# Investigate a dataset – Oliver Phipps

The dataset I will be investigating will be ‘Soccer Database’ (original source on Kaggle). I am using this dataset because I have an interest in football and I am a lifelong Arsenal fan, and also because I was interested in improving the SQL skills that I have learnt whilst taking part in this course.

My first steps were to download the data and also ‘DB Browser for SQL Lite’, which enabled me to view the data.

After taking a look at the database and getting to grips with the program I decided I wanted to look at betting trends in football. I personally do not bet on football but it is interesting to see how bookmakers may price their odds and if there are any trends. The full questions are as follows:

- **Are there certain bookmakers that may give better odds than others consistently?**
- **Are there certain teams that may defy bookmakers consistently?**

## Extracting data from the database

Because this dataset is housed within an SQL database, the first thing I needed to do was to look at what data I needed, and to join the necessary tables to ensure I could perform analysis upon the data. My SQL, along with comments is below:

```
SELECT c.name country_name, l.name league_name, m.season, m.home_team_goal, m.away_team_goal,m.B365H Bet365_Home, m.B365D Bet365_Draw, m.B365A Bet365_Away,

m.LBH Ladbrokes_Home, m.LBD Ladbrokes_Draw, m.LBA Ladbrokes_Away, m.WHH WilliamHill_Home, m.WHD WilliamHill_Draw, m.WHA WilliamHill_Away,

t1.team_long_name home_team, t2.team_long_name away_team

FROM match m

/* to get the country name instead of ID*/

JOIN country c

ON c.id = m.country_id

/* to get the home team name, use T1 so that we can use this table again for away teams*/

JOIN team t1

ON m.home_team_api_id = t1.team_api_id

/* to get the away team name, use T2 as we have joined the table already for the home team*/

JOIN team t2

ON m.away_team_api_id = t2.team_api_id

/* to get the league names instead of ID*/

JOIN league l

ON l.country_id = m.league_id

/* filtering so that we only see leagues in England or Spain as I see these are the biggest leagues in European football */

WHERE l.id = '1729' AND m.season = '2015/2016' OR l.id ='21518' AND m.season = '2015/2016';
```

Using this query, I extracted the columns that I believed I needed to answer the questions I posed. I chose to only look at 2 different leagues, the Spanish league (BBVA) and the English league (Premier League) as I view these are the two biggest leagues in European football.

I also filtered this further and just looked at the 2015/2016 season, as I thought this made the size of the dataset (760 rows) more reasonable. I also removed some of the betting companies to just show bigger companies. I made sure that all column names had easy to understand titles to make it easier for the reader and easier to perform the analysis.

I made a conscience effort to try and clean data within SQL before getting into Python, for example when choosing what betting companies to remove I made sure that the ones that had an observable amount of NULL values were not included.

## Data Wrangling

When I read the data in Python and performed some initial analysis I found that the cleaning via SQL that I did was helpful, and in the wrangling phase I focused more on adding columns and masks that I thought would help me answer the questions I posed previously.

My goal here was to get the data to a state in which I could perform analysis cleanly and without manipulating the source data much more. When I went to answer my second question around teams that consistently beat the bookmakers I did end up adding more columns to the dataset to make this easier.

I did of course perform checks on the data to ensure that the data was of the quality that I thought. I used the 'info' function in pandas and got the below:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 760 entries, 0 to 759
Data columns (total 16 columns):
country_name        760 non-null object
league_name         760 non-null object
season              760 non-null object
home_team_goal      760 non-null int64
away_team_goal      760 non-null int64
Bet365_Home         760 non-null float64
Bet365_Draw         760 non-null float64
Bet365_Away         760 non-null float64
Ladbrokes_Home      760 non-null float64
Ladbrokes_Draw      760 non-null float64
Ladbrokes_Away      760 non-null float64
WilliamHill_Home    760 non-null float64
WilliamHill_Draw    760 non-null float64
WilliamHill_Away    760 non-null float64
home_team           760 non-null object
away_team           760 non-null object
dtypes: float64(9), int64(2), object(5)
memory usage: 95.1+ KB
```

The above is a step that lets us confirm that there are no null values that need to be resolved. We identified from the 'shape' code that this data has 760 rows, so if any of the above had less than 760 results we would have to investigate.

Looking at the above we can see that all are showing 760 rows. Next let's make some charts to see the distribution of data. If we are seeing a large number of outliers it would be a call to investigate.

