A logo for college computing

Description automatically generated

**Assessment Cover Page**

|  |  |
| --- | --- |
| *Student Full Name* | David Mooney |
| *Student Number* | sbs24066 |
| *Module Title* | Strategic Thinking |
| *Assessment Title* | CA2 - Can data be used to gain a statistical advantage in sports betting? |
| *Assessment Due Date* | 19/05/24 |
| *Date of Submission* | 18/05/24 |

**Declaration**

By submitting this assessment, I confirm that I have read the CCT policy on academic misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source.

I declare it to be my own work and that all material from third parties has been appropriately referenced.

I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Table of Contents

[Introduction 3](#_Toc166884211)

[Objectives 3](#_Toc166884212)

[**Market Identification:** 3](#_Toc166884213)

[**Create Predictive Models** 3](#_Toc166884214)

[**Testing Models** 3](#_Toc166884215)

[**Applying Other Metrics** 4](#_Toc166884216)

[**Develop Betting Strategy** 4](#_Toc166884217)

[Problem Definition 4](#_Toc166884218)

[Potential Challenges 4](#_Toc166884219)

[**Ensuring the data I acquire is the most up to date and accurate**: 4](#_Toc166884220)

[**Adhering to the CRISP-DM workflow protocol:** 4](#_Toc166884221)

[**Finding and circumventing bias:** 4](#_Toc166884222)

[Scope 5](#_Toc166884223)

[Semester 1 5](#_Toc166884224)

[Semester 2 5](#_Toc166884225)

[Possible Data Sources 5](#_Toc166884226)

[Ethical Considerations 6](#_Toc166884227)

[CA2 – Examining A Dataset and building a Machine Learning Model 8](#_Toc166884228)

[Project Management And Planning 8](#_Toc166884229)

[Market Identification 8](#_Toc166884230)

[Statistical Analysis 9](#_Toc166884231)

[EDA & Descriptive Statistics 11](#_Toc166884232)

[Correlation Map 12](#_Toc166884233)

[Feature Engineering 13](#_Toc166884234)

[Machine Learning Implementation and Evaluation 14](#_Toc166884235)

[Model Implementation 14](#_Toc166884236)

[Random Forest on Match Result 14](#_Toc166884237)

[Linear Regression on Total Expected Goals 17](#_Toc166884238)

[References 20](#_Toc166884239)

\**word count - 2231(not including all tables and headings).*

# Introduction

Sports betting has always been a prominent past time in Ireland. The first gambling laws were established here in 1926 and after several amendments, most recently in 2015, all forms of regulated sports betting are now legal. *(Brent, H., 2024)* According to (*Health Research Board, August, 2023),* online gambling in Ireland has increased by 300% since 1998.

It’s no secret the house always wins when it comes to gambling. Book makers have structured their markets such that they will always have an advantage over the punter. This advantage is known as bookie edge. (*TryPod, 2018)*. Book maker edge is expressed as a percentage and represents the amount of profit made by the bookmaker over the length of time the punter is betting on a particular market. *(TryPod, 2018)*.

My intention from the outset of this project is to identify particular markets where the book maker edge is the lowest such as over/under goals per game. The area in particular I will focus on is football. Using historical football data, I will analyse patterns and hopefully make predictions about upcoming games and potential goals scored. By studying a team’s goal scored and goals conceded I will be able to calculate the expected goals per game. Originally, I use the English Premier League as there is an abundance of data available relating to it but moving forward I intend to apply my algorithm to different football league and maybe sports.

*“According to a December 2022 report from Variety Intelligence, 56% of gamblers said betting is entertaining and 42% determined it makes games more exciting.”* (*Chris Bumbaca, 2024)*

This project aims to enhance this entertainment by making it profitable.

# Objectives

The following objectives will be achieved upon completion of this project:

**Market Identification:**  Research and identify particular markets where the bookies edge is at its lowest. Knowing this will give me a better insight into what kind of data I will need to gather.

**Create Predictive Models:** Applying machine learning algorithms will enable me to develop predictive models. In their implementation, these models will attempt to predict match results and individual player stats. They will give me an insight into team performances while also casting light on how certain teams perform against other particular teams.

**Testing Models:** Ensure the model is performing accurately by acquiring the F1 score. Once the model is adequately tested, new data will be applied to ensure it can retain accuracy.

**Applying Other Metrics:** Extracting other metrics from the data, such as expected goals, will provide a more in depth analysis of how each team performs on a weekly basis. These metrics will help create an estimation of how many goals a team should score and/or concede, thus aiding in the prediction of the outcome.

**Develop Betting Strategy:** To ensure profitability, I will develop a betting strategy based off the results of the data acquired. This will most likely come in the form of a spreadsheet that details stake size, edge percentage of particular markets and market fluctuations. \**I will not be doing any real gambling. All bets will be fictional.*

# Problem Definition

For me to gain an edge over the bookie I will have to find value bets . A value bet is when an event, such a team winning or a certain amount of goals to be scored, has a better chance of occurring than the book makers odds reflect. *(Sacha Alche, 2024)* Using player and team data to calculate expected goals and overall team performance, I will be able to calculate my own odds of that particular event occurring. By comparing my interpreted odds with that of the book maker, I will hopefully be able to identify scenarios where my algorithms odds are greater than the bookies odds, thus generating value and shifting the edge in my favour. Odds in this case refers to the probability of the event occurring.

