Football Markets Betting Strategy using Machine Learning

Sean Drummond

Abstract

This paper describes the methods used to develop a quantitative strategy to bet on markets in professional football (soccer) matches. The strategy has three unique characteristics; feature engineering, use of look-up-tables for confident bet selection, and a statistical method to maximize return. Machine Learning algorithms, probabilistic & statistical methods and football knowledge were all used to develop this strategy.

Raw data was obtained from www.footballdata.co.uk on nine of the top European Football Leagues and odds pricing on upcoming matches in those leagues from BetFair's API. Data on each team's match performance was used, rather than data on individual players. The strategy analysed eleven different markets for each match.

The strategy was trialled over twelve matchweeks from the 09/02/2018 to the 27/04/2018 and resulted in 89% of the 27 placed bets winning. This generated an accumulative return of 38%.

This strategy was developed and tested by Sean Drummond. I am a MSc in Computing (Machine Learning) student studying at Imperial College London with a keen interest in sports and data science. I am currently seeking employment in a data science role based in London beginning in September 2019.

(https://www.linkedin.com/in/sean-drummond-a83918a7/)

Technical Specifications

Programming Language: R.

Libraries: dplyr, readr, randomForest, caret, betfaiR, tidyr, tibble, ggplot2.

Hardware: Intel Core i3-6100U CPU @ 230GHz, 4GB RAM.

Introduction

The Sports Gambling Industry operates on a model of offering gamblers a certain return if an outcome in a sports event occurs. This price is calculated by the betting company to attempt to ensure a profit and to entice gamblers to bet. A successful sports gambler could be defined as one who consistently makes a profit by performing well in two tasks:

(1) Predicting the outcome of sports events.
(2) Choosing to place bets on outcomes with good returns.

Machine learning (ML) algorithms are used widely to find patterns in data and build models to make predictions. Supervised learning is one of two categories of ML algorithms and involves learning a model which best maps input features to labelled outputs. ML has become prominent in a wide range of industries for successfully predicting outcomes. This ranges from the automobile industry where ML in computer vision is being used in the development of driverless cars, to the financial industry where ML is being used in commercial banking to predict loan defaults, to the medical industry where ML is being used to predict illnesses from medical images. There is no reason why, with the right data, ML algorithms could not be used to attempt to predict sporting event outcomes.

This system has focused solely on developing a betting strategy for football. There are a number of reasons for this:

(1) Data

There is a large amount of open-source data readily available online. The accuracy of ML algorithms generally increases with the amount of data.

(2) Frequency of Games

Ideally, the betting strategy should be tested as often as possible to determine its accuracy. Football matches across the major European leagues occur weekly for approximately 40 weeks.

(3) Team Sport

Football is an 11-aside team sport. This makes the outcome of a game relatively uninfluenced by an injury/absence of a certain individual player.

(4) Assumption of Desire to Win

The major European football leagues are very competitive and it can be assumed that every team goes out with the intent to win every match.

(5) Betting Markets

Football is the most popular sport in the world. This is reflected in the large number of markets on each game. A large number of markets gives more choice on bets to place.

The datasets obtained contain statistics on the events of football matches. From these statistics, features can be extracted which capture a team's form in the lead up to a game. Football is played over 90 minutes and inherently, there are a large number of events that happen over the course of a game that have a major influence on the outcome and are not captured in the datasets. There are also many other factors that influence the outcome that are not likely to be captured in any dataset, e.g. how each player is feeling both mentally and physically in the lead up to the game. For these reasons, it was unlikely that the machine learning models developed would have a high classification accuracy. This coincides with sport being relatively unpredictable and exciting. Fortunately, to be a successful gambler you do not have to correctly predict the outcome of every market of every game. A successful gambler would only choose to bet on games and in markets in which they are most certain of the outcome.

ML classification algorithms output a metric for each possible output class which indicates the confidence the model has in that prediction. As the confidence level increases, the accuracy of the model does too. It is very useful to know the relationship between these two metrics. For this reason the betting strategy utilizes the concept of Confidence Accuracy Proportion Look up Tables (CAP LUTs). The accuracy metric is calculated as the percentage of correct predictions when the confidence is above a certain threshold and the proportion metric gives the percentage of data on which the model is confident. These three metrics are used to select confident bets, minimizing risk.

A successful gambler will not only make correct predictions on the outcome of sporting events, but also place money on bets with returns high enough to justify the risk. The betting strategy uses a combination of the metrics above to rank bets and then selects an optimum combination of the bets which maximises return while keeping the risk above a certain level. The total expected return of this combination and knowledge of the teams involved in the bets are used to make a decision on whether to place the bets.

Methods

Data Preparation and Feature Engineering

The raw data contains the following relevant statistics on every match played over the course of a season:

Division	Date	Home Team	Away Team

Full-time Home Goals	Full-time Away Goals	Full-time Result	Half-time Home Goals
Half-time Away Goals	Half-time Result	Home Shots	Away Shots
Home Shots on Target	Away Shots on Target	Home Fouls	Away Fouls
Home Corners	Away Corners	Home Yellow Cards	Away Yellow Cards
Home Red Cards	Away Red Cards		

Table 1-Raw Data Statistics

From these statistics, a number of labels (markets) can be calculated. These include:

- Full-Time Result
- Half-Time Result
- Full-Time Under/Over 0.5 Goals
- Full-Time Under/Over 1.5 Goals
- Full-Time Under/Over 2.5 Goals
- Full-Time Under/Over 3.5 Goals
- Full-Time Under/Over 4.5 Goals
- Full-Time Under/Over 5.5 Goals
- Full-Time Both Teams to Score
- First-Half Both Teams to Score
- Second-Half Both Teams to Score

The statistics describe the events of each match, but are obviously not available prior to the match and are therefore not able to be used to predict outcomes on that match.

