# Soccer Game Outcome & Player Value Prediction Modeling

*Authors: Beni Bienz, Avra Saslow*

Github

## Summary

Analytics have famously influenced the tactics in professional baseball, basketball, and football in the last decade. Utilizing data has fundamentally changed the way that teams and managers think about how games are won, what successful tactics look like and what makes a valuable player. While other sports have been enjoying a peak of success due to data analysis, until recently, nobody in soccer cared much about using a range of statistics to assess teams and players...they mostly just looked at who scored goals. Ultimately, it seems that these metrics can have just as great of an impact on soccer as any other sport, and can provide an understanding of the role of luck and skill in winning.

However, the caveat is that soccer is harder to predict than other sports - the New York Times explored this challenge in their article *How Data (and Some Breathtaking Soccer) Brought Liverpool to the Cusp of Glory.* They explain, "In soccer, pure chance can influence outcomes to a much greater extent than in other sports. Goals are relatively rare, fewer than three per game in England's Premier League. So whether a ball ricochets into the net or misses it by a few inches has, an average, far more of an effect upon the final result than whether, say, a potential home run in baseball lands far or foul or an N.F.L. running back grinds out a first down."

It is with that in mind that we set off to assess soccer metrics and possibly discover soccer insights that go beyond traditional statistics (i.e. goals scored). With an enormous dataset available to us, we had access to data about how many shots players had taken, what percentage of the time each team had control over the ball, and dozens of other metrics. We wanted to test if any of these statistics could help us piece together a clearer picture of what's happening on the field, as well as play around with predicting which team will win and which players will be successful. Thus, we approached this exploration in both a macro and micro sense by evaluating data associated with the team level as well as the player level.

# Predicting Player Value

## Preprocessing :

We wanted to look outside of traditional statistics to predict player value; one interesting question is if a player's zodiac sign can predict what their overall rating as a player might be. Perhaps it is true that when Mercury is in retrograde, the player with the associated zodiac sign doesn't get the ball into more advantageous positions as often. Either way, it's a good starting question that requires SQL to query columns from different tables and a classifier to predict player value; both of which were our initial learning goals.

One of the complex parts of using zodiac signs, instead of months, is that they start and end at random parts of the month.

To resolve this challenge, we utilized the datetime python library and built two helper functions that determined the zodiac sign for each player. The first helper function, get_zodiac_for_players, parses the birthday object given for each player. Initially, each birthday looks like 'YYYY-MM-DD 00:00:00', so the helper function separates the date from the time, and then puts it in a specific format such that get_zodiac_of_date can read it appropriately.

get_zodiac_of_date has a list of all of the zodiac signs and associated dates, and returns the correct zodiac sign for a specific date.

Both of these helper functions are called when we create the column 'zodiac' in the dataframe.

While preparing the data for Logistic Regression and SVM, we had to convert the the zodiac column from a string to a float, which was surprisingly a task that wasn't easy to solve. We didn't want to totally go back to the beginning and rename the zodiac signs, as they were useful in understanding the analysis.

The solution to this problem was using apply and lambda; a method in pandas which allowed us to build complex logic for the zodiac integer column. We defined a function that provided an integer associated with each zodiac sign, and then called apply lambda on that function to go through each row and add the appropriate integer for each zodiac sign in a new column. That made it easy to apply the classifiers to the data.

## Implementing Machine Learning Model:

Initially, we wanted to explore CatBoost and Logistic Regression as a means to predict player value, as we had read about their use in the academic paper *Actions Speak Louder than Goals: Valuing Player Actions in Soccer*. However, when it came time to actually implement the classifiers, we realized that they might not be the right machine learning models for determining if a player's zodiac sign predicts their rating as a player. Well, truthfully, we first implemented Logistic Regression, and found the results to be completely chaotic and a cacophony of confusion.

We stepped back and utilized the machine learning map [Figure 1] to identify that we had more than 50 samples, we weren't trying to predict a category but a quantity, and we had less than 100k samples. That led to Stochastic Gradient Descent Regression, which determines the derivative (gradient) of a single function at each iteration.