I will create an excel spreadsheet to document my findings and the book makers findings on a particular event. As I will not be placing any real bets. Instead, I will use a column to display fictional bets to see if my algorithm generates profit over time.

## Potential Challenges

**Ensuring the data I acquire is the most up to date and accurate**: so my predicted probabilities are true. This may require using a data scraping API to obtain the most recent information.

**Adhering to the CRISP-DM workflow protocol:** Applying the six main CRISP-DM headings will ensure I am on task while maintaining the integrity of the project.

**Finding and circumventing bias:** Use only the data to draw any conclusions made about particular events. Don’t allow outside opinion affect any decision making based around the project.

# Scope

The project will take place over two semesters with the workload spread evenly throughout. The following topics will be explored ion each semester:

### Semester 1

* Establishing markets and which leagues to build project around.
* Collect all the relevant data.
* Strategise how best to extract the appropriate data.
* Begin building models to predict certain outcomes.
* Undergo testing determine accuracy.
* Perform evaluation.

### Semester 2

* Continue model experimentation and begin testing
* Acquire further data
* Provide other statistical approaches to predicting outcomes in football matches
* Formalise model selection
* Implementation of strategy.
* Begin making fictional bets.
* Develop spreadsheet to document all fictional wins and losses.

### Possible Data Sources

|  |  |  |
| --- | --- | --- |
| Name | Source | Permission |
| Premier League Matches 1993 - 2023 | <https://www.kaggle.com/datasets/evangower/premier-league-matches-19922022> | Open Source |
| European Soccer Database | <https://www.kaggle.com/datasets/hugomathien/soccer> | Open Source |
| Historical Results/ Betting Odds Data | <https://www.kaggle.com/datasets/mexwell/historical-football-resultsbetting-odds-data> | Open Source |
| Data Scraping API | [https://www.zyte.com/zyte-api-extraction-lp/?kw=website data extractor&cpn=20986803769&utm\_source=ADW&utm\_medium=PAI&utm\_campaign=Automatic\_Extraction\_SaaS\_EMEA\*\*web\_extractor\*\*website data extractor&gad\_source=1&gclid=Cj0KCQjwqpSwBhClARIsADlZ\_TlJmVRzQZ\_Jq1l3IDt4tIyR5OM9vW0P169HBx3v6RJrqrw\_nYubAtkaAj-KEALw\_wcB](https://www.zyte.com/zyte-api-extraction-lp/?kw=website%20data%20extractor&cpn=20986803769&utm_source=ADW&utm_medium=PAI&utm_campaign=Automatic_Extraction_SaaS_EMEA**web_extractor**website%20data%20extractor&gad_source=1&gclid=Cj0KCQjwqpSwBhClARIsADlZ_TlJmVRzQZ_Jq1l3IDt4tIyR5OM9vW0P169HBx3v6RJrqrw_nYubAtkaAj-KEALw_wcB) | Open Source |

# Ethical Considerations

Gambling can be extremely addictive activity. There is an extremely fine line between a person enjoying an occasional bet to developing a gambling disorder. For this reason, I will not actually be placing any bets. This project is a hypothetical experiment to see if using data to gain an advantage is possible.

*“Many people may take gambling lightly, not realizing that it may be addictive and harmful in many of the same ways as drugs are.“ (Yale Medicine, 2024)*

In terms of the data itself, all my data sheets are open source. Should I find anymore that I deem beneficial to the project, I will ensure they adhere to the appropriate guidelines, Ie., all personal/ sensitive data redacted to adhere to GDPR regulations, correct Harvard referencing etc.

I will also ensure that the data I am testing is free of bias and my results and predictions are based off correct decision making procedures and are undertaken in a logical way.

# Examining A Dataset and building a Machine Learning Model

## Project Management And Planning

My plan for the second part of this project is to establish a market to build my predictive model around. It’s no secret that the English Premier League is the most volatile league in the sense that any team can turn up on any day and beat any team. Ie., the bottom team can beat the best team with no prior warning. In terms of betting, this doesn’t bode well for being able to gain an edge on the bookies. However, with the correct game selection and research it will hopefully alleviate any major upsets. Another reason for selecting the Premier League is there is a lot more up to date datasets available*.*

### Market Identification

I will build my model to predict two things:

* The match result / who will win. (Categorical task)
* The total expected goals per game. (Regression Task)

The main challenges surrounding this come in the form of what models to use and establishing how to implement them on my dataset. It is of paramount importance to select the correct features and ensure the appropriate testing has taken place to confirm accuracy. I will have to create two or more models as they are both predicting different outcomes. Upon early examining of the dataset, I can see there are expected goal scored and expected goal conceded features. Calculating the expected goals for both sides will give me the match result. However, I will create a separate model for the result feature.

