**Universidade do Minho**
Escola de Engenharia

António Guerra
José Barros

# Análise de Dados para prever o resultado de um jogo de futebol

Mestrado em Engenharia Informática

Unidade Curricular de
Mineração de Dados

junho 2023

# Conteúdo

# Capítulo 1

# Introdução

## 1.1 Contexto

A análise de dados no desporto tem evoluído significativamente nos últimos anos, com o surgimento de novas tecnologias e a disponibilidade de dados detalhados. No futebol, em particular, os clubes e as equipas técnicas têm reconhecido o valor dessa abordagem para obter insights sobre o desempenho dos jogadores, padrões táticos e estratégias vencedoras.

Nesse sentido, a Premier League, como uma das ligas de futebol mais prestigiadas e competitivas do mundo, oferece uma rica fonte de dados históricos para análise. Com mais de duas décadas de informações sobre jogos, equipas, jogadores e estatísticas, é possível extrair conhecimentos valiosos que podem ser aplicados tanto no campo como fora dele.

## 1.2 Motivação

A motivação por trás deste projeto baseia-se na procura por uma vantagem competitiva no mundo das apostas desportivas. O futebol atrai um grande número de entusiastas que desejam transformar o seu conhecimento e intuição em ganhos financeiros. No entanto, prever resultados de jogos de forma consistente é um desafio, especialmente numa liga tão imprevisível como a Premier League.

Ao utilizar dados históricos da Premier League e técnicas avançadas de análise, este projeto procura fornecer uma abordagem mais fundamentada para as apostas desportivas. Acreditamos que a análise de dados pode revelar padrões ocultos, relações estatísticas e tendências relevantes para prever resultados com maior precisão.

## 1.3 Objetivos

O objetivo principal deste projeto é desenvolver um modelo com base na análise de dados dos jogos da Premier League dos últimos 20 anos, capaz de prever resultados futuros com uma taxa de acerto elevada. Este modelo pode ser uma ferramenta valiosa tanto para treinadores, que podem utilizá-lo para tomar decisões estratégicas e táticas, como para apostadores desportivos, que procuram informações fiáveis para realizar as suas apostas.

## 1.4 Trabalho Relacionado

O artigo Herbinet [2018] apresenta uma abordagem abrangente para prever resultados de futebol usando técnicas de aprendizado de máquina. Ao incorporar uma métrica de 'golos esperados' e classificações ofensivas e defensivas, o modelo proposto visa fornecer uma previsão mais precisa e diferenciada dos resultados da partida. A utilização de uma recolha extensa de dados e o foco nos eventos durante o jogo demonstram um compromisso em melhorar a eficácia dos modelos de previsão.

O artigo Prasetio et al. [2016] apresenta um modelo para previsão de partidas de futebol utilizando regressão logística e destaca a importância de variáveis como *Home Offense*, *Home Defense*, *Away Offense*, e *Away Defense*. Os autores também apresentam o software Football Predictor, que permite aos utilizadores personalizar o modelo de previsão adicionando os seus próprios dados de treino. Embora os experimentos e resultados específicos não sejam fornecidos nos textos fornecidos, o artigo conclui com uma perspetiva positiva sobre a utilização da regressão logística na previsão dos resultados das partidas de futebol.

# Capítulo 2

# Metodologia

## 2.1 Fontes de dados

Os dados para este estudo foram extraídos do site Football-Data.co.uk, que é uma plataforma de dados relacionados com futebol. O site oferece diversos datasets para cada season do campeonato de futebol inglês, que foi o foco do nosso trabalho. A escolha deste site foi baseada na relevância e facilidade em obter os dados fornecidos. Não usamos nenhum dataset do kaggle como tinha sido erradamente dito na primeira apresentação

## 2.2 Processamento dos dados

Iniciamos o nosso trabalho com os dados em formato raw. Foi preciso uniformizar os dados para garantir a consistência ao longo de todas as seasons. Começamos por remover as colunas 'DIV', que eram redundantes já que todos os jogos se referiam à primeira divisão inglesa, e 'attendance', 'hhw' (home hit woodwork / acertou na barra), 'ahw' (away hit woodwork), 'offsides', que estavam presente apenas nos dois primeiros datasets. Também foi removida a coluna 'hora do jogo', visto ser irrelevante para o nosso objetivo de prever o resultado de jogos de futebol.