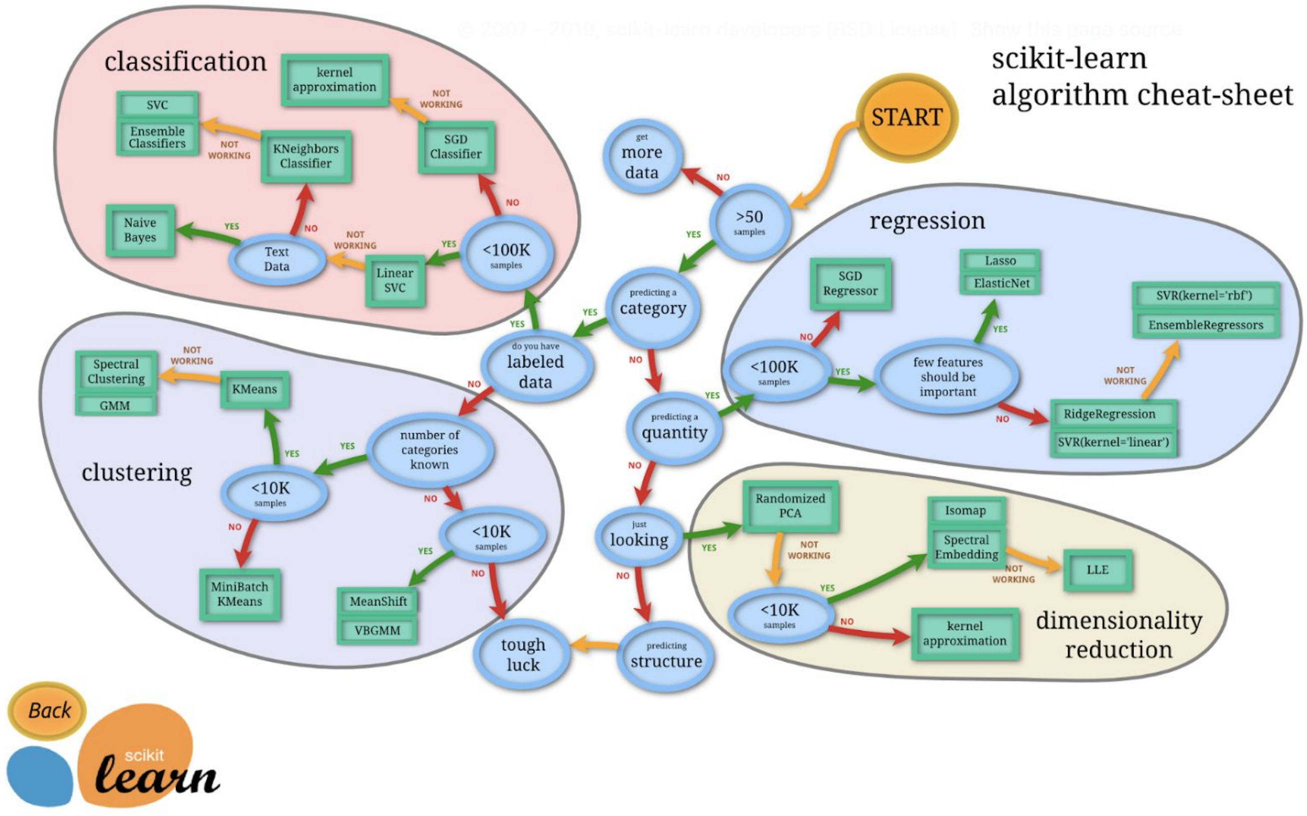
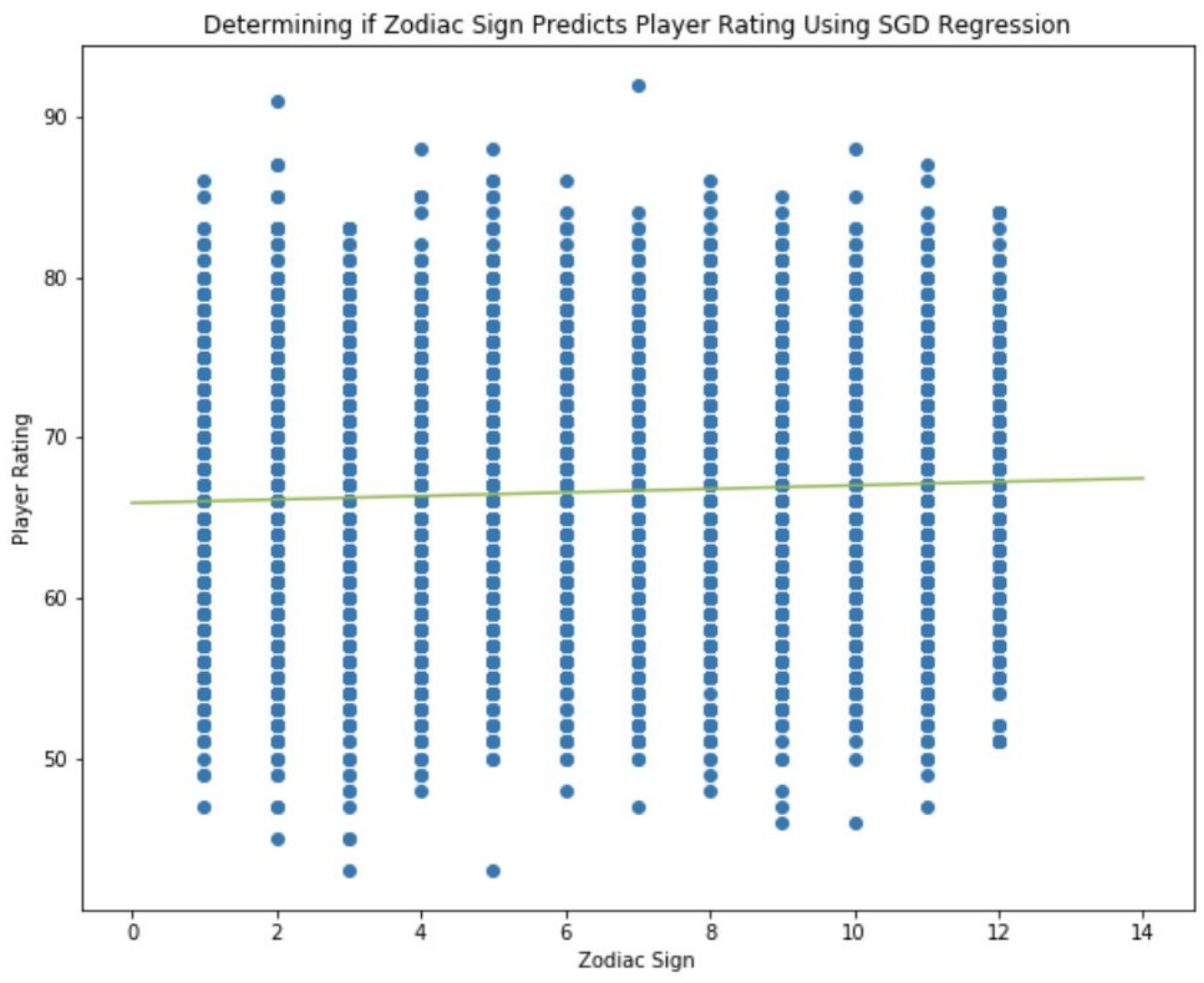