**Below is a data dictionary detailing each feature in the dataset. *(Not included in wordcount)***

|  |  |
| --- | --- |
| **Unnamed:** 0 : Column especially for index | **PKatt:** Penalty kicks attempted. |
| **Date:** Date of match | **Poss:** Possession per centage in the game. |
| **Time:** Match kick off time | **Attendance:**  Stadium attendance. |
| **Comp:** What competition the game took place in. | **Captain:** Captain name. |
| **Round:** How many match weeks have taken place | **Formation:** How the teams lined up. |
| **Day:**  Day of week the match took place. | **Referee:** Who refereed the match. |
| **Venue:** What stadium the match took place in. | **Match Report:** Link to the official match report |
| **Result:** Match result (W, L , D) *(Categorical Target)* | **Notes:** additional notes about the match. |
| **GF:** Goals for. *(Regression Target)* | **Sh:** Number of shot taken by the team. |
| **GA:** Goals Against. *(Regression Target)* | **SoT:** Shots on target by the team |
| **Opponent:** Who the “Team” opponent was. | **Dist:** Average distance the shots were taken |
| **xG:** Expected Goals | **FK:** Number of free kicks taken. |
| **xGA:** Expected goals against. | **Season:** The season year | |
| **PK:** Penalty kicks scored. | **Team:** The team each data row is based around. | |

## Statistical Analysis

To begin, I conducted some statistical analysis on certain aspects within the dataset to ascertain if I could use any other features to gain an edge over the bookies.

**Team home form –** gauging how a team plays in their home stadium is vitally important when trying to predict the outcome of a match and will definitely provide valuable betting analysis. I have only conducted the analysis on each teams winning and drawing records in their home stadium as I will only betting on a team to win rather than lose.

I used the .groupby function. I took all the “W” from the “Result” feature for each “Team” game that was played from the “Home” string value in the “Venue” feature. I used the .sum() function because I am adding all the values together. I used the .value() function to count all the home games played. Divided by each other and times by 100 provides a per centage. The same was repeated for the draws.

A graph showing the results of a team

Description automatically generated

**Overall Team Possession**

A graph of different colored lines

Description automatically generated

An indication of team possession is important for insight into team performances.

## EDA & Descriptive Statistics

Upon uploading the dataset in Jupyter Notebook, I discovered the following:

* 606 observations
* 28 features

Of the 28 features:

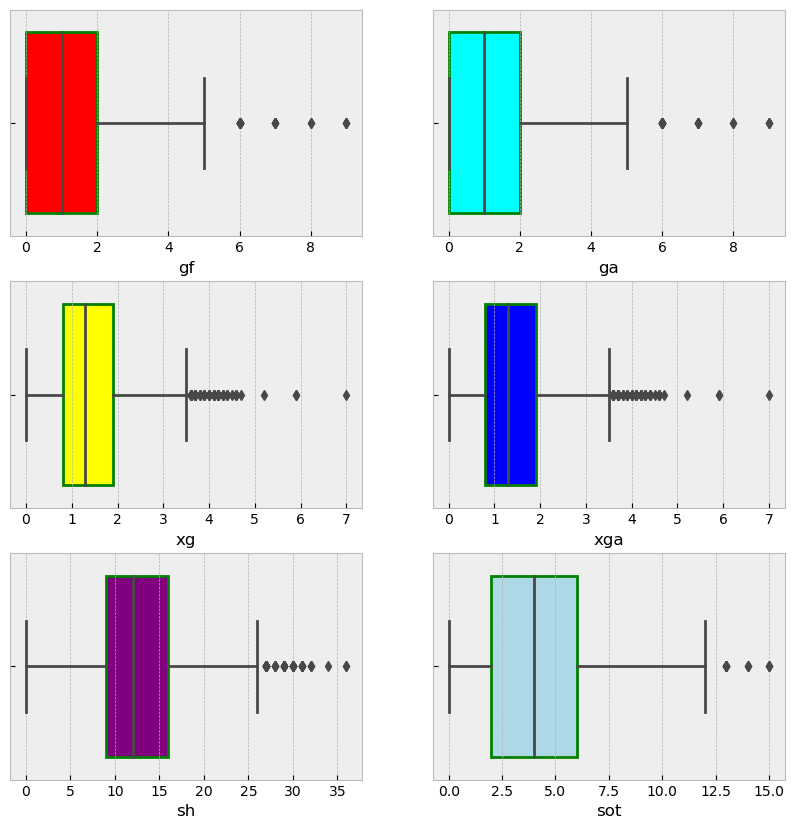
* 11 features containing floating point values
* 4 features containing int point values
* 13 features containing object point values

One feature has missing values but the rest looks to be devoid of Nan values.

**The distribution of all the numerical columns are as follows:**

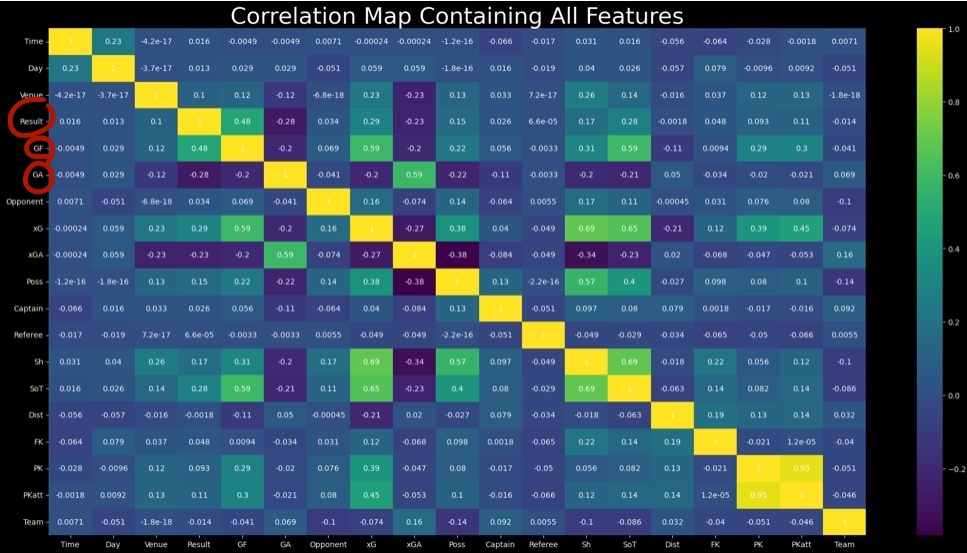
A group of blue and white graphs

Description automatically generated

I’m ignoring features “Unmamed 0”, “Notes” and “Season” as they have no relevance to the target variables. The rest of the features are relatively evenly distributed. There are some outliers in features “GA”, “GF”, “xG” and “xGA”.

