|
| 0 | 1872-11-30 | Scotland | England | 0 | 0 | Friendly | Glasgow | Scotland | False |
| 1 | 1873-03-08 | England | Scotland | 4 | 2 | Friendly | London | England | False |
| 2 | 1874-03-07 | Scotland | England | 2 | 1 | Friendly | Glasgow | Scotland | False |
| 3 | 1875-03-06 | England | Scotland | 2 | 2 | Friendly | London | England | False |
| 4 | 1876-03-04 | Scotland | England | 3 | 0 | Friendly | Glasgow | Scotland | False |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 43416 | 2022-03-30 | Mexico | El Salvador | 2 | 0 | FIFA World Cup qualification | Mexico City | Mexico | False |
| 43417 | 2022-03-30 | Costa Rica | United States | 2 | 0 | FIFA World Cup qualification | San José | Costa Rica | False |
| 43418 | 2022-03-30 | Panama | Canada | 1 | 0 | FIFA World Cup qualification | Panama City | Panama | False |
| 43419 | 2022-03-30 | Jamaica | Honduras | 2 | 1 | FIFA World Cup qualification | Kingston | Jamaica | False |
| 43420 | 2022-03-30 | Solomon Islands | New Zealand | 0 | 5 | FIFA World Cup qualification | Doha | Qatar | True |

43421 rows x 9 columns

2.2 Tabelas Histórica do Ranking de Seleções da Fifa

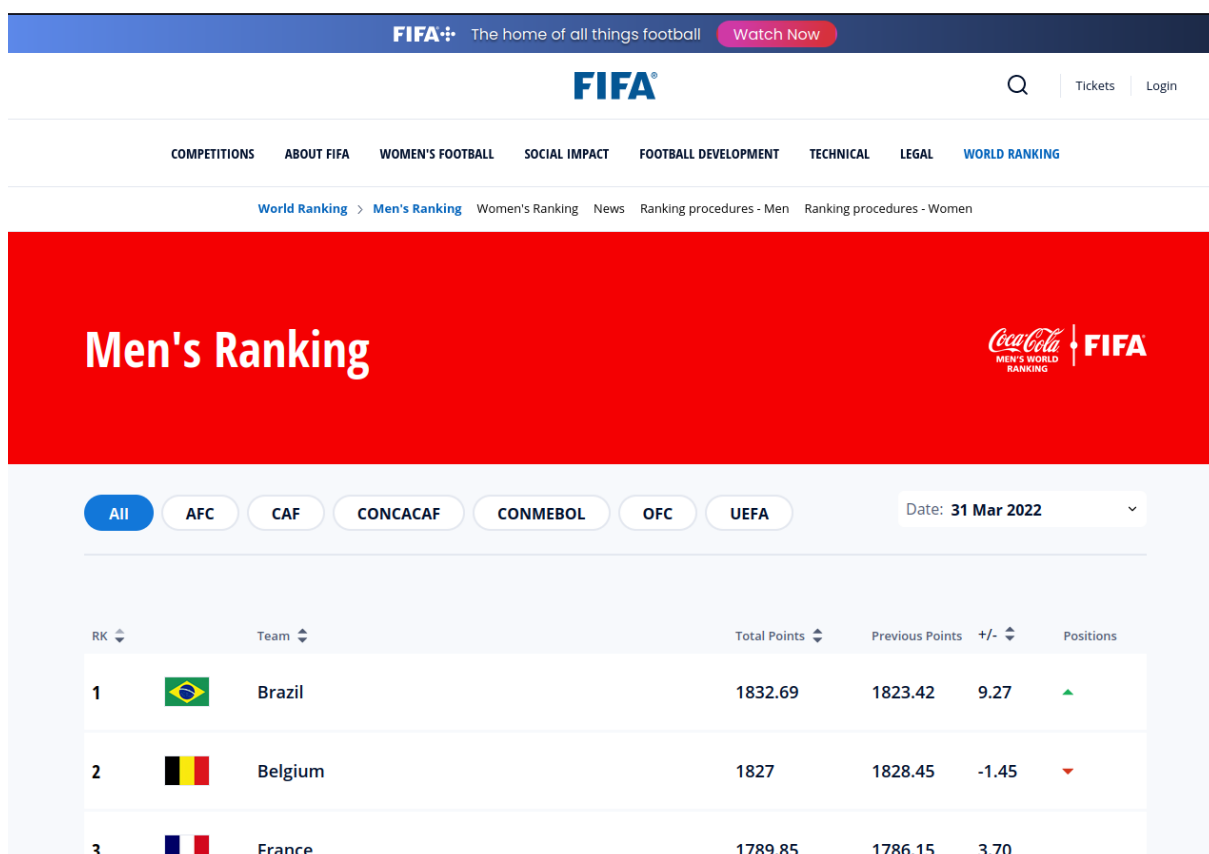
As tabelas históricas do ranking oficial da Fifa foram adquiridas aplicando-se uma técnica de scrapping no site da própria instituição. Nenhuma base de dados foi encontrada na internet com os dados que eram necessários, nem mesmo em bases de dados como a do Kaggle. A aplicação da técnica utilizada será melhor detalhada na Seção 3 deste documento, percorrendo as soluções adotadas e as motivações.

Após aplicação do scrapping, foi gerado 1 arquivo no formato “.csv”, com o caractere “,” (vírgula) como separador dos campos e o caractere “.” (ponto) como separador decimal. A tabela abaixo traz um detalhamento dos campos.

| Nome do campo | Descrição | Tipo |
|---------------|---|---------|
| Date | Data da liberação do ranking no formato ISO “ano-mês-dia” | String |
| Pos | Posição ordinal da seleção no ranking | Integer |
| Team | Nome da seleção no padrão Fifa | String |
| Iso-alfa-3 | Sigla da seleção contendo 3 caracteres no padrão iso-alfa-3 | String |

| | | |
|-----------------|---|--------|
| Total_points | Pontuação da seleção no ranking atual | Float |
| Previous_points | Pontuação da seleção no ranking anterior | Float |
| Diff | Diferença de pontos entre total e previous points | Float |
| Var | Campo informativo se a seleção ganhou ou perdeu posições desde o ranking anterior | String |

Para obtenção dos rankings da Fifa de forma histórica, foi necessário o desenvolvimento de uma aplicação python para realizar o scrapping dos dados da tabela no site: <https://www.fifa.com/fifa-world-ranking/men>.






The screenshot shows the FIFA Men's World Ranking page as of 31 Mar 2022. The page features a red header with the 'Men's Ranking' title and the Coca-Cola FIFA logo. Below the header, there are navigation tabs for various football confederations: All, AFC, CAF, CONCACAF, CONMEBOL, OFC, and UEFA. The 'All' tab is selected. The main content area displays a table of the top 3 ranked teams.

| RK | Team | Total Points | Previous Points | +/- | Positions |
|----|---------|--------------|-----------------|-------|-----------|
| 1 | Brazil | 1832.69 | 1823.42 | 9.27 | ▲ |
| 2 | Belgium | 1827 | 1828.45 | -1.45 | ▼ |
| 3 | France | 1789.85 | 1786.15 | 3.70 | |

Durante o desenvolvimento algumas soluções precisaram ser adotadas. Utilizando a linguagem de programação Python foram utilizadas em conjunto as bibliotecas Selenium e BeautifulSoup.

O mecanismo de exibição dos rankings utilizados pela instituição envolve o uso de “Caixa de Seleção ou Dropdown” e paginação a cada 50 registros, que

precisam ser acionados através de JavaScript, sendo esse o principal problema a se resolver.

| | | | | | | |
|----|---|--------------|---------|---------|-------|---|
| 48 |  | Romania | 1446.54 | 1453.18 | -6.64 | ▼ |
| 49 |  | Saudi Arabia | 1444.69 | 1433.95 | 10.74 | ▲ |
| 50 |  | Paraguay | 1443.3 | 1440.53 | 2.77 | |

< 1 2 3 ... 5 >

Para contornar esse problema foi utilizado a biblioteca Selenium, muito comum para desenvolvimento de testes automatizados em aplicações web, e também em aplicações de webscrap/webcrawler. O Selenium utiliza o driver de navegadores do mercado como Google Chrome, Safari e Firefox (que foi o escolhido nesta solução), para o possamos interagir com o JavaScript de determinada página.

Com o Selenium foi possível manipular o JavaScript para interagir com a paginação e com o dropdown. Para cada página o seu conteúdo HTML foi completamente copiado e salvo como arquivo “.html”, aproximadamente 5 arquivos para cada “dia” do ranking.

Com os arquivos HTML gerados, se deu a utilização do BeautifulSoup para buscar os dados da tabela, convertendo cada linha da tabela em um registro “.csv”, e agregando cada arquivo em um único, agrupando-os por dia. Dentre os tratamentos necessários algumas medidas foram tomadas:

1. O segundo campo da tabela do site, o Brasão, foi convertido para o campo “iso-alfa-3”, utilizando da string identificadora do Brasão a nível de css, conseguindo capturar assim a informação e convertendo-a para UPPERCASE, mas também o próprio brasão foi baixado para posterior utilização;
2. O último campo da tabela do site, o var, foi identificado a nível de CSS, qual a classe utilizada para informar se o time subiu ou desceu alguma posição desde a última atualização do ranking;

- O nome final do arquivo, foi recuperado com ajuda do texto no dropdown, porém para processamento mais facilitado mais a frente, o texto foi convertido de “31 Mar 2022” para “2022_03_31”, traduzindo o mês abreviado do inglês para o número do mesmo.

O dataframe final ficou no seguinte formato:

| | date | team | iso-alfa-3 | total_points | previous_points | diff | var |
|-----|------------|------------------------|------------|--------------|-----------------|-------|------|
| pos | | | | | | | |
| 1 | 1992-12-31 | Germany | GER | 57.00 | 0.00 | 57.00 | - |
| 2 | 1992-12-31 | Italy | ITA | 57.00 | 0.00 | 57.00 | - |
| 3 | 1992-12-31 | Brazil | BRA | 56.00 | 0.00 | 56.00 | - |
| 4 | 1992-12-31 | Sweden | SWE | 56.00 | 0.00 | 56.00 | - |
| 5 | 1992-12-31 | England | ENG | 55.00 | 0.00 | 55.00 | - |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 207 | 2022-03-31 | Guam | GUM | 838.33 | 838.33 | 0.00 | down |
| 208 | 2022-03-31 | US Virgin Islands | VIR | 816.13 | 816.13 | 0.00 | down |
| 209 | 2022-03-31 | British Virgin Islands | VGB | 812.94 | 812.94 | 0.00 | down |
| 210 | 2022-03-31 | Anguilla | AIA | 792.34 | 792.34 | 0.00 | down |
| 211 | 2022-03-31 | San Marino | SMR | 776.97 | 780.33 | -3.36 | down |

63283 rows × 7 columns

O algoritmo completo, bem como o dataset gerado por ela, encontra-se disponível no github, estando o link disponível na seção 7, a aplicação é livre para ser baixada, utilizada e modificada. Por não haver um dataset como este disponível no Kaggle, o mesmo também foi disponibilizado lá, o link encontra-se na seção 7.

Obs: O algoritmo também faz o download dos ranking relacionados à seleções femininas de futebol, utilizando o mesmo site e estratégia.

3. Processamento/Tratamento de Dados

As bases de dados precisaram passar por adequações realizando ajustes tanto dos registros internos, quanto para permitir a integração entre si, promovendo uma agregação da informação que será utilizada para treinar os modelos de machine learning.

O dataset de histórico de resultado de partidas, foi importado, referenciado pelo nome de arquivo “results.csv”.

Carregando o DataSet de Resultados Históricos

```
[4] full_match_history = pd.read_csv(datasets_folder_location+'results.csv')
full_match_history.head()
```

| | date | home_team | away_team | home_score | away_score | tournament | city | country | neutral |
|---|------------|-----------|-----------|------------|------------|------------|---------|----------|---------|
| 0 | 1872-11-30 | Scotland | England | 0 | 0 | Friendly | Glasgow | Scotland | False |
| 1 | 1873-03-08 | England | Scotland | 4 | 2 | Friendly | London | England | False |
| 2 | 1874-03-07 | Scotland | England | 2 | 1 | Friendly | Glasgow | Scotland | False |
| 3 | 1875-03-06 | England | Scotland | 2 | 2 | Friendly | London | England | False |
| 4 | 1876-03-04 | Scotland | England | 3 | 0 | Friendly | Glasgow | Scotland | False |

Como os dados do dataset passaram por um pré-processamento, muitos dos ajustes das informações já foram feitas no momento de criação dos datasets. Algumas consultas foram realizadas, buscando identificar possíveis pontos de atenção. Tentando encontrar registros duplicados, ou campos com valores null/na, nenhum foi encontrado.