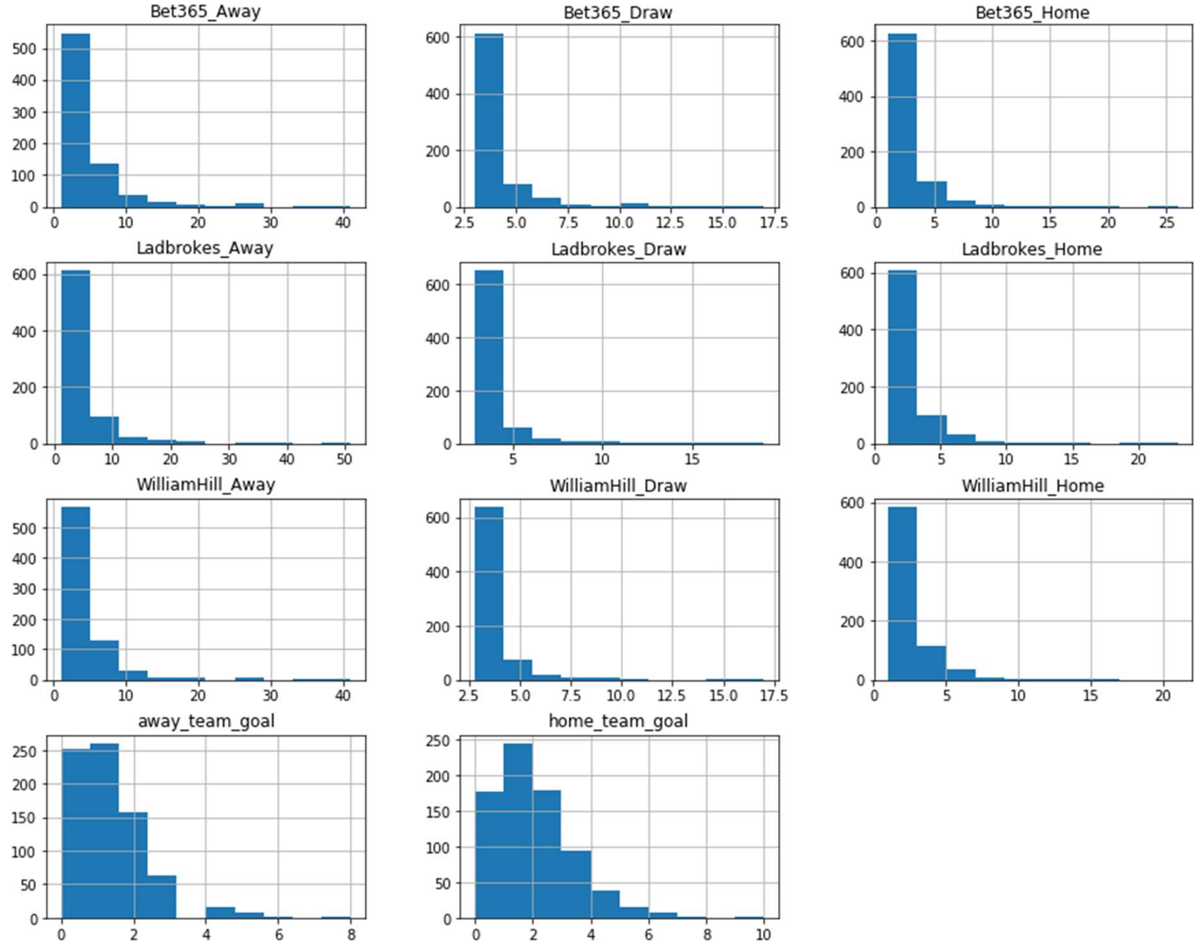
We can also see here the data types.

- We would expect any name variables to show as 'objects' - pandas name for a string.

- Any data about goals would be an integer as it would be an absolute value and could not have decimal points (you cannot score half a goal).

- Odds would be floats as they would include decimal places.

After using the above to confirm that we had the right data types and no missing values, I plotted some histograms to see visually if we had any outliers.