I feel no threat of skewed results from the outliers present as there so few and they are contained in the target variable for the regression task. However, I noted it and will return to handle them if the model is under performing.

## Correlation Map



I removed any feature that has less than 1% correlation to the target variables.

## Feature Engineering

I used one hot encoding to convert the categorical data into binary vectors. It alleviates hierarchical assumption within the categories while improving the accuracy of the model. *(Analytics Vidhya, 2024)*

I created a new dataset of the encoded values and concatenated it with the original dataset using pd.concat function. Finally, I dropped the original features.

The new dataset looks as follows:

A close up of a number

Description automatically generated

Finally, prior to model building I rounded all the floating value features to integers as it makes it easier to interpret the results.

A screenshot of a computer

Description automatically generated

## Machine Learning Implementation and Evaluation

What models am I using and why?

**Categorical Prediction Of Match Result – Random Forest**

Reasons for Random Forest are:

* Adept at when confronted with overfitting
* Excellent at handling high dimensional data
* Allow the ability to extract feature importance

**Expected Goals/Goals for/Goals Against Predictions -Linear Regression**

Reasons for Linear Regression are:

* Easiest models to implement
* Using linear regression will provide a foundation level approach to the model and allow me to build and explore other regressor models such as random forest

## Model Implementation

### Random Forest on Match Result

With a train/ test split of 70/30 - standard random forest test split as the decision tree will always not see 30% of the data. (*R, S.E., (2021)*

The model results were as follows:



**Results breakdown**

* Perfect training score can indicate overfitting.
* Test score is minimally less but still quite high. This is evidence that the model is generalising the data well.
* Hyperparameter Tuning and Cross Validation
* I tuned the max features and n\_estimator based off the results of using the best estimator function.
* n\_estimator is the number of trees used in the random forest.
* Max\_features provides the appropriate amount of features to gain accurate results when testing.

A screenshot of a computer program

Description automatically generated

#### **Model Evaluation**

**Confusion Matrix**

A number with numbers on it

Description automatically generated with medium confidence

|  |  |  |
| --- | --- | --- |
| Class 1 | Class 2 | Class 3 |
| 41 True Positives | 76 True Positives | 63 True Positives |
| 2 False Positives | 0 False Positives | 0 False Positves |

A black text with numbers

Description automatically generated

1. 98.93 % Precision score: Shows the model doesn’t make many mistakes. It rarely identifies negative instances as positive.
2. 98.9%Recall score indicates the model doesn’t often miss positive instances and correctly detects a significant number of them. This is evidence of a low false negative rate.
3. 98.9% is a combination of both the Recall and F1 score. This high score implies good classification of instances.

A diagram of a confusion matrix

Description automatically generated

#### **Summary of Random Forest**

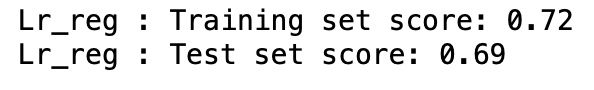
Overall, the model is performing well but in my opinion the high accuracy results indicates slight overfitting. Moving forward, I will try a different approach to feature engineering, use some feature scaling.

### Linear Regression on Expected Goals (xG)

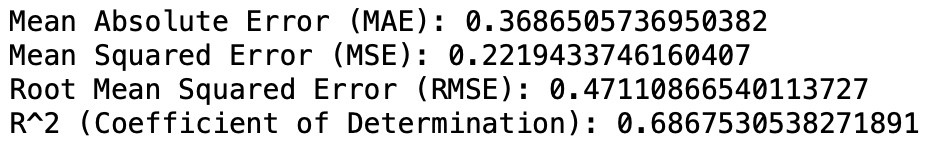
Before splitting data, I removed some more columns to reduce the possibility of bias. As the Result, GF,GA all contain goal data from the match I will remove them before implementing the model.

I am attempting to calculate expected goals in a game. This is an important metric to predicting the result of the game as it gives an insight into how many goals a team may score, thus shedding light on what the final score might be.

