Predicting the Premier League

Jake Haggard

Goal

The goal of the project is to accurately predict results using data derived from past seasons of the English Premier League.

Prediction Model Pipeline

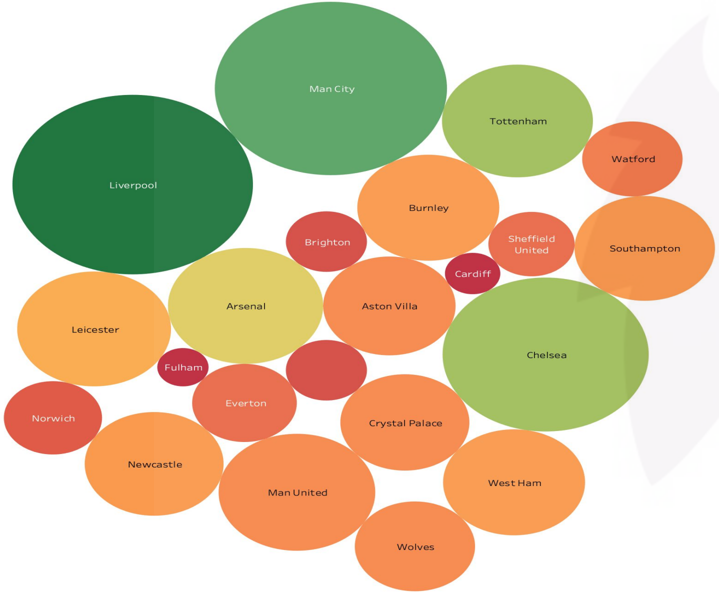
- ETLT - Extrating, Loading, and Transforming the data
- Exploring - Exploring characteristics of the data to get a better understanding of it
- Fitting the model - Fitting the data to the models
- Model Evaluation - Evaluating the effectiveness of the model
- Insights - Building insights from the most effective models

ETLT

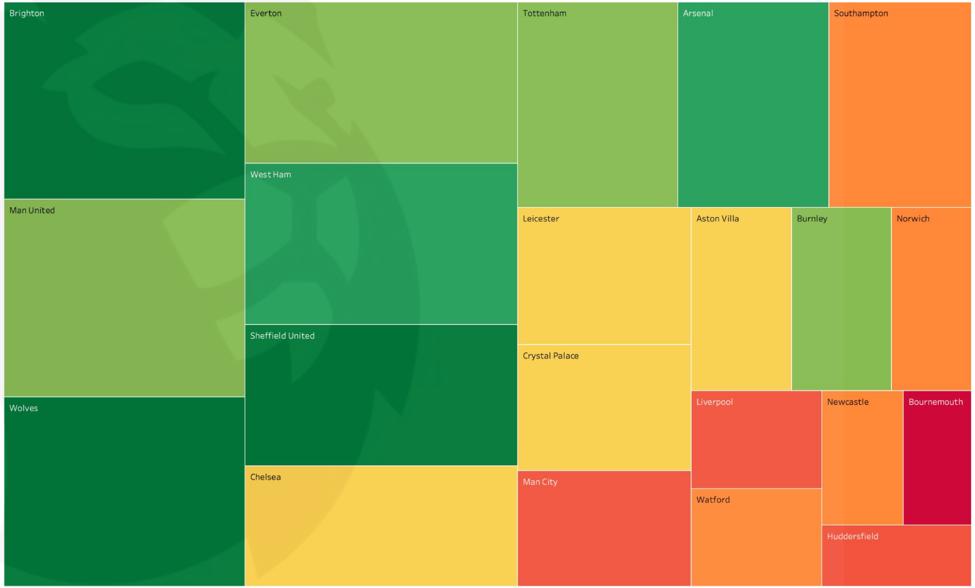
- The data was extracted from football-data.co.uk
- Transformed variables such as
  - Season Encoding - Added a variable for the four seasons
  - Team Encoding - Given each team a numerical ID
  - Year of Season - Year of the following season
  - Full Time Result Encoding - Variable dedicated in explaining the result of the game