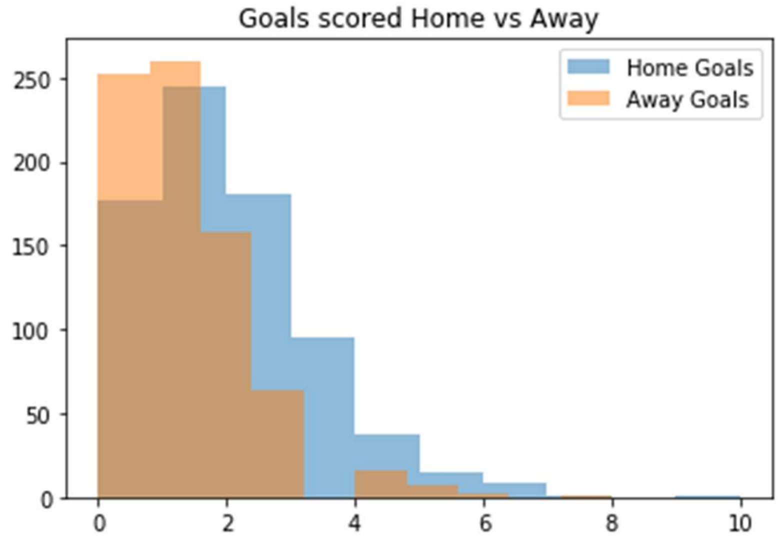
This visual representation is really helpful in seeing if there are any obvious errors and getting an idea of what you may find in your analysis. For example, if there was a huge amount of games which had 10 goals, or a large amount of games with betting odds at 20 it would be a call to do some analysis to clean the data before we preceded. Looking at the above, the data looks sensible.

There isn't much we didn't already know here, e.g. goals are skewed to right towards one, although we can see that there are more zero values for goals for away teams.

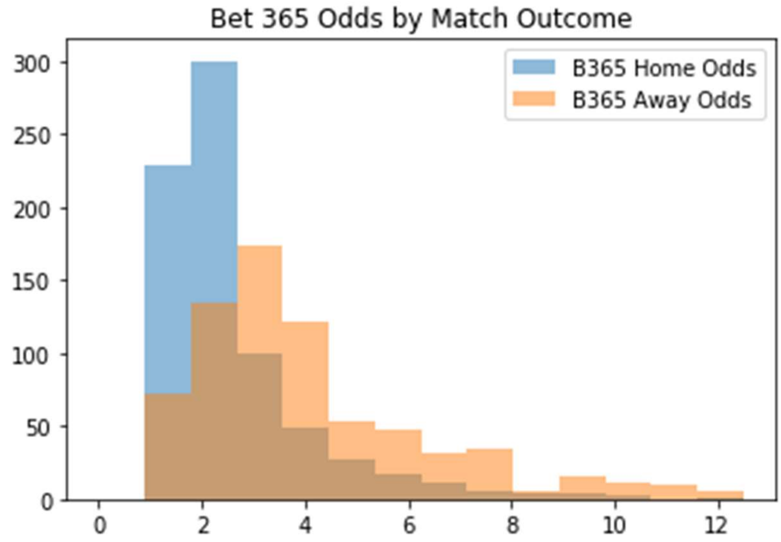
Now that we are happy with this data, we can move on to performing the analysis and to plotting some graphs.

## Plots

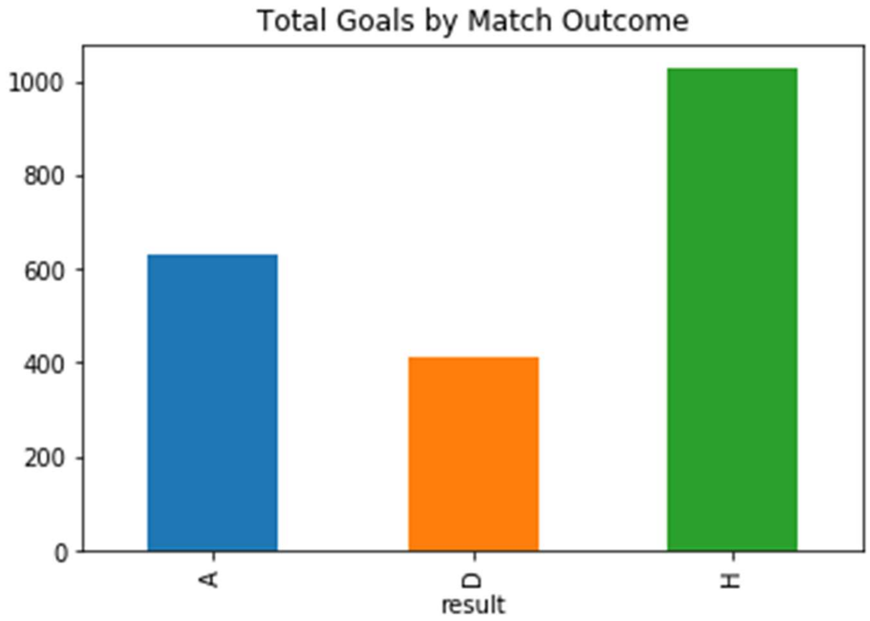
Full commentary can be found on the html that accompanies this project, but the charts below help give an idea of the results that I found:



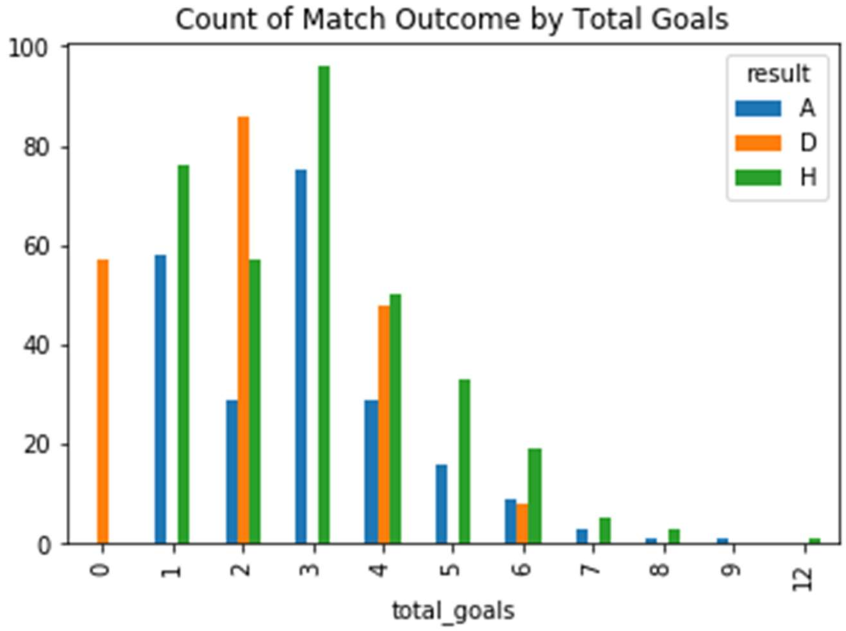