I employed the linear regression model with 80/20 train/test split and the accuracy was as follows:

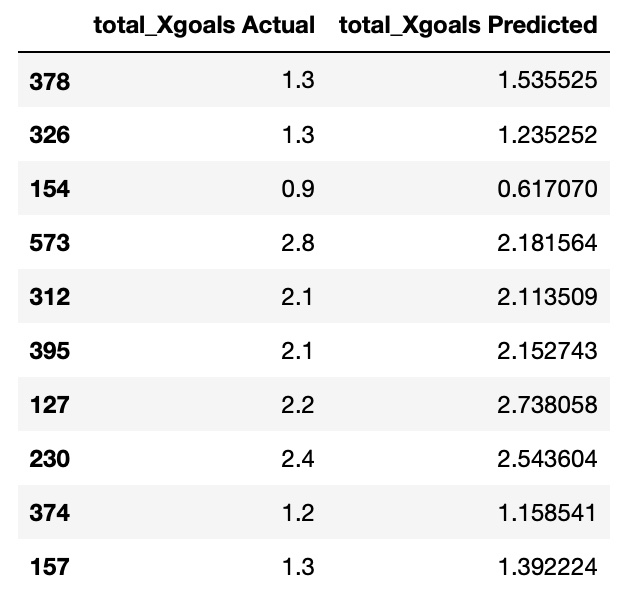


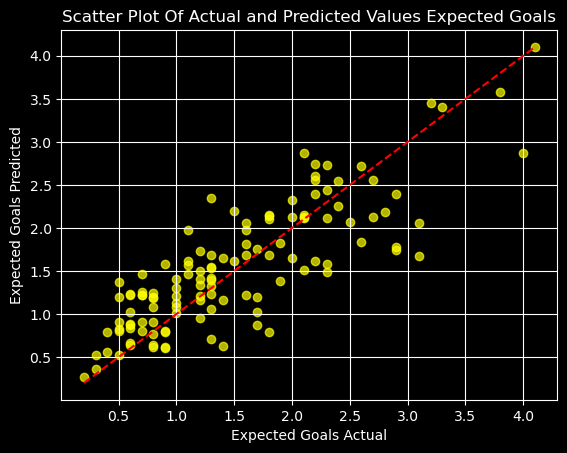
#### **Model Evaluation**



1. Mean Absolute Error - measures the distance between predicted values and actual values.
2. Mean Squared Error – measures the average squared distance between predicted and actual values.
3. Root Mean Squared Error - is the square root of the mean squared error.
4. R^2 – scores the model on a 0 – 1 scale based on the explained variance.

#### **Model Visualisation**





#### **Summary of Expected Goals Linear Regression**

The r^2 error of 0.68675 indicates that about 69% of the variance is being explained in the model. Overall I am happy with these results given the limited amount of features I have to work with. Expected is calculated on many factors such as, position on the pitch when shot is taken(shot angle), goal keepers position, body part the shot is taken with and type of assist provided. Having only one of those parameters, it is no surprise the model is not breaking 70%.

### Linear & Random Forest Regression on Goals For Feature

The following are two more experimental regressor models to verify accuracy and see which one, if any, will be suitable for what I’m looking to achieve. Both were run using the goals for feature as the target variable. Secondary to linear regression, I am using a random forest regressor for the ability to tune the model using hyper parameters. I employed splits of 70/30 and 80/20 on each model. The results of the two models are as follows:

#### Linear Regression 70/30 Split:

The model accuracy was as follows:



The predicted results were as follows:

A table with numbers and a black text

Description automatically generated

After plotting data for 70/30 Split:

A graph showing the growth of values

Description automatically generated

#### Linear Regression 80/20 Split:

The model accuracy was as follows:



The predicted results were as follows:

A screenshot of a graph

Description automatically generated

After plotting data 80/20 Split:

A graph showing a graph of values

Description automatically generated with medium confidence

#### Random Forest Before Gird Search 70/30 Split:

The model accuracy was as follows:



#### Random Forest Before Gird Search 80/20 Split:

The model accuracy was as follows:



Implementing Grid Search Cross Validation using the following Hyper Parameters:

A computer code with red green and black text

Description automatically generated

#### Random Forest After Gird Search 70/30 Split:

The model accuracy was as follows:



#### Random Forest After Gird Search 80/20 Split:

The model accuracy was as follows:



Comparison Table for Evaluation Metrics:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Mean Absolute Error (MAE):** | **Mean Squared Error (MSE):** | **Root Mean Squared Error (RMSE):** | **R^2 Score (Coefficient of Determination):** |
| **Linear Regression 70/30 Splits** | 0.47327849 | 0.35489246 | 0.59572851 | 0.72320886 |
| **Linear Regression 80/20 Splits** | 0.47932587 | 0.36225422 | 0.6018755 | 0.72455261 |
| **Random Forest 70/30 Splits** | 0.36227011 | 0.31239243 | 0.5589207 | 0.75635589 |
| **Random Forest 80/20 Splits** | 0.3753304 | 0.32720531 | 0.5720186 | 0.75120277 |
| **Random Forest**  **Grid Search 70/30 Splits** | 0.3651519 | 0.30949877 | 0.5563261 | 0.7586127 |
| **Random Forest**  **Grid Search 80/20 Splits** | 0.37431510 | 0.32356258 | 0.5688256 | 0.75397259 |

Comparison Table of Evaluation Metrics:

A graph of different colored bars

Description automatically generated with medium confidence

Having looked at the results across all the splits, I can see the results are very close. However, the model with the best performance is the random forest before grid search on a 70/30 split. It has a mean absolute error of 0.36. This means every prediction is out by 0.36 goals from the actual values. The difference between Linear and Random Forest Regressors was 2/3 percent.

A comparison of the best linear model vs the best random forest model plotted on a graph.

