Predicting Soccer Outcomes With Multi-View Learning

Pim Hoeven[1], Omar Elbaghdadi[2], Robert Smit[3], Krijn van den Berg[4], Frank Laeven[5]

April 2, 2017

p.s.l.hoeven@student.vu.nl[1], o.n.elbaghdadi@student.vu.nl[2], r.c.smit@student.vu.nl[3], k.a.vanden.berg@student.vu.nl[4], f.s.laeven@student.vu.nl[5]

Student Registration Numbers: 2542987[1], 2577566[2], 2555333[3], 2558600[4], 2581235[5]

Abstract

In this report, we try to forecast outcomes of soccer matches using various Machine Learning techniques. Our forecasts are based on data on European soccer matches from 2008 to 2016 and FIFA player rankings. We find that our models perform better than simply guessing the most prevalent outcome, 'home wins' and equaling the accuracy bookmakers achieve, indicating that it is possible to identify patterns in a soccer game that is subject to many random factors. The most important variables to predict match outcomes are the win ratio's of the teams. The logistic ensemble of base models performed best with 53% accuracy.

Contents

1 Introduction

In this research project, we use various Machine Learning algorithms to predict the outcome of soccer matches. The exact score of matches is not within the scope of our research, we focus only on win for the home or away team, or a draw. The data we use is from kaggle.com and contains results and information on almost 26.000 professional matches played between 2008 and 2016 in a number of European competitions. It also contains information on teams and players from popular soccer computer game FIFA. We will train various models and ensembles of models to see what works best in predicting those outcomes. Our objective is to use machine learning techniques to maximize the accuracy of our predicted outcomes, i.e. maximize the percentage of soccer matches correctly predicted.

1.1 Motivation

Soccer is one of the most beloved and popular sports in the world. The outcome of matches seem to rely many different factors, making the outcome of a match very hard to predict. Together with the abundance of data and our personal interest in soccer, predicting soccer outcomes is a very interesting research topic.

Another reason to try to predict the outcome of soccer matches is a possible economical gain. Since soccer is as popular as it is, the gambling on matches is also a very popular practice. Many betting offices would be very interested in a model that can predict the outcome of soccer matches accurately. By leveraging the power of machine learning, we hope to be able to beat the bookmakers.

1.2 Hypothesis

We will be building various models in R and Python, as well as a number of ensembles. We expect that the neural network performs best, because we do think the complexity of a soccer match cannot be described by linear models. Furthermore we think that the ensembles will outperform the base models, because they can use the qualities of each individual model to predict the outcome. Our expectation is that all ensembles outperform the base models.
The performance of our model is, like soccer matches, hard to predict. With randomly guessing the three outcomes (Home Win, Draw, Away Win), we would get about 33% correct. However, in 46% of the cases in our data the home team wins, making always predicting 'Home' and achieving an accuracy of 46% the base case. According to the data owner[1] "bookies get it right about 53% of the time" and that was also his score. Therefore, we expect an accuracy of about 53%.

1.3 Literature Review

There have been several approaches to predicting soccer results. Some researchers try to predict the complete score of the match while others solely look at wins, draws and losses. Part of every approach however is that researchers look at team strength, home advantage and team form. The way that all these subjective traits are being calculated and represented vary from research to research. Some even combining a few into one indicator. When looking how previous researches tried to represent team strength one can see that there is a wide variety of information

[1]Hugo Mathien, *European Soccer Database*, `https://www.kaggle.com/hugomathien/soccer`

taken into account.

Dyte & Clarke (2000) used the FIFA ratings of the 2 competing national teams for their simulation model of the World Cup. The FIFA ratings are calculated using all the matches between senior national teams. Points are awarded based on relative strength. Points are allocated for every result based on the relative strength of the teams before the match with extra points for away wins and goals scored. Also the importance of the match are taken into account. All points earned are then gradually reduced to zero over a period of eight years.

Goddard & Asimakopoulos (2004) used a combination of win ratio variables in their model for matches across different divisions in England and Wales where they allowed for individual components to have different contributions when calculating the team quality by looking for example in which division they were and are playing in. *Constantinou, Fenton & Neil (2012)* however used results of previous seasons with higher uncertainty for older seasons, the current season and a subjective team strength indicator to calculate the team strength.