Fortunately, the statistics from previous matches may be predictive of future match results. From each of these statistics, a number of features can be generated which extract information on the teams form in that statistic over time. The method chosen extracts a single number for each feature to represent all the previous values of that feature. This accounts for the time series nature of the data, so that machine learning models with a one-to-one mapping from features to labels can be used.

The ideal format for the dataset to be input to ML models, is that each row corresponds to a game played and the columns to the labels, general features specific to the game (Matchweek, Season, etc) and features specific to each team (League Position and form features specific to each team, which describe each team’s form coming into that game).

Brainstorming about the nature of football and competitive sport, it is known that performance in consecutive games has an impact on performance in future games and therefore there should be features which take into account consecutive game form. It is also known that whether a team is playing at home or away has a large influence on performance and therefore there should be features which take into account consecutive home form and consecutive away form. It is also known that certain teams consistently perform better or worse against certain teams and therefore there should be features which take into account consecutive form, consecutive home form and consecutive away form in head-to-head meetings.

To generate these features, two data structures are created, which are used with the raw data:

1. Form DataFrame

Structure: DataFrame.

Rows: Row for each team in a league in a season.

Columns: Column for each Consecutive Form Feature, Consecutive Home Form Feature, Consecutive Away Form Feature.

2. Head-to-Head (H2H) Form List

Structure: List of DataFrames.

Length: One DataFrame for each team in a league across all seasons.

Rows: Row for each team in a league across all seasons, excluding team whose DataFrame it is.

Columns: Column for each H2H Consecutive Form Feature, H2H Consecutive Home Form Feature, H2H Consecutive Away Form Feature.

Initially, all Form DataFrames features are initialized to the value of $\frac{1}{Number\ of\ Teams\ in\ the\ League}$. All H2H Form features are initialized to the value of 0.5.

The raw data is then iterated through from the earliest season to the latest season. The following two steps are taken for each game:

Data Extraction:
Form DataFrame:

- Consecutive Form Features for the two teams are extracted into the raw data.
- Consecutive Form Home and Away Features are extracted from the corresponding Home and Away Teams into the raw data.

H2H Form List

- H2H Form Features from the Home Team's H2H DataFrame with the row corresponding to the Away Team are extracted into the raw data.

Data Update
Based on the statistics in the raw data, the form features are updated:

Form DataFrame:

- The two teams that were playing's consecutive form features and respective home and away features are updated proportionally to the corresponding statistic. (See appendix I for update formulas).
- Every team's form features are normalized by feature (by column) so that they sum to 1 and that each team's feature is greater than zero and less than one.

H2H Form List

- The two teams that were playing's H2H DataFrames are found and the corresponding row for the other team is selected.
- Each team's consecutive form features and respective home and away features are updated proportionally to the corresponding statistic.
- Each team's form features are normalized across the two teams that are involved so that they sum to 1 and that each team's feature is greater than zero and less than one.

The Data Extraction and Data Update processes are repeated for each game throughout an entire season. At the end of each season, the Form DataFrame is re-initialized. Labels are also generated while iterating through each game.

There are in total 93 features which are shown in Appendix II.

Feature Generation Example for Form DataFrame:

First game of the 18/19 EPL season with Man United playing Leicester.

	Div	Date	HomeTeam	AwayTeam	FTHG	FTAG	FTR	HTHG	HTAG	HTR	HS	AS	HST	AST	HC	AC	HF	AF	HY	AY	HR	AR
1	E0	10/08/18	Man United	Leicester	2	1	H	1	0	H	8	13	6	4	2	5	11	8	2	1	0	0

As this is the first game, all variables in the form table are initialized to 0.05 (20 teams in the EPL). Man United is shown.

	Team	FT_Form	HT_Form	FT_Att_Form	FT_Mid_Form	FT_Def_Form	HT_Att_Form	HT_Mid_Form	HT_Def_Form
1	Man United	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05

These variables are extracted into raw data dataframe.

	Div	Date	HomeTeam	AwayTeam	PH_FT_Form	PH_HT_Form	PH_FT_Att_Form	PH_FT_Mid_Form	PH_FT_Def_Form
1	E0	10/08/18	Man United	Leicester	0.05	0.05	0.05	0.05	0.05

And the form variables in the Form DataFrame are updated.

	Team	FT_Form	HT_Form	FT_Att_Form	FT_Mid_Form	FT_Def_Form	HT_Att_Form	HT_Mid_Form	HT_Def_Form
1	Man United	0.06	0.06	0.05911330	0.05688376	0.05418719	0.05472637	0.05925926	0.04975124

See appendix I for details on the formulas used to update form variables.

Data Modelling

The data was modelled using a Random-Forest Classifier. An initial investigation on a subset of the data indicated that Random-Forest outperformed Logistic Regression (Softmax Regression in non-binary classification) and Support-Vector Classifier.