Exploring

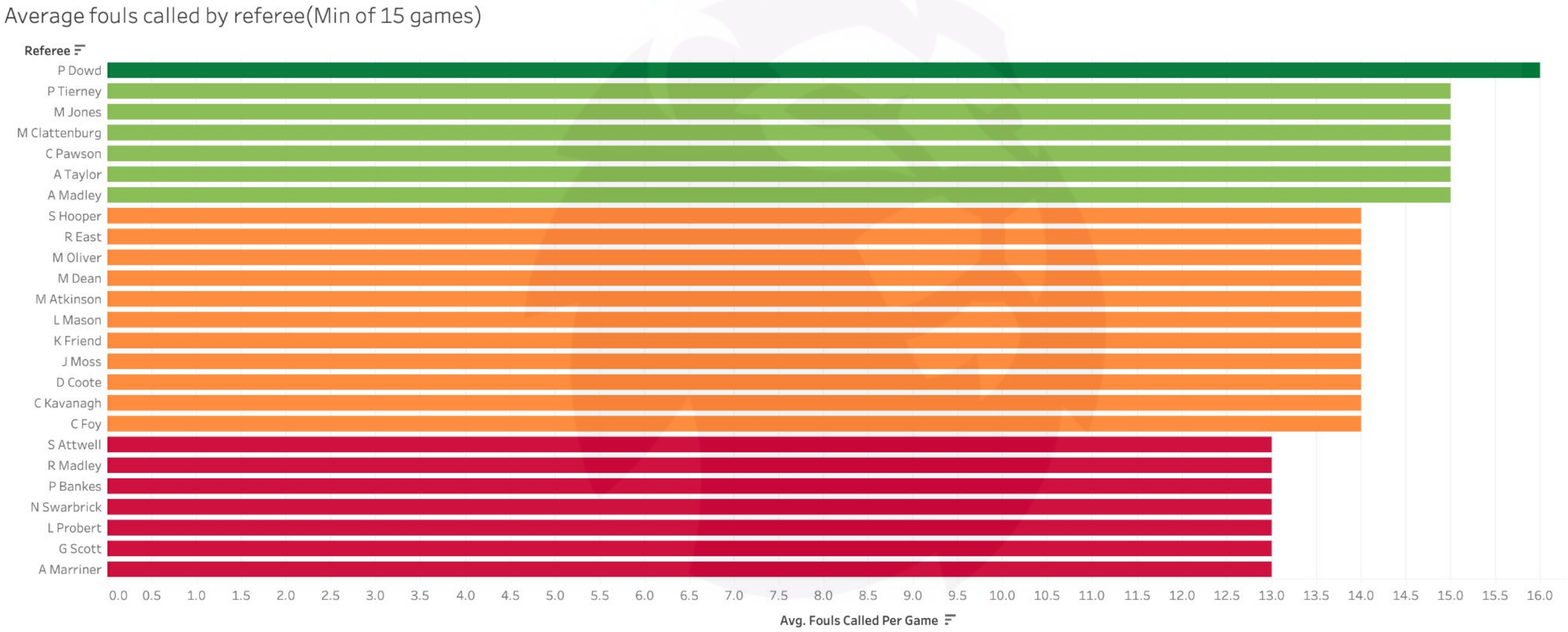
Relationship between shots on target from home and results



