Selected Topics in Economics:
Machine Learning

ECON 420/706

Sébastien Nunes (260668247)

Jacob Gervais-Chouinard (260588157)

Match Outcome Predictions in Association Football

Prof. Russell Davidson

Department of Economics

McGill University

December 24th, 2018

Match Outcome Prediction in European Association Football

Executive Summary

In this project, we test multiple machine learning algorithms on an extensive FIFA football dataset, in an attempt to predict match outcomes (win, loss or draw). We find that the ADA boost and Gradient boost classifiers perform best, with approximately 55% accuracy. This is much better than always guessing a home team win, which happens 46% of the time.

1 Introduction

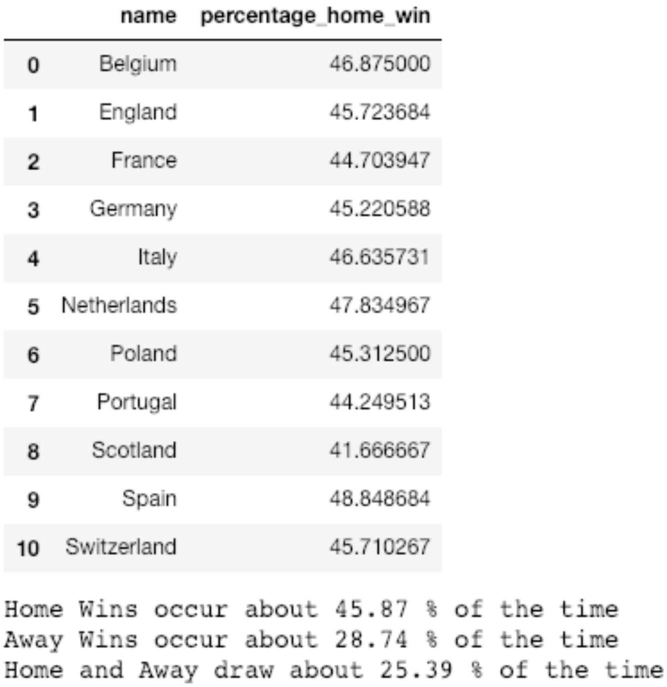
Gambling has been a human practice since before written history. It can be traced all the way back to 3000 B.C. in Mesopotamia, where a six-sided dice was discovered (Schwartz, 2013). Gambling houses were also around in China in the last millennium B.C., where betting on animal fights was common. All kinds of betting games have found life in different areas of the planet, such as horse racing, poker, and the lottery.

A notable segment is sports betting, which has since evolved into a much more mathematics-driven environment. Bookmakers use more and more refined models to calculate odds and generate profits from gamblers. More recently, with the incredible developments in computing, machine learning has become an invaluable tool in constructing the most refined models. The growing availability of high-quality data has made this refined modelling accessible to anyone with access to an Internet connection. Websites such as *Kaggle* have an immense variety of datasets which are accessible for free, with the mission of advancing data science to new heights.

Association football (or soccer in North America), due in part to its popularity—it is considered the most popular sport in the world by fan count (World Atlas, 2018)—boast incredible amounts of information about teams, players and coaches, going back decades. This enables data scientists and fans alike to construct models of all sorts, including match outcome prediction. The results of running these models on a given match can then be compared to professional bookmaker odds, with the goal of beating those odds and generate a profit from the gambler's side—or simply for bragging rights.

Another growing use of data analytics in sports is to help team managers craft the best team possible, at the lowest cost possible. One can think of the famous 2003 book by Michael Lewis, *Moneyball: The Art of Winning an Unfair Game*, explaining how data analytics for individual player performance turned a small-budget professional baseball team into a championship contender just in its first year under this new data-driven strategy.

In this project, we attempt to pin down the most important variables in professional European football, in order to build the most optimal model to predict match outcomes. We will walk the reader through each different model that we have tested, with different features of players and teams, and present the best overall one. Football matches in our datasets are regular season matches, so they face three different outcomes: win, loss or tie. Estimates from games dating back to 2008 (Figure 1) show that on average, home teams will win 46% of the time, loses 29% of the time, and the rest is ties (approx. 25%). Our main goal is to do better than a simple guess, which would be accurate at approximately the rates mentioned above. That is, our model should predict an outcome with a higher accuracy.



	name	percentage_home_win
0	Belgium	46.875000
1	England	45.723684
2	France	44.703947
3	Germany	45.220588
4	Italy	46.635731
5	Netherlands	47.834967
6	Poland	45.312500
7	Portugal	44.249513
8	Scotland	41.666667
9	Spain	48.848684
10	Switzerland	45.710267

```
Home Wins occur about 45.87 % of the time
Away Wins occur about 28.74 % of the time
Home and Away draw about 25.39 % of the time
```

Figure 1: From 2008 to 2016, the observed probability of home team winning, by country.