O “describe()” em conjunto com comando “count()” nos permitiu verificar que todos os campos principais estão preenchidos. Campos como “data”, “city”, “country” não serão relevantes para o treino do modelo de machine learning, portanto serão descartados na ocasião oportuna.

```

full_match_history[['date', 'home_team', 'away_team', 'neutral']].count()

```

```

date      43421
home_team  43421
away_team  43421
neutral    43421
dtype: int64

```

```

[ ] full_match_history.describe()

```

| | home_score | away_score |
|-------|--------------|--------------|
| count | 43421.000000 | 43421.000000 |
| mean | 1.741876 | 1.180972 |
| std | 1.751876 | 1.397932 |
| min | 0.000000 | 0.000000 |
| 25% | 1.000000 | 0.000000 |
| 50% | 1.000000 | 1.000000 |
| 75% | 2.000000 | 2.000000 |
| max | 31.000000 | 21.000000 |

Foi verificado também quanto a equiparidade de dados únicos nos campos “home_team” e “away_team”, o objetivo é identificar se há algum dado que não está em um deles, mas não em outro. Tivemos um achado aqui, existem 7 registros em “home_team”, mas a mesma equipe não se encontra em “away_team”. O mesmo foi achado no caminho oposto, mas dessa vez contendo 5 registros.

```

# Verificar quantidade de registros únicos em home_team e away_team
home_team_unique = full_match_history['home_team'].unique()
away_team_unique = full_match_history['away_team'].unique()

len(home_team_unique), len(away_team_unique)

(306, 304)

[ ] # Registros de away_team que não são encontrados em home_team
diff_list = full_match_history.loc[~(full_match_history['away_team'].isin(home_team_unique))]
diff_list['away_team'].unique()

array(['Asturias', 'Crimea', 'Two Sicilies', 'Surrey',
       'Brunei Darussalam'], dtype=object)

[ ] # Registros de home_team que não são encontrados em away_team
diff_list = full_match_history.loc[~(full_match_history['home_team'].isin(away_team_unique))]
diff_list = diff_list['home_team'].unique()
diff_list

array(['Silesia', 'Niue', 'Palau', 'Canary Islands',
       'Republic of St. Pauli', 'Gãgãuzia', 'Madrid'], dtype=object)

```

Apesar de não ser o que queremos, não será tratado agora, pois esses registros irão desaparecer quando relacionarmos esse dataset com o ranking de seleções e remover as seleções que não estão no ranking.

Vamos dar uma olhada agora no ranking da Fifa, deste dataset vamos nos atentar apenas aos campos “team” e “total_points”, pois são os dois únicos que serão de real importância para treinamento do modelo de Machine Learning, por isso por enquanto iremos ignorar o campo var com valores “-”.

```
last_ranking_fifa = pd.read_csv([ranking_folder+'2022_03_31.csv', index_col='pos'])
last_ranking_fifa
```

| | team | iso-alfa-3 | total_points | previous_points | variation | positions |
|-----|------------------------|------------|--------------|-----------------|-----------|-----------|
| pos | | | | | | |
| 1 | Brazil | BRA | 1832.69 | 1823.42 | 9.27 | up |
| 2 | Belgium | BEL | 1827.00 | 1828.45 | -1.45 | down |
| 3 | France | FRA | 1789.85 | 1786.15 | 3.70 | NaN |
| 4 | Argentina | ARG | 1765.13 | 1766.99 | -1.86 | NaN |
| 5 | England | ENG | 1761.71 | 1755.52 | 6.19 | NaN |
| ... | ... | ... | ... | ... | ... | ... |
| 207 | Guam | GUM | 838.33 | 838.33 | 0.00 | down |
| 208 | US Virgin Islands | VIR | 816.13 | 816.13 | 0.00 | down |
| 209 | British Virgin Islands | VGB | 812.94 | 812.94 | 0.00 | down |
| 210 | Anguilla | AIA | 792.34 | 792.34 | 0.00 | down |
| 211 | San Marino | SMR | 776.97 | 780.33 | -3.36 | down |

211 rows × 6 columns

```
last_ranking_fifa[['team', 'total_points']].count()
```

```
team          211
total_points   211
dtype: int64
```

Pensando na agregação dos dados, precisamos tirar todos os registros de partidas em que “home_team” e “away_team” não tem uma correspondência no ranking da Fifa, sendo assim foram identificadas 113 seleções que estão ou em “home_team” ou “away_team”, mas não estão no ranking da Fifa. Precisamos fazer essa limpeza pois, como iremos utilizar a pontuação no ranking com uma métrica para o modelo de Machine Learning avaliar, caso o deixemos, podemos induzir o nosso algoritmo a pensar diferente.

last_ranking_fifa.describe()

| | pos | total_points | previous_points | variation |
|-------|-----------|--------------|-----------------|------------|
| count | 211.00000 | 211.000000 | 211.000000 | 211.000000 |
| mean | 106.00000 | 1220.580284 | 1216.265118 | 4.315166 |
| std | 61.05462 | 252.134235 | 264.432733 | 62.550487 |
| min | 1.00000 | 776.970000 | 0.000000 | -35.420000 |
| 25% | 53.50000 | 1006.940000 | 1016.025000 | -4.015000 |
| 50% | 106.00000 | 1174.040000 | 1176.500000 | 0.000000 |
| 75% | 158.50000 | 1429.415000 | 1429.460000 | 3.620000 |
| max | 211.00000 | 1832.690000 | 1828.450000 | 899.330000 |

Poderia seguir na estratégia de preencher todos esses valores como 0, para todos que não fossem encontrados. Mas não seria uma abordagem muito viável, pois até mesmo o último time do ranking da Fifa “San Marino” tem uma pontuação, 776.97. Seguindo e analisando:

```
home_team_unique_with_away_diff = np.append(away_team_unique, diff_list)

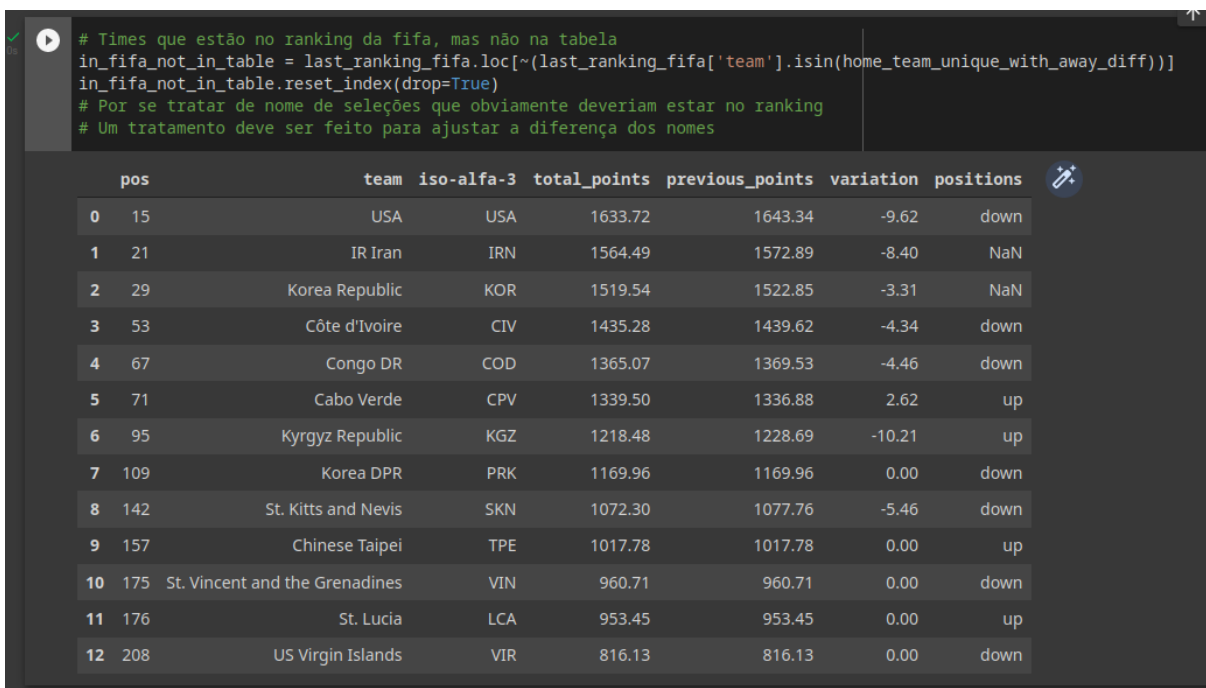
# Lista com todos os times do histórico de partidas
home_team_unique_with_away_diff.sort()

in_fifa = last_ranking_fifa['team'].unique()

# Todos os teams que não estão no ranking da fifa
not_in_fifa = list(filter(lambda team: team not in in_fifa, home_team_unique_with_away_diff))
print(len(not_in_fifa))
not_in_fifa
```

```
113
['Abkhazia',
 'Alderney',
 'Andalusia',
 'Arameans Suryoye',
 'Artsakh',
 'Asturias',
 'Barawa',
 'Basque Country',
 'Bonaire',
 'Brittany',
 'Brunei',
 'Canary Islands',
 'Cape Verde',
 'Cascadia',
 'Catalonia',
 'Central Spain',
 'Chagos Islands',
 'Chameria',
 'Corsica',
 'County of Nice',
 'Crimea',
 'Czechoslovakia',
```


Em teoria, essa seria a coleção de nomes de seleções cujos registro poderiam ser excluídos do histórico de partidas, mas ao fazer o caminho inverso, “Seleções do Ranking da Fifa que não estão no histórico de partidas”, entendemos o porquê não podemos fazer simplesmente assim.



```
# Times que estão no ranking da fifa, mas não na tabela
in_fifa_not_in_table = last_ranking_fifa.loc[~(last_ranking_fifa['team'].isin(home_team_unique_with_away_diff))]
in_fifa_not_in_table.reset_index(drop=True)
# Por se tratar de nome de seleções que obviamente deveriam estar no ranking
# Um tratamento deve ser feito para ajustar a diferença dos nomes
```

| | pos | team | iso-alfa-3 | total_points | previous_points | variation | positions |
|----|-----|--------------------------------|------------|--------------|-----------------|-----------|-----------|
| 0 | 15 | USA | USA | 1633.72 | 1643.34 | -9.62 | down |
| 1 | 21 | IR Iran | IRN | 1564.49 | 1572.89 | -8.40 | NaN |
| 2 | 29 | Korea Republic | KOR | 1519.54 | 1522.85 | -3.31 | NaN |
| 3 | 53 | Côte d'Ivoire | CIV | 1435.28 | 1439.62 | -4.34 | down |
| 4 | 67 | Congo DR | COD | 1365.07 | 1369.53 | -4.46 | down |
| 5 | 71 | Cabo Verde | CPV | 1339.50 | 1336.88 | 2.62 | up |
| 6 | 95 | Kyrgyz Republic | KGZ | 1218.48 | 1228.69 | -10.21 | up |
| 7 | 109 | Korea DPR | PRK | 1169.96 | 1169.96 | 0.00 | down |
| 8 | 142 | St. Kitts and Nevis | SKN | 1072.30 | 1077.76 | -5.46 | down |
| 9 | 157 | Chinese Taipei | TPE | 1017.78 | 1017.78 | 0.00 | up |
| 10 | 175 | St. Vincent and the Grenadines | VIN | 960.71 | 960.71 | 0.00 | down |
| 11 | 176 | St. Lucia | LCA | 953.45 | 953.45 | 0.00 | up |
| 12 | 208 | US Virgin Islands | VIR | 816.13 | 816.13 | 0.00 | down |

O motivo é que a nomenclatura utilizada para o nome de algumas seleções diverge entre o apresentado pelo ranking da Fifa e o disponível no dataset do histórico de partidas. Algumas que inclusive estarão na Copa do Mundo 2022, estão sendo desconsideradas, como é o caso dos Estados Unidos, Irã, Coreia do Sul, Costa do Marfim e Congo que estão na Copa, e as outras seleções na lista acima, num total de 12.

Isso acontece porque no ranking da Fifa, os Estados Unidos são identificados por “USA”, enquanto que no dataset de histórico de partidas “United States”.

A Seleção do Estados Unidos encontra-se como USA no ranking da Fifa, porém na tabela histórica de jogos, é referenciado como 'United States'. Assim como outras seleções, o ajuste deve ser feito para equiparação dos nomes, pois será de extrema importância no momento de buscar a pontuação no ranking.

```
full_match_history.loc[(full_match_history['home_team']=='United States') | (full_match_history['away_team']=='United States')].head()
```

| | date | home_team | away_team | home_score | away_score | tournament | city | country | neutral |
|-----|------------|---------------|---------------|------------|------------|------------|------------|---------------|---------|
| 41 | 1885-11-28 | United States | Canada | 0 | 1 | Friendly | Newark | United States | False |
| 48 | 1886-11-25 | United States | Canada | 3 | 2 | Friendly | Newark | United States | False |
| 450 | 1916-08-20 | Sweden | United States | 2 | 3 | Friendly | Stockholm | Sweden | False |
| 451 | 1916-09-03 | Norway | United States | 1 | 1 | Friendly | Kristiania | Norway | False |
| 812 | 1924-06-10 | Poland | United States | 2 | 3 | Friendly | Warsaw | Poland | False |

Uma intervenção manual foi necessária para identificar as 12 seleções que têm o nome divergente, e precisarão ser corrigidas.