This graph makes it clear to us that the away goals are more skewed towards 0-2, whilst we can actually see clearly that there are actually more instances where the home teams scores one goal, than the home team not scoring at all. This is also true for the away team, but by a much smaller margin.



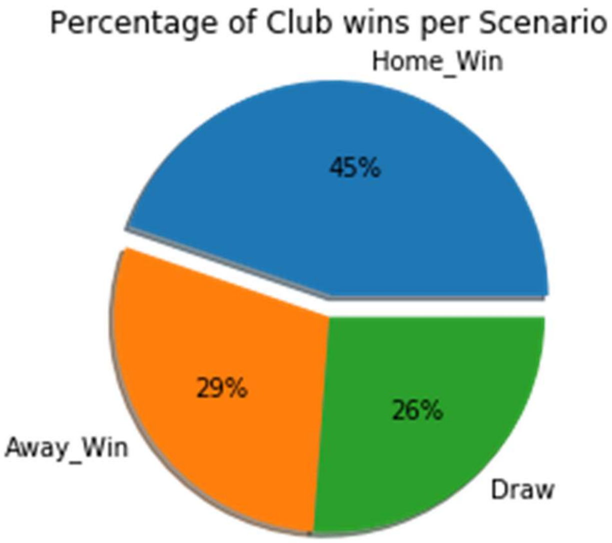
This histogram has been manipulated and cuts out any odds that are over 12.5 - this makes the graph more concise and clear. We have also not shown the odds for a draw so the graph is comparable to that of the above.  We can see that Home odds are more skewed to the right and are concentrated around 1 to 2, whilst the odds for an Away win are more spread out and peak at around 3. This shows us that bookmakers believe that a home win is more likely and therefore want to limit what they want to pay out by offering lower odds.



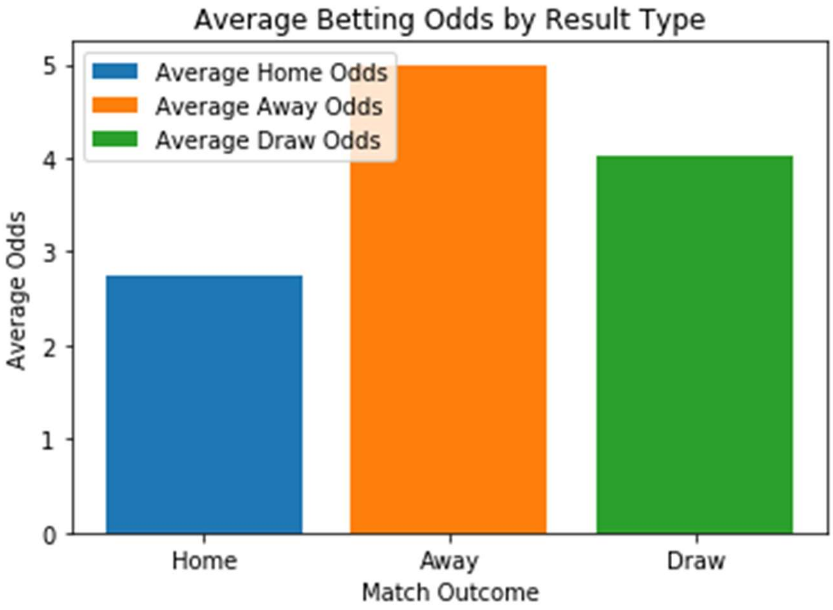
The graph above shows us the total amount of goals per match outcome. We can see that draws had the least amount of goals, whilst wins for the side at home had by far the most amount of goals here. This would suggest to us that the home side is likely to score the most amount of goals.