Nesta etapa, procedemos à manipulação dos dados através da definição de várias funções para extrair informações pertinentes dos nossos dataframes. As funções desenvolvidas incluíram:

- `calculate_goals(df)`: Esta função foi criada para calcular os golos marcados e sofridos por cada equipa.
- `calculate_points(df)`: Esta função tinha como objetivo calcular os pontos obtidos por cada equipa.
- `calculate_positions(df)`: Esta função foi usada para determinar a posição de cada equipa.

- `games_played_between(df, team1, team2)`: Esta função foi criada para calcular o número de jogos disputados entre duas equipas.

- `_5_ultimos_jogos_entre_equipes(equipe1, equipe2, data_limite)`: Esta função tinha como objetivo determinar os resultados dos últimos cinco jogos entre duas equipas até uma data limite específica.

- `_5_ultimos_jogos_uma_team(equipe1, data_limite)`: Esta função teve como objetivo principal determinar os resultados dos últimos cinco jogos de uma equipa específica até uma data limite.

- `calculate_h2hString_and_last5GameString_andPoints(df)`: Esta função é a mais potente e calcula uma variedade de estatísticas e resultados recentes para cada jogo no dataframe fornecido. Para cada jogo no dataframe, a função faz o seguinte:

  Identifica as equipas que jogam em casa e fora e a data do jogo. Obtém os últimos cinco jogos entre as duas equipas (head-to-head, ou H2H), bem como os últimos cinco jogos de cada equipa, independentemente do oponente. Calcula várias estatísticas para esses conjuntos de jogos, incluindo o número de golos marcados e sofridos, os pontos ganhos, e os resultados dos jogos (vitória, derrota ou empate). Determina estatísticas adicionais, como a diferença de golos, o número de cartões amarelos e vermelhos e outras estatísticas como remates, remates à baliza, faltas e cantos. Adiciona todas essas estatísticas ao dataframe original como novas colunas, para cada jogo.

Após a etapa de limpeza e processamento inicial,as funções acima foram aplicadas a cada dataframe de época usando o seguinte ciclo:

```
for season in datasetsPARAmanipular.keys():
    print(f"a comecar {season}")
    datasetsPARAmanipular[season] = calculate_goals(datasetsPARAmanipular[season])
    datasetsPARAmanipular[season] = calculate_points(datasetsPARAmanipular[season])
    datasetsPARAmanipular[season] = calcula_posicao(datasetsPARAmanipular[season])
    datasetsPARAmanipular[season] = calculate_h2hString_and_last5GameString_andPoints
```

Assim, ao final deste processo, cada dataframe contém uma grande quantidade de informações calculadas com base no desempenho recente de cada equipa. Estas estatísticas podem ser utilizadas na análise subsequente para ajudar a prever o resultado de futuros jogos.

Depois os dataframes individuais foram reunidos num único dataframe. Isto foi conseguido através da concatenação dos dataframes. O código para este processo é apresentado a seguir:

```
dataframes = []
for ano in datasetsPARAmanipular:
    dataframe = datasetsPARAmanipular[ano]
    dataframes.append(dataframe)

combined_df = pd.concat(dataframes)
```

Este processo resultou num dataframe combinado, que reúne todos os dados de todas as épocas num único local. Este dataframe consolidado é mais fácil de manipular e oferece uma visão mais completa do conjunto de dados como um todo.

O dataframe final foi então exportado para um ficheiro CSV para uso posterior. Isto foi feito com o seguinte comando:

```
combined_df.to_csv('GIGAset.csv', index=False)
```

Esta exportação permite que o dataframe seja facilmente carregado para futuras sessões de análise, evitando a necessidade de repetir os passos de pré-processamento.