Após análise manual, foram identificados os nomes de seleções da tabela que estão em desacordo com o nome na Fifa

```
[38] nome_a_serem_corrigidos = {'United States': 'USA', 'South Korea': 'Korea Republic', 'DR Congo': 'Congo DR', 'North Korea': 'Korea DPR'}
print(nome_a_serem_corrigidos)

{'United States': 'USA', 'South Korea': 'Korea Republic', 'DR Congo': 'Congo DR', 'North Korea': 'Korea DPR', 'Ivory Coast': 'Côte d'Ivoire'}
```

```
# Correção dos nomes
full_match_history['home_team'].replace(nome_a_serem_corrigidos,inplace=True)
full_match_history['away_team'].replace(nome_a_serem_corrigidos,inplace=True)
```

Após correção aplicada:

```
[41] # verificando se ainda há resquícios do nome United States na coluna home_team em partidas
full_match_history.loc[(full_match_history['home_team']=='United States') | (full_match_history['away_team']=='United States')]
```

| | date | home_team | away_team | home_score | away_score | tournament | city | country | neutral |
|--|------|-----------|-----------|------------|------------|------------|------|---------|---------|
|--|------|-----------|-----------|------------|------------|------------|------|---------|---------|

```
# Verificando se ainda há resquícios de outros nomes divergentes
last_ranking_fifa.loc[~(last_ranking_fifa['team'].isin(full_match_history['home_team']))]
```

| | team | iso-alfa-3 | total_points | previous_points | variation | positions |
|-----|------|------------|--------------|-----------------|-----------|-----------|
| pos | | | | | | |

Nenhum registro consta mais como faltante entre o ranking da Fifa e o dataset. Restando apenas remover aquelas seleções que não estão no ranking da Fifa:

```
[123] full_match_history.drop(full_match_history.index[full_match_history["home_team"].isin(not_in_fifa)], axis=0, inplace=True)
full_match_history.drop(full_match_history.index[full_match_history["away_team"].isin(not_in_fifa)], axis=0, inplace=True)
```

Neste momento a lista com todos os times que participarão da Copa do Mundo foi definida. Alguns algoritmos e funções criados para que possamos identificar as seleções no ranking da Fifa, e agregar com as informações do ranking na partida.

```
Realizando o Teste da funcionalidade

# Testing functions to find ranking
find_ranking_points_team_for_match('Morocco', '2022-03-29')

1547.42
```

O algoritmo foi testado puxando o ranking da seleção de Marrocos, simulando que a partida estivesse ocorrendo no dia 29/03/2022

A parte de processamento mais pesada nessa parte de tratamento de dados, fica por conta do algoritmo de agregação dos dados. Para cada partida, o algoritmo deve pegar o ranking mais atualizado até a data da mesma, e identificar a pontuação de cada time, um novo dataset será criado, agora sim, pronto para ser aplicado a um modelo de Machine Learning.

```
# Cleaned
full_match_history
```

| | date | home_team | away_team | home_score | away_score | tournament | city | country | neutral |
|-------|------------|-----------------|--------------|------------|------------|------------------------------|-------------|------------|---------|
| 0 | 1993-01-01 | Ghana | Mali | 1 | 1 | Friendly | Libreville | Gabon | True |
| 1 | 1993-01-02 | Gabon | Burkina Faso | 1 | 1 | Friendly | Libreville | Gabon | False |
| 2 | 1993-01-02 | Kuwait | Lebanon | 2 | 0 | Friendly | Kuwait City | Kuwait | False |
| 3 | 1993-01-03 | Burkina Faso | Mali | 1 | 0 | Friendly | Libreville | Gabon | True |
| 4 | 1993-01-03 | Gabon | Ghana | 2 | 3 | Friendly | Libreville | Gabon | False |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 24529 | 2022-03-30 | Mexico | El Salvador | 2 | 0 | FIFA World Cup qualification | Mexico City | Mexico | False |
| 24530 | 2022-03-30 | Costa Rica | USA | 2 | 0 | FIFA World Cup qualification | San José | Costa Rica | False |
| 24531 | 2022-03-30 | Panama | Canada | 1 | 0 | FIFA World Cup qualification | Panama City | Panama | False |
| 24532 | 2022-03-30 | Jamaica | Honduras | 2 | 1 | FIFA World Cup qualification | Kingston | Jamaica | False |
| 24533 | 2022-03-30 | Solomon Islands | New Zealand | 0 | 5 | FIFA World Cup qualification | Doha | Qatar | True |

24534 rows x 9 columns

Algoritmo agregação entre as partidas e o ranking de seleções:

```

# Devido a baixa performance do algoritmo o arquivo foi pre-carregado e disponibilizado separadamente

match_history = pd.DataFrame()
match_history['date'] = full_match_history['date']
match_history['home_team'] = full_match_history['home_team']
match_history['away_team'] = full_match_history['away_team']
match_history['neutral'] = full_match_history['neutral']

ranking_home_list = []
ranking_away_list = []

for idx in full_match_history.index:
    # print(full_match_history['date'][idx], full_match_history['home_team'][idx], ' x ', full_match_history['away_team'][idx])
    ranking_home_list.append(find_ranking_points_team_for_match(full_match_history['date'][idx], full_match_history['home_team'][idx]))
    ranking_away_list.append(find_ranking_points_team_for_match(full_match_history['date'][idx], full_match_history['away_team'][idx]))

match_history['home_rk_points'] = ranking_home_list
match_history['away_rk_points'] = ranking_away_list

### MELHORAR AQUI COM O .iterrows()

def populate_winners(match_df):
    winner_list = []
    for idx in full_match_history.index:
        home_score = full_match_history['home_score'][idx]
        away_score = full_match_history['away_score'][idx]

        if home_score == away_score:
            winner_list.append('DRAW')
        elif home_score > away_score:
            winner_list.append('HOME')
        else:
            winner_list.append('AWAY')

    match_history['winner'] = winner_list

populate_winners(match_history)

# Negative points_diff means that away_team has better ranking position than home_team
match_history['points_diff'] = match_history['home_rk_points'] - match_history['away_rk_points']

```

Devido a alta carga de processamento que há no algoritmo de agregação, por motivos práticos, um dataset novo e pré-processado dessa fase de agregação foi criado e disponibilizado em arquivo separadamente sob o nome de `result_with_points.csv`, isso evita que tenhamos que rodar o algoritmo sempre que quisermos realizar os testes, e evitamos isso devido a sua baixa performance.

Após a agregação, o nosso histórico de partidas ficou da seguinte maneira:

| | date | home_team | away_team | neutral | home_rk_points | away_rk_points | winner | points_diff |
|-------|------------|-----------------|--------------|---------|----------------|----------------|--------|-------------|
| 0 | 1993-01-01 | Ghana | Mali | True | 34.00 | 22.00 | DRAW | 12.00 |
| 1 | 1993-01-02 | Gabon | Burkina Faso | False | 27.00 | 11.00 | DRAW | 16.00 |
| 2 | 1993-01-02 | Kuwait | Lebanon | False | 21.00 | 0.00 | HOME | 21.00 |
| 3 | 1993-01-03 | Burkina Faso | Mali | True | 11.00 | 22.00 | HOME | -11.00 |
| 4 | 1993-01-03 | Gabon | Ghana | False | 27.00 | 34.00 | AWAY | -7.00 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 25870 | 2022-03-30 | Mexico | El Salvador | False | 1647.90 | 1346.04 | HOME | 301.86 |
| 25871 | 2022-03-30 | Costa Rica | USA | False | 1464.06 | 1643.34 | HOME | -179.28 |
| 25872 | 2022-03-30 | Panama | Canada | False | 1375.56 | 1497.82 | HOME | -122.26 |
| 25873 | 2022-03-30 | Jamaica | Honduras | False | 1378.62 | 1303.96 | HOME | 74.66 |
| 25874 | 2022-03-30 | Solomon Islands | New Zealand | True | 1072.78 | 1161.66 | AWAY | -88.88 |

25875 rows x 8 columns

Foram adicionados 4 campos novos:

- `home_rk_points` = Pontuação do time da casa no ranking da Fifa referente a data da partida
- `away_rk_points` = Pontuação do time visitante no ranking da Fifa referente a data da partida
- `winner` = Campo que representa que time venceu a partida, HOME, AWAY ou DRAW (em caso de empate), foi criado relacionados os campos `home_score` e `away_score`
- `points_diff` = Campo calculado sobre a diferença entre a pontuação do time da casa e do visitante, números negativos, indicam que o ranking do time visitante é superior ao do mandante.

Trabalhando como o dataframe novo, ainda precisamos realizar alguns ajustes no dataframe atual, o campo `winner`, passará a apresentar valores entre -1,0,1. Em que -1 significa vitória do visitante, 0 empate e 1 vitória do time mandante. A nomenclatura de mandante e visitante aparece pois o campo `neutral` vai ser interessante para indicar aqueles times que têm performance melhor em casa do que fora, isso pode computar pontos importantes, já que a Copa será disputada em campo neutro.

As demais mudanças estão no próprio campo `neutral`, que passará a ser 1, caso a partida se dê em campo neutro, e 0 caso não.

```
match_history.loc[match_history['winner']=='HOME', 'winner'] = 1
match_history.loc[match_history['winner']=='AWAY', 'winner'] = -1
match_history.loc[match_history['winner']=='DRAW', 'winner'] = 0

match_history.loc[match_history['neutral']==True, 'neutral'] = 1
match_history.loc[match_history['neutral']==False, 'neutral'] = 0

match_history['winner'] = pd.to_numeric(match_history['winner'])
match_history['neutral'] = pd.to_numeric(match_history['neutral'])

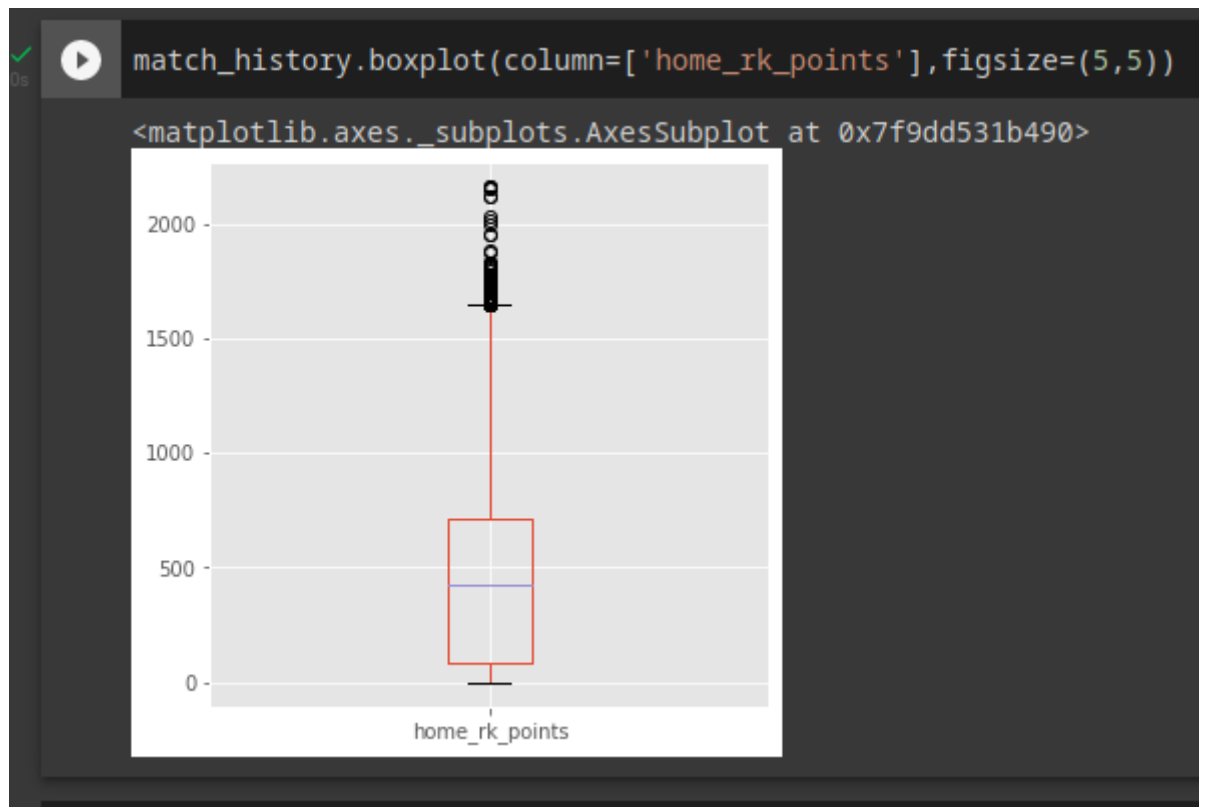
match_history.head()
```