Relationship between shots on target from away side and results

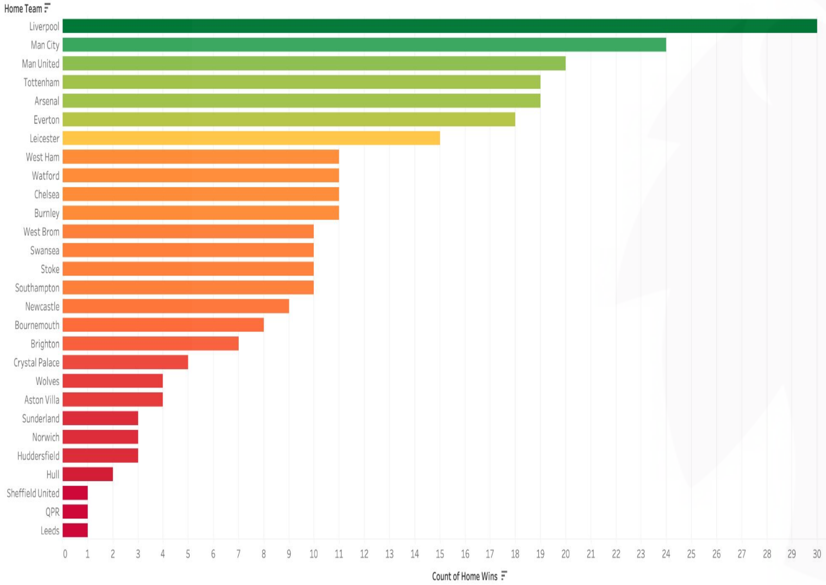


Exploring

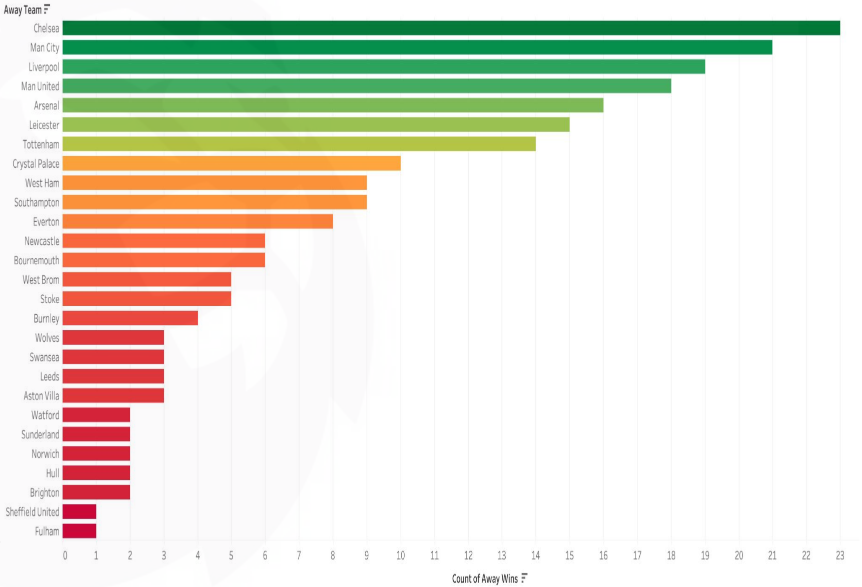

Average fouls called by referee(Min of 15 games)
Referee
P Dowd
P Tierney
M Jones
M Clattenburg
C Pawson
A Taylor
A Madley
S Hooper
R East
M Oliver
M Dean
M Atkinson
L Mason
K Friend
J Moss
D Coote
C Kavanagh
C Foy
S Attwell
R Madley
P Bankes
N Swarbrick
L Probert
G Scott
A Marriner
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
9.0
9.5
10.0
10.5
11.0
11.5
12.0
12.5
13.0
13.5
14.0
14.5
15.0
15.5
16.0
Avg. Fouls Called Per Game

Exploring

Home wins when more freq. foul calling referees officiating



Away wins when more freq. foul calling referees officiating



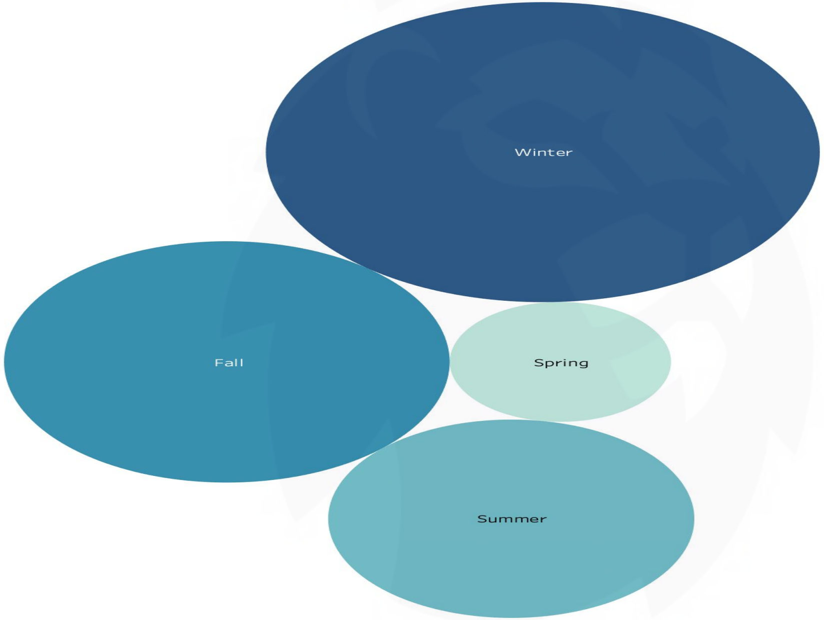
Exploring

Home Team & Seasons

Man United
Newcastle
Sheffield United
Liverpool
Brighton
Burnley
Southampton
Leicester
Wolves
Arsenal
Aston Villa
Chelsea
Leeds
West Ham
Fulham
Crystal Palace
Tottenham
Norwich
Everton
Watford
Man City
West Brom