2 Data and Methodology

We use a dataset from the open-source data science website *Kaggle.com*. The dataset includes over 25,000 matches from seasons 2008 to 2016, across 11 European countries. Match data includes, both team and player information.

The dataset has information on 10,000 individual players, including their *attributes*, as developed by EA Sports' FIFA video game. In addition to personal information such as age, weight and height, player attributes include, among others: overall rating, acceleration, shot power, sliding tackle, and more. These attributes are given on a scale of 0-100 and are a reasonably close representation of reality. Every year, multiple thousands of experts report their opinion on player attributes, including less-known players. Ratings are then adjusted according to the league in which a player is in (Lindberg, 2016). This should not bias our results as matches in the dataset are intra-league. Hence, this specific adjustment effectively cancels out—as opposed to World Cup or Euro tournaments, where this would be unfair to top players that happen to play in weaker leagues.

Team data includes the starting players, number of shots, fouls, corners, possession time, whether the game was away or home, and aggregated player attributes by position.

The first challenge is to identify which variable should be kept in the models, in order to get rid of some unnecessary noise. First, since a team's attributes are an aggregated version of its underlying player attributes, we should not include both due to multicollinearity issues. Thus, we choose to keep overall team attributes, which encompass its starting players' attributes. These team-level attributes are broken down into specific factors, such as attacker, defender, midfielder and goalie rating, and they are updated to reflect the players actually taking part in that specific match. This allows for more accurate information when injuries or other events forces aside a regularly-starting player, which could have a major impact on match outcome if the missing player is a top-player. Other team information on match-day are included in the model, such as the team's record in the last five games, their winning rate this season up to that match, the historical head-to-head winning rate against that specific opponent, and more. We choose to include the following variables, stated per match:

- Win Rate: historical team winning rate
- Win Rate This Season: the current period winning rate
- Draw Rate: historical tying rate
- Draw Rate This Season: current period tying rate
- Number of top players: # of players in the top 1% of the ratings distribution
- Number of bottom players: # of players in the lowest portion of the ratings distribution.
- Midfielder rating: score 0-100 for overall team strength at midfield
- Goalkeeper rating: score 0-100 for overall team strength in goalkeeping
- Defender rating: score 0-100 for overall team strength at defenders
- Attacker rating: score 0-100 for overall team strength in attackers
- Head-to-Head home team winning rate: historical winning record when the two teams play against each other
- Away team winning rate at this ground: visiting team historical winning rate when playing in that stadium.
- Away team tying rate at this ground: visiting team historical tying rate when playing in that stadium.

Now, it is important to note that some predictors should be heavily correlated. Indeed, there is a high probability that if a team has really good attacker rating, its defenders should also be good since it probably means the team is a top-franchise with a lot of money to afford top players across the field. Same goes for midfielders and goalies. Therefore, we run a correlation matrix (Appendix 1) and we see that some of the variables are indeed correlated. We assume that it should not cause any issues since our dataset has many observations. Therefore, there should be enough instances when even highly correlated variables are sometimes not going in the same direction, enabling the model to calculate partial effects of each variable.

Also, even though we do not have interaction or polynomial terms, we choose to standardize our variables as some are not on the same scale. This will help the model identify the effect of each variable correctly. Next, for all of the models, we will use randomly shuffled training and test sets generated from a Scikit-Learn tool, each holding 80% and 20% of the data, respectively. Finally, note that we always look at predictions from the home team's standpoint. For example, when a model predicts a win, it means the home team is predicted to win the match.