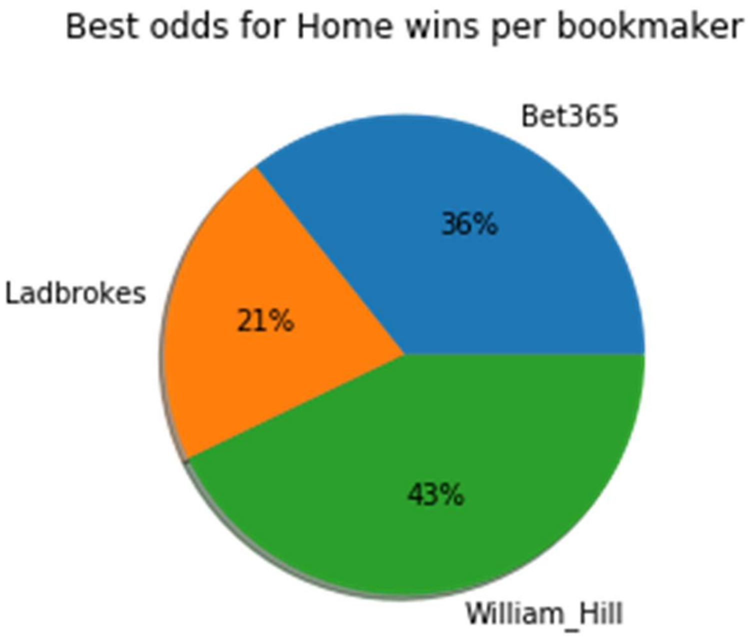
This graph shows us a count of matches and how many goals they had by result. We can see that obviously goals with no goals are all draws, and that the most common is a home draw with 3 goals in it.



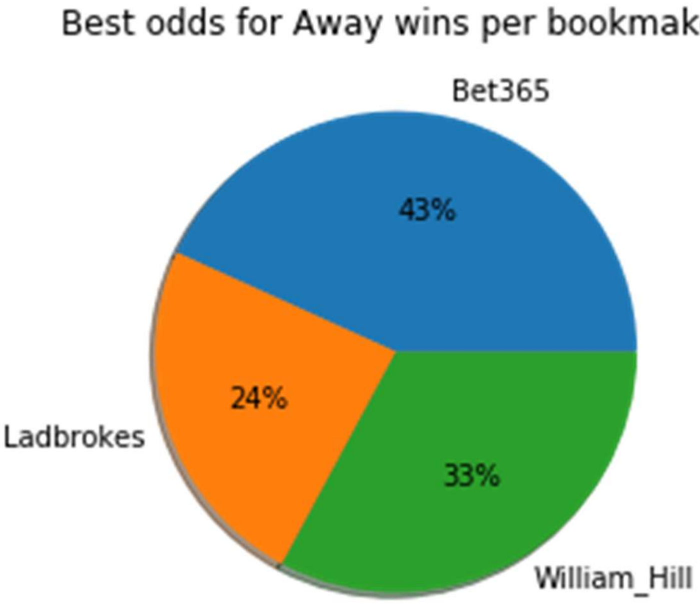
This chart here shows the match results as a percentage of all outcomes in the dataset. We can see that a home win is most likely and a draw is the most unlikely result.



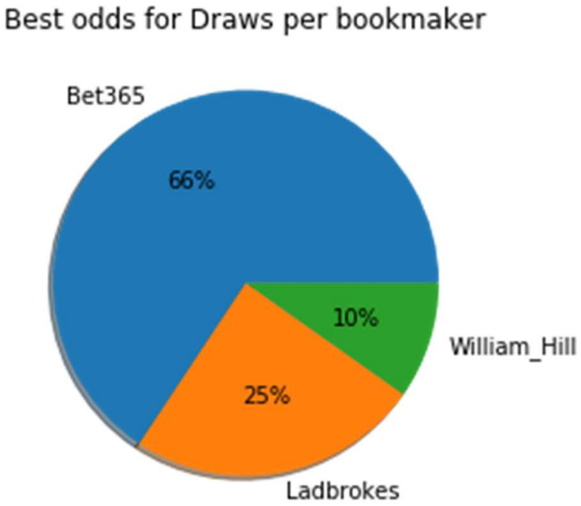
Perhaps contradicting what we saw in the pie chart above – even though a draw is most unlikely, on average the longest odds come from away wins.