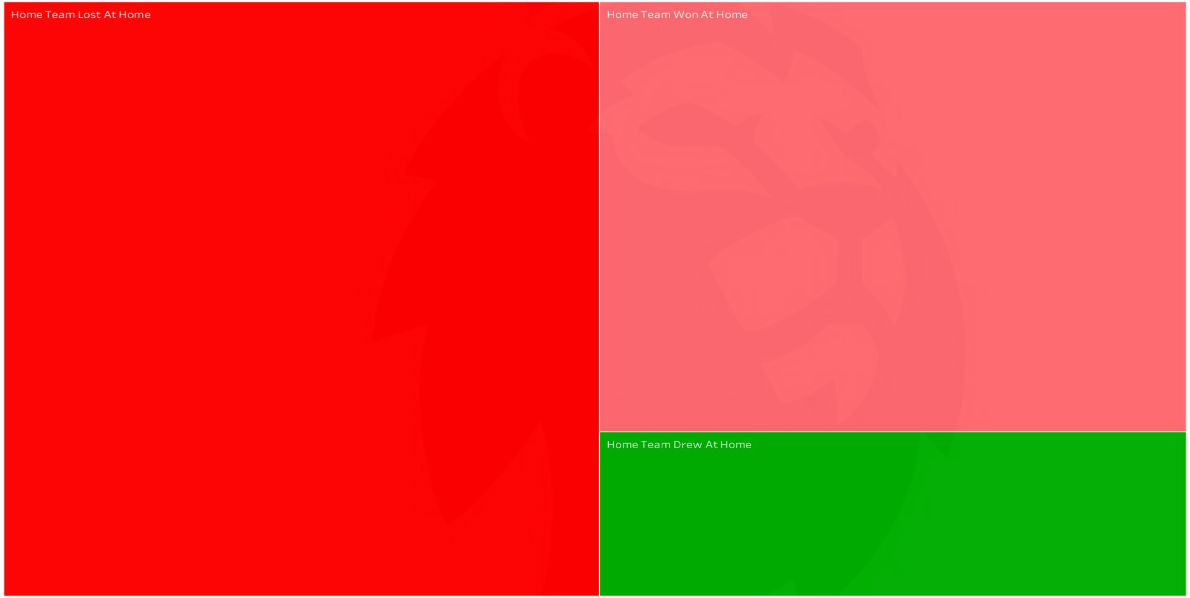
Exploring

Time of Day



Exploring

Home Team Aggression



Fitting Models

Created multiple models that consist of:

- Logistic Regression
- Decision Tree
- Random Forest
- K Nearest Neighbor
- Neural Network
- Ensemble Method
- SVM
- Etc.