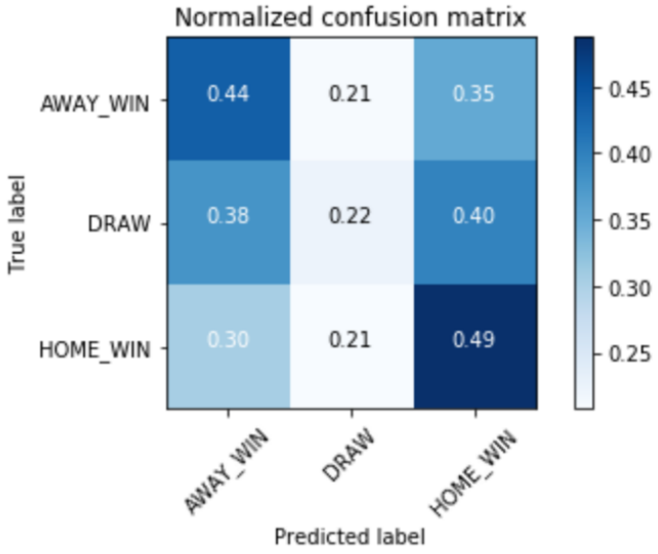
3 Models and Discussion

3.1 *K*-Nearest Neighbor Classifier

Our first model is a K-nearest neighbours (KNN) classifier. We can summarize the KNN classifier as a supervised learning model that will classify a test data point in the same category as that of its *k*-closest neighbours in the training set, i.e. the *k* data points that are the most similar to it. The similarity is usually in terms of Euclidean distance. Scikit-Learn's KNN classifier chooses the optimal parameter *k* for us.

We run a 3-fold cross-validation, which returns scores of 40%, 40% and 39%. In other words, the model correctly predicts the match outcome only around 40% of the time. The model's precision, recall and f-score are also quite bad. The model correctly predicts both true positives (precision) and true negatives (recall) 40.7% of the time. The confusion matrix is shown below.

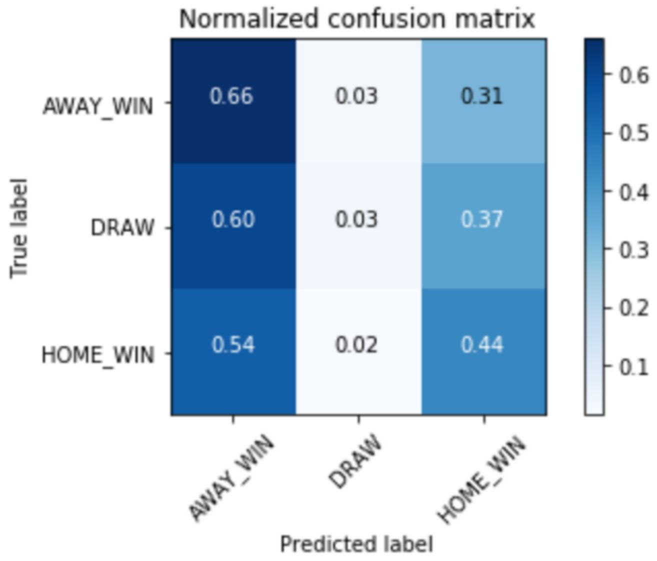
We see that the model is fair at predicting home losses (or away wins), since 44% is much higher than its observed mean of 29%. It fares slightly better than a random guess for home wins, with 49% vs. 46% observed probability. However, it does not beat a random guess for both home wins and draws. In fact, a drawback of the KNN classifier is that in skewed distributions, the most frequent category is so common among the *k*-nearest neighbours that it tends to do well for those labels but not well for less frequent labels. It seems to be the opposite in our data. Thus, we conclude that the KNN classifier performs poorly, and could not generate any edge in a betting context.



3.2 SGD Classifier

Our second model is a classifier that uses Stochastic Gradient Descent to minimize its cost function. In this case, it works in a similar fashion as multinomial logistic regression, but instead of returning probabilities it returns an actual category: win, loss or draw. This model randomly chooses weights in a linear model in order to make the Euclidean distance smaller and smaller at every iteration, and it does so until it feels it has reached a local minimum (Goodfellow, 2016).