## 2.3 Desenvolvimento

Ao preparar os dados para a análise, o primeiro passo foi remover todas as colunas relacionadas com apostas. Esta decisão foi tomada com base na nossa avaliação inicial de que as estatísticas disponíveis no conjunto de dados seriam suficientes para a previsão dos resultados dos jogos. Tal como os professores indicaram, este podia ter sido um caminho interessante a seguir, mas que não exploramos por inicialmente termos pensado que as estatísticas do nosso gigaset fossem suficientes para prever os futuros resultados.

Após a remoção das colunas de apostas, procedemos à construção da matriz de correlação para identificar e eliminar as colunas que estavam fortemente correlacionadas.

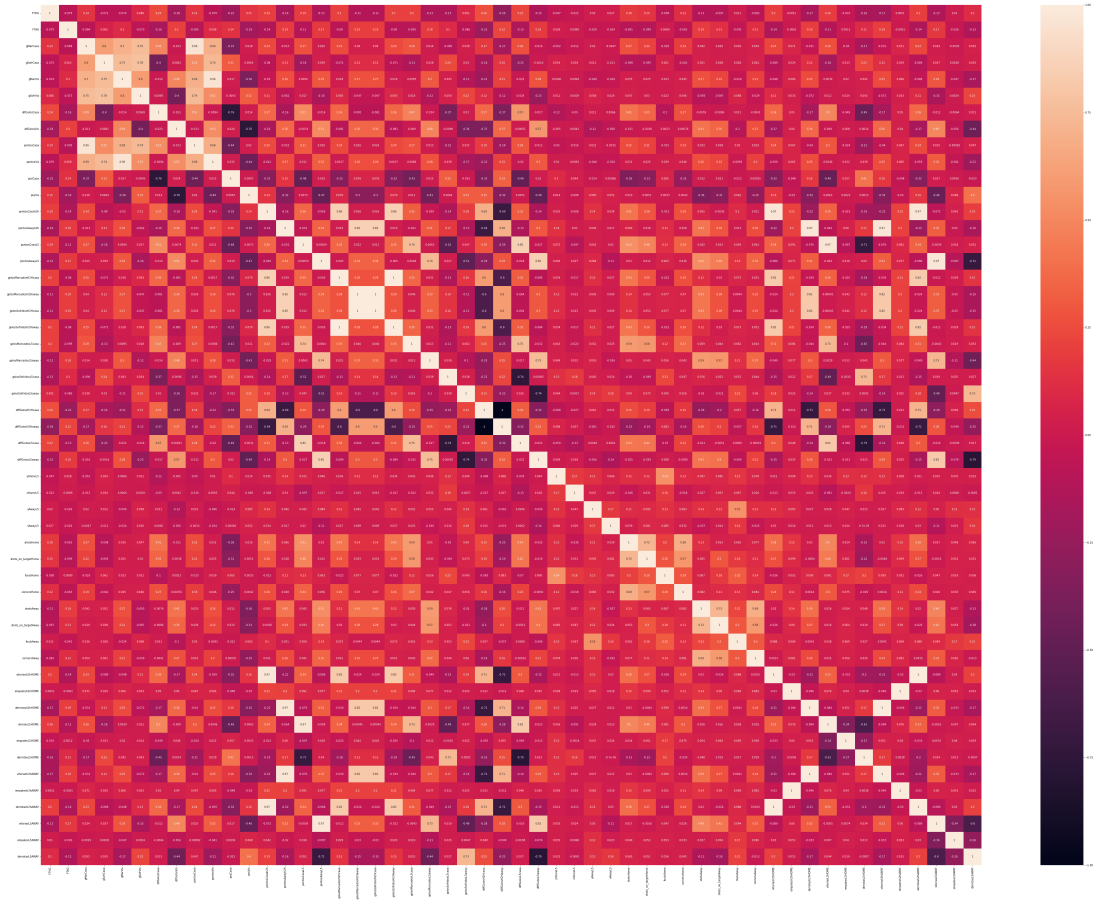

Figura 1: Matriz de correlação antes de remover colunas

```
gMarCasa e pontosCasa: 0.9588939404416029
gMarVis e pontosVis: 0.9594637046344794
pontosCasah2h e golosMarcadosH2Hcasa: 0.8571791893255722
pontosCasah2h e golosSofridosH2Haway: 0.8571791893255722
pontosCasah2h e vitoriash2hHOME: 0.9692411495427115
pontosCasah2h e derrotash2hAWAY: 0.9692411495427115
pontosAwayh2h e golosMarcadosH2Haway: 0.852380065376603
pontosAwayh2h e golosSofridosH2Hcasa: 0.852380065376603
pontosAwayh2h e derrotash2hHOME: 0.9692432280419339
pontosAwayh2h e vitoriash2hAWAY: 0.9692432280419339
pontosCasaL5 e vitoriasL5HOME: 0.9686453253568388
pontosAwayL5 e vitoriasL5AWAY: 0.9693138386727392
golosMarcadosH2Hcasa e golosSofridosH2Haway: 1.0
golosMarcadosH2Haway e golosSofridosH2Hcasa: 1.0
diffGolosH2Hcasa e diffGolosH2Haway: -1.0
vitoriash2hHOME e derrotash2hAWAY: 1.0
empatesh2hHOME e empatesh2hAWAY: 1.0
derrotash2hHOME e vitoriash2hAWAY: 1.0
```

Figura 2: Print das colunas com mais do que 0.85 ou -0.85 de correlação

A correlação é uma medida que descreve o grau de relação entre duas variáveis. A existência de correlações fortes entre as características pode levar à redundância de informação, que pode tornar alguns modelos menos eficientes ou menos precisos.