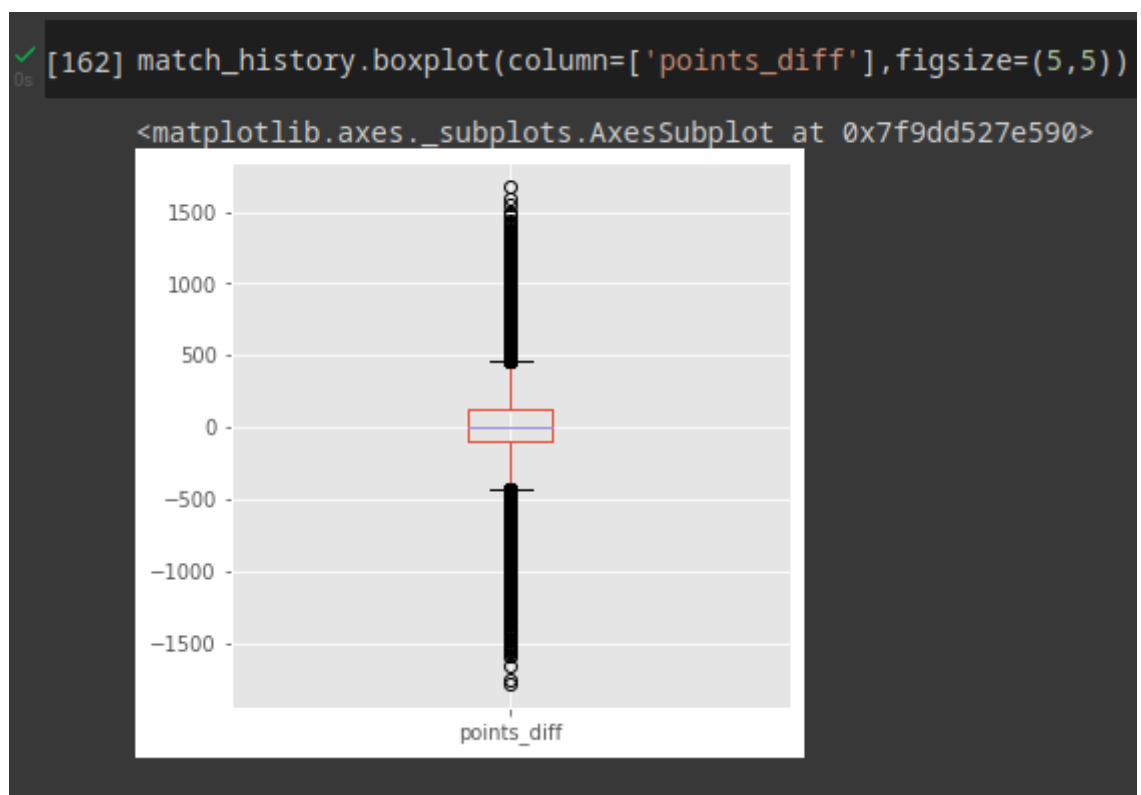
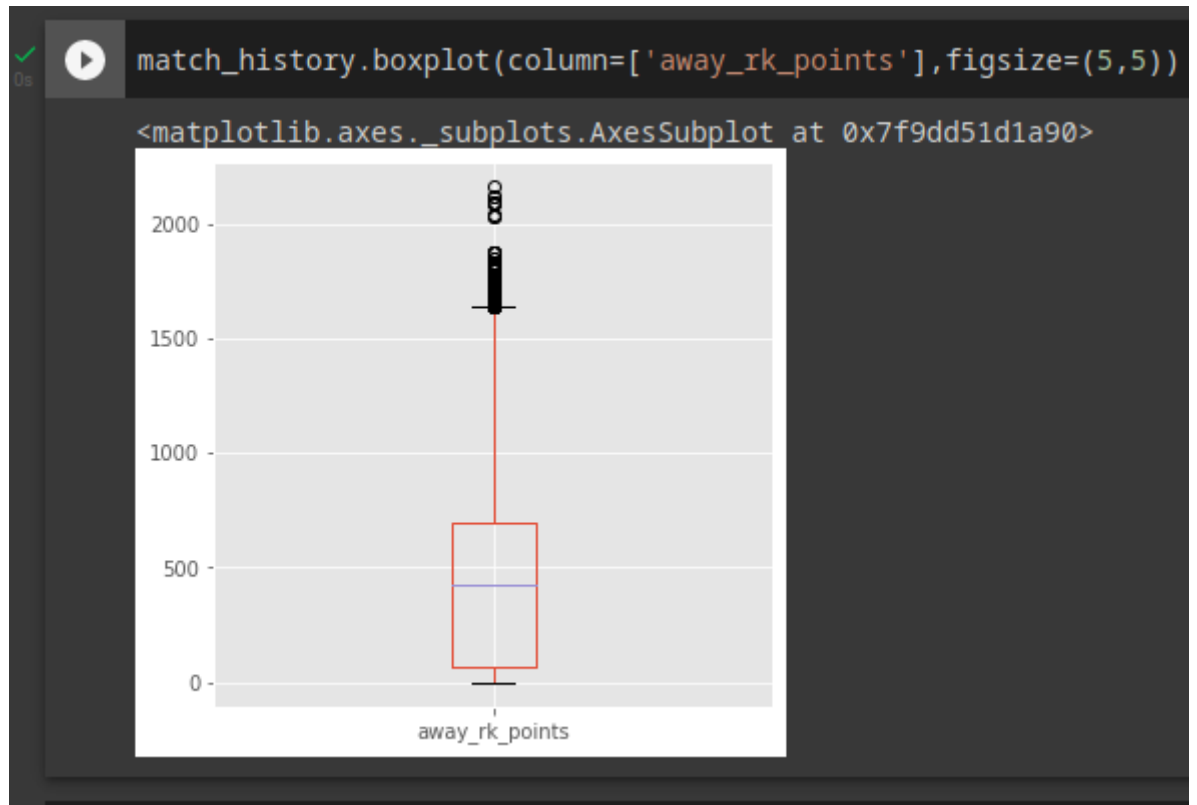
| | date | home_team | away_team | neutral | home_rk_points | away_rk_points | winner | points_diff |
|---|------------|--------------|--------------|---------|----------------|----------------|--------|-------------|
| 0 | 1993-01-01 | Ghana | Mali | 1 | 34.0 | 22.0 | 0 | 12.0 |
| 1 | 1993-01-02 | Gabon | Burkina Faso | 0 | 27.0 | 11.0 | 0 | 16.0 |
| 2 | 1993-01-02 | Kuwait | Lebanon | 0 | 21.0 | 0.0 | 1 | 21.0 |
| 3 | 1993-01-03 | Burkina Faso | Mali | 1 | 11.0 | 22.0 | 1 | -11.0 |
| 4 | 1993-01-03 | Gabon | Ghana | 0 | 27.0 | 34.0 | -1 | -7.0 |

Processamentos/Tratamentos finalizados, podemos agora iniciar uma análise sobre os dados e ir para a criação dos modelos de Machine Learning.

4. Análise e Exploração dos Dados


Iniciamos a nossa análise em busca de outliers, qualquer valor que esteja fora dos padrões aceitáveis:





Sem valores com discrepância, e dentro dos valores esperados, nossos campos principais demonstram estar livres de interferências para seguirmos em frente.

A execução do comando “describe()” sobre o nosso dataframe principal nos ajuda a corroborar com o que os boxplots já nos mostraram.


✓ 0s  match_history.describe()

| | neutral | home_rk_points | away_rk_points | winner | points_diff |
|-------|--------------|----------------|----------------|--------------|--------------|
| count | 25875.000000 | 25875.000000 | 25875.000000 | 25875.000000 | 25875.000000 |
| mean | 0.270222 | 498.362811 | 486.563349 | 0.207614 | 11.799462 |
| std | 0.444083 | 438.591234 | 433.505524 | 0.849152 | 281.085048 |
| min | 0.000000 | 0.000000 | 0.000000 | -1.000000 | -1775.000000 |
| 25% | 0.000000 | 87.000000 | 69.000000 | -1.000000 | -93.000000 |
| 50% | 0.000000 | 427.000000 | 422.000000 | 0.000000 | 3.000000 |
| 75% | 1.000000 | 712.000000 | 697.000000 | 1.000000 | 128.000000 |
| max | 1.000000 | 2164.000000 | 2164.000000 | 1.000000 | 1669.000000 |

Sem campos Nulos ou faltantes:

✓ 0s [255] match_history.isnull().sum()

| | |
|----------------|---|
| home_team | 0 |
| away_team | 0 |
| neutral | 0 |
| home_rk_points | 0 |
| away_rk_points | 0 |
| winner | 0 |
| points_diff | 0 |
| dtype: int64 | |

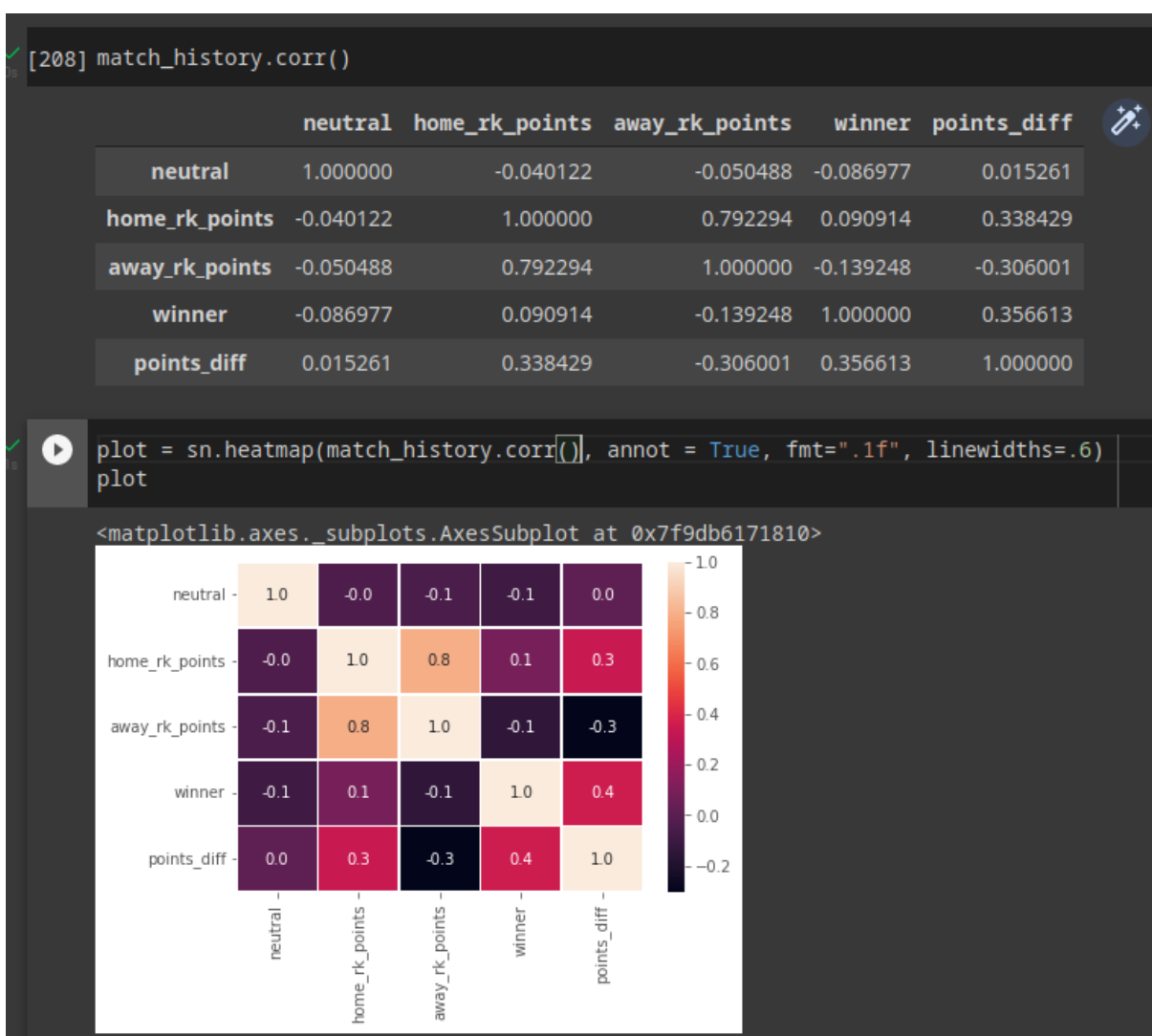
✓ 0s  match_history.isna().sum()

| | |
|----------------|---|
| home_team | 0 |
| away_team | 0 |
| neutral | 0 |
| home_rk_points | 0 |
| away_rk_points | 0 |
| winner | 0 |
| points_diff | 0 |
| dtype: int64 | |

Com o gráfico a seguir, vemos que há uma tendência muito grande, quase o dobro, de vitória do time da casa em relação ao visitante, que ganha apenas para o número de empates, porém ainda assim bem próximos.