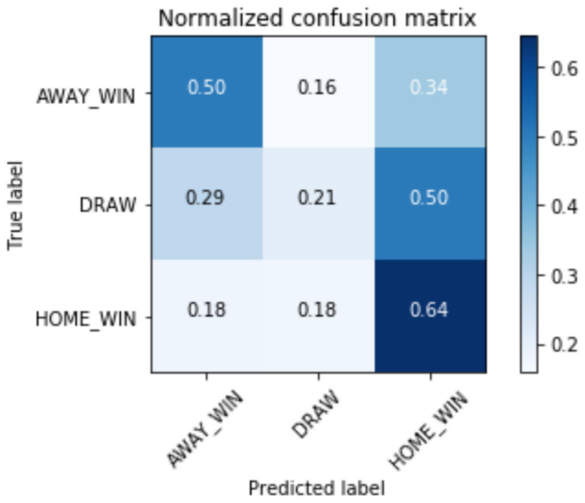
This simple model does worse than the KNN model, and does not beat a random guess. In a 3-fold cross-validation, its prediction accuracy sits at 46%, 25% and 29%. Coincidentally, both its precision and recall are low. We see from the confusion matrix how the model does poorly for ties especially. An interesting fact supporting this is the large variability in the accuracy of the model when running it on differently shuffled training and test sets, as it often restricts itself to predicting home wins and losses, almost completely disregarding draws. This could be due to the fact that the SGD reaches a local minimum that is not closely similar to the absolute minimum.



3.3 Random Forest Classifier

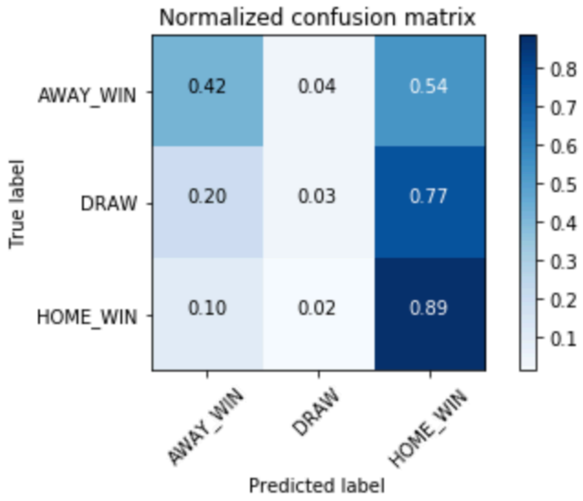
We then turn to a third model: the Random Forest classifier. Decision trees break down the input space into smaller Boolean components in an attempt to construct an exhaustive model to predict an outcome. The random forest then averages multiple decision trees on sub-samples of the training set in order to minimize over-fitting and improve accuracy (Goodfellow, 2016).

The accuracy of this model is much better than the previous models: a 3-fold cross-validation returns accuracies hovering very close to 50%, with an accuracy in the test set of 49.5%. This is slightly higher than our base case of pure guess of home win (46% probability). The confusion matrix below shows that the model is quite good at predicting home wins and losses, but not as good at correctly predicting draws. However, even though draw predictions produce many errors, the correct predictions are still quite close to the actual empirical average of 25%. Therefore, we conclude that this model fares much better than the previous two and should be considered in predicting activities.



3.4.1 Voting Classifier

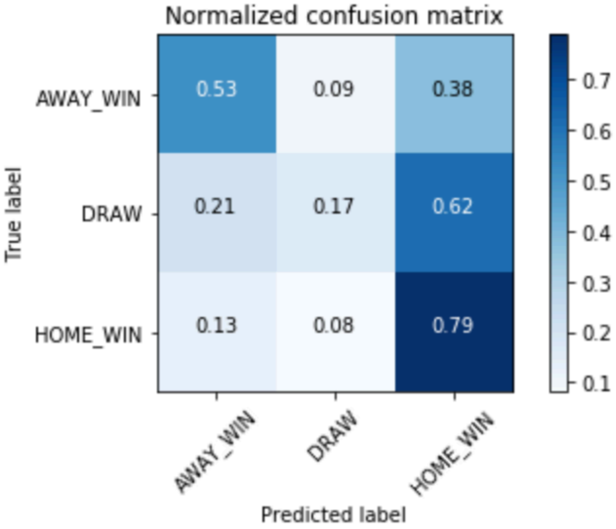
Next, we build a voting classifier out of three unique algorithms: Random Forests as seen above, Logistic Regression and Support Vector Machines (for Classification). We used a technique called *soft voting*, which allows the Voting Classifier to average the class probabilities generated by the individual classification algorithms and then itself predict the class that has the highest probability. Using this method, the Voting Classifier has an accuracy score of 54.07%, greater than any of the three individual classifiers. However, by judging its Confusion Matrix, we notice that this classifier is not effective at predicting draws. It almost always correctly predicts a Home Win (TP = 89%) but misclassifies almost all Draws as wins. This may be due in part by the fact that the SVM Classifier consistently votes that the match is a win, with 100% certainty. Furthermore, the Logistic Regression Classifier never predicts a Draw (it essentially sorts games into "Win" or "Loss" categories). This effect may be mitigated by adding a classifier that predicts draws more often, such as the KNN classifier above.