According to *Carron, Loughhead & Bray (2005)*, we can see in how many ways playing at home can give an advantage to that team. From the effects of the location, psychological to behavioural effects. When it comes to home advantage some like *Dyte & Clarke (2000)* just use a parameter for the venue (home, away or neutral) where they modelled the parameter for the effect of away to be zero. *Baio & Blangiardo (2010)* on the other hand have a single set parameter representing the home advantage for all teams throughout the season which was significantly different from zero within their research. But in other researches and models home advantage comes out in other ways mainly when taking form into consideration. The advantage of location is something *Goddard & Asimakopoulos (2004)* had in their model by having a parameter with the geographical distance between the competing teams. Possibly catching the effects of local derbies and long-distance travel as was described by *Carron, Loughhead & Bray (2005)*. While the psychological and behavioural effects are taken into consideration by parameters like cup elimination, match significance, (*Goddard & Asimakopoulos (2004)*) or managerial impact, tiredness and confidence (Constantinou, Fenton & Neil (2012)).

Lastly an often used predictor is team form. Team form can be based on various available information from looking to the last few matches, looking at past home and away matches separately to the availability of (subjectively) decided key players. While most researches look at the recent results of both teams through experimentation *Goddard & Asimakopoulos (2004)* came to the conclusion that recent home results are better predictors than its recent away results for the home team and similarly for the away team. While *Constantinou, Fenton & Neil (2012)* used weights of [2/3, 1/3] for home form and away form respectively. They combined this with key player availability and a returning effect of first team players.

1.4 Description of data and intuition for leaving variables out

For our research we used a database from Kaggle. This database contains a dataset of 25979 European soccer matches from the seasons 2008 to 2016, of which more than 10000 have detailed match events. The dataset contains match dates, league, teams, player ID and position and match outcomes. The detailed matches contain, in addition to that, match events like what type of goal was scored, possession, cards given and fouls. It also contains datasets of the attributes of 11060 players and 299 teams. These attributes are sourced from the EA Sports' FIFA video game series. Examples of these attributes are player ratings for different stats, like

passing, sprint speed and dribbling, and overall ratings and things like build up and chance creating type and rating per team. We will now look at some descriptive statistics of some of the variables that seem important.
The descriptive statistics of match outcomes can be seen in figure 1. In our data the outcome value equals the difference in goals, so a positive outcome means that the home team has won the match and zero and negative outcome means ‘draw’ or ‘away team win’ respectively. Important to notice here is that the mean is positive, this indicates that there is some sort of home advantage. Furthermore, notice that the Jarque-Bera test for normality has a p-value of 0%, so the hypothesis that the match outcomes are normally distributed is rejected.

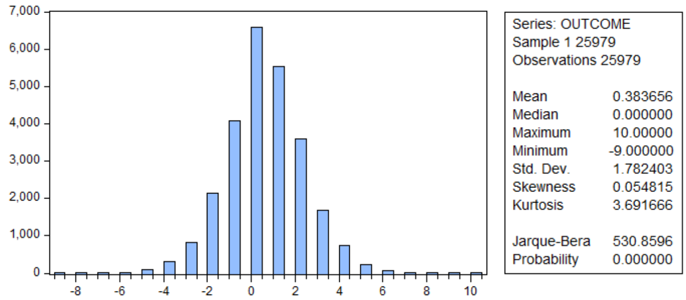

Figure 1: descriptive statistics and histogram of match outcomes

The total frequencies of the different match outcomes can be found in table 1.

Table 1: match outcomes

	frequency	percentage
home win	11917	45.9%
away win	7466	28.7%
draw	6596	25.4%

If we compare the win ratios of all teams for matches played at home and matches played away, of which the descriptive statistics can be seen in figure 2 and 3 respectively, we can see the chance to win when playing a home match seems higher than for an away match. This may be another indication of some sort of home advantage.