The above chart shows us the which bookmaker typically gave the best, or joint best odds for a home win. We can see that William Hill was the top here.

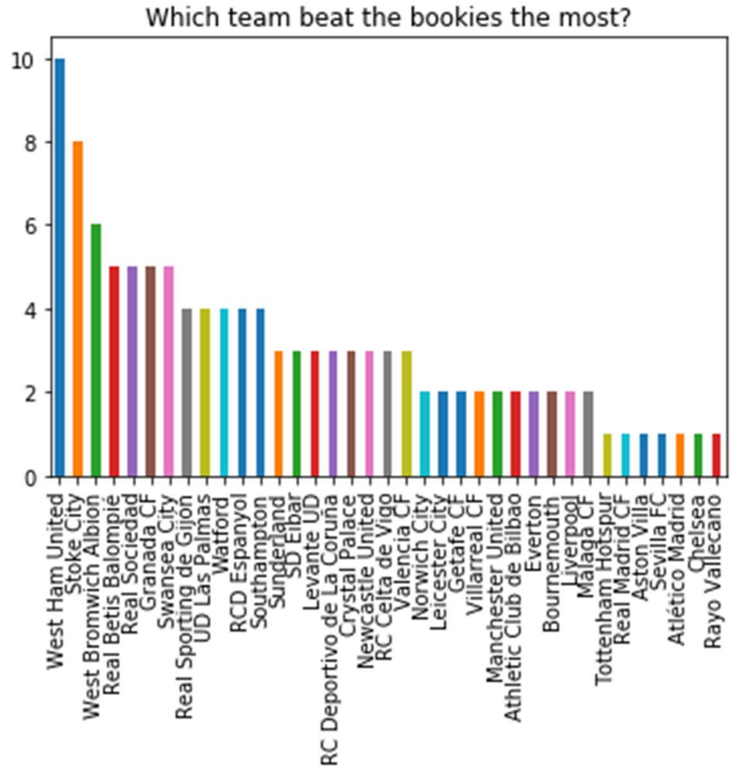


The above chart shows us the which bookmaker typically gave the best, or joint best odds for an away win. We can see that Bet365 was the top here.

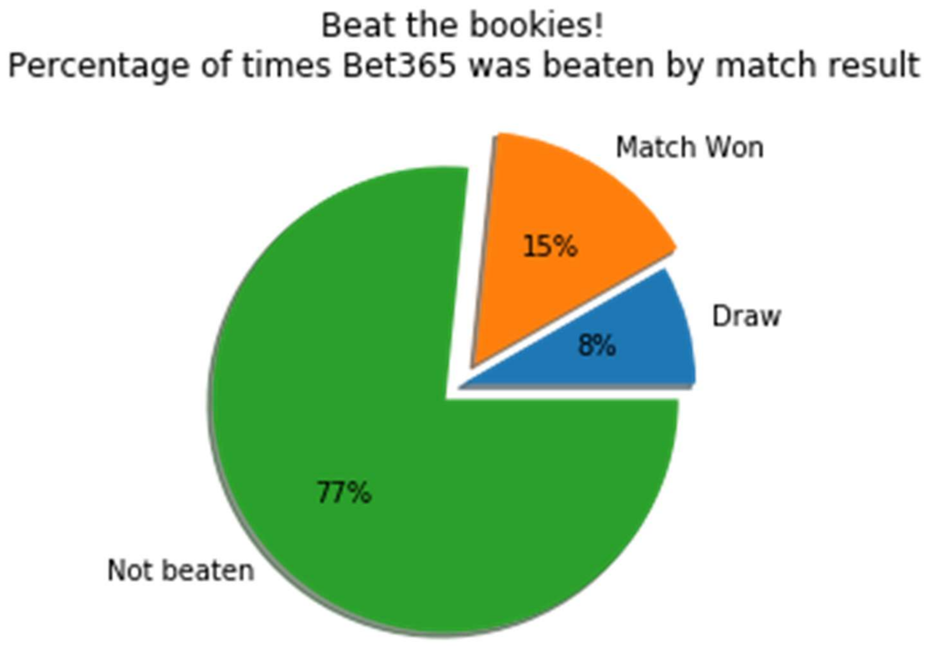


The above chart shows us the which bookmaker typically gave the best, or joint best odds for draw. We can see that Bet365 was the top here.