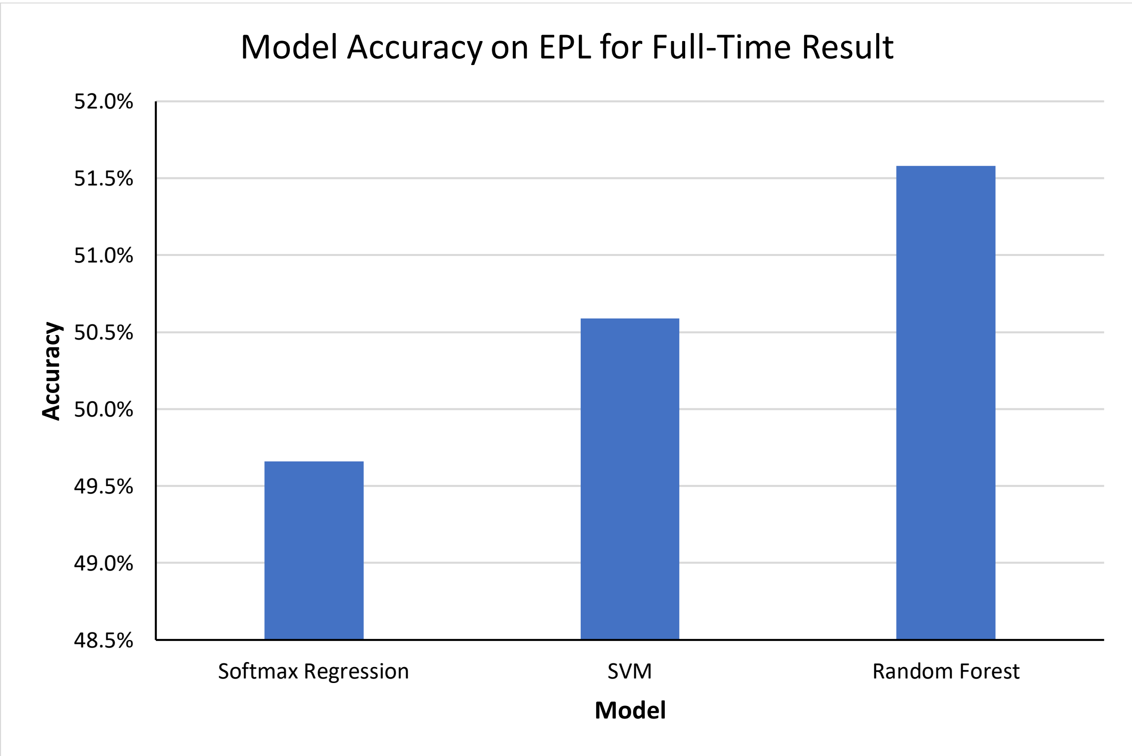

Figure 1- Classifier Accuracy

Random-Forest (RF) Classifiers are an ensemble method of machine learning classification that constructs multiple decision trees (DT) and takes the mode of the DT outputs. RF models are able to find non-linear relationships in data and do not require a large amount of data to achieve promising results.

A second investigation discovered that in general, when a model was trained on each league rather than over all leagues, the average accuracy was higher. For this reason, a model was trained for each market on each league's data.

Recursive Feature Elimination and hold-out cross-validation were used to select the optimum features. The following table shows the test accuracies achieved across each market across each league by the optimum model.

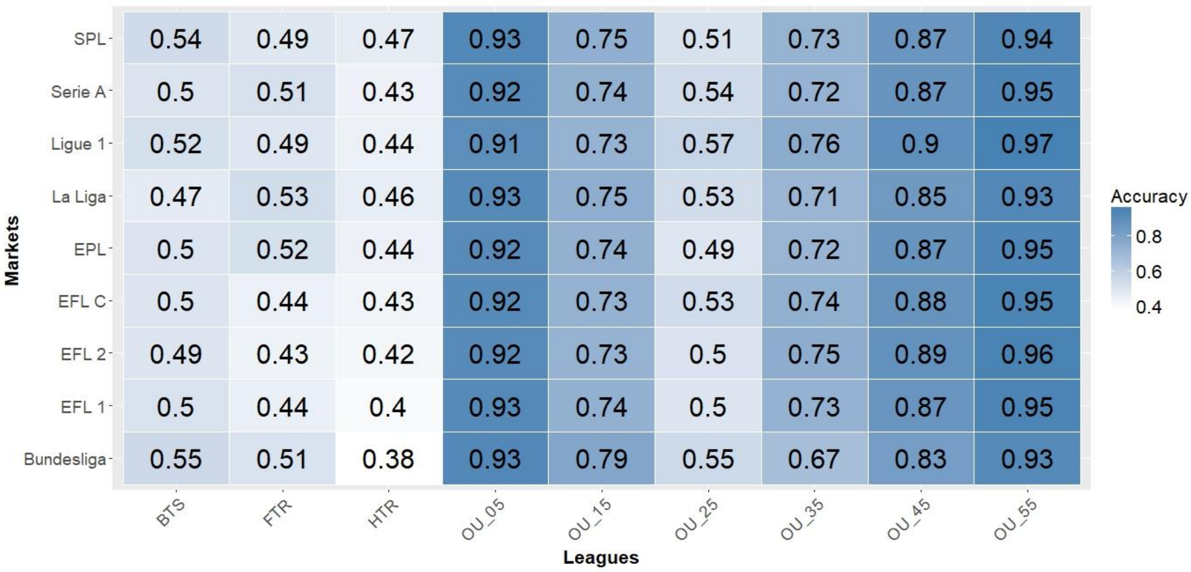

Figure 2-Test Accuracy (70:30 train:test split) of Optimum Models for Each Market and League

CAP LUTs

The models were used to generate Confidence Accuracy Proportion Look up Tables. The purpose of these tables were that once the model generated a certain confidence in a prediction, the test accuracy could be found. This metric is important to the betting strategy. The proportion metric indicates what percentage of the data is being predicted on at that level of confidence and is also a useful metric.