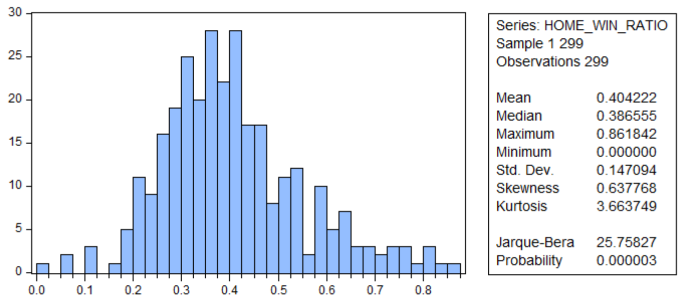

Figure 2: descriptive statistics and histogram of win ratios for home matches

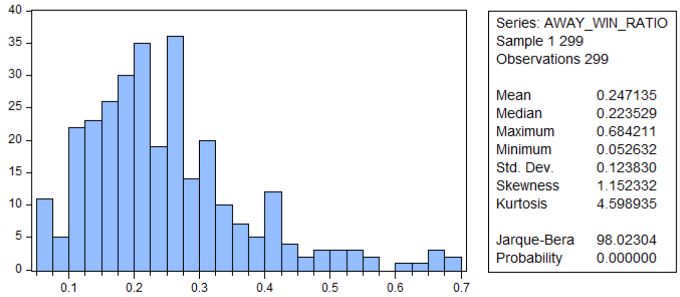

Figure 3: descriptive statistics and histogram of win ratios for away matches

Another interesting variable to look at is the overall rating of the players. EA uses a team of 9,000 members, including professional scouts as well as season-ticket holders of clubs around the world. This team watch players submit the feedback on players to EA. 5.4 million pieces of small information are put into creating a player's attributes. Overall rating is then calculated by a formula that weighs these attributes for each particular position. The descriptive statistics of the overall player ratings can be found in figure 4. Note that the overall rating is not normally distributed as well.

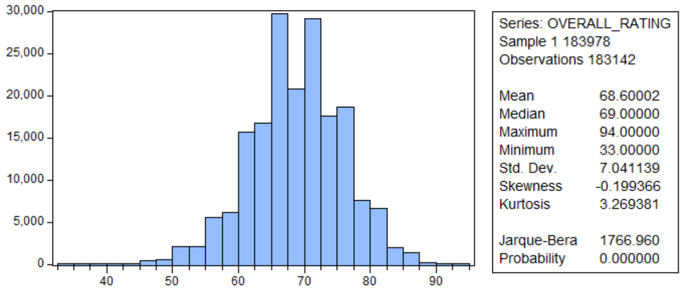

Figure 4: descriptive statistics and histogram of the overal player ratings

Figure 5 shows the scatter plot of the player IDs and player age. In this plot there seems to be negative correlation between the player ID and the age of the player. To determine the significance of this correlation we did a linear regression of which the results can be found in figure 6. The relation is significant and it seems that players with higher IDs are younger and could therefore have less experience, thus we assume that player ID's are assigned based on their first appearance on the pitch.

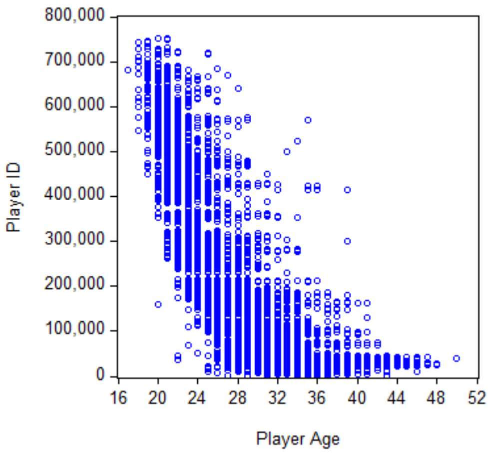

Figure 5: the scatter plot of player age and ID

Dependent Variable: PLAYER_API_ID
Method: Least Squares
Date: 03/31/17 Time: 12:03
Sample: 1 11060
Included observations: 11060

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	815061.7	5552.490	146.7921	0.0000
AGE	-22191.89	184.0411	-120.5812	0.0000

R-squared	0.568010	Mean dependent var	156582.4
Adjusted R-squared	0.567971	S.D. dependent var	160713.7
S.E. of regression	105635.4	Akaike info criterion	25.97356
Sum squared resid	1.23E+14	Schwarz criterion	25.97488
Log likelihood	-143631.8	Hannan-Quinn criter.	25.97400
F-statistic	14539.82	Durbin-Watson stat	1.971208
Prob(F-statistic)	0.000000		

Figure 6: results of regression of ID with regressors a constant and age

2 Method

Our method is visually described in the flowchart in figure 7. We used three separate input datasets: Match data (describing the match statistics like teams, date, scores and teams), Team Attributes (with tactics according to FIFA) and Player Attributes (with FIFA information on the players. We handle the data as described in section 2.2 and then split the data in two sets: matches played before 2016 (A) and during 2016 (B) as our train and test set respectively. The reason we use this split is that if we want to predict future matches, the best approach is to try to predict future matches. All our base models are trained on this train set A. Then, our training set is randomly split in two sets of equal size ($A1, A2$). One of those, say $A1$ is used to train the base models for use in the ensemble models, the predictions of those models for set $A2$ are used to train our ensemble models. Our final ensemble models are built on the prediction made for B by the base models trained on the entire train set A. With this strategy we can use the entire 2016 set to test our results.
Since our goal is to predict as many matches correctly as possible, the best metric to assess the quality of our models is the accuracy the model has on the test set.