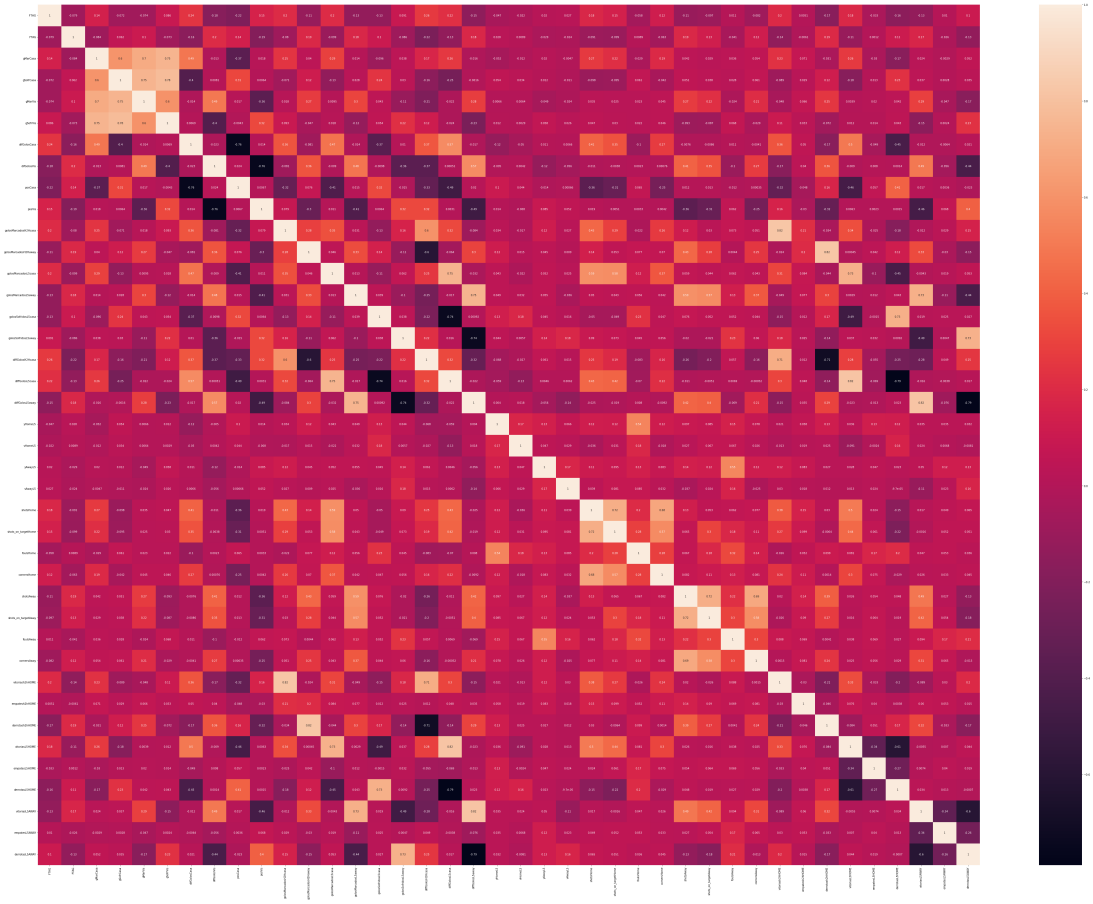

Figura 3: Matriz de correlação depois de remover colunas

Após a limpeza e preparação dos dados, passamos para o treino dos modelos de Machine Learning. O target selecionado para este projeto foi FTR (Full Time Result), que representa o resultado final de cada jogo. Exploramos modelos como Regressão Logística e Random Forest.

```
Num de features: 39

Features: ['FTR', 'gMarCasa', 'gSofrCasa', 'gMarVis', 'gSofrVis', 'difGolosCasa', 'difGolosVis',
'posCasa', 'posVis', 'golosMarcadosH2Hcasa', 'golosMarcadosH2Haway', 'golosMarcadosL5casa',
'golosMarcadosL5away', 'golosSofridosL5casa', 'golosSofridosL5away', 'diffGolosH2Hcasa',
'diffGolosL5casa', 'diffGolosL5away', 'yHomeL5', 'vHomeL5', 'yAwayL5', 'vAwayL5', 'shotsHome',
'shots_on_targetHome', 'foulsHome', 'cornersHome', 'shotsAway', 'shots_on_targetAway', 'foulsAway',
'cornersAway', 'vitoriash2hHOME', 'empatesh2hHOME', 'derrotash2hHOME', 'vitoriasL5HOME', 'empatesL5HOME',
'derrotasL5HOME', 'vitoriasL5AWAY', 'empatesL5AWAY', 'derrotasL5AWAY']


Accuracy: 53.06004618937644
```

Figura 4: Features usadas

```python
# modelo Random Forest
model = RandomForestClassifier()