*Figure 1: Machine Learning Map*

The SGD Regressor model produced the following correlation - obviously it visually appears that there is very little to no correlation.



The r-square value was 0.00256, thus, the model explains none of the variability in player rating around its mean. This tells us that the relative positions of celestial bodies don't explain player ratings; they aren't correlated.

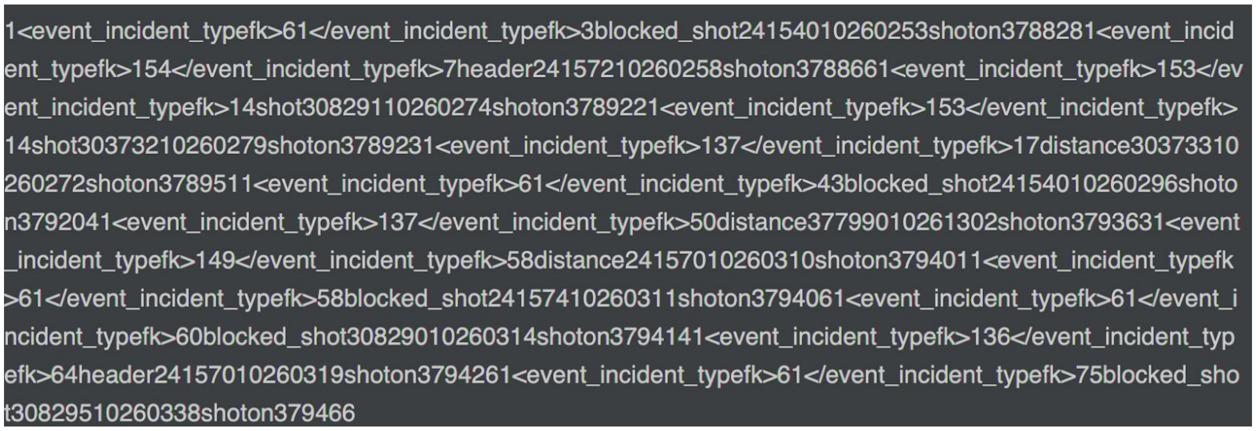
However, some of the most important insights about soccer are finding valuable players, and this model is a step in exploring whether or not specific metrics correlate with player value. Perhaps instead of zodiac signs, we could search for players who continually work to move the ball into more advantageous positions. Ultimately, the model has the advantage of finding players that provide value to the success of the team (a higher r^2 such that the model explains most/all of the variability in player rating).

## Predicting Game Outcomes

The ‘Ultimate’ dataset features game data from all European league games over 8 seasons, including match stats and bookie odds. We were interested in predicting the outcomes of matches using this data, but first we had to extract some features.