3.4.2 Adaptive (ADA) Boost Classifier

Next, we consider the ADA boost classifier, which is an ensemble learning method that uses a number of weaker classifiers (accuracy lower than the benchmark random guess) in an attempt to construct a stronger one (more accuracy, less error). It has been identified to work best with decision trees as a base classifier (Gandhi, 2018). However, when comparing empirical results, we achieved approximately 53% accuracy with Decision Trees compared to nearly 56% accuracy when using Random Forests as the base estimator.

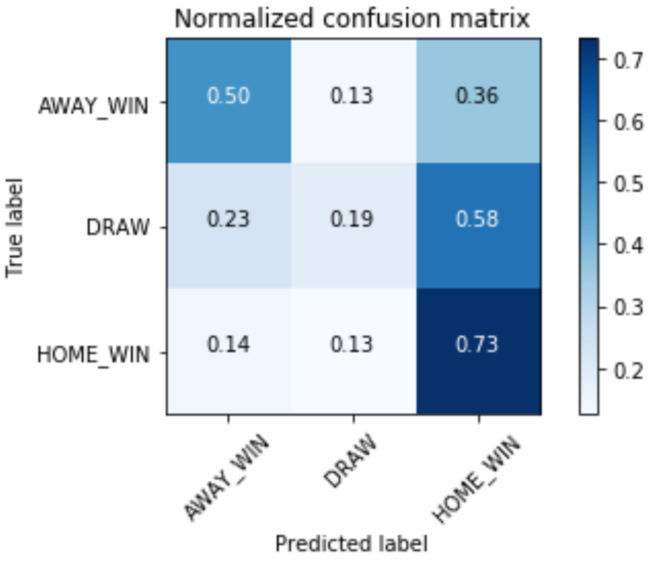
The ADA Boost classifier based on Random Forest correctly predicts 55.96% of match outcomes. This is the best model that our dataset can produce with the tools we have explored so far. It is limited when predicting draws, however, which is expected due to its smaller number of instances. The model is very good at predicting wins and losses, as shown in the confusion matrix below. The precision of the model is 53.2%, its recall is 55.96% and its overall accuracy 55.96%.



3.4.3 Gradient Boost Classifier

Similar to the ADA boost classifier, the Gradient boost classifier draws on an increasing number of models in order to minimize the loss function, but the models are all in the same family (e.g. logistic regression).

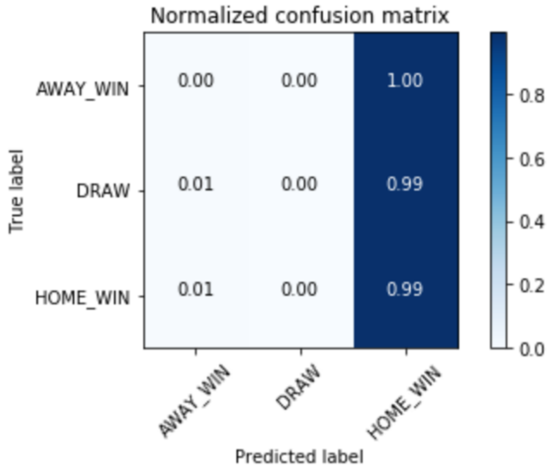
Based on logistic regressions, the Gradient Boost classifier correctly predicts 54.84% of the match outcomes. This is very close to its cousin ADA as it uses the same type of intuition. The confusion matrix is also similar, with a weakness in predicting draws.



3.4.4 Multilayer Perceptron

Finally, we turn to a Multi-Layer Perceptrons (MLP) classifier. MLP classifiers are used to approximate a function that takes our selected features as an input vector and returns a category—win, loss or draw—using a feedforward network. This final "output-layer" function is in fact a composition of multiple functions, as it embeds a number of "hidden-layer" functions comprising the network. The model learns this optimal hidden-layer mapping in order to produce the best output-layer function. The "deeper" the model, that is, the more hidden-layers and the wider each layer, the more "connections" the model can use to optimize itself (Goodfellow, 2016).