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

# Calcular a Accuracy do modelo
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy * 100)
```

Após treinar e prever com ambos os modelos, optou-se por tentar um modelo de Random Forest com múltiplas variáveis, tendo como alvo Home Goals e Away Goals em vez de apenas considerar o resultado final do jogo.

```
Num de features: 40

Features: ['FTHG', 'FTAG', 'gMarCasa', 'gSofrCasa', 'gMarVis', 'gSofrVis', 'difGolosCasa', 'difGolosVis',
'posCasa', 'posVis', 'golosMarcadosH2Hcasa', 'golosMarcadosH2Haway', 'golosMarcadosL5casa',
'golosMarcadosL5away', 'golosSofridosL5casa', 'golosSofridosL5away', 'diffGolosH2Hcasa',
'diffGolosL5casa', 'diffGolosL5away', 'yHomeL5', 'vHomeL5', 'yAwayL5', 'vAwayL5', 'shotsHome',
'shots_on_targetHome', 'foulsHome', 'cornersHome', 'shotsAway', 'shots_on_targetAway', 'foulsAway',
'cornersAway', 'vitoriash2hHOME', 'empatesh2hHOME', 'derrotash2hHOME', 'vitoriasL5HOME', 'empatesL5HOME',
'derrotasL5HOME', 'vitoriasL5AWAY', 'empatesL5AWAY', 'derrotasL5AWAY']


MSE for 'FTHG': 1.5181701308699
MSE for 'FTAG': 1.2613294090646652
```