CAP LUTs were generated for each result of each market within each league. To calculate the values, a predefined set of confidence thresholds were chosen. They were different for bets with three possible outcomes and two possible outcomes. The model then made predictions on the test set and the accuracy of the predictions with confidence greater than each of the thresholds was calculated.

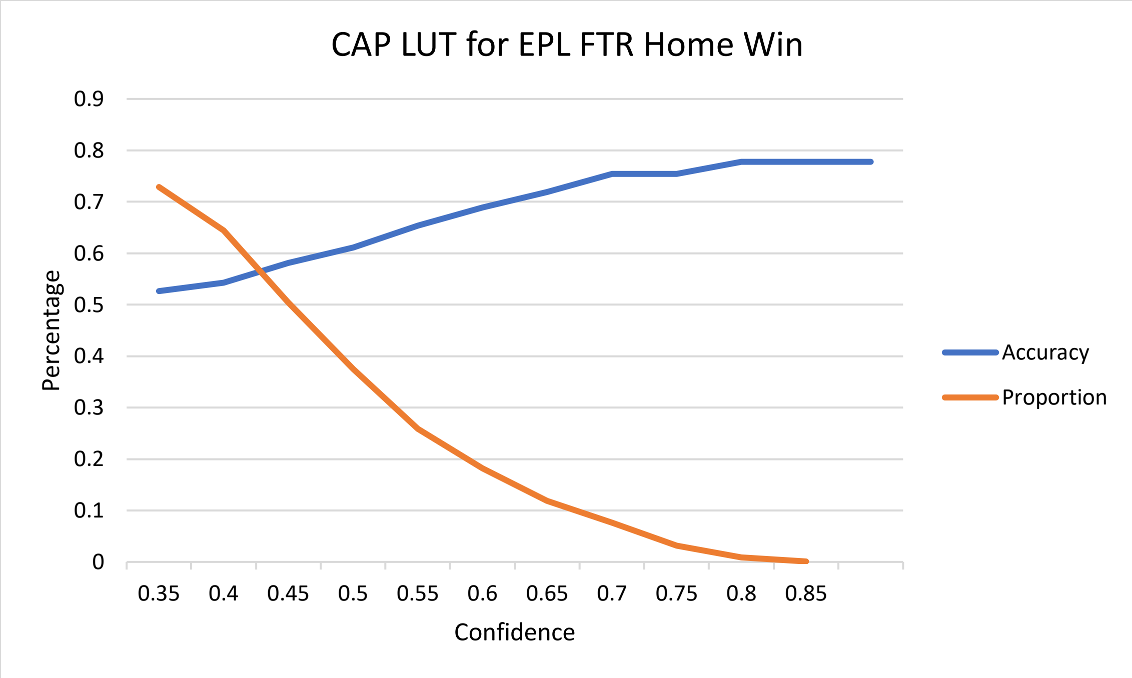

Figure 3-CAP LUT for English Premier League Full-time Result Home Win

Prediction

The prediction process involved producing a function which read in the league code of a certain league and produced a list of confident bets on matches in that league in the upcoming matchweek.

This function could then be iterated through all of the league codes and a definitive list of confident bets across all leagues could be generated.

This prediction process involved a number of key steps:

1. Latest Season Data Preparation

The function would read in the raw data from the current season and prepare it using the same process outlined in the Data Preparation and Feature Selection section, generating labels, creating and updating the Form DataFrame and updating the H2H List.

2. Next Gameweek Fixtures

The function would pull the fixtures from the BetFair API.

3. Feature Extraction

The function pulls features from the Form DataFrame and H2H list, corresponding to the upcoming fixtures.

4. Prediction

The function reads in the corresponding models and makes predictions creating a Bet DataFrame.

5. CAP LUTs

Function reads in the corresponding CAP LUTs and pulls the accuracy and proportion metrics. Bets from the Bet DataFrame with a confidence below a certain value are not added to the Confident Bet DataFrame.

6. BetFair Odds

Function pulls the odds from the BetFair API for confident bets, calculates the returns (odds + 1) and binds data to the Confident Bet DataFrame.

Bet Selection Strategy

The fixture prediction function was iterated over each league to produce a Confident Bet DataFrame across all markets and leagues for the upcoming gameweek. The following method was used to reduce the number of bets in this dataframe to approximately 20. The reason for reducing to 20 bets is that the next step in the process used an algorithm which computed values for every combination of bets and with more than 20 bets this would take a long time.

1. Remove Low Return Bets

Bets with a return below 10% were removed.

2. Create ConfAcc Metric

Create a metric which is the bet confidence multiplied by the bet accuracy.

3. Quantile Bets by ConfAcc

Divide the bets in 10 quantiles ranked by the ConfAcc metric.

4. Select best market per Match

Betfair do not allow you to bet on multiple markets on the same match within an accumulator. Therefore select best market per match by the ConfAcc metric.

5. Remove High Proportion Bets

Remove bets which have a proportion greater the 50%. These bets are not particularly rare and accuracy and confidence metrics are likely to have been skewed by the high occurrence rates of the event (e.g Over 0.5 Goals prediction will have a very high confidence and accuracy as it occurs in most games).