### Preprocessing

Preprocessing was quite challenging for this dataset. Most of the stats fields were populated by complex xml strings such as:



```
1<event_incident_typefk>61</event_incident_typefk>3blocked_shot24154010260253shoton3788281<event_incid
ent_typefk>154</event_incident_typefk>7header24157210260258shoton3788661<event_incident_typefk>153</ev
ent_incident_typefk>14shot30829110260274shoton3789221<event_incident_typefk>153</event_incident_typefk>
14shot30373210260279shoton3789231<event_incident_typefk>137</event_incident_typefk>17distance30373310
260272shoton3789511<event_incident_typefk>61</event_incident_typefk>43blocked_shot24154010260296shoto
n3792041<event_incident_typefk>137</event_incident_typefk>50distance37799010261302shoton3793631<event
_incident_typefk>149</event_incident_typefk>58distance24157010260310shoton3794011<event_incident_typefk
>61</event_incident_typefk>58blocked_shot24157410260311shoton3794061<event_incident_typefk>61</event_i
ncident_typefk>60blocked_shot30829010260314shoton3794141<event_incident_typefk>136</event_incident_typ
efk>64header24157010260319shoton3794261<event_incident_typefk>61</event_incident_typefk>75blocked_sho
t30829510260338shoton379466
```

The curator of the dataset provided no information about the format of these strings and no tools for parsing them. It appears they encode the event stream data for the given stat (the above image is shots on target). To parse these fields, we adapted some tools from this Kaggle kernel and transformed the dataset into a pandas DataFrame where each match is recorded twice, with stats included for the home and away teams respectively. We also limited our dataset to English Premier League games only.

### Classifier Setup

We chose three classifier models using the sklearn flowchart shown earlier:

- Linear SVC - Support Vector Classification with a linear kernel
- K Nearest Neighbors Classifier - a simple clustering model
- AdaBoost Classifier - a gradient boosting ensemble method

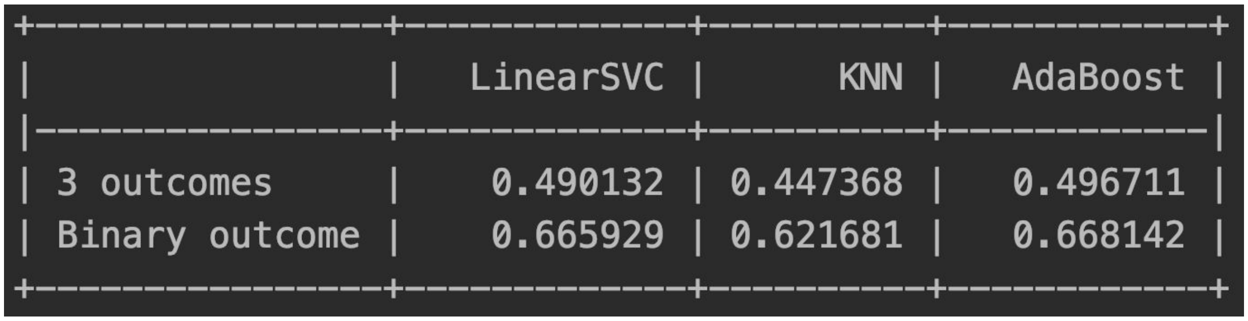
We wrote a function to use grid search cross-validation to sweep various hyperparameter values for each classifier and select the best performing model. We were also interested in the difference between a binary set of classes (win or lose) or three categories (win, lose or tie) - binary classes are typically favored by classifiers, but a significant proportion of league

games end in a tie, and you are only able to place a bet on one of 3 outcomes at the bookies.
We chose a train-test split of 80-20, with no shuffling of the dataset - ideally we want to predict future games from past data.

## Baselining with Betting Odds

To test our training setup and to establish baseline accuracies, we tried training on the odds for winning, losing and drawing as set by Bet365. We would expect each classifier to simply select the best odds when making predictions, but the results actually showed some curious behaviour. LinearSVC and AdaBoost ignored ties completely when training on three outcomes, while KNN predicted for all three (see next page). Test accuracy was not particularly good for any classifier, with none breaking 0.5 when ties were accounted for.