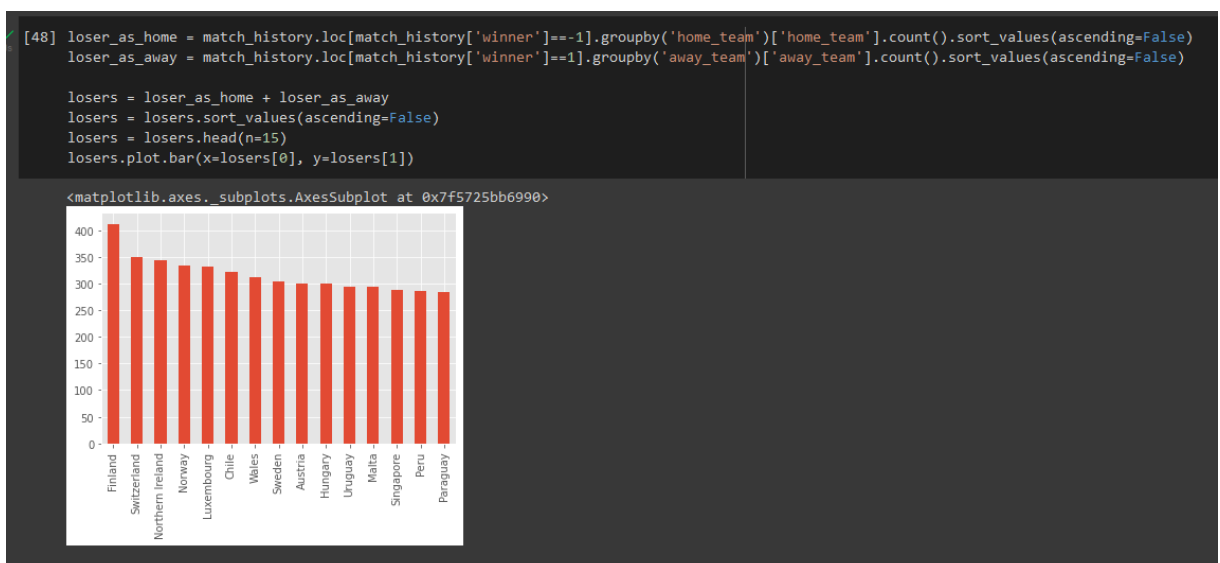
A análise de correlação mostra que há correlação positiva entre winner e “points_diff” apesar de fraca, entre winner e “home_rk_points”, “away_points” e “neutral” é quase nula. Significa que times com pontuação muito distantes entre os times, gera uma tendência do time com mais pontos a vencer. Seleções no alto do ranking da Fifa tendem a ser beneficiadas ao jogar contra times menores, mas à medida que pega times com diferenças de pontos mais baixas, o jogo passa a ser mais equilibrado, pois a correlação apesar de existir, é fraca.



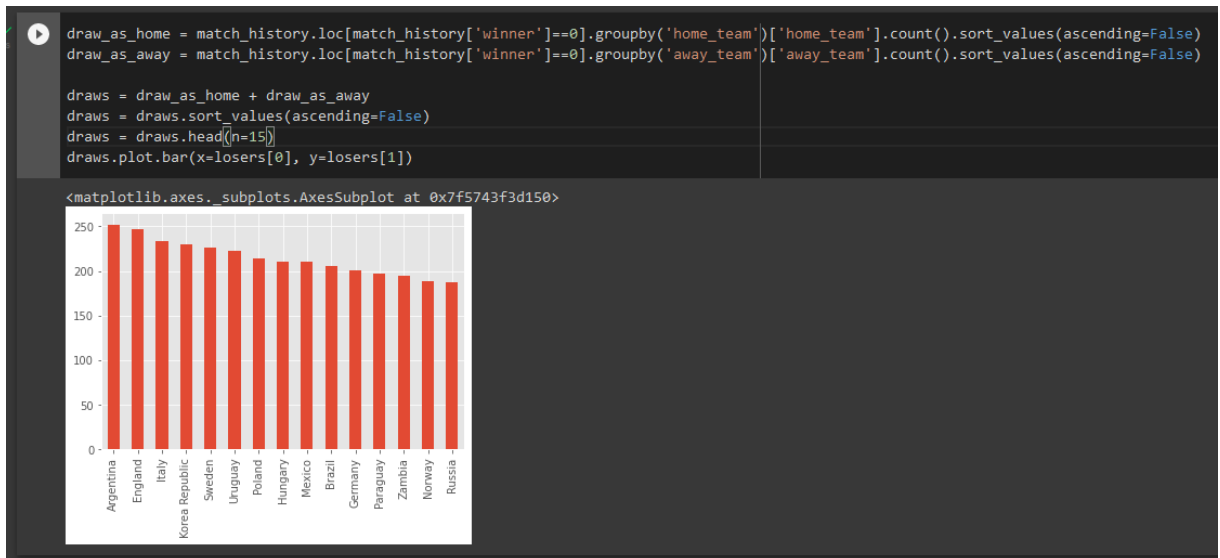
A Suécia aparece como o time que mais venceu na história, seguido do México e dos Estados Unidos. Isso pode impactar positivamente essas seleções.



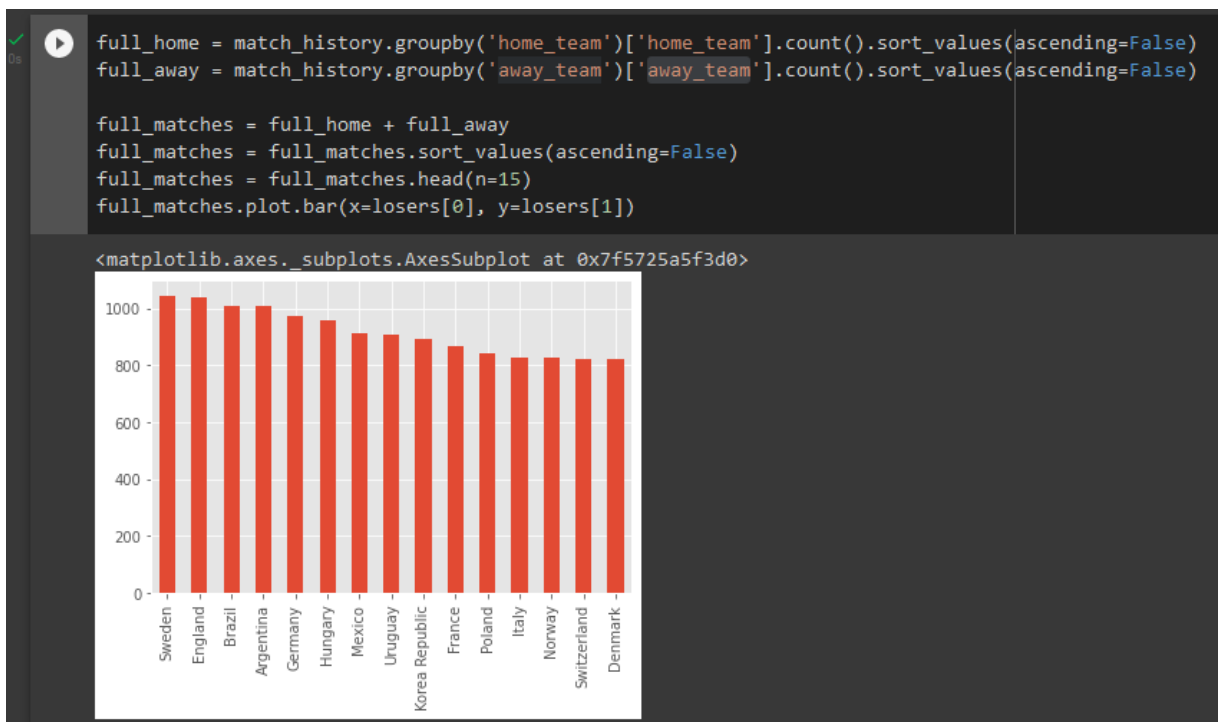
Do outro lado da tabela, Finlândia, Suíça e Irlanda do Norte encabeçam os times com mais derrotas na história.



Entre os times que mais empatam, apresentam-se, Argentina, Inglaterra e Itália.



E analisando os times com mais jogos na história temos Suécia, Inglaterra e Brasil como líderes do ranking:



5. Criação dos Modelos de Machine Learning

Para a criação do modelo de machine learning vamos fazer a divisão do nosso dataset em dataset de treino e de teste, os parâmetros utilizados foram, `test_size = 0.30`, `random_state=49`, e `shuffle=True`.

```
[56] from sklearn.model_selection import train_test_split

dummed = pd.get_dummies(match_history, prefix=['home_team', 'away_team', 'tournament'], columns=['home_team', 'away_team', 'tournament'])
x = dummed.drop(['winner'], axis = 1)
y = dummed['winner'].astype('int')

X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.30, random_state=49, shuffle=True)
```

Com essa divisão o nosso dataset de um total de 25875 registros, foi dividido em dataset de treino com um total de 18112 e dataset de teste com um total de 7763.

```
len(dummed), len(X_train), len(X_test), len(y_train), len(y_test)

(25875, 18112, 7763, 18112, 7763)
```

Nossa variável target, será o campo winner, por esse motivo, ele foi removido do dataset de treino, e adicionado ao campo de respostas para os determinados registros.

5.1 Escolha dos Modelos

Para o problema proposto, selecionamos alguns algoritmos de Classificação, dentre vários que foram testados, alguns tiveram uma performance muito abaixo e 7 tiveram a performance nos níveis aceitáveis.

Os algoritmos escolhidos para realização dos testes foram: LogisticRegression, RidgeClassifier, XGBClassifier, KNeighborsClassifier, GradientBoostClassifier e DecisionTreeClassifier.

Os algoritmos serão testados utilizando a mesma base de dados, ou seja, mesmo dataset de treino e teste entre eles. Para cada execução, será criado o “classification_report” e a “confusion_matrix” do pacote sklearn.metrics, e será realizado um “Cross Validation” utilizando “StratifiedKFold” como estratégia.

Para aplicação dos modelos será utilizado a biblioteca sklearn no Python na sua versão 3 utilizando o Google Colab para execução.

5.1.1 Logistic Regression

Apesar de ser um algoritmo de Regressão, o algoritmo “LogisticRegression” trabalha de forma a dar um percentual de chance para duas ou mais valores ocorrerem na variável target, sendo assim ele consegue ser utilizado com eficiência para realizar um papel de classificação como o nosso.

Com base nas métricas do modelo, o algoritmo tende a acertar com precisão de 57%, segundo o cross validation k-fold, o resultado da partida, com base nos campos que foram disponibilizados no modelo. A vitória do mandante foi o atributo mais acertado pelo algoritmo, chegando a uma precisão de 84%. Sobre a vitória do time visitante o algoritmo obteve um acerto de cerca de 57%, enquanto que os empates só foram previstos em 6% dos casos, o que derrubou o F1-Score para 11%.

```

Algorithm -> LogisticRegression
StratifiedKFold Accuracy: 57.10% (0.52%)
[[1226  533  452]
 [ 141  110  144]
 [ 784 1133 3240]]

```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| -1 | 0.57 | 0.55 | 0.56 | 2211 |
| 0 | 0.06 | 0.28 | 0.10 | 395 |
| 1 | 0.84 | 0.63 | 0.72 | 5157 |
| accuracy | | | 0.59 | 7763 |
| macro avg | 0.49 | 0.49 | 0.46 | 7763 |
| weighted avg | 0.73 | 0.59 | 0.64 | 7763 |

```

<matplotlib.axes._subplots.AxesSubplot at 0x7f31cc6d5d10>