The Model

The best performing model of the final models was the Random Forest model

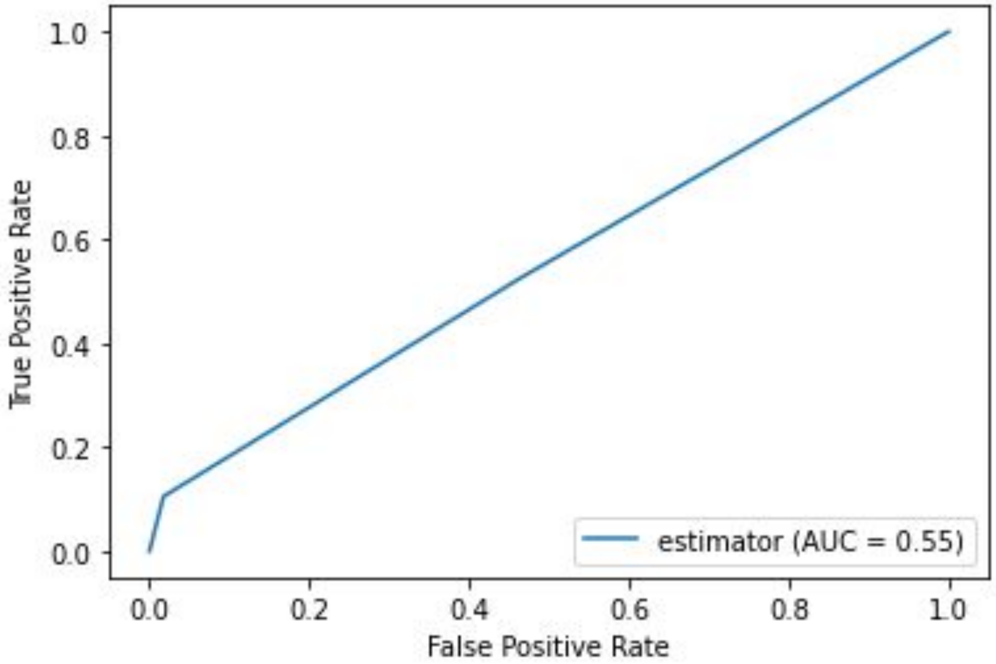
- N Estimators: 10
- Min Samples Split: 3
- Max Features: 10
- Max Depth: 5

Model Evaluation

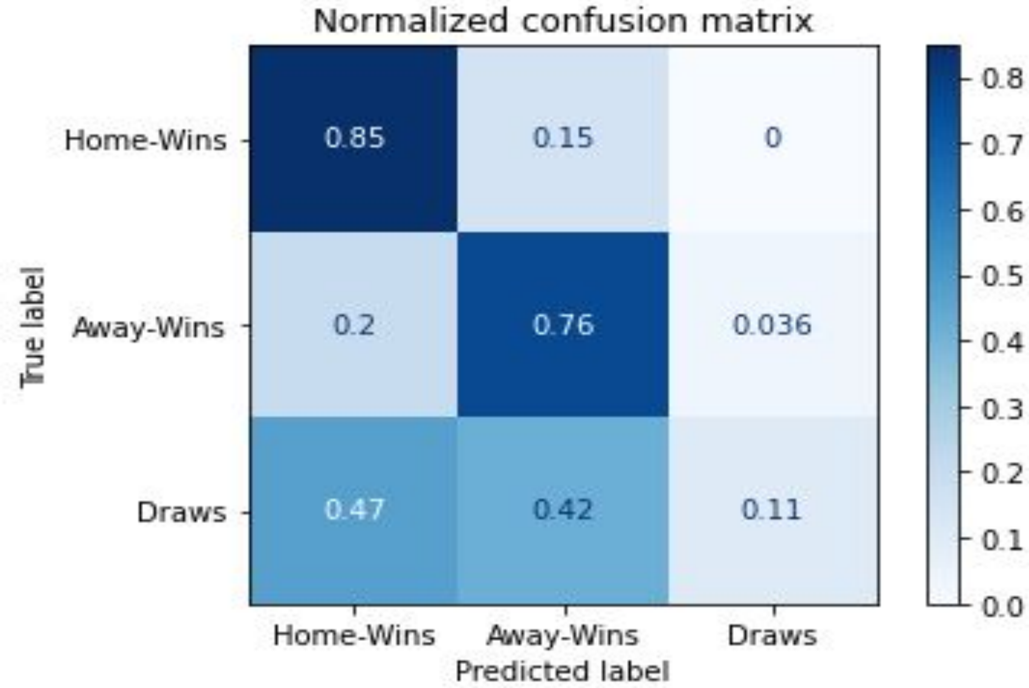
Accuracy Score: 63.15789%

	Precision	Recall	F1-Support	Support
0 (Home Win)	0.63	0.85	0.72	59
1 (Away Win)	0.63	0.76	0.69	55
2 (Draw)	0.67	0.11	0.18	38
Accuracy			0.63	152
Macro Avg.	0.64	0.57	0.53	152
Weighted Avg.	0.64	0.63	0.58	152