6. Remove Low ConfAcc Bets

Remove bets which have a ConfAcc below 0.5.

These methods reduced the Confident Bet Dataframe to approximately 20 entries. A sample from the gameweek of the 20/04/2018 is shown below:

HomeTeam	AwayTeam	Bet	BetfairMarket	Prediction	BF_Pred	Accuracy	Returns	League	Confidence	Proportion	ConfAcc
Marseille	Lille	OU_55	Over/Under 5.5 Goals	1	Under 5.5 Goals	0.9696181	1.18	FRA1	0.966	90.923441	0.9366510
Kilmarnock	Aberdeen	OU_05	Over/Under 0.5 Goals	2	Over 0.5 Goals	0.9331158	1.16	SCO1	0.968	73.297214	0.9032561
Doncaster	Oxford	OU_05	Over/Under 0.5 Goals	2	Over 0.5 Goals	0.9279077	1.14	E3	0.960	72.314578	0.8907914
AFC Wimbledon	Oldham	OU_05	Over/Under 0.5 Goals	2	Over 0.5 Goals	0.9279077	1.21	E3	0.946	72.314578	0.8778007
Lincoln	Colchester	OU_45	Over/Under 4.5 Goals	1	Under 4.5 Goals	0.9194444	1.26	E4	0.902	23.017903	0.8293389
Ipswich	Aston Villa	OU_45	Over/Under 4.5 Goals	1	Under 4.5 Goals	0.8850829	1.13	E2	0.932	29.955243	0.8248972
Preston	Norwich	OU_45	Over/Under 4.5 Goals	1	Under 4.5 Goals	0.8850829	1.13	E2	0.930	29.955243	0.8231271
Cardiff	Nott'm Forest	OU_45	Over/Under 4.5 Goals	1	Under 4.5 Goals	0.8850829	1.20	E2	0.910	29.955243	0.8054254
M'gladbach	Wolfsburg	OU_45	Over/Under 4.5 Goals	1	Under 4.5 Goals	0.8961938	1.16	GER1	0.892	25.757576	0.7994048
Bristol City	Hull	OU_45	Over/Under 4.5 Goals	1	Under 4.5 Goals	0.8850829	1.16	E2	0.902	29.955243	0.7983448
Hamburg	Freiburg	OU_15	Over/Under 1.5 Goals	2	Over 1.5 Goals	0.7888601	1.45	GER1	0.792	51.693405	0.6247772
Barnet	Newport County	OU_35	Over/Under 3.5 Goals	1	Under 3.5 Goals	0.7685644	1.26	E4	0.810	10.805627	0.6225371
RB Leipzig	Hoffenheim	OU_15	Over/Under 1.5 Goals	2	Over 1.5 Goals	0.7888601	1.18	GER1	0.788	51.693405	0.6216218
Stuttgart	Werder Bremen	OU_15	Over/Under 1.5 Goals	2	Over 1.5 Goals	0.7888601	1.32	GER1	0.778	51.693405	0.6137332
Sheffield Weds	Reading	OU_35	Over/Under 3.5 Goals	1	Under 3.5 Goals	0.7898659	1.32	E2	0.776	21.451407	0.6129359
Rangers	Hearts	OU_35	Over/Under 3.5 Goals	1	Under 3.5 Goals	0.7731707	1.32	SCO1	0.782	31.733746	0.6046195
Gillingham	Blackpool	OU_35	Over/Under 3.5 Goals	1	Under 3.5 Goals	0.7629630	1.22	E3	0.778	12.947570	0.5935852
Hibernian	Celtic	OU_15	Over/Under 1.5 Goals	2	Over 1.5 Goals	0.7620915	1.35	SCO1	0.778	41.099071	0.5929072
Rochdale	Bradford	OU_35	Over/Under 3.5 Goals	1	Under 3.5 Goals	0.7629630	1.31	E3	0.768	12.947570	0.5859556
Swindon	Grimsby	OU_15	Over/Under 1.5 Goals	2	Over 1.5 Goals	0.7409241	1.29	E4	0.760	36.892583	0.5631023
Leganes	La Coruna	OU_15	Over/Under 1.5 Goals	2	Over 1.5 Goals	0.7618567	1.43	SPA1	0.730	65.197368	0.5561554

Figure 4-Confident Bet DataFrame from 20/04/2018

An iterative loop was used to calculate the combined returns and combined accuracy of every possible combination of each of the 20 bets and stored them in a data structure. Then another loop iterated through each of the combined returns and stored the index of the bet combination (accumulator) that maximised combined returns, while keeping the combined accuracy above a certain threshold. This was done for a number of thresholds, so a range of combinations with a level of risk from low to high was calculated.