```



5.1.2 Ridge Classifier

O modelo de Machine mais rápido nos testes, dentro todos, teve uma acurácia levemente superior à do LogisticRegression, e também apresentou um desvio padrão menor 0.49%. apesar F1-Score de 11%, acima do LogisticRegression, na prática o algoritmo só conseguiu prever empates com os mesmos 6% de precisão, 84% de precisão em acerto de vitórias do mandante, e 58% de acerto em vitórias do visitante, colocando-os em um empate técnico. Foi o algoritmo que apresentou o melhor desvio padrão dentre todos.

```

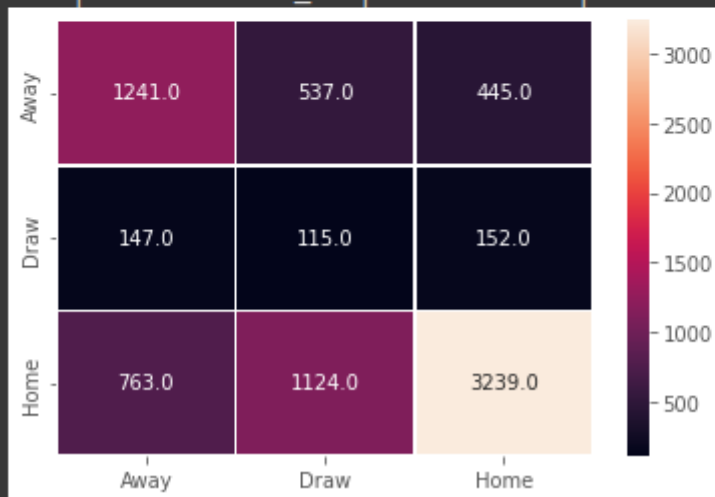
Algorithm -> RidgeClassifier
StratifiedKfold Accuracy: 57.98% (0.49%)
      precision    recall  f1-score   support

      -1         0.58         0.56         0.57         2223
         0         0.06         0.28         0.11          414
         1         0.84         0.63         0.72         5126

 accuracy
macro avg         0.50         0.49         0.47         7763
weighted avg         0.73         0.59         0.65         7763

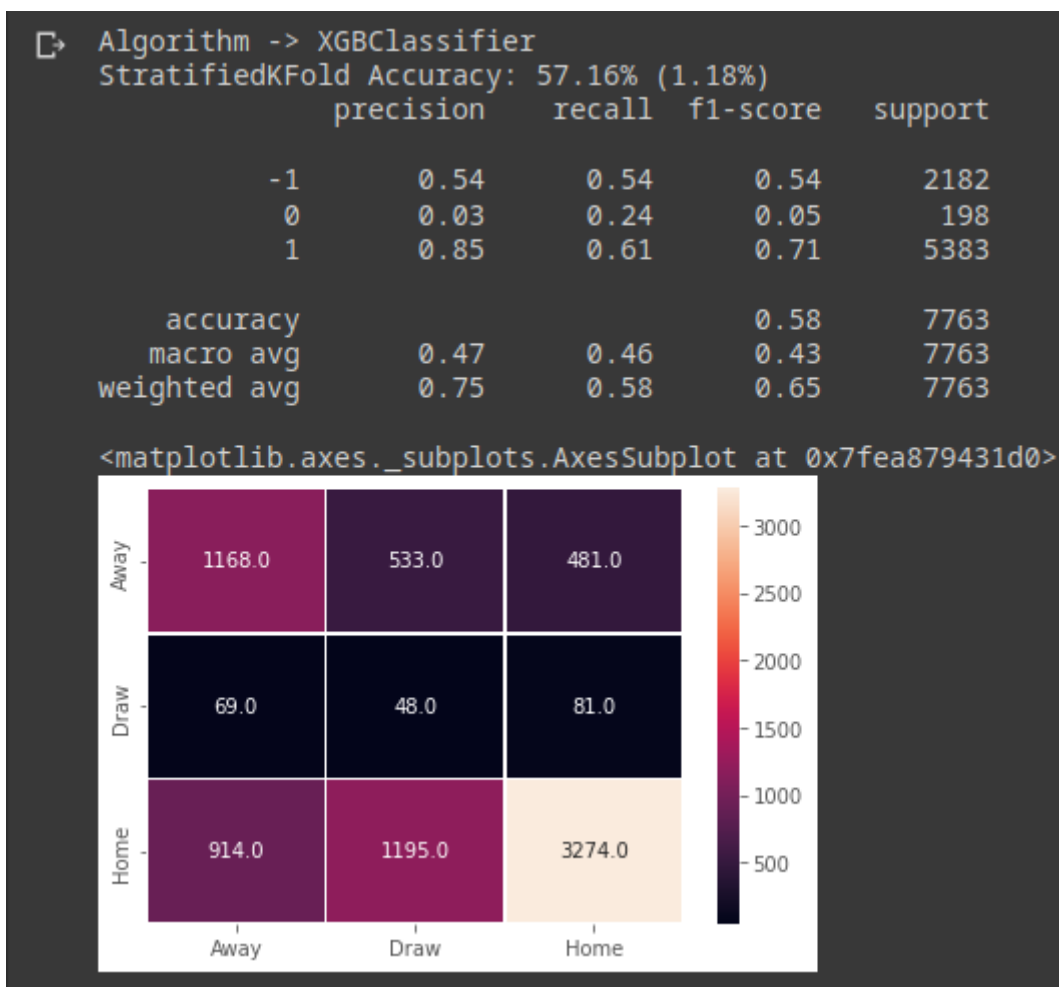
<matplotlib.axes._subplots.AxesSubplot at 0x7fea899df490>

```



5.1.3 XGB Classifier

Modelo de Machine Learning XGB Classifier, foi o modelo que mais demorou para realizar o treinamento, ainda mais quando aplicado ao Cross Validation K-Fold. Preparado para rodar em cluster e processamento distribuído, trabalhar com poucos dados em uma máquina talvez não seja o melhor cenário para ele. Apesar dos 57.16% de acurácia, foi inferior ao LogisticRegression no F1-Score e na precisão de acerto de empates, que foi menor ainda, chegando a apenas 3% de acerto.



5.1.4 KNeighbors Classifier

O modelo de classificação que utiliza a estratégia de vizinhos próximos, foi até o momento o algoritmo com a menor precisão dentre todos, nas métricas do Cross Validation K-Fold, no entanto foi o que teve a melhor capacidade de identificar empates com precisão de 19%, muito superior aos anteriores. Com o desvio padrão de 1.60% foi o algoritmo menos eficiente.

```

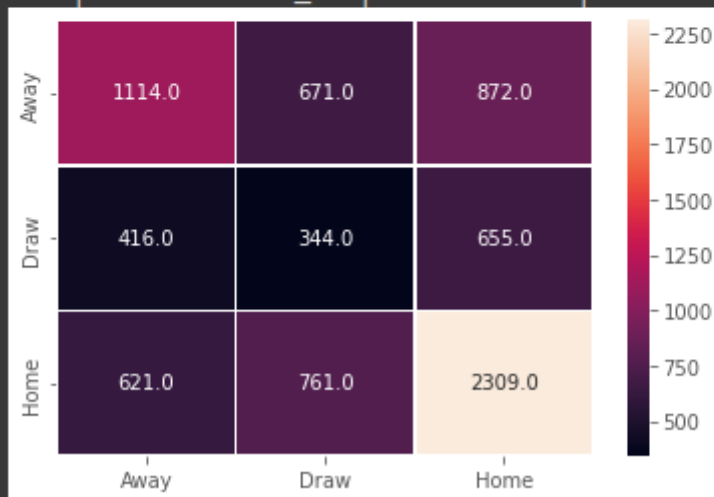
Algorithm -> KNeighborsClassifier
StratifiedKFold Accuracy: 48.99% (1.60%)
      precision    recall  f1-score   support

      -1         0.52      0.42      0.46       2657
         0         0.19      0.24      0.22       1415
         1         0.60      0.63      0.61       3691

 accuracy
macro avg         0.44      0.43      0.43       7763
weighted avg      0.50      0.49      0.49       7763

```

<matplotlib.axes._subplots.AxesSubplot at 0x7fea878fa7c



5.1.5 Gradient Boosting Classifier

Mais um algoritmo que peca ao acertar os empates, apenas 3% dos testes foram acertados, apesar disso a sua precisão segundo o Cross Validation K-Fold, é mais alta que até mesmo o LogisticRegression. O seu F1-Score foi 5%. Porém com desvio padrão de 1.02% foi um dos algoritmos menos eficientes.

```

Algorithm -> GradientBoostingClassifier
StratifiedKFold Accuracy: 57.39% (1.02%)
      precision    recall  f1-score   support

      -1         0.53      0.53      0.53       2152
         0         0.03      0.29      0.05        182
         1         0.85      0.60      0.71       5429

 accuracy
macro avg      0.47      0.47      0.43       7763
weighted avg    0.75      0.58      0.64       7763

```

<matplotlib.axes._subplots.AxesSubplot at 0x7fea86d59790>



5.1.6 Decision Tree Classifier

O algoritmo que teve a melhor performance para prever empates, com 22% de precisão conseguiu ter uma taxa de acertos na variável, maior que o KNeighborsClassifier. Porém, houve uma diminuição na taxa de acertos tanto para identificar mandantes como vencedores, quanto visitantes. Segundo o Cross Validation K-Fold, apresentou 49% de acurácia com 0.65% de desvio padrão.

```

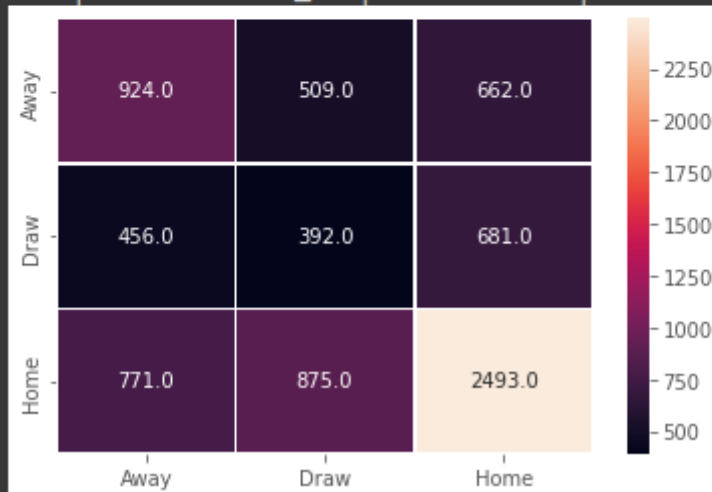
Algorithm -> DecisionTreeClassifier
StratifiedKFold Accuracy: 49.40% (0.65%)
      precision    recall  f1-score   support

      -1         0.43      0.44      0.44       2095
       0         0.22      0.26      0.24       1529
       1         0.65      0.60      0.63       4139

 accuracy
macro avg      0.43      0.43      0.43       7763
weighted avg    0.51      0.49      0.50       7763

```

<matplotlib.axes._subplots.AxesSubplot at 0x7fea84e40bd0>



5.2 Decisão

Os dois algoritmos escolhidos foram o RidgeClassifier e o LogisticRegression, ambos apresentaram baixa taxa de acerto no empate o que geralmente é realmente muito difícil de acertar, porém o desvio padrão foram os melhores dentre os algoritmos, apesar do F1-Score ter sido baixo, se mantiveram como um custo benefício, enquanto outros algoritmos com um F1-Score maior, tiveram um desvio muito mais alto, quase que o dobro dos demais.

5.2.1 O Problema do Empate

O empate realmente é um problema difícil de prever, mesmo uma pessoa, com seu pensamento crítico raramente irá escolher o empate como o resultado final de uma partida de futebol por ser estatisticamente mais raro.

A vitória um **Time A** sobre um **Time B** implica em o **Time A** marcar mais gols que o **Time B**, a quantidade de diferença de gols não importa neste momento, se o **Time A** marcar mais vezes que o **Time B**, ele será vencedor.

Ao olhar para o empate há uma regra que aparentemente está explícita, mas não está. **Time A** precisa marcar a exata quantidade de gols do **Time B**, não importa se a partida será 0x0, 1x1 ou até 20x20, ambos os times precisam marcar a mesma quantidade de gols. Isso por si só já mostra a complexidade de se acertar um empate entre duas seleções.

Além de ser mais difícil de acertar os empates, é mais raro também de ocorrer. Em um total de 43.421 partidas entre as seleções, apenas aproximadamente 10.000 delas ocasionaram um empate, cerca de $\frac{1}{4}$ das ocorrências. Sendo assim temos 2 cenários, (1) é muito fácil para algoritmo chutar algo diferente de um empate, pois ele terá uma probabilidade maior de acertar, e (2) com menos casos de empate, isso implica em menos casos de treino para o algoritmo, de modo a ele não saber com eficiência como identificar um empate.

Por esse motivo é até que plausível a ideia de que os algoritmos tenham tanta dificuldade em ter uma alta taxa de acerto em empates, pois com apenas o estudo mental acima, podemos ver que é muito mais raro de acontecer que uma vitória do **Time A** ou do **Time B**.

5.2.2 O problema da Base de Dados

Para uma tentativa de análise mais precisa, mais variáveis preditoras seriam necessárias, algo como pontuação dos jogadores que atuam sobre os times, detalhes sobre o clima no dia da partida, entre outros. O problema é que muitas dessas informações só são conhecidas em momentos antes da partida.

Uma previsão do tempo, por exemplo, poderia identificar qual seleção gosta de jogar no clima do dia da partida, mas só seria possível analisar dias antes da mesma acontecer. O mesmo acontece ao tentar analisar a escalação dos times, que muitas vezes são divulgados até mesmo horas antes da partida começar, para evitar uma contra tática do time adversário.

Por isso o estudo se baseia apenas em conhecer os algoritmos, e a sua aplicação real é muito mais complexa e completa do que puramente identificar os vencedores de partidas através de resultados anteriores, apesar deste também ser uma das características que pesam, o chamado “Fator Histórico”.

6. Apresentação dos Resultados

Nesta seção, vamos aplicar os dois modelos de Machine Learning escolhidos na etapa anterior, e ambos irão ser aplicados às partidas da Copa do Mundo, para analisarmos a distribuição dos dados e como se comportam com dados novos. Observando também se os mesmos se aproximam da realidade.

Antes de tudo é necessário contexto. Até a presente data, existem 3 vagas em aberto para a Copa do Mundo Qatar 2022. As Eliminatórias Europeias foram atrasadas devido à guerra entre Rússia e Ucrânia, assim como as Eliminatórias da Oceania e Ásia. Portanto uma vaga há para ser disputada na Europa, e mais duas vagas em repescagens.

O próprio algoritmo treinado será utilizado para simular as partidas faltantes da Eliminatória e das repescagens, fase de grupos completa, oitavas de final, quartas de final, semifinal e final. Avaliando quanto será o percentual de cada possibilidade “Vitória do mandante”, “Empate” e “Vitória do Visitante”.

A nomenclatura será mantida, mesmo os jogos da Copa do Mundo não sendo realizados com o quesito de “dentro ou fora de casa”

6.1 RidgeClassifier

Partidas das Eliminatórias da Europa, a Ucrânia vence a primeira contra a Escócia, e vence a segunda partida contra País de Gales, com chances de 44.18% e 42.05% respectivamente. Juntando-se ao Grupo B.

```
Partida entre Scotland x Ukraine -> Predictions ( Home: 29.22 % | Draw: 26.60 % | Away: 44.18 % )
Partida entre Wales x Ukraine -> Predictions ( Home: 28.43 % | Draw: 29.52 % | Away: 42.05 % )
Ukraine Classificada para o grupo B da Copa do Mundo 2022
```

Na Eliminatória Asiática, a Austrália venceu a seleção de Emirados Árabes Unidos, com chances de 49.00%. Na partida da Repescagem, enfrentou a seleção Peruana e voltou a vencer com chances de 43.91%. A Austrália se classifica e se junta ao Grupo D da Copa

```
Partida entre United Arab Emirates x Australia -> Predictions ( Home: 24.85 % | Draw: 26.15 % | Away: 49.00 % )
Partida entre Australia x Peru -> Predictions ( Home: 43.91 % | Draw: 24.66 % | Away: 31.43 % )
Australia Classificada para o grupo D da Copa do Mundo 2022
```

Costa Rica e Nova Zelândia se enfrentaram pela última vaga da Copa, com vitória da Costa Rica com 52.49% de chances, se classificando assim para a ultima vaga da Copa no Grupo E.