2.1 Implementation

The goal of our research is to eventually be able to predict the outcome of soccer matches as good as possible. Therefore we will use accuracy to assess model performance. Our best performing model can, if it performs good enough, be implemented for different purposes. The most obvious one is to use our model to bet on soccer matches, but another purposes of our research could be to use our model to give advise to soccer teams. Our model could then be used to determine their winning chance and factors that should be changed in their team to increase their chances to win.

2.2 Preprocessing the Data

In the data handling stage (pre-processing stage), the first thing that was done was removing all the data that was not known before the match was played. This includes, but isn't limited to, the amount of yellow/red cards and the goals scored.

The players were matched with their ratings by their player api id. Even though the line-up of a team is not known days in advance of a match, it is known before the match starts and thus could be included in a betting strategy. The data was therefore included in the models. In order to reduce the complexity of the models the choice was made to only use the overall ratings. Since it is a representation of their overall skill, this shouldn't drastically change the final results of the models. For simplicity's sake, some other variables were removed. For instance, coordinates of player positions are replaced by team formations and bookmaker's quotes are removed. Variables we assumed to be important but were not already included were calculated. This included a total win/loss ratio (to assess the quality of a team), the result of the last five matches (win and loss ratio, goals scored and received) as a measure of fitness. Besides, dates were transformed to epoch time. Finally, all the matches that had missing data in the columns that are to be used were excluded. The way the data was handled can be seen in figure 7.

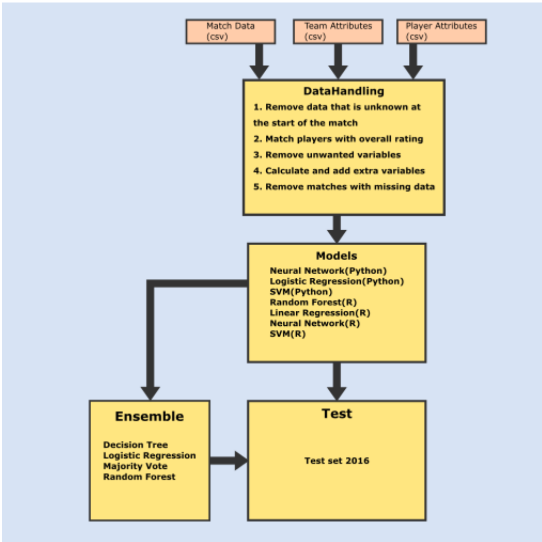

Figure 7: Layout of the program architecture

2.2.1 Feature Selection

In the data set, a lot of features to fit on are available. However, having too many features introduces higher variance and over-fitting. We want to select the features that have the most effect on our explained variable. We do this by using a Recursive Feature Elimination (RFE) algorithm.

The goal of RFE is to select features by recursively considering smaller and smaller sets of features. First, an estimator is trained on the initial set of features and weights are assigned to each one of them. Then, features whose absolute weights are the smallest are pruned from the current set features. That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached.

3 Models

We have used both R and Python in this project. This way, everyone was able to use their preferred language and we were able to compare the performance of packages for R and Python. Both software packages were used to build three models, R was also used for preprocessing the data and ensembling models. Our strategy is to build and tune three base models extensively (a Neural Network, a Logistic Regression and a SVM, in Python) and three base models quickly (with default parameters, in R), so we have more input for our ensemble models.