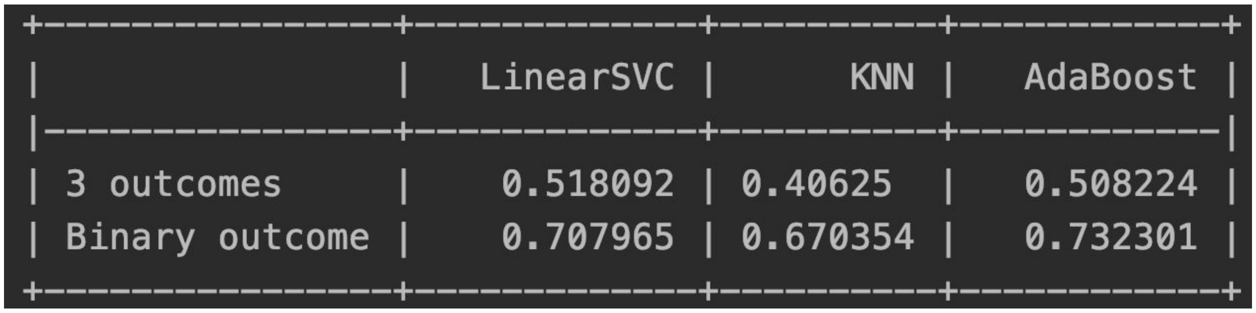
**Test accuracy table:**



```
+----------------+-------------+----------+------------+
|                |   LinearSVC |      KNN |   AdaBoost |
|----------------+-------------+----------+------------|
| 3 outcomes     |    0.490132 | 0.447368 |   0.496711 |
| Binary outcome |    0.665929 | 0.621681 |   0.668142 |
+----------------+-------------+----------+------------+
```

Interestingly, they predict better when the data is shuffled before the test-train split. Maybe football has become more unpredictable?

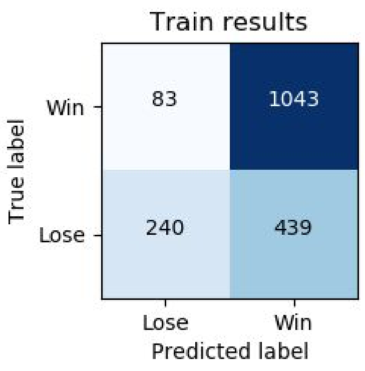
**Accuracy with shuffled training data:**

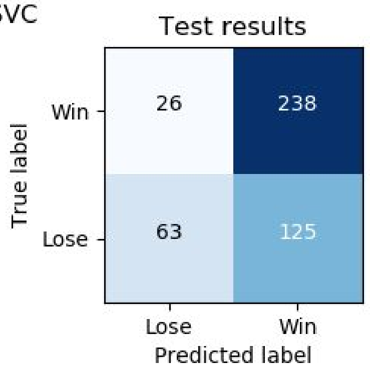


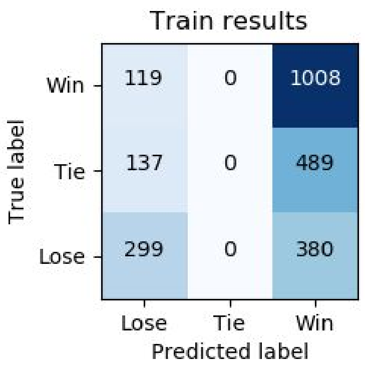
```
+----------------+-------------+----------+------------+
|                |   LinearSVC |      KNN |   AdaBoost |
|----------------+-------------+----------+------------|
| 3 outcomes     |    0.518092 | 0.40625  |   0.508224 |
| Binary outcome |    0.707965 | 0.670354 |   0.732301 |
+----------------+-------------+----------+------------+
```

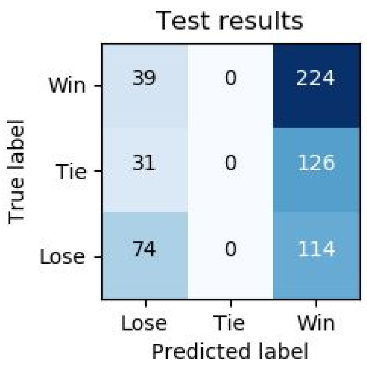
**Confusion matrices for unshuffled data:**