Model Evaluation



Model Evaluation



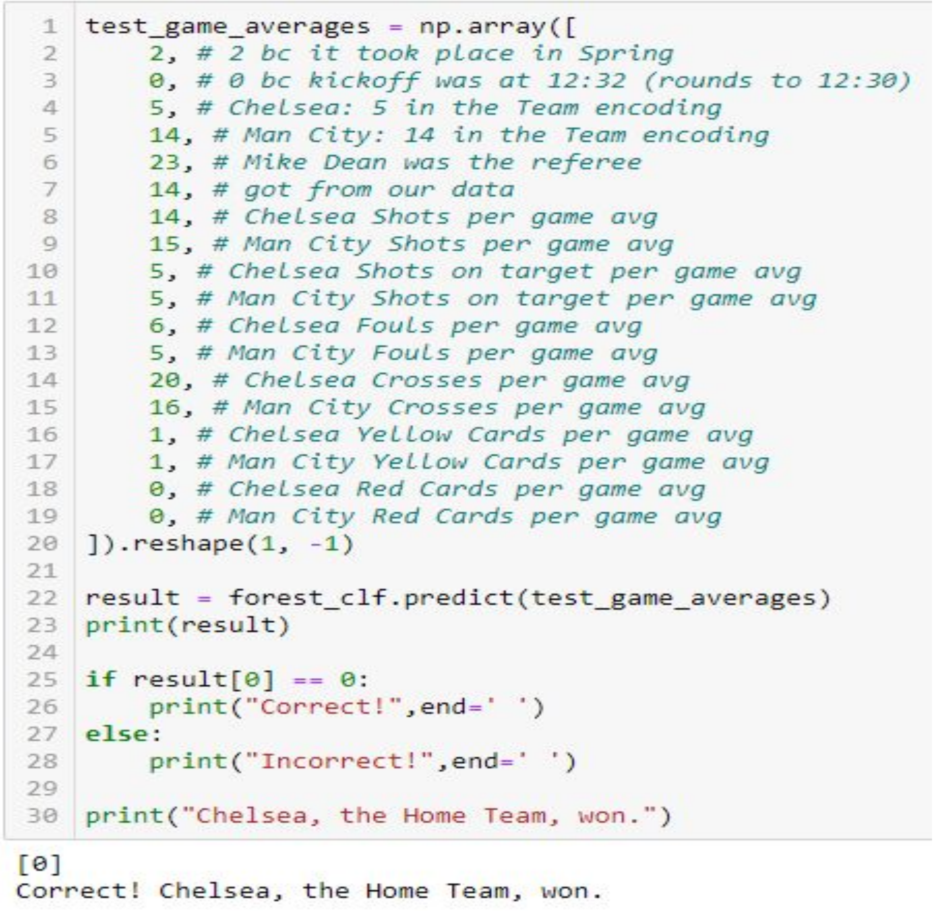
Insights

Insights that were recovered from the analysis explains that through the model it was able to predict home and away wins with an accuracy of 80.5%. An outstanding number only to be dampened by the accuracy of predicting draws. After completing this project, I'd issue the following recommendations:

- Only wins are more reliant than when the model predicts draws.
- Further research need to be done to introduce more variables to help the accuracy of the model predictions.
  - Eg: Managers, Formation, Team's Form, etc.

Use Case

- I designed this model to allow people use team's averages to predict future games results.



```
test_game_averages = np.array([
      2, # 2 bc it took place in Spring
      0, # 0 bc kickoff was at 12:32 (rounds to 12:30)
      5, # Chelsea: 5 in the Team encoding
      14, # Man City: 14 in the Team encoding
      23, # Mike Dean was the referee
      14, # got from our data
      14, # Chelsea Shots per game avg
      15, # Man City Shots per game avg
      5, # Chelsea Shots on target per game avg
      5, # Man City Shots on target per game avg
      6, # Chelsea Fouls per game avg
      5, # Man City Fouls per game avg
      20, # Chelsea Crosses per game avg
      16, # Man City Crosses per game avg
      1, # Chelsea Yellow Cards per game avg
      1, # Man City Yellow Cards per game avg
      0, # Chelsea Red Cards per game avg
      0, # Man City Red Cards per game avg
]).reshape(1, -1)

result = forest_clf.predict(test_game_averages)
print(result)

if result[0] == 0:
      print("Correct!",end=' ')
else:
      print("Incorrect!",end=' ')

print("Chelsea, the Home Team, won.")
```

```
[0]
Correct! Chelsea, the Home Team, won.
```