3.1 Neural Network

A neural network was built using the python package sci-kit learn. In order to test what features added to the accuracy of the model, multiple models were tested using different features. The first step was testing what features contributed to the model, and what features did not. 50-fold cross-validation was used in order to get a good estimate on the true accuracy of the model. Since this is so computationally heavy, the features were divided into groups in order to minimize computations. In order to group the features, the following assumptions were made:
Assumption 1 (Uniform Role Importance)**.** *Either all of the player ratings have a positive effect on the model or none of them do.*
Assumption 2 (Uniform Win-Rate Importance)**.** *Either all of the win rates have a positive effect on the model or none of them do.*

The features were grouped in the following way: Player ratings(both teams), Team win/loss ratios both home and away and in the their last 5 matches. In order to test the possible combination of features all 7 combinations were tested. The results can be seen in figure 8. It is clear that only including win rates gives the best accuracy. Therefore, the eventual model will only use the win rates of the teams.

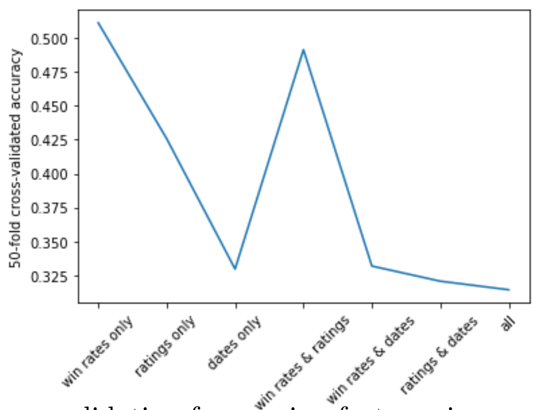

Figure 8: Cross-validation for varying features in a neural network

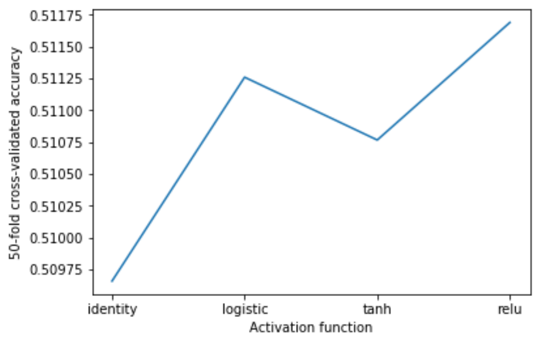

Figure 9: Cross-validation for varying activation functions in a neural network

Next multiple activation function were tested in order to see which one gave the best results.

Again 50-fold cross-validation was used to measure the accuracy. The function that were considered are the tanh, the logistic function, the relu function and the identity function. These are the activation function that sci-kit learn supports. The results can be seen in figure 9. the The relu function has the highest accuracy so that will be used in the model.

Finally, in order to decide how many neurons the hidden layer should have, models with 1 to 40 neurons were tested using 50-fold cross-validation. The results can be seen in figure 10. As can be seen the accuracy rises from 1-5 neurons but flattens out from there on. The absolute maximum accuracy was attained at 19 neurons with an accuracy of 51.4%. Therefore the eventual model will use 19 neurons and one hidden layer.

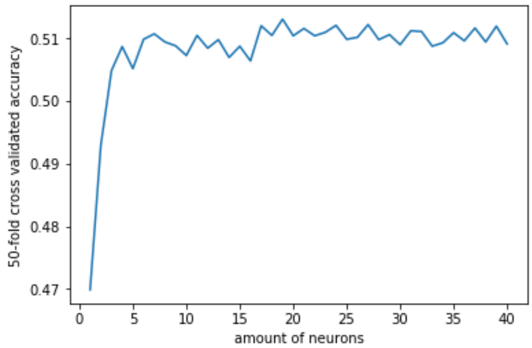

Figure 10: Cross-validation for varying amounts of neurons in a neural network

3.2 Logistic Regression

A logistic regression was performed to predict the outcome of the match using the package sci-kit learn in Python. In order to test the features that added to the accuracy of the model, a 50-fold cross validation of the features was performed using assumption 1 and assumption 2 as mentioned in the section Neural Network. The results can be seen in figure 11. It shows clearly that the model performs best when only including the win rates of the teams.