A graph with blue and red lines

Description automatically generated

## Selecting A Model And Formulating A Strategy

At this point, I had been building and testing models for quite some time. I thought it was wise to take stock and asses my needs. The various different models all require different parameters to predict on the appropriate variable. I noticed that depending on the feature selection, some models would be either incredibly or no good at all. For example, I was getting 100% results on the classification task of predicting the result until I noticed that the “goals for” and “goals against” features were still included before running the model. The reason it was doing so well with its predictions was it already had the final score of the game baked into its DNA. This raised a problem. For a model to work on new data, it must be fed with the same data that it has been trained on. If I am trying to classify what kind of a result it was and I am including the goals that both teams have scored, when it comes to predicting on new data I will have to provide that criteria again. As I am predicting on football matches that haven’t yet taken place, I will have to guess the goals and all that would render the model somewhat useless. I could do several different tests per game to see which one was more realistic and then come with a solution to pick the most likely result but I felt that it wasn’t in keeping with what I wanted to achieve so I had to think of something else.

Subsequently, I concluded the best metric in which to predict and bet on is the amount of goals per game/per team etc. There is an abundance of bets cantered the goals per game. The main markets I will be sticking to are:

Home Goals – How many goals home team will score

Away Goals – How many goals away team will score

Total Goals – Total goals per game

Over/Under – Over under amount of goals set by the bookies.

As previously discussed, the criteria needed to produce accurate predictions of the target variable is quite extensive. Having tested both the linear regression and random forest models using “goals for” as the target variable, I could see both accuracy scores plummeted with “goals against” removed from the training data.





A screenshot of a graph

Description automatically generated

I trained the models on the following statistcal features:

xG – Expected Goals

xGa - Expected goals Against

Possession

Shots

Shots on Target

Result

The only thing that could be difficult to input is the result but if I only stick to games where one team is more likely than the other to win, it will make choosing the result a bit easier. Alternatively, I can run the model three times on the new data while choosing a different result each time (W,D,L – Win, Draw, Lost) and compare each prediction to determine which one would be most likely. To predict to other features data, I produced an abundance of statistics which I will discuss in the next section.

# Gathering Statistics and Other Metrics

I began by downloading a more complete dataset which spanned across 5 seasons from 2019-2024. I wanted to have the most up to date data so the stats would be as relevant and accurate as possible so I also gathered data from this season from the [Global Sports Archive](https://globalsportsarchive.com/) and complied it into a dataset using Microsoft Excel.

Moving forward, I will refer to the new data I created as “this seasons data” and the rest of the data as “all years data”.

I decided to keep this seasons data separate from all years data to make comparisons between them easier. I created a function that allowed me to split all years data into the separate years or keep it altogether.

To specify what stats to pull up, I used three global variables that I can call throughout the notebook. The idea being that when two teams are playing against each other, its very useful to have all the stats related to each team at hand. The following example will pull up everything I have on Everton as the home team and Crystal Palace as the away team from the Premier league in 2023/2024.

A close up of red text

Description automatically generated

I opted to split the stats into home and away groups. The reason behind this is most teams play differently in their home stadium as oppose to their oppositions stadium. As a result, their form differs.