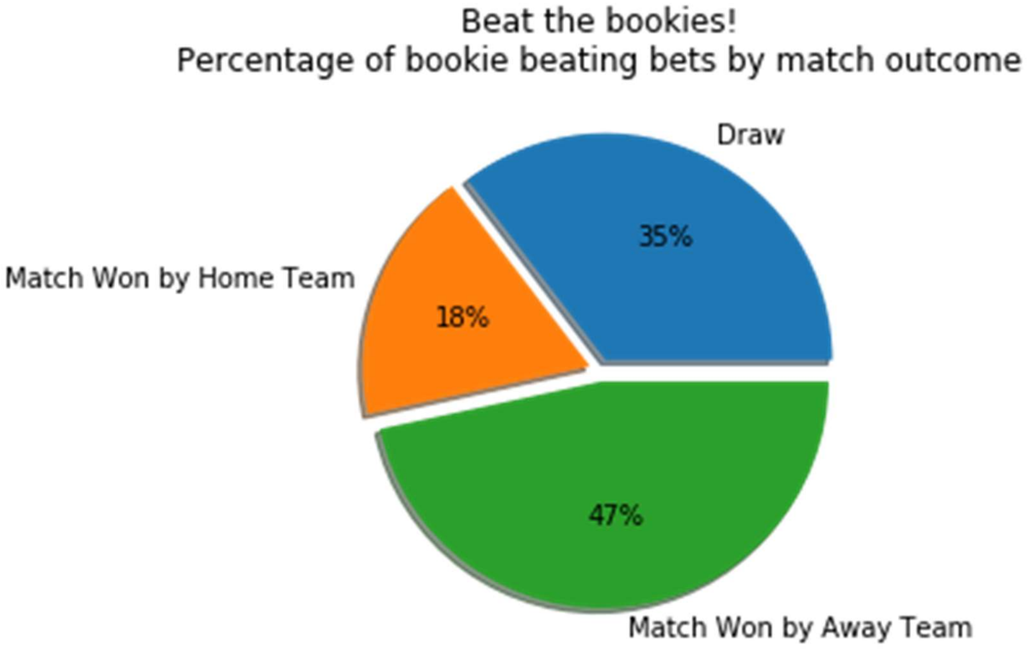
These results show that if you want to get the best odds you should be varying the bookmaker that you use depending on the outcome that you believe will happen.



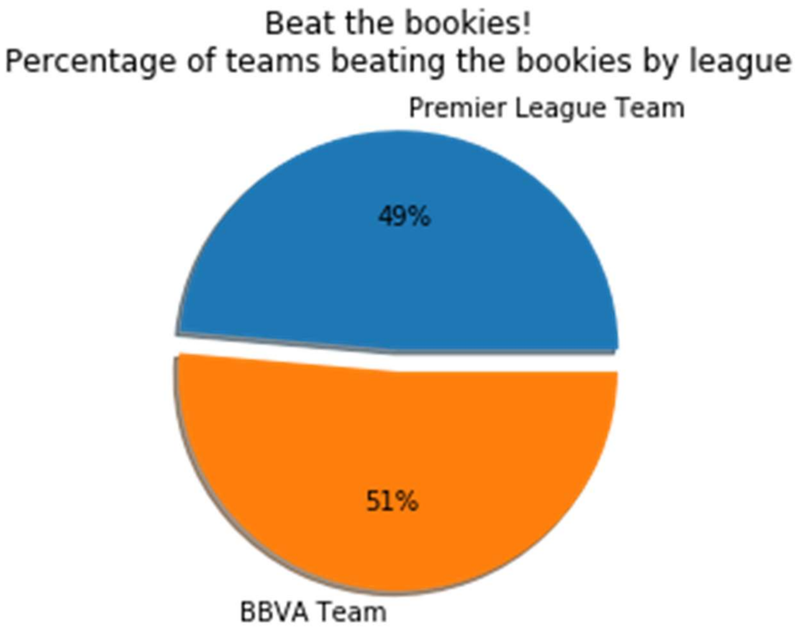
Now let’s see what teams beat the bookies the most! For this chart we created new data frames removing and draws and the times where the bookmakers won (most of the time, as we can see from the chart below) and its clear that West Ham are on top by quite a way. With many teams grouping together further along the table.