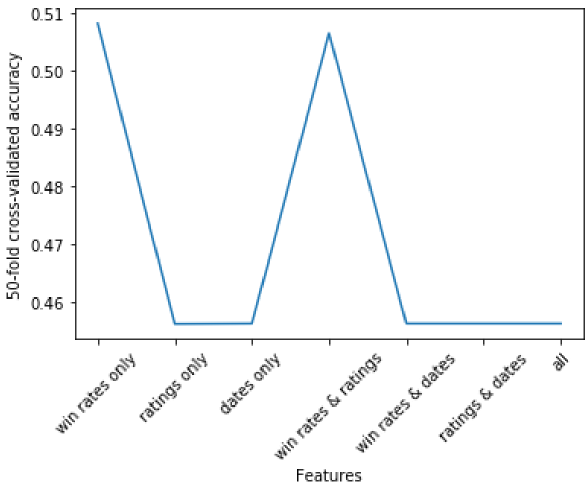

Figure 11: Cross-validation for varying features for a logistic regression

3.3 Support Vector Machine (SVM)

We have used two different models for SVM's. One in Python and one in R.

A support vector machine constructs a hyper-plane or set of hyper-planes in a high or infinite-dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class, so-called functional margin, since in general the larger the margin the lower the generalization error of the classifier.

Oftentimes, sets of classes are not linearly separable. Using the 'kernel trick', it is possible to map the data to higher-dimensional space, presumably making the separation easier in that space. We have used a radial basis function (rbf) kernel as opposed to for example a linear kernel, since the classes were clearly not linearly separable. (see Fig 12).

The linear and rbf kernels $K(x, x')$ on two samples x and x', represented as feature vectors in some input space, are defined as:

- linear: $K(x, x') = \langle x, x' \rangle$
- rbf: $K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) = \exp\left(-\gamma \|x - x'\|^2\right)$, where $\gamma = \frac{1}{2\sigma^2}$

The rbf kernel depends on a parameter $\gamma = \frac{1}{2\sigma^2}$.

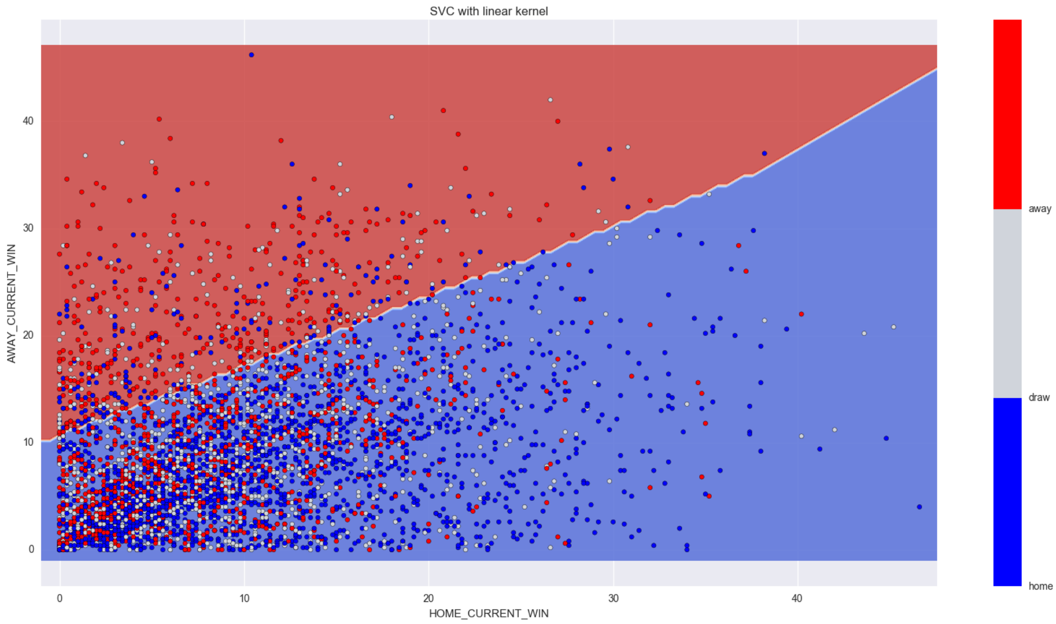

Figure 12: An SVC with a linear kernel. The classifier was fit on two features: the home current win ratio and the away current win ratio.

Tuning the Hyper-Parameters

The SVM with an rbf kernel is parametrized by hyper-parameters C and γ.

- γ

  Intuitively, the γ parameter defines how far the influence of a single training example reaches. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors. The behavior of the model is very sensitive to the gamma parameter.

- C

  The C parameter trades off misclassification of training examples against simplicity of the decision surface. A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly by giving the model freedom to select more samples as support vectors.

We tune these parameters by performing an exhaustive gridsearch, using 5-fold cross-validation for each set of parameters. We will use the set of parameters with the highest hit rate. The parameters have both been searched in the range $[1 \times 10^{-3}, 1 \times 10^{-2}, \cdots, 1 \times 10^{3}]$.