In this case, we set the dimensions of our MLP at 30-by-30. In other words, there are 30 hidden layers and they each have 30 unique activation functions. We use the Limited Broyden–Fletcher–Goldfarb–Shanno (LBFGS) solving method, which is an iterative "hill-climbing" method seeking a stationary point as a rule for optimality. Using a 3-fold cross-validation yields an accuracy of 46.32%. The model fails miserably at making intelligent predictions, opting instead to always predict a win. As a result, the confusion matrix, aptly named for this confused neural network, looks like this:



We tried playing with all sorts of hyper-parameters, changing the solver method, the hidden layer sizes, the initial learning rate as well as its behavior, changing the activation function and batch size, all to no avail. It seems the MLP is simply ill-suited for this task.

3.6 Discussion and Limitations/Drawbacks

Our results imply that the models predicting the highest number of correct occurrences across the three categories are the ADA boost and Gradient boost classifiers. This is certainly due to these models' ability to convert weak learners to strong ones. However, it's worth noting that the sequential nature of the Boosting algorithms means that they are quite time-consuming to train (they can't be pipelined, as each iteration is entirely dependent on the results of the previous iteration). However, in a case where the size of the data set is relatively small, like ours (less than 50,000 instances), this is not a severely limiting factor.

Although these two models fare quite well compared to a random guess, one limitation to their use in a betting context is the constant up-dating of the data. In football, players sometimes get transferred around so this needs to be accounted for. The same applies for unexpected injuries, where one would have to adjust team attributes to reflect the loss of a top player. One potential improvement would be to use a dynamic database connected to EA's "live stats" feature, which updates team lineups and players' ratings continuously through a season to more accurately reflect real life.

4 Conclusion

In this project, we test many popular machine learning models on an extensive dataset of Association Football match statistics. We benchmark the results to a simple random guess. A random guess will be correct 46% of the time when predicting a home team win, 29% of the time when predicting a home team loss, and 25% of the time when predicting a tie. The SGD classifier performs the worst with a low of only 28% accuracy and high variability. The *K*-Nearest Neighbour classifier also does poorly, with a 40% accuracy. The Random Forest classifier fares much better, with a 50% accuracy. This is starting to become interesting as it gains an edge on the random guess. Next, the MLP classifier does not seem to be fit for our dataset, as it sticks to predicting only home wins, thus returning a 46% accuracy. Finally, our ADA boost *ensemble* classifier does even better, as it embeds multiple models, with a 55.3% accuracy. Its cousin, the Gradient boost classifier, does almost as good, with 54.8% accuracy. We conclude that these last two models are the better options to predict match outcomes in our dataset.

Another interesting area for research in applying Machine Learning models to Association Football is the use of in-game feedback. Stadiums that are equipped with sensors and special cameras can be used to track on and off-the-ball player movement, identify which areas of the pitch are under the most pressure, notify the manager when a certain player is underperforming, or even recommend a strategy that could help a team win given an opposing team's style of play. These ideas may sound futuristic, but in sports like basketball, this is already the norm. Spectrum, a Los-Angeles based startup, is already delivering services similar to the ones mentioned to NBA teams.

To take this analysis one step further, one could take the results from these machine learning models and use bookmaker odds as a more advanced benchmark, in contrast with a simple guess. This could be a viable investment strategy if the model predicts outcomes with significantly more accuracy than bookmakers, by placing small bets on many matches at a time to take advantage of the law of large numbers. However, one would have to keep the dataset up-to-date with injuries, player transfers, and so on, in order to capture the true team attributes for any given match. This could become a quite tedious task when keeping track of multiples leagues at a time, unless one has access to EA Sports' live stats platform.

Finally, a different application of machine learning in football could be for team managers to use Principal Component Analysis in an attempt to identify which main components seem to affect winning rates. These data could then be used to tailor their management strategy, similar to what has been done in Baseball as illustrated in the book *Moneyball*.

References

Gandhi, R. (2018). “Boosting Algorithms: AdaBoost, Gradient Boosting and XGBoost”. Retrieved from: https://hackernoon.com/boosting-algorithms-adaboost-gradient-boosting-and-xgboost-f74991cad38c (accessed on December 20th, 2018).

Gandhi, R. (2018). “Gradient Boosting and XGBoost”. Retrieved from: https://hackernoon.com/gradient-boosting-and-xgboost-90862daa6c77 (accessed on December 20th, 2018).