Partida entre Costa Rica x New Zealand -> Predictions (Home: 52.49 % | Draw: 25.83 % | Away: 21.68 %)
 Costa Rica Classificada para o grupo E da Copa do Mundo 2022

6.1.1 - Fase de Grupos

Grupo A:

| | name | wins | losses | draw | seed | points |
|---|-------------|------|--------|------|------|--------|
| 0 | Netherlands | 3.0 | 0.0 | 0.0 | 4 | 9.0 |
| 1 | Ecuador | 2.0 | 1.0 | 0.0 | 2 | 6.0 |
| 2 | Senegal | 1.0 | 2.0 | 0.0 | 3 | 3.0 |
| 3 | Qatar | 0.0 | 3.0 | 0.0 | 1 | 0.0 |

Grupo B:

| | name | wins | losses | draw | seed | points |
|---|---------|------|--------|------|------|--------|
| 0 | England | 3.0 | 0.0 | 0.0 | 1 | 9.0 |
| 1 | Ukraine | 2.0 | 1.0 | 0.0 | 4 | 6.0 |
| 2 | USA | 1.0 | 2.0 | 0.0 | 3 | 3.0 |
| 3 | IR Iran | 0.0 | 3.0 | 0.0 | 2 | 0.0 |

Grupo C:

| | name | wins | losses | draw | seed | points |
|---|--------------|------|--------|------|------|--------|
| 0 | Argentina | 3.0 | 0.0 | 0.0 | 1 | 9.0 |
| 1 | Mexico | 2.0 | 1.0 | 0.0 | 3 | 6.0 |
| 2 | Poland | 1.0 | 2.0 | 0.0 | 4 | 3.0 |
| 3 | Saudi Arabia | 0.0 | 3.0 | 0.0 | 2 | 0.0 |

Grupo D:

| | name | wins | losses | draw | seed | points |
|---|-----------|------|--------|------|------|--------|
| 0 | France | 3.0 | 0.0 | 0.0 | 1 | 9.0 |
| 1 | Denmark | 2.0 | 1.0 | 0.0 | 3 | 6.0 |
| 2 | Australia | 1.0 | 2.0 | 0.0 | 2 | 3.0 |
| 3 | Tunisia | 0.0 | 3.0 | 0.0 | 4 | 0.0 |

Grupo E:

| | name | wins | losses | draw | seed | points |
|---|------------|------|--------|------|------|--------|
| 0 | Germany | 3.0 | 0.0 | 0.0 | 3 | 9.0 |
| 1 | Spain | 2.0 | 1.0 | 0.0 | 1 | 6.0 |
| 2 | Japan | 1.0 | 2.0 | 0.0 | 4 | 3.0 |
| 3 | Costa Rica | 0.0 | 3.0 | 0.0 | 2 | 0.0 |

Grupo F:

| | name | wins | losses | draw | seed | points |
|---|---------|------|--------|------|------|--------|
| 0 | Croatia | 3.0 | 0.0 | 0.0 | 4 | 9.0 |
| 1 | Belgium | 2.0 | 1.0 | 0.0 | 1 | 6.0 |
| 2 | Canada | 0.0 | 2.0 | 1.0 | 2 | 1.0 |
| 3 | Morocco | 0.0 | 2.0 | 1.0 | 3 | 1.0 |

Grupo G:

| | name | wins | losses | draw | seed | points |
|---|-------------|------|--------|------|------|--------|
| 0 | Brazil | 3.0 | 0.0 | 0.0 | 1 | 9.0 |
| 1 | Switzerland | 1.0 | 1.0 | 1.0 | 3 | 4.0 |
| 2 | Serbia | 1.0 | 2.0 | 0.0 | 2 | 3.0 |
| 3 | Cameroon | 0.0 | 2.0 | 1.0 | 4 | 1.0 |

Grupo H:

| | name | wins | losses | draw | seed | points |
|---|----------------|------|--------|------|------|--------|
| 0 | Portugal | 3.0 | 0.0 | 0.0 | 1 | 9.0 |
| 1 | Uruguay | 2.0 | 1.0 | 0.0 | 3 | 6.0 |
| 2 | Korea Republic | 1.0 | 2.0 | 0.0 | 4 | 3.0 |
| 3 | Ghana | 0.0 | 3.0 | 0.0 | 2 | 0.0 |

6.1.2 - Oitavas de Final

Partida entre Netherlands x Ukraine -> Predictions (Home: 45.84 % | Draw: 25.88 % | Away: 28.28 %)
 Partida entre Argentina x Denmark -> Predictions (Home: 46.12 % | Draw: 24.84 % | Away: 29.04 %)
 Partida entre Germany x Belgium -> Predictions (Home: 36.61 % | Draw: 28.13 % | Away: 35.26 %)
 Partida entre Brazil x Uruguay -> Predictions (Home: 55.22 % | Draw: 22.09 % | Away: 22.69 %)
 Partida entre England x Ecuador -> Predictions (Home: 48.92 % | Draw: 26.51 % | Away: 24.57 %)
 Partida entre France x Mexico -> Predictions (Home: 48.90 % | Draw: 22.24 % | Away: 28.87 %)
 Partida entre Croatia x Spain -> Predictions (Home: 26.52 % | Draw: 26.55 % | Away: 46.94 %)
 Partida entre Portugal x Switzerland -> Predictions (Home: 34.91 % | Draw: 31.78 % | Away: 33.31 %)

| | home_team | away_team | neutral | home_rk_points | away_rk_points | winner | points_diff | stage | group_name |
|---|-------------|-------------|---------|----------------|----------------|--------|-------------|-------|------------|
| 0 | Netherlands | Ukraine | 1.0 | 1658.66 | 1535.08 | 1 | 123.58 | 4.0 | ROUND16 |
| 1 | Argentina | Denmark | 1.0 | 1765.13 | 1653.60 | 1 | 111.53 | 4.0 | ROUND16 |
| 2 | Germany | Belgium | 1.0 | 1650.53 | 1827.00 | 1 | -176.47 | 4.0 | ROUND16 |
| 3 | Brazil | Uruguay | 1.0 | 1832.69 | 1635.73 | 1 | 196.96 | 4.0 | ROUND16 |
| 4 | England | Ecuador | 1.0 | 1761.71 | 1452.63 | 1 | 309.08 | 4.0 | ROUND16 |
| 5 | France | Mexico | 1.0 | 1789.85 | 1658.82 | 1 | 131.03 | 4.0 | ROUND16 |
| 6 | Croatia | Spain | 1.0 | 1621.11 | 1709.19 | -1 | -88.08 | 4.0 | ROUND16 |
| 7 | Portugal | Switzerland | 1.0 | 1674.78 | 1635.32 | 1 | 39.46 | 4.0 | ROUND16 |

6.1.3 - Quartas de Final

Partida entre Netherlands x Argentina -> Predictions (Home: 30.45 % | Draw: 26.44 % | Away: 43.11 %)
 Partida entre Germany x Brazil -> Predictions (Home: 23.48 % | Draw: 25.58 % | Away: 50.94 %)
 Partida entre England x France -> Predictions (Home: 26.71 % | Draw: 24.01 % | Away: 49.28 %)
 Partida entre Spain x Portugal -> Predictions (Home: 44.09 % | Draw: 24.45 % | Away: 31.46 %)

| | home_team | away_team | neutral | home_rk_points | away_rk_points | winner | points_diff | stage | group_name |
|---|-------------|-----------|---------|----------------|----------------|--------|-------------|-------|---------------|
| 0 | Netherlands | Argentina | 1.0 | 1658.66 | 1765.13 | -1 | -106.47 | 5.0 | QuarterFinals |
| 1 | Germany | Brazil | 1.0 | 1650.53 | 1832.69 | -1 | -182.16 | 5.0 | QuarterFinals |
| 2 | England | France | 1.0 | 1761.71 | 1789.85 | -1 | -28.14 | 5.0 | QuarterFinals |
| 3 | Spain | Portugal | 1.0 | 1709.19 | 1674.78 | 1 | 34.41 | 5.0 | QuarterFinals |

6.1.4 - Semifinal

Partida entre Argentina x Brazil -> Predictions (Home: 30.78 % | Draw: 23.29 % | Away: 45.93 %)
 Partida entre France x Spain -> Predictions (Home: 34.27 % | Draw: 24.17 % | Away: 41.56 %)

| | home_team | away_team | neutral | home_rk_points | away_rk_points | winner | points_diff | stage | group_name |
|---|-----------|-----------|---------|----------------|----------------|--------|-------------|-------|------------|
| 0 | Argentina | Brazil | 1.0 | 1765.13 | 1832.69 | -1 | -67.56 | 5.0 | SemiFinals |
| 1 | France | Spain | 1.0 | 1789.85 | 1709.19 | -1 | 80.66 | 5.0 | SemiFinals |

6.1.5 - Final

Partida entre Brazil x Spain -> Predictions (Home: 42.90 % | Draw: 22.25 % | Away: 34.85 %)

| | home_team | away_team | neutral | home_rk_points | away_rk_points | winner | points_diff | stage | group_name |
|---|-----------|-----------|---------|----------------|----------------|--------|-------------|-------|------------|
| 0 | Brazil | Spain | 1.0 | 1832.69 | 1709.19 | 1 | 123.5 | 6.0 | Final |

Segundo algoritmo de RidgeClassifier a final será disputada entre Brasil e Espanha, com vitória do Brasil com chance de 42.90%

6.2 LogisticRegression

Partidas das Eliminatórias da Europa, a Ucrânia vence a primeira contra a Escócia, e vence a segunda partida contra País de Gales, com chances de 52.08% e 47.48% respectivamente. Juntando-se ao Grupo B.

Partida entre Scotland x Ukraine -> Predictions (Home: 24.67 % | Draw: 23.25 % | Away: 52.08 %)
 Partida entre Wales x Ukraine -> Predictions (Home: 25.52 % | Draw: 27.00 % | Away: 47.48 %)
 Ukraine Classificada para o grupo B da Copa do Mundo 2022

Na Eliminatória Asiática, a Austrália venceu a seleção dos Emirados Árabes Unidos, com chances de 58.19%. Na partida da Repescagem, enfrentou a seleção Peruana e voltou a vencer com chances de 55.77%. A Austrália se classifica e se junta ao Grupo D da Copa

Partida entre United Arab Emirates x Australia -> Predictions (Home: 19.30 % | Draw: 22.51 % | Away: 58.19 %)
 Partida entre Australia x Peru -> Predictions (Home: 55.77 % | Draw: 20.29 % | Away: 23.94 %)
 Australia Classificada para o grupo D da Copa do Mundo 2022

Costa Rica e Nova Zelândia se enfrentaram pela última vaga da Copa, com vitória da Costa Rica com 61.47% de chances, se classificando assim para a última vaga da Copa no Grupo E.