This process was repeated with each of the combinations of bets (accumulators), to obtain an optimum combination of accumulators. Usually the algorithm outputted a combination of three single bets. With three bets, there are 8 possible outcomes depending on whether each bet wins or loses. Taking the accuracy metric as the probability of the bet coming true, an expected return for each outcome was calculated and then a total expected return over all outcomes was calculated. If this expected return was above 5% and there was no reason not to bet on the teams playing, the bet was placed.

```
[[5]]
[[5]][[1]]
  Total Returns Total Accuracy
1        1.5428      0.7127428

[[5]][[2]]
    HomeTeam   AwayTeam        BetfairMarket         BF_Pred Returns  Accuracy League Confidence
192 Toulouse       Nice Over/Under 3.5 Goals Under 3.5 Goals    1.33 0.8047337   FRA1      0.802
232    Lazio Fiorentina Over/Under 4.5 Goals Under 4.5 Goals    1.16 0.8856877   ITA1      0.922


[[6]]
[[6]][[1]]
  Total Returns Total Accuracy
1        1.7685       0.619263

[[6]][[2]]
   HomeTeam  AwayTeam        BetfairMarket        BF_Pred Returns  Accuracy League Confidence
95  Wycombe    Burton Over/Under 1.5 Goals Over 1.5 Goals    1.31 0.7850099     E3      0.782
92 Hannover Stuttgart Over/Under 1.5 Goals Over 1.5 Goals    1.35 0.7888601   GER1      0.824


[[7]]
[[7]][[1]]
  Total Returns Total Accuracy
1       1.61172      0.5166266

[[7]][[2]]
        HomeTeam   AwayTeam        BetfairMarket        BF_Pred Returns  Accuracy League Confidence
91 Bayern Munich M'gladbach Over/Under 1.5 Goals Over 1.5 Goals    1.11 0.7888601   GER1      0.832
93      Dortmund   Augsburg Over/Under 1.5 Goals Over 1.5 Goals    1.21 0.7888601   GER1      0.820
90     Tottenham    Cardiff Over/Under 1.5 Goals Over 1.5 Goals    1.20 0.8301887     E1      0.838
```

Figure 5-Examples of Accumulators with Highest Total Return and Total Accuracy above thresholds of 70%,60%,50%

	Var1	Var2	Var3	Var4	Return
1	1	1	1	1	1.2133333
2	2	1	1	1	1.0116667
3	1	2	1	1	0.9850000
4	2	2	1	1	0.7833333
5	1	1	2	1	0.9483333
6	2	1	2	1	0.7466667
7	1	2	2	1	0.7200000
8	2	2	2	1	0.5183333
9	1	1	1	2	1.0283333
10	2	1	1	2	0.8266667
11	1	2	1	2	0.8000000
12	2	2	1	2	0.5983333
13	1	1	2	2	0.7633333
14	2	1	2	2	0.5616667
15	1	2	2	2	0.5350000
16	2	2	2	2	0.3333333

Figure 6-Table of the Expected Return off all Possible Outcomes for 4 Accumulators

The total amount of money placed on all bets is 67% of the total account value. This is because, unlike equities, if a bet fails the value of the bet is 0. Therefore, if only 67% of the total is placed and the bet fails, the maximum fall in value is 67%. The 67% is split evenly on each individual accumulator.

Results

Over the 12 week period the system selected a total of 117 confident bets. The accuracy on these bets was 67%. Out of these 117 bets, 27 bets were selected to place money on. Accuracy on these bets was 85.2%.

Figure 4 shows the week on week returns and the week on week accumulative returns for the period. The accumulative return over the entire period was 38.4%.

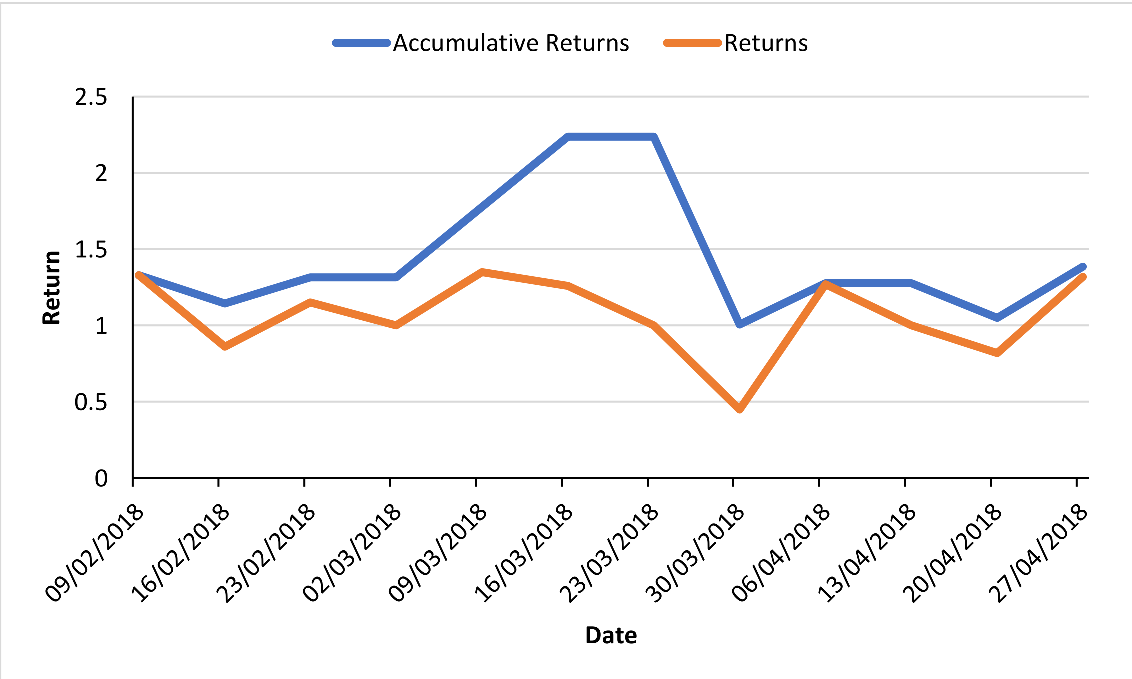

Figure 7-Returns over 12 Week Period

Figure 5 shows the value in the account over the same period. The total amount invested over the period was €861 and the final amount accrued was €946, resulting in a monetary profit of €85 which equates to a 9.9% return.