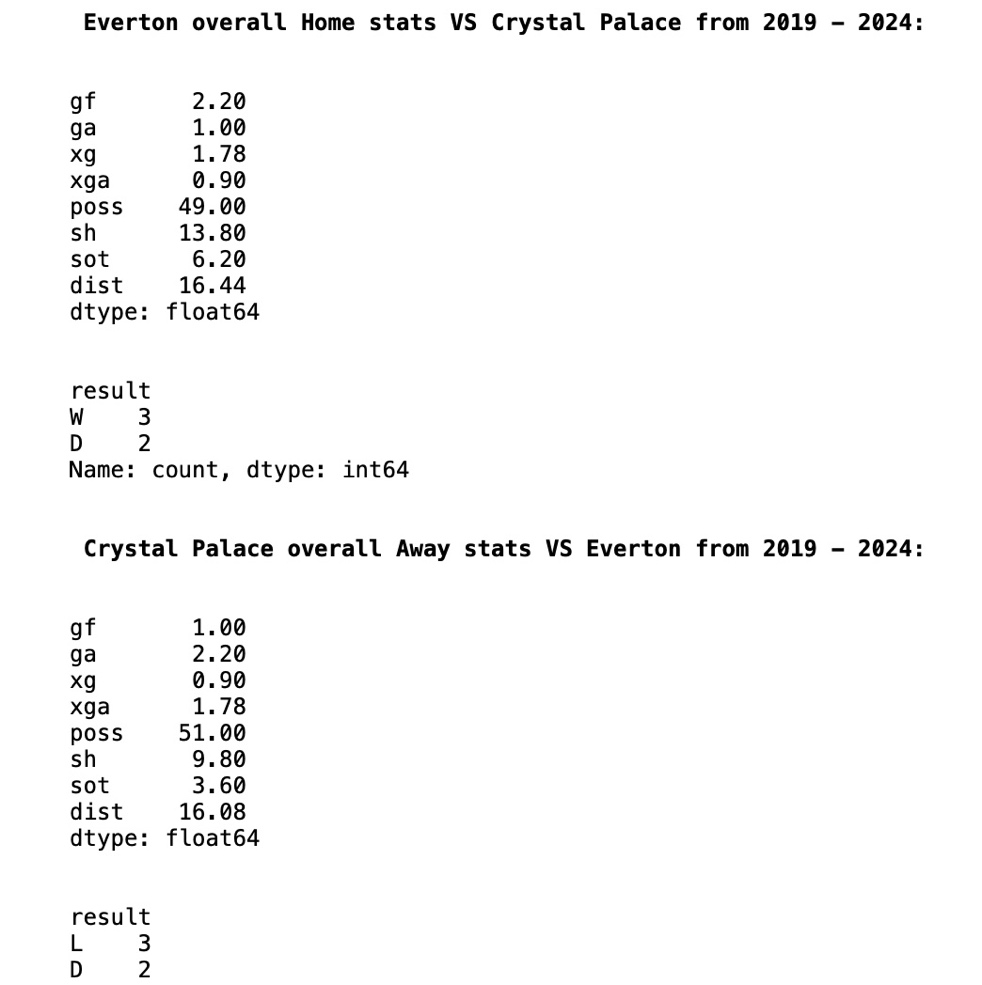
## Important Stats

*\*All the following metrics change depending on what home team or away team is selected.*

**Teams record against each other –** All the times the home team has played the away team and the relevant stats that pertain to each particular meeting. A schedule of time and date

Description automatically generated with medium confidence

**Both teams mean stats from the entire last 5 years and from the year selected in chosen season and this season so far –** The averages of each feature is a very important as it’s how each team performs more often than not.



**Home teams home form and away teams away from this season and last season only –** A recent metric of how both teams performing.

A screenshot of a score

Description automatically generated

A screenshot of a computer screen

Description automatically generated

**Shots to Goal Comparisons –**  This seasons stats on percentage of shots on target and percentage that shot on target was a goal.

A graph with numbers and points

Description automatically generated

**Possession to shots/shots on target/ expected goals/goals for –** I wanted to see how a team’s possession relates to their form in front of goal. In theory, a team with more possession should have more shots which leads to more shots on goal which should lead to more expected goals and finally which leads to more goals.

A group of graphs with blue dots

Description automatically generated

**How a team performs against other opponents (in terms of last year’s league position) –** who the opponent is plays a huge part in how a team plays. By knowing where the opponent finished each season, I can gauge their level in terms of opposition. The level of opponents are split into four groups of teams:

* Top four (finished 1-4)
* Upper mid table (finished 5-10)
* Lower mid table (finished 11-16)
* Bottom four teams (finished 17-20)

*\*finished in the positions specified in the previous year of the current chosen season*

Eg., Manchester City finished 1st in last season’s league so they would be considered a top 4 opponent. If a team’s opponent is Manchester City, I wanted to have a metric of how a team has performed against all other top 4 opponents from the years specified.

For the example in question in this report, Everton are the home team and Crystal Palace are the away team. Crystal Palace finished 10 so they are an upper mis table club.



Stats are then generated for Everton’s home record against other upper mid table clubs from the chosen season function.

Result record:

A graph with blue squares

Description automatically generated

Shots on target against other upper mid table clubs:

A graph showing different colored bars

Description automatically generated

Possession versus other upper mid table:

A graph showing a number of different colored bars

Description automatically generated

**Possession to home win percentage –** when a home team has more than 50% possession, how often do they win and when a home team has more than 60% possession, how often do they win. Stats taken from the last five years.

A graph of a number of people

Description automatically generated with medium confidence

**What day and time the game is being played -** I know this may sound frivolous but certain days /times don’t suit some teams. Although, it is not that statistical, it is still an important factor to take into consideration when predicting the outcome of a football match. For example, Everton have lost 100% of the games they played on Sunday at 4pm since 2020.

A screenshot of a football match

Description automatically generated

# What is the Strategy?

**Model Selection:**

Although the random forest produced slightly better results, I have opted to use both linear and random forest regression models so I can compare the results from both.

**Target Variable Selection:**

I am predicting the home teams goals exclusively.

**How it’s going to work:**

1. Enter home and away team.
2. Gather all relevant statistics.
3. Use all the statistics I have gathered to fill the dataset that the model will predict on.
4. Make prediction with model.
5. Find and place a suitable bet that fits prediction.

# Implementation and Testing

#### **Making an Excel spreadsheet to keep track of everything**

Keeping up to date with progress, what I’m betting on and financials is very important which is why I made an spreadsheet. I can keep track of all the matches that are on while also staying clued in to my predictions. Below is an example of a week in the spredsheet:

A green and black calendar

Description automatically generated

I have split it into weeks as there is 10 games on in most given weeks. I have also given myself a starting balance of 10 euro. So the profit and loss metrics are in relation to that figure. In any given week, I can bet as I please. I could do ten 1 euro bets or as you can see above, four bets of various value adding up to 10 euro. The darker rows are games that I made bets on.

I saved each model using the joblib library. This meant I could load it into a new python work book to do all the predictions outside of the other two main notebooks to save CPU and RAM. I make a new dataset based on the stats I have gathered on each team and then run the model on that data to predict the target variable of goals scored.

#### **Testing Example – Fulham VS West Ham**

I want to predict how many goals Fulham will score so I have to produce the following metrics in order for the model to work:

* Goals against – how many goals West Ham will score
* Expected goals for and against Fulham
* Possession (Fulham Possession)
* Shots (Fulham Shots)
* Shots on target – (Fulham shots on target)
* Result – who do I think will win (will try both and compare)
* Venue – Home or away game

**How I broke it down based on the gathering stats and other metrics section:**

Fulham have a mediocre record against West Ham at home – out of two games they’ve won 1 and lost 1.