Figura 5: Features usadas Multivariavel

```
#RandomFOREST


# modelo Random Forest
model = RandomForestClassifier()

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

# Calcular a Accuracy do modelo
#accuracy = accuracy_score(y_test, y_pred)
#print("Accuracy:", accuracy * 100)



from sklearn.metrics import mean_squared_error



# Calcula o MSE para cada coluna
mse_fthg = mean_squared_error(y_test['FTHG'], y_pred[:, 0])
mse_ftag = mean_squared_error(y_test['FTAG'], y_pred[:, 1])

print(f"MSE for 'FTHG': {mse_fthg}")
print(f"MSE for 'FTAG': {mse_ftag}")
```

```
# Separar o dataframe em features e target
X = dataset_final_pmanipular.drop(['FTHG', 'FTAG'], axis=1)
y1 = dataset_final_pmanipular['FTHG']
y2 = dataset_final_pmanipular['FTAG']


X_encoded = pd.get_dummies(X)


X_train, X_test, y1_train, y1_test, y2_train, y2_test = train_test_split(X_encoded, y1, y2, test_size=0.2, random_state=42)


# num de features
num_features = dataset_final_pmanipular.shape[1]

# lista de features
feature_list = dataset_final_pmanipular.columns.tolist()

print("Num de features:", num_features)
print("\nFeatures:", feature_list)
print("\n")


# modelos Random Forest
model1 = RandomForestRegressor()
model2 = RandomForestRegressor()

# Treinar os modelos
model1.fit(X_train, y1_train)
model2.fit(X_train, y2_train)

# Previsões
y1_pred = model1.predict(X_test)
y2_pred = model2.predict(X_test)

# Calcular o MSE para cada modelo
mse_fthg = mean_squared_error(y1_test, y1_pred)
mse_ftag = mean_squared_error(y2_test, y2_pred)

print(f"MSE for 'FTHG': {mse_fthg}")
print(f"MSE for 'FTAG': {mse_ftag}")
```

# Capítulo 3

# Conclusões e trabalho futuro

Este trabalho apresentou o desenvolvimento e a avaliação de modelos de previsão de resultados de jogos de futebol, usando Logistic Regression e a Random Forest. O nosso target de previsão foram os golos marcados pelas equipas de casa e fora, bem como o resultado final do jogo.

A performance dos modelos foi, no entanto, moderada. A melhor accuracy alcançada nos nossos testes não ultrapassou os 55 por cento. Isto significa que, apesar de nossos esforços para preparar e ajustar os modelos com nossos dados, há um limite para o quão preciso um modelo de previsão pode ser quando se trata de prever os resultados do futebol com base nas estatísticas dos jogos anteriores. Não vamos ficar ricos com este projecto infelizmente.

No que diz respeito à previsão dos golos marcados, a MSE não baixou de 1.5 e 1.2. Isto significa que, em média, nossas previsões estavam erradas mais ou menos por 1.22 a 1.10 golos. Mesmo que estes números pareçam baixos, eles podem ter um impacto enorme em contextos como as apostas desportivas, onde a precisão é de extrema importância.

Para trabalhos futuros pode ser útil considerar outros tipos de dados ou características que não foram incluídas neste estudo. Por exemplo, informações sobre lesões, jogadores ou por exemplo factores psicológicos que podem talvez melhorar a precisão das previsões.

Também poderia ser benéfico explorar outros tipos de modelos ou usar técnicas de deep Learning talvez.

# Bibliografia

Corentin Herbinet. Predicting football results using machine learning techniques. 2018.

Darwin Prasetio et al. Predicting football match results with logistic regression. In *2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA)*, pages 1–5. IEEE, 2016.