LinearSVC

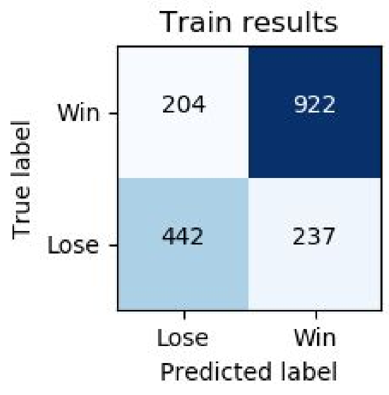


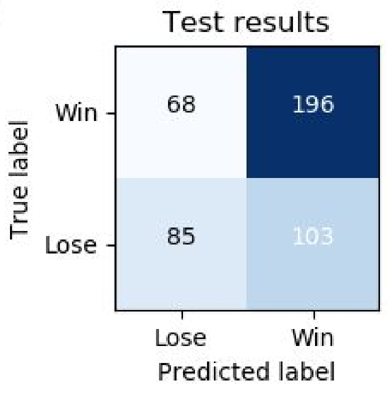


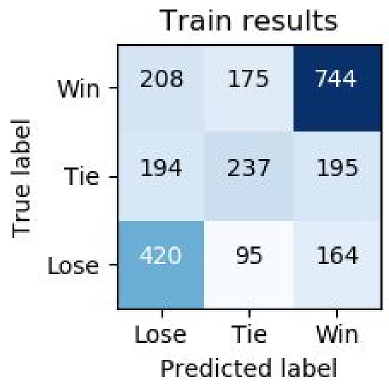


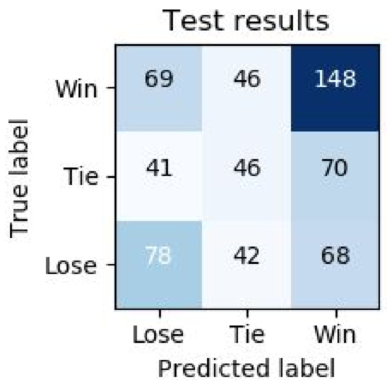


KNN









## Trying Our Own Features

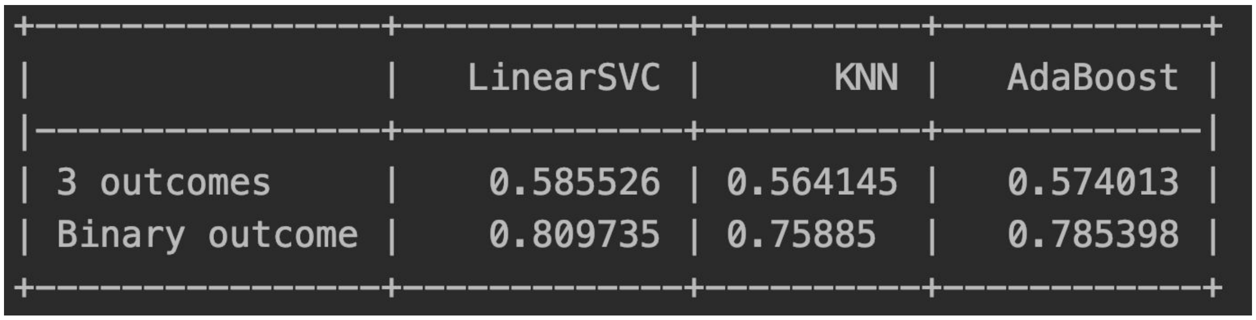
Our next step was to try and extract a few features and see if we could build a rudimentary predictive model. As there is not a lot of data, we tried to keep the number of features low. We needed features which stayed in a consistent range (i.e. a teams cumulative points would change over a season so the absolute value is meaningless). The obvious answer was to take differences between the two competing teams; we chose:

- Goals For Difference - difference in total amount of goals each team has scored so far
- Goals Against Difference - difference in total amount of goals each team has conceded so far
- Points Difference - difference in cumulative points

Although these numbers are likely to increase over the course of a season, this should be in line with the relative skill difference between the teams -. teams that are greatly unmatched should have larger differences. We also expect the current difference will have a psychological effect on the teams, which would be represented by this feature choice.

Running these features through our training code produces better results than the betting odds.