Goodfellow, I., Bengio Y., Courville A. (2016). *Deep Learning*. Published: MIT Press. Retrieved from: http://www.deeplearningbook.org/. (accessed on December 20th, 2018).

Lewis, M. (2003). *Moneyball: The Art of Winning an Unfair Game*. Published: W. W. Norton & Company.

Lindberg, A. (2016). “FIFA 17's player ratings system blends advanced stats and subjective scouting”. ESPN FC: Retrieved from: http://www.espn.com/soccer/blog/espn-fc-united/68/post/3727011/sorrypep-possession-isnt-everything-just-ask-atletico-and-dortmund-footballs-medieval-minimalists (accessed on December 17th, 2018).

Schwartz, D. (2013). *Roll The Bones: The History of Gambling*. Winchester Books. ISBN 978-0615847788.

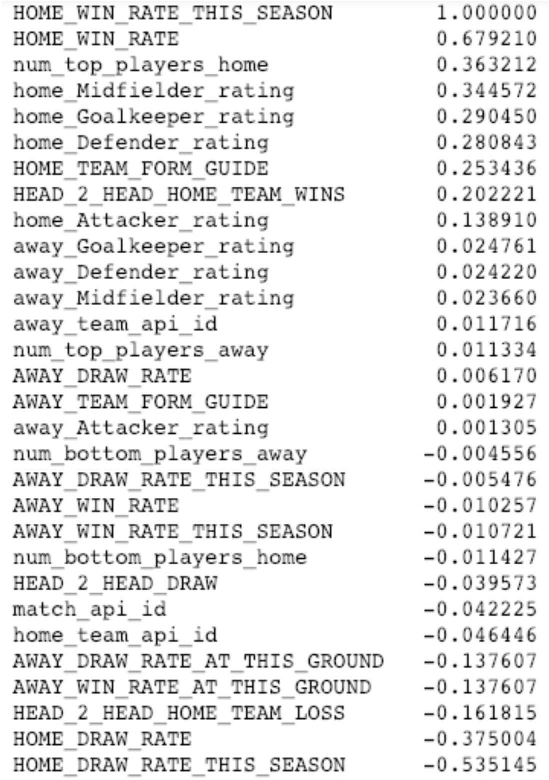
Second Spectrum Official Website: https://www.secondspectrum.com/

Statista (2018). “Global gambling market gross gaming yield (GGY) from 2001 to 2019 (in billion U.S. dollars)”. Retrieved from: https://www.statista.com/statistics/253416/global-gambling-market-gross-win/ (accessed on December 18th, 2018).

World Atlas (2018). “The Most Popular Sports in the World”. Retrieved from: https://www.worldatlas.com/articles/what-are-the-most-popular-sports-in-the-world.html (accessed on December 18th, 2018.)

Appendix

1 Correlation table (between Historical Home Winning Rate and each features)



```
HOME_WIN_RATE_THIS_SEASON       1.000000
HOME_WIN_RATE                   0.679210
num_top_players_home            0.363212
home_Midfielder_rating          0.344572
home_Goalkeeper_rating          0.290450
home_Defender_rating            0.280843
HOME_TEAM_FORM_GUIDE            0.253436
HEAD_2_HEAD_HOME_TEAM_WINS      0.202221
home_Attacker_rating            0.138910
away_Goalkeeper_rating          0.024761
away_Defender_rating            0.024220
away_Midfielder_rating          0.023660
away_team_api_id                0.011716
num_top_players_away            0.011334
AWAY_DRAW_RATE                  0.006170
AWAY_TEAM_FORM_GUIDE            0.001927
away_Attacker_rating            0.001305
num_bottom_players_away        -0.004556
AWAY_DRAW_RATE_THIS_SEASON     -0.005476
AWAY_WIN_RATE                  -0.010257
AWAY_WIN_RATE_THIS_SEASON      -0.010721
num_bottom_players_home        -0.011427
HEAD_2_HEAD_DRAW               -0.039573
match_api_id                   -0.042225
home_team_api_id               -0.046446
AWAY_DRAW_RATE_AT_THIS_GROUND  -0.137607
AWAY_WIN_RATE_AT_THIS_GROUND   -0.137607
HEAD_2_HEAD_HOME_TEAM_LOSS     -0.161815
HOME_DRAW_RATE                 -0.375004
HOME_DRAW_RATE_THIS_SEASON     -0.535145
```