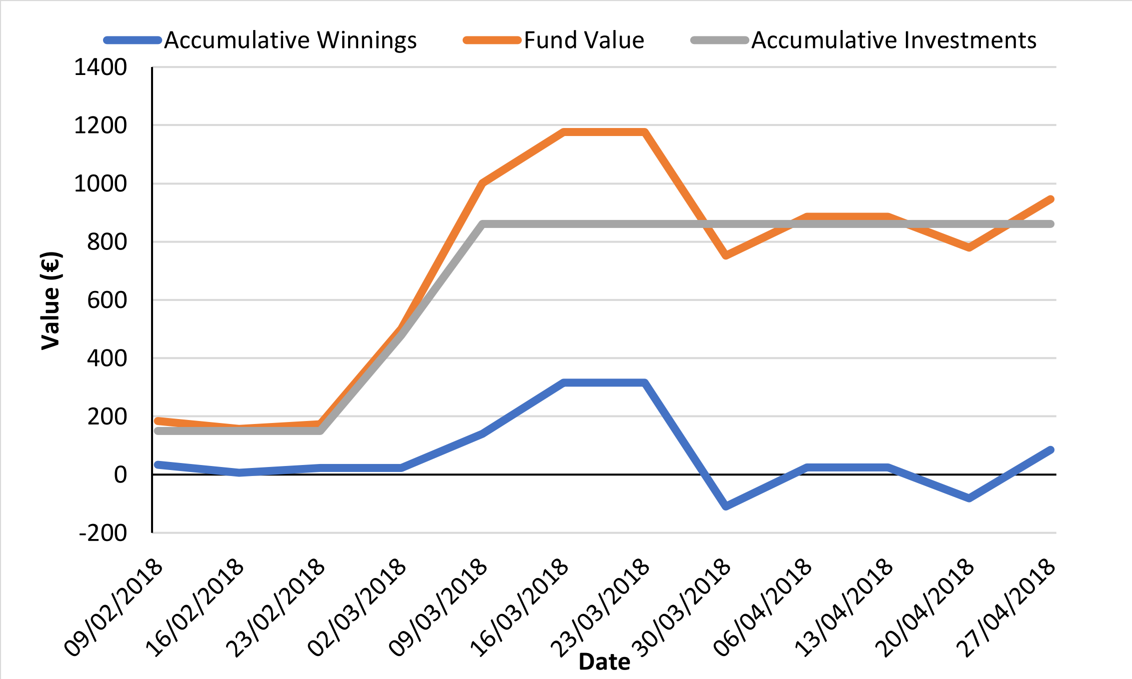

Figure 8-Account Value over 12 week period

Unfortunately the majority of the successful bets were made when only a small amount was invested and this caused the disparity between the monetary return and accumulative return.

A significant loss occurred on the week of the 30/03/2018 of 55%. The bets that lost were Napoli to beat Sassuolo away in the Italian Serie A and Hertha Berlin vs Wolfsburg FC Over 1.5 goals in the German Bundesliga. In hindsight, a simple rule could have been employed to avoid this large loss – not placing bets on away teams to win. If this rule had been employed, the accumulative return over the period would have been 110% with a monetary return of €345 (28.6%). This is Figure 6:

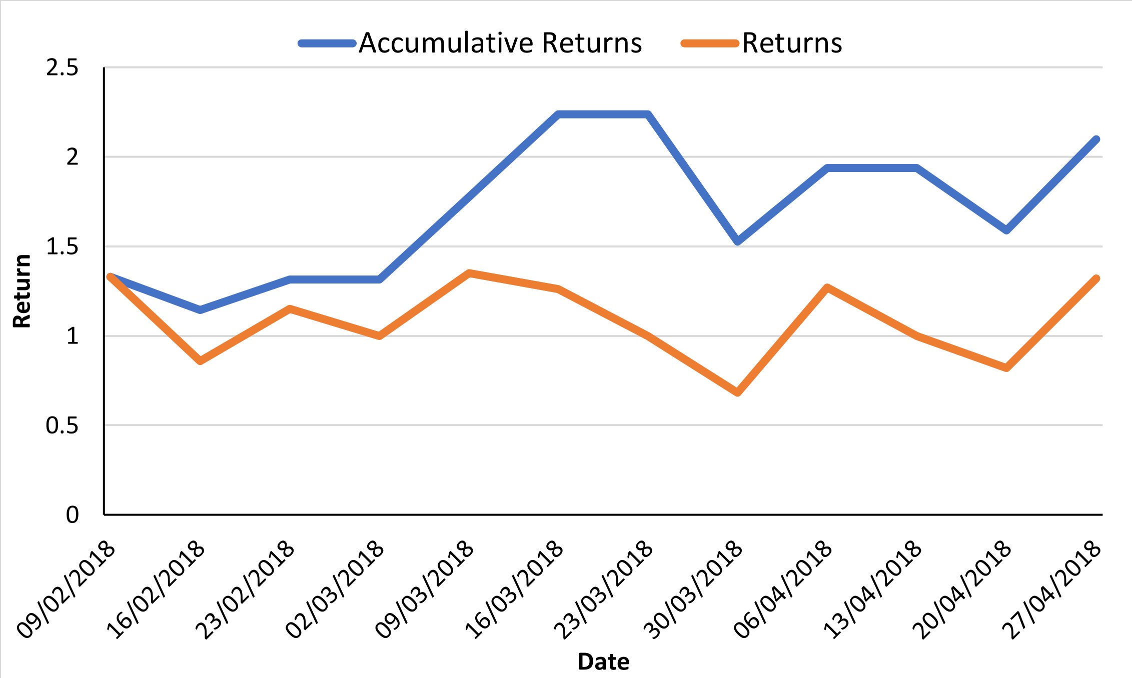

Figure 9-Potential Returns

Conclusion

A return of almost 40% over 12 weeks is a very significant amount by any standard. The S&P 500 return over the same period was approximately 2%. The system's results were successful, but there are a number of areas in which the system could be improved.