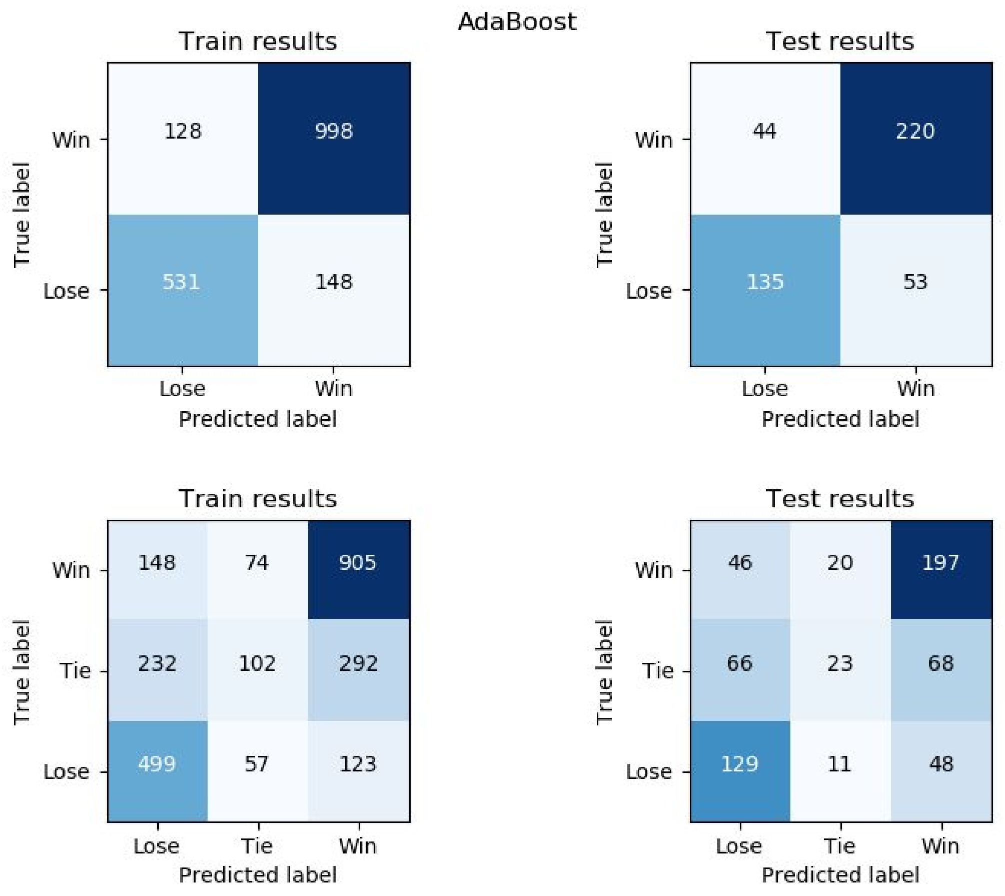
**Test accuracy table (unshuffled data):**



```
+----------------+-------------+----------+------------+
|                |   LinearSVC |      KNN |   AdaBoost |
|----------------+-------------+----------+------------|
| 3 outcomes     |    0.585526 | 0.564145 |   0.574013 |
| Binary outcome |    0.809735 | 0.75885  |   0.785398 |
+----------------+-------------+----------+------------+
```

Now all three classifiers predict some ties (see AdaBoost example on next page). This is an unexpectedly good result. Perhaps the bookies are overcomplicating things? Another explanation is the data preprocessing for these features worked better for these models than the betting odds. Maybe we shouldn't bet our life savings just yet...

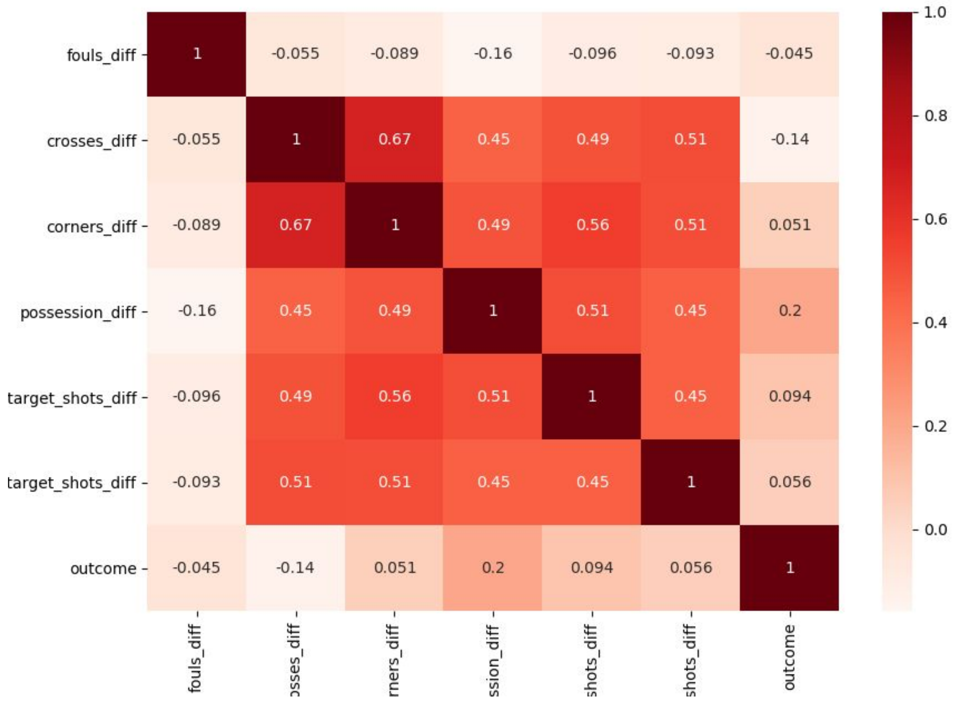
**AdaBoost results for our “differences” feature set:**