Partida entre Costa Rica x New Zealand -> Predictions (Home: 61.47 % | Draw: 23.92 % | Away: 14.61 %)
 Costa Rica Classificada para o grupo E da Copa do Mundo 2022

6.2.1 Fase de Grupos

Com os grupos definidos, as partidas da fase de grupo foram realizadas, ficando assim a final da fase de grupos:

Por motivos de legibilidade apenas as partidas da rodada 1 será postada:

Partida entre Qatar x Ecuador -> Predictions (Home: 16.33 % | Draw: 26.61 % | Away: 57.06 %)
 Partida entre Senegal x Netherlands -> Predictions (Home: 8.49 % | Draw: 25.23 % | Away: 66.28 %)
 Partida entre England x IR Iran -> Predictions (Home: 50.99 % | Draw: 30.33 % | Away: 18.68 %)
 Partida entre USA x Ukraine -> Predictions (Home: 40.61 % | Draw: 21.06 % | Away: 38.33 %)
 Partida entre Argentina x Saudi Arabia -> Predictions (Home: 78.97 % | Draw: 14.71 % | Away: 6.32 %)
 Partida entre Mexico x Poland -> Predictions (Home: 39.33 % | Draw: 28.56 % | Away: 32.11 %)
 Partida entre France x Australia -> Predictions (Home: 66.82 % | Draw: 18.44 % | Away: 14.73 %)
 Partida entre Denmark x Tunisia -> Predictions (Home: 40.16 % | Draw: 37.47 % | Away: 22.37 %)
 Partida entre Spain x Costa Rica -> Predictions (Home: 71.66 % | Draw: 17.07 % | Away: 11.27 %)
 Partida entre Germany x Japan -> Predictions (Home: 53.13 % | Draw: 19.97 % | Away: 26.90 %)
 Partida entre Belgium x Canada -> Predictions (Home: 59.51 % | Draw: 18.50 % | Away: 21.98 %)
 Partida entre Morocco x Croatia -> Predictions (Home: 17.84 % | Draw: 25.03 % | Away: 57.13 %)
 Partida entre Brazil x Serbia -> Predictions (Home: 66.77 % | Draw: 19.31 % | Away: 13.93 %)
 Partida entre Switzerland x Cameroon -> Predictions (Home: 31.43 % | Draw: 37.83 % | Away: 30.74 %)
 Partida entre Portugal x Ghana -> Predictions (Home: 60.02 % | Draw: 30.78 % | Away: 9.20 %)
 Partida entre Uruguay x Korea Republic -> Predictions (Home: 44.83 % | Draw: 38.67 % | Away: 16.50 %)
 Partida entre Qatar x Senegal -> Predictions (Home: 31.85 % | Draw: 37.86 % | Away: 50.07 %)

Grupo A:

| | name | wins | losses | draw | seed | points |
|---|-------------|------|--------|------|------|--------|
| 0 | Netherlands | 3.0 | 0.0 | 0.0 | 4 | 9.0 |
| 1 | Ecuador | 2.0 | 1.0 | 0.0 | 2 | 6.0 |
| 2 | Senegal | 1.0 | 2.0 | 0.0 | 3 | 3.0 |
| 3 | Qatar | 0.0 | 3.0 | 0.0 | 1 | 0.0 |

Grupo B:

| | name | wins | losses | draw | seed | points |
|---|---------|------|--------|------|------|--------|
| 0 | England | 3.0 | 0.0 | 0.0 | 1 | 9.0 |
| 1 | USA | 2.0 | 1.0 | 0.0 | 3 | 6.0 |
| 2 | Ukraine | 1.0 | 2.0 | 0.0 | 4 | 3.0 |
| 3 | IR Iran | 0.0 | 3.0 | 0.0 | 2 | 0.0 |

Grupo C:

| | name | wins | losses | draw | seed | points |
|---|--------------|------|--------|------|------|--------|
| 0 | Argentina | 3.0 | 0.0 | 0.0 | 1 | 9.0 |
| 1 | Mexico | 2.0 | 1.0 | 0.0 | 3 | 6.0 |
| 2 | Poland | 1.0 | 2.0 | 0.0 | 4 | 3.0 |
| 3 | Saudi Arabia | 0.0 | 3.0 | 0.0 | 2 | 0.0 |

Grupo D:

| | name | wins | losses | draw | seed | points |
|---|-----------|------|--------|------|------|--------|
| 0 | France | 3.0 | 0.0 | 0.0 | 1 | 9.0 |
| 1 | Denmark | 2.0 | 1.0 | 0.0 | 3 | 6.0 |
| 2 | Australia | 1.0 | 2.0 | 0.0 | 2 | 3.0 |
| 3 | Tunisia | 0.0 | 3.0 | 0.0 | 4 | 0.0 |

Grupo E:

| | name | wins | losses | draw | seed | points |
|---|------------|------|--------|------|------|--------|
| 0 | Spain | 3.0 | 0.0 | 0.0 | 1 | 9.0 |
| 1 | Germany | 2.0 | 1.0 | 0.0 | 3 | 6.0 |
| 2 | Japan | 1.0 | 2.0 | 0.0 | 4 | 3.0 |
| 3 | Costa Rica | 0.0 | 3.0 | 0.0 | 2 | 0.0 |

Grupo F:

| | name | wins | losses | draw | seed | points |
|---|---------|------|--------|------|------|--------|
| 0 | Croatia | 3.0 | 0.0 | 0.0 | 4 | 9.0 |
| 1 | Belgium | 2.0 | 1.0 | 0.0 | 1 | 6.0 |
| 2 | Canada | 0.0 | 2.0 | 1.0 | 2 | 1.0 |
| 3 | Morocco | 0.0 | 2.0 | 1.0 | 3 | 1.0 |

Grupo G:

| | name | wins | losses | draw | seed | points |
|---|-------------|------|--------|------|------|--------|
| 0 | Brazil | 3.0 | 0.0 | 0.0 | 1 | 9.0 |
| 1 | Switzerland | 1.0 | 1.0 | 1.0 | 3 | 4.0 |
| 2 | Serbia | 1.0 | 2.0 | 0.0 | 2 | 3.0 |
| 3 | Cameroon | 0.0 | 2.0 | 1.0 | 4 | 1.0 |

Grupo H:

| | name | wins | losses | draw | seed | points |
|---|----------------|------|--------|------|------|--------|
| 0 | Portugal | 3.0 | 0.0 | 0.0 | 1 | 9.0 |
| 1 | Uruguay | 2.0 | 1.0 | 0.0 | 3 | 6.0 |
| 2 | Korea Republic | 1.0 | 2.0 | 0.0 | 4 | 3.0 |
| 3 | Ghana | 0.0 | 3.0 | 0.0 | 2 | 0.0 |

6.2.2 Oitavas de Final

| Partida entre Netherlands x USA -> Predictions (Home: 54.42 % Draw: 19.61 % Away: 25.97 %) | | | | | | | | | |
|---|-------------|-------------|---------|----------------|----------------|--------|-------------|-------|------------|
| Partida entre Argentina x Denmark -> Predictions (Home: 56.32 % Draw: 22.49 % Away: 21.18 %) | | | | | | | | | |
| Partida entre Spain x Belgium -> Predictions (Home: 50.53 % Draw: 20.75 % Away: 28.71 %) | | | | | | | | | |
| Partida entre Brazil x Uruguay -> Predictions (Home: 69.89 % Draw: 19.03 % Away: 11.08 %) | | | | | | | | | |
| Partida entre England x Ecuador -> Predictions (Home: 57.86 % Draw: 25.36 % Away: 16.79 %) | | | | | | | | | |
| Partida entre France x Mexico -> Predictions (Home: 57.35 % Draw: 18.57 % Away: 24.07 %) | | | | | | | | | |
| Partida entre Croatia x Germany -> Predictions (Home: 23.43 % Draw: 24.08 % Away: 52.49 %) | | | | | | | | | |
| Partida entre Portugal x Switzerland -> Predictions (Home: 37.59 % Draw: 35.12 % Away: 27.28 %) | | | | | | | | | |
| | home_team | away_team | neutral | home_rk_points | away_rk_points | winner | points_diff | stage | group_name |
| 0 | Netherlands | USA | 1.0 | 1658.66 | 1633.72 | 1 | 24.94 | 4.0 | ROUND16 |
| 1 | Argentina | Denmark | 1.0 | 1765.13 | 1653.60 | 1 | 111.53 | 4.0 | ROUND16 |
| 2 | Spain | Belgium | 1.0 | 1709.19 | 1827.00 | 1 | -117.81 | 4.0 | ROUND16 |
| 3 | Brazil | Uruguay | 1.0 | 1832.69 | 1635.73 | 1 | 196.96 | 4.0 | ROUND16 |
| 4 | England | Ecuador | 1.0 | 1761.71 | 1452.63 | 1 | 309.08 | 4.0 | ROUND16 |
| 5 | France | Mexico | 1.0 | 1789.85 | 1658.82 | 1 | 131.03 | 4.0 | ROUND16 |
| 6 | Croatia | Germany | 1.0 | 1621.11 | 1650.53 | -1 | -29.42 | 4.0 | ROUND16 |
| 7 | Portugal | Switzerland | 1.0 | 1674.78 | 1635.32 | 1 | 39.46 | 4.0 | ROUND16 |

6.2.3 - Quartas de Final

| Partida entre Netherlands x Argentina -> Predictions (Home: 28.15 % Draw: 24.19 % Away: 47.67 %) | | | | | | | | | |
|--|-------------|-----------|---------|----------------|----------------|--------|-------------|-------|---------------|
| Partida entre Spain x Brazil -> Predictions (Home: 25.34 % Draw: 19.57 % Away: 55.09 %) | | | | | | | | | |
| Partida entre England x France -> Predictions (Home: 20.74 % Draw: 21.90 % Away: 57.36 %) | | | | | | | | | |
| Partida entre Germany x Portugal -> Predictions (Home: 36.45 % Draw: 29.30 % Away: 34.25 %) | | | | | | | | | |
| | home_team | away_team | neutral | home_rk_points | away_rk_points | winner | points_diff | stage | group_name |
| 0 | Netherlands | Argentina | 1.0 | 1658.66 | 1765.13 | -1 | -106.47 | 5.0 | QuarterFinals |
| 1 | Spain | Brazil | 1.0 | 1709.19 | 1832.69 | -1 | -123.50 | 5.0 | QuarterFinals |
| 2 | England | France | 1.0 | 1761.71 | 1789.85 | -1 | -28.14 | 5.0 | QuarterFinals |
| 3 | Germany | Portugal | 1.0 | 1650.53 | 1674.78 | 1 | -24.25 | 5.0 | QuarterFinals |

6.2.4 - Semifinal

Partida entre Argentina x Brazil -> Predictions (Home: 26.81 % | Draw: 21.39 % | Away: 51.80 %)
 Partida entre France x Germany -> Predictions (Home: 39.65 % | Draw: 20.06 % | Away: 40.29 %)

| | home_team | away_team | neutral | home_rk_points | away_rk_points | winner | points_diff | stage | group_name |
|---|-----------|-----------|---------|----------------|----------------|--------|-------------|-------|------------|
| 0 | Argentina | Brazil | 1.0 | 1765.13 | 1832.69 | -1 | -67.56 | 5.0 | SemiFinals |
| 1 | France | Germany | 1.0 | 1789.85 | 1650.53 | -1 | 139.32 | 5.0 | SemiFinals |

6.2.5 - Final

Partida entre Brazil x Germany -> Predictions (Home: 57.83 % | Draw: 18.16 % | Away: 24.01 %)

| | home_team | away_team | neutral | home_rk_points | away_rk_points | winner | points_diff | stage | group_name |
|---|-----------|-----------|---------|----------------|----------------|--------|-------------|-------|------------|
| 0 | Brazil | Germany | 1.0 | 1832.69 | 1650.53 | 1 | 182.16 | 6.0 | Final |

Segundo algoritmo de LogisticRegression a final será disputada entre Brasil e Alemanha, com o Brasil vencendo com chance de 57.83%

```
print('The Fifa World Cup 2022 - Qatar winner is', get_winner_from_match(grand_final.iloc[0]), '!!!')
The Fifa World Cup 2022 - Qatar winner is Brazil !!!
```


7. Conclusões

Dentre todos os algoritmos comparados, o LogisticRegression e o RidgeClassifier foram os que melhor se adaptaram ao grupo de dados disponibilizados e que poderiam facilmente ser incorporados com outras variáveis preditoras para identificar vencedores de partidas de futebol entre seleções.

Ambos algoritmos apontam uma grande vantagem do Brasil ser vencedor da Copa do Mundo.

8. Atualizações Pós Eliminatórias

Após a realização do estudo, as partidas das eliminatórias Europeia e Asiática e das Repescagens 1 e 2 foram realizadas, 5 partidas ainda faltavam se concretizar e haviam sido simuladas com os algoritmos LogisticRegression e RidgeClassifier, que foram os escolhidos.

8.1 Escócia / Ucrânia / País de Gales

A partida entre Escócia e Ucrânia havia sido adiada por conta da guerra entre Rússia e Ucrânia. Realizada no dia 01/06/2022 a Ucrânia obteve vitória sobre a Escócia por 3x1, fazendo os algoritmos realmente acertarem a previsão.



O mesmo não aconteceu, porém, com a partida entre Ucrânia e País de Gales, enquanto os algoritmos indicavam uma vitória Ucrâniana, e com relativa facilidade, a seleção de País de Gales se sagrou vencedora e ganhou a vaga na Copa do Mundo de 2022.



8.2 Emirados Árabes / Austrália / Peru

Outra eliminatória que tinha sido adiada, foi a eliminatória Asiática, a partida entre Emirados Árabes Unidos enfrentou a Austrália e confirmou o acerto dos dois algoritmos de Machine Learning. A Austrália venceu e seguiu para a repescagem 1, para enfrentar o Peru.



Na repescagem, mais uma vez os algoritmos acertaram a previsão, apesar da partida entre Austrália e Peru ter sido empate, a Austrália se classificou nos pênaltis, e ganhando a vaga para a Copa do Mundo.



8.3 Costa Rica / Nova Zelândia

Mais uma vez os algoritmos se mostraram corretos ao preverem a vitória da Costa Rica sobre a Nova Zelândia. Apesar da alta probabilidade de vitória da Costa Rica (Calculada pelos algoritmos), a partida terminou em 1x0. Porém, vitória é vitória.



9. Conclusão

As partidas faltantes e que foram simuladas pelos algoritmos mostraram uma taxa de acerto de % ou 80%, mostrando que os resultados adquiridos através dos mesmos podem e devem ser levados em consideração como válidos para a análise em busca de um vencedor em partidas de futebol envolvendo seleções.

10. Links

Downloader Ranking Fifa

https://github.com/FabricioGSC/fifa_ranking_repository

Kaggle do Dataset Partidas

<https://www.kaggle.com/datasets/martj42/international-football-results-from-1872-to-2017>

Github do Notebook

https://github.com/FabricioGSC/fifa_world_cup_2022_predictions

Youtube da apresentação

https://youtu.be/atiVg6IW_Go