Future Work

Deep Learning

The system was built in R, which doesn't have a neural network library. A future version could be built using python and the keras or tensorflow libraries, to avail of deep neural network algorithms.

Instead of hand-crafting the feature engineering method, a time-series model such as a RNN could be used. This end-to-end deep learning approach could derive better results.

Feature Extraction

The feature extraction method gives the same weighting to less recent and more recent events. In reality, more recent events have more influence over future events than less recent. A weighting factor could be introduced in the algorithm which more significantly weights recent events.

Binary Prediction False Positive.

For some of the binary predictions where there were a large number of positives and a small number of negatives (e.g. Over Under 0.5 goals), a different cut-off value could be used in order to reduce the number of false positives.

Machine Learning to Develop a Betting Strategy.

The betting strategy using the metrics of accuracy, confidence, proportion and returns was heuristically chosen and produced good results, but a machine learning model trained on there metrics could be employed to make predictions.

Appendices

I - Formulas for Calculating Form Updates

Result Form – Win

$$F_{t+1} = F_t * 1.2$$

Result Form – Draw

$$F_{t+1} = F_t * 1.1$$

Result Form – Loss

$$F_{t+1} = F_t * 0.8$$

All Other Form Features

$$F_{t+1} = F_t * (\frac{f_t}{10} + 1)$$

Where:

F_t is the form feature value at time t.

f_t is the corresponding statistics (e.g. number of shots).

II – Features

Coding:

- PH = Team Playing at Home
- PA = Team Playing Away
- FT = Full-time
- HT = Half-time
- Att = Attack
- Mid = Midfield
- Def = Defence
- Shot Target P = Shot on Target %
- YC = Yellow Card
- RC = Red Card

Matchweek	Home Position	Away Position
PH_FT_Form	PH_HT_Form	PH_FT_Att_Form
PH_FT_Mid_Form	PH_FT_Def_Form	PH_HT_Att_Form
PH_HT_Mid_Form	PH_HT_Def_Form	PH_Shot_Form
PH_Shot_Target_Form	PH_Shot_Target_P_Form	PH_Corner_Form
PH_Foul_Form	PH_YC_Form	PH_RC_Form
PA_FT_Form	PA_HT_Form	PA_FT_Att_Form
PA_FT_Mid_Form	PA_FT_Def_Form	PA_HT_Att_Form
PA_HT_Mid_Form	PA_HT_Def_Form	PA_Shot_Form
PA_Shot_Target_Form	PA_Shot_Target_P_Form	PA_Corner_Form
PA_Foul_Form	PA_YC_Form	PA_RC_Form
PH_FT_Home_Form	PH_HT_Home_Form	PH_FT_Att_Home_Form
PH_FT_Mid_Home_Form	PH_FT_Def_Home_Form	PH_HT_Att_Home_Form
PH_HT_Mid_Home_Form	PH_HT_Def_Home_Form	PH_Home_Shot_Form
PH_Home_Shot_Target_Form	PH_Home_Shot_Target_P_Form	PH_Home_Corner_Form
PH_Home_Foul_Form	PH_Home_YC_Form	PH_Home_RC_Form
PA_FT_Away_Form	PA_HT_Away_Form	PA_FT_Att_Away_Form

PA_FT_Mid_Away_Form	PA_FT_Def_Away_Form	PA_HT_Att_Away_Form
PA_HT_Mid_Away_Form	PA_HT_Def_Away_Form	PA_Away_Shot_Form
PA_Away_Shot_Target_Form	PA_Away_Shot_Target_P_Form	PA_Away_Corner_Form
PA_Away_Foul_Form	PA_Away_YC_Form	PA_Away_RC_Form
H2H_FT_Form	H2H_HT_Form	H2H_FT_Att_Form
H2H_FT_Mid_Form	H2H_FT_Def_Form	H2H_HT_Att_Form
H2H_HT_Mid_Form	H2H_HT_Def_Form	H2H_Shot_Form
H2H_Shot_Target_Form	H2H_Shot_Target_P_Form	H2H_Corner_Form
H2H_Foul_Form	H2H_YC_Form	H2H_RC_Form
H2H_FT_Venue_Form	H2H_HT_Venue_Form	H2H_FT_Att_Venue_Form
H2H_FT_Mid_Venue_Form	H2H_FT_Def_Venue_Form	H2H_HT_Att_Venue_Form
H2H_HT_Mid_Venue_Form	H2H_HT_Def_Venue_Form	H2H_Venue_Shot_Form
H2H_Venue_Shot_Target_Form	H2H_Venue_Shot_Target_P_Form	H2H_Venue_Corner_Form
H2H_Venue_Foul_Form	H2H_Venue_YC_Form	H2H_Venue_RC_Form

Figure 10-Features Extracted