Jumping straight to this seasons stats, I can see that West Ham are scoring on average a goal a game away from home. goals against = [1].

For the expected goals for and against, I take an average of the average of both metrics for both teams. West Hams home xG is 2.5 and Fulhams away xGA is 1.7 so after I average those xG = [2.1].

The xGA for Fulham is done the same way. West Hams away xG is 0.9 while Fulhams homw xGA is 0.9 so after averging, XGA = [0.9].

For Possession I can see that Fulham are averaging 55% for home games and West Ham are averaging 44% for away games so I will keep this as is. Possession = [55]

To calculate shots I go the poss to shots scatter plots from the stats section previously. I can see that when Fulham have 55% possession, they have between 17 – 22 shots. I average this. Shots = [19.5]

For shots on target, I know that Fulham are averaging 34% of shots going on target so I take 34% of 19.5/ Shots on Target = [6.63]

As both teams are struggling this season a draw is the most likely outcome. Result = [D]

Both models predicted around 1.34 goals for Fulham which rounded down to 1. As I have put West Ham on 1 goal my prediction for the score is 1-1.

The spread for that is as follows:

A screenshot of a football match

Description automatically generated

Compared to the actual score my prediction was correct.

The betting for this game was as follows:

A screenshot of a scoreboard

Description automatically generated

The market that best suited my prediction was over/under. More specifically, under 2.5 market. Essentially, I am betting there will be under 2.5 goals in the game based off my prediction of 1-1. I placed 2 euro at odds of 1.85 which returned 5.70 euro giving me a profit of 3.70 euro.

#### **Progress To Date**

\*PERCENTAGE OF PROGRESS HERE make table !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

# CONCLUSION

To conclude, I am pleased at how the project has planned out. After extensive testing, I identified a market that would best suit the models I produced. I am satisfied that I have built the best models I could’ve given my current experience and knowledge. That being said, there’s always room for improvement and I would like to improve from the 76% accuracy score. I hope to achieve by conducting more EDA and searching for more data to give a more complete picture of the variance. Furthermore, the models training features are something that I’d like to reduce. I will explore the possibility of employing Principal Component Analysis moving forward.

During the testing of the model and strategy building, I noticed some flaws. As I need to have a result feature in the training data, it can lead to some incorrect predictions. However, there is some leeway there as I can tailor the prediction to various bets so the prediction may be off by a goal but that can still fall within a winning a bet. This also coincides with the r^2 score of 0.37 for the random forest model.

Another minor negative aspect I noticed was when testing the model I was using the mean values. The mean is the safest way to predict something but the premier league is one of the most volatile league s in terms of crazy results and anomalies occurring. So while the mean will work a lot of the time, I will still be wrong a lot of the time when used for predicting. Moving forward, it would be nice to employ multiple machine learning models to predict different metrics.

Overall I am up …… so I have answered the question that data can be used to gain an advantage in sports betting.

# References

*(Online Sports Betting - Global | Statista Market Forecast)* [online] (2024) *Statista*, available: <https://www.statista.com/outlook/amo/online-gambling/online-sports-betting/worldwide> [accessed 20 Mar 2024].

*Khan, Md.M.I. (2023)* *Cricket-Based Betting and University Students: A New Income Source or a Curse for University Students in Bangladesh?*, preprint, Sociology, available: <https://doi.org/10.33774/coe-2023-mb6xv>.

*TryPod (2018)*, ‘HOW TO CALCULATE BOOKIE EDGE’ available: <https://www.trypodbetting.com/how-to-calculate-bookie-edge/> [accessed 21 Mar 2024].

*Chris Bumbaca (2024)* ‘Sports Betting’s Rise Is a Cash Cow. Are States Doing Enough to Curb Gambling Addiction?’ [online] *USA TODAY*, available: <https://www.usatoday.com/story/sports/sports-betting/2023/05/25/sports-betting-popularity-creates-gambling-addiction-concerns/70228634007/> [accessed 21 Mar 2024].

*Brent, H. (2024)* A Complete Guide to Irish Gambling Laws as a Tourist in 2020 [online], *The Irish Post*, available: <https://www.irishpost.com/entertainment/complete-guide-irish-gambling-laws-tourist-2020-178350> [accessed 23 Mar 2024].

*Health Research Board (August, 2023)* - Gambling\_factsheet.pdf – available - <https://www.drugsandalcohol.ie/37130/1/HRB_NDL_Gambling_factsheet.pdf>

*Sacha Alche (2024)* - Understanding Value Betting in Football | Goal.Com South Africa [online] (2024) available: <https://www.goal.com/en-za/betting/understanding-value-betting-in-football/blt1930709889af9e7c> [accessed 26 Mar 2024].

*Yale Medicine* *(2024) -* *Gambling Disorder,*  > Fact Sheets > Yale Medicine [online] available: <https://www.yalemedicine.org/conditions/gambling-disorder> [accessed 27 Mar 2024].

*Analytics Vidhya*, (2024) - One Hot Encoding Using Categorical Data | Analytics available: <https://www.analyticsvidhya.com/blog/2023/12/how-to-do-one-hot-encoding/> [accessed 16 May 2024].

*R, S.E., (2021)* ‘Understand Random Forest Algorithms With Examples (Updated 2024)’, *Analytics Vidhya*, available: <https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/> [accessed 16 May 2024].