## General Analysis

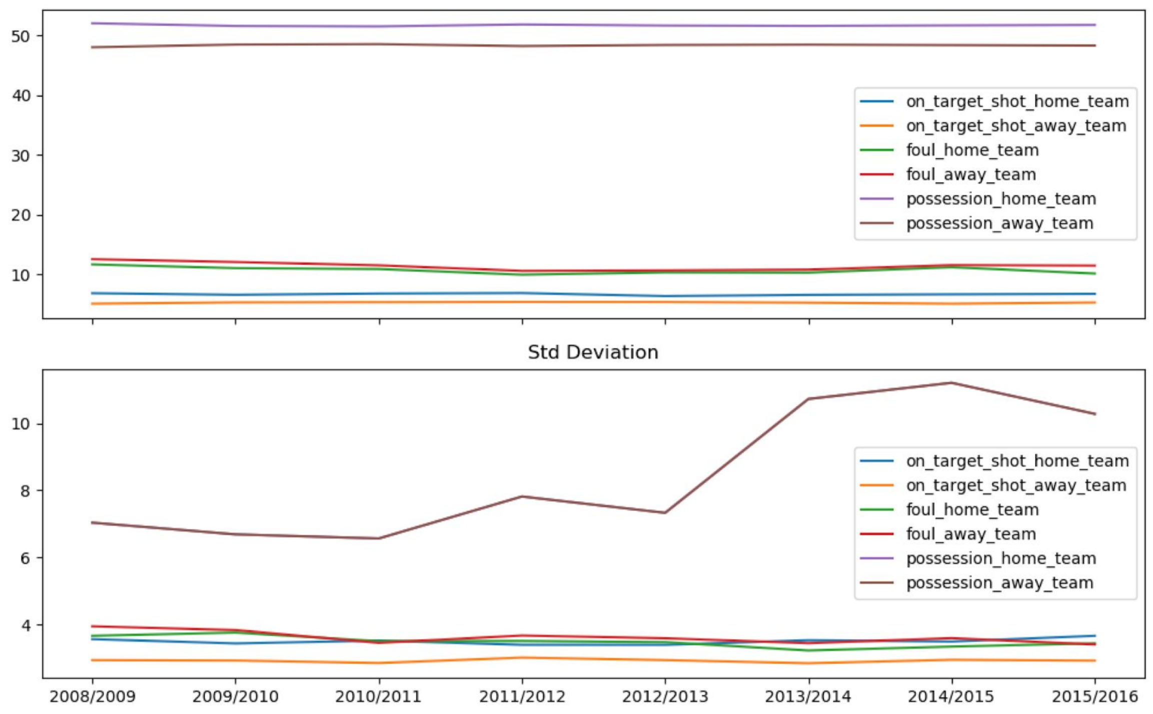
In playing around with this dataset we also found some other cool stuff. We performed a correlation study on the major match statistics that are typically reported, to see which of these were most important for game outcomes. To do this we took the difference between each stat for every game of the EPL (e.g. home team fouls minus away team fouls) and produced a Pearson correlation matrix (next page). Staggeringly, *none* of the stats have *any* correlation with game outcome apart from difference in possession, which has a weak positive correlation of 0.2. This gives us insight into why soccer is such a difficult game to

predict; essentially none of the commonly accepted proxy statistics for summarizing which team played better actually correlate with match outcome.
**Pearson Correlation Matrix of Match Statistics:**



|  | fouls_diff | crosses_diff | corners_diff | possession_diff | shots_diff | shots_diff | outcome |
|---|---|---|---|---|---|---|---|
| fouls_diff | 1 | -0.055 | -0.089 | -0.16 | -0.096 | -0.093 | -0.045 |
| crosses_diff | -0.055 | 1 | 0.67 | 0.45 | 0.49 | 0.51 | -0.14 |
| corners_diff | -0.089 | 0.67 | 1 | 0.49 | 0.56 | 0.51 | 0.051 |
| possession_diff | -0.16 | 0.45 | 0.49 | 1 | 0.51 | 0.45 | 0.2 |
| target_shots_diff | -0.096 | 0.49 | 0.56 | 0.51 | 1 | 0.45 | 0.094 |
| target_shots_diff | -0.093 | 0.51 | 0.51 | 0.45 | 0.45 | 1 | 0.056 |
| outcome | -0.045 | -0.14 | 0.051 | 0.2 | 0.094 | 0.056 | 1 |

Finally, we had a look at how stats change over the seasons and found this interesting demonstration of home team advantage:



## Who Did What and What We Learned

This project was a collaborative effort. Every decision was made as a team and most of the data preprocessing work (the hard bit) was done by both teammates. Avra then performed the analyses on player value, while Beni did match prediction and general analysis.

We both learned the following:

- SQL
- XML parsing
- Classifier types and how to select for your data
- Sklearn preprocessing, grid search CV, model evaluation
- Various plotting functions
- Soccer is difficult to predict!