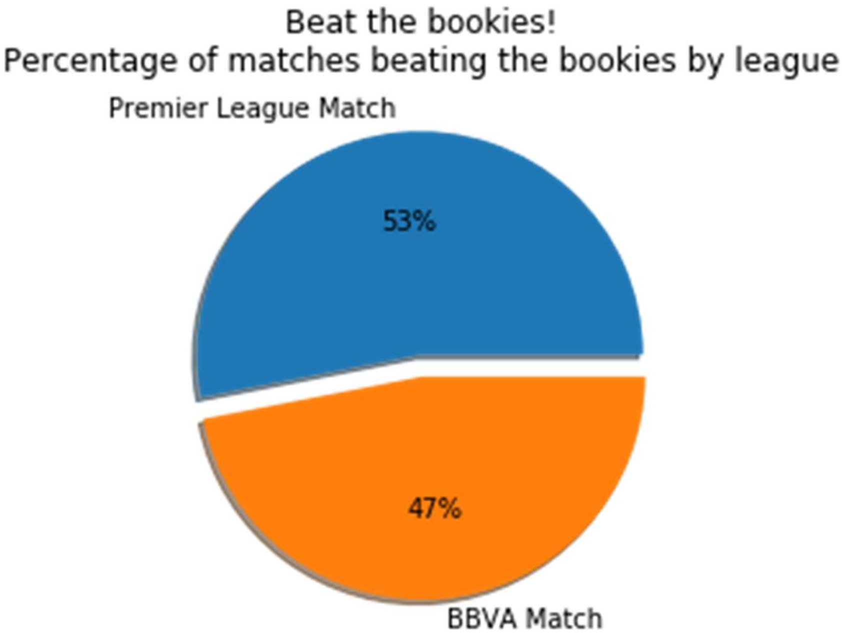
We are now looking at data in which the bookmaker was beaten, we have taken that to mean cases in which the outcome with the longest odds was correct. We can see that bookmakers were not beaten the vast majority of the time – given how many bookmakers there are in existence, this result was to be expected!



We have now filtered the data just to look at times in which the bookmaker was beaten. We can see here that they were beaten most when the away team won. The fact that the Home team winning was the smallest segment suggests that it is not often that the home team is the underdog.



We have now looked at some league specific data. This chart shows us the number of teams, by league, that beat the bookmakers. We can see that the Spanish league (BBVA) just edges out the Premier League here.



We are now looking at the number of matches in which the bookmaker was beaten by league. We can see in this scenario the Premier League has more matches and would possibly be a better option if you want to win betting on the longest odds.

## Conclusion

In conclusion this analysis has helped us understand the best bookmakers to use depending on the outcome we are expecting and also which scenarios would most consistently give the longest odds.

It was interesting to see that even though a draw is the most unlikely scenario - it was an away win that gave the longest odds on average.

If we look at the league data there were just about more teams in the Spanish league (BBVA) that did beat the bookies, there were more games in the premier league in which the longest odds paid out. This may suggest slightly more volatility in the Spanish league, with more underdog teams winning, whilst the data suggesting that a smaller amount of underdog premier league teams caused upsets.

Using this data, I would look to use William Hill is I was betting on a home team to win and Bet 365 for a draw or an away win. I would also turn to the Spanish league if I wanted to put on more varied bets on smaller teams causing an upset, whilst in the premier league I would maybe want to spend more time studying smaller teams which could beat the favourite.

If I were to continue this analysis I would look at the betting odds trends between leagues to understand if the bookmakers who gave the best odds overall consistently did between leagues or if there were differences between the two.

In regards to limitations, I would say that I would like to have carried out some predictive modelling using regression or something similar but I do not yet have the knowledge. If I had more time this is something I would have tried to research so that I could have implemented it. I would also say that it would have been good to have details on domestic cup competitions within the data set as this meant we could have perhaps analysed how a team was given odds in the league vs. domestic cups even if they played the same teams.

## Appendix – Helpful sources

https://stackoverflow.com/questions/5994485/is-it-possible-to-rename-a-joined-column-during-an-inner-join - how to call from the same table twice (when getting home and away team names)

https://stackoverflow.com/questions/19913659/pandas-conditional-creation-of-a-series-dataframe-column - how to create a Panda column based on results from other columns

https://matplotlib.org/gallery/pie_and_polar_charts/pie_demo2.html#sphx-glr-gallery-pie-and-polar-charts-pie-demo2-py – matplotlib pie charts refresher

https://stackoverflow.com/questions/8364674/how-to-count-the-number-of-true-elements-in-a-numpy-bool-array - counting amount of True values in an array

https://stackoverflow.com/questions/29919306/find-the-column-name-which-has-the-maximum-value-for-each-row - how to find the column name which has the highest value in a number of rows

https://stackoverflow.com/questions/29947574/splitting-at-underscore-in-python-and-storing-the-first-value - changing a column name to remove everything after an underscore

https://stackoverflow.com/questions/30631841/pandas-how-do-i-assign-values-based-on-multiple-conditions-for-existing-columns - how to reassign column values based on other column results