The optimal parameters were

Parameter	Value
C	100
γ	0.001

If we plot the points with these parameters, we see that the resulting graph is vastly different from the last one (Fig. 13).

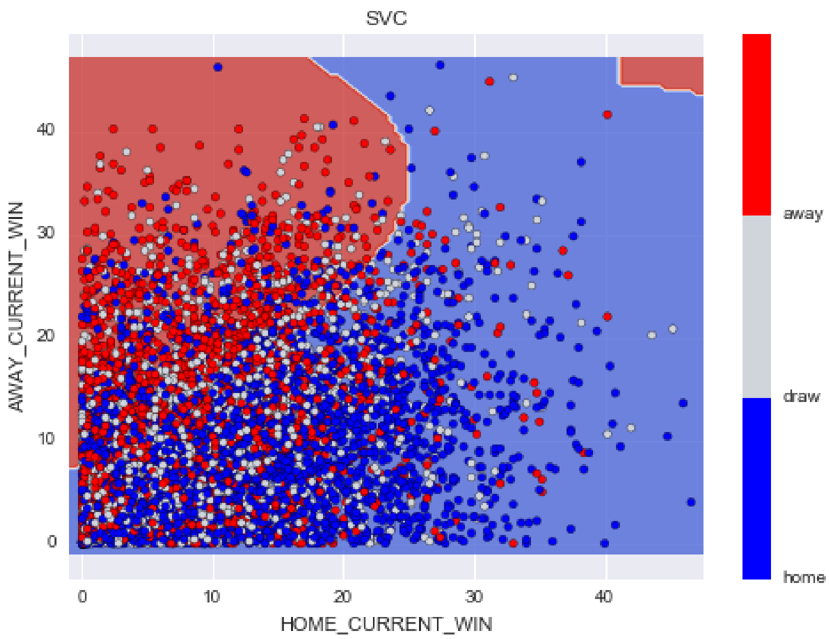

Figure 13: An SVC with an rbf kernel. The classifier was fit on two features: the home current win ratio and the away current win ratio, using parameters $\gamma = 0.001$, $C = 100$.

3.4 Tree Based Models

Due to the interpretability of decision tree's, we hoped this model would give us valuable insights. Unfortunately, when trying to predict on a test set with various parameters, the accuracy was far below the base case prediction (home wins). This is probably due to the large amount of features, making the decision tree fit the noise in the training data (e.g. in one case the id of player 4 was more important than the results of the previous matches). To overcome the problem of overfitting, we tried a random forest using the randomForest package in R. We used the standard configuration because the results of the first tests of this algorithm using default settings where very good and we expected that even slight improvements of this model would take a lot of time and thus improving this model would be inefficient given the purpose of this model.

3.5 Linear Regression

A linear regression was primarily built to assist in the ensemble models. We noticed that our elaborate models did not perform very well at predicting draws, so we tried to build a model that would. The regression was built to predict the difference in goals between both teams. When the absolute difference was smaller than 0.5, the model would predict a draw.

3.6 Stacked ensemble models

We used four stacking ensembles on six base learners to improve our model. The basic idea of a stacking ensemble is to use the predictions of base models as input features to predict the same variable that the base models tried to predict. The base models we used are all models described above: a Neural Network, two SVM's, a Linear and a Logistic Regression and a Random Forest. The meta-learners we used are voting (according to scheme 2), a Decision Tree, a Random Forest and a Logistic Regression.

Table 2: Voting scheme

Situation	Prediction
$\#away = \#home$	draw
$\#draw > max(\#away, \#home)$	draw
$\#home > \#away$	home
$\#away > \#home$	away

Problems to avoid

When modelling a problem there are certain pitfalls to avoid. One of those pitfalls is over-fitting. If we take the neural network for example, it is important not to use too many hidden layers because this might lead to the model fitting the noise of the test set instead of the useful patterns. This is of course true for most of these models. The way this was dealt with was by cross-validating the data and using grid search in order to minimize the potential of over-fitting.

Another potential problem is including too many independent variables. The more variables included, the more complex the models will become and the more data is needed to find their patterns. The way this was handled was by analysing the effect of excluding certain features from the model and seeing what effect this has on the accuracy. It is important to note that two independent variables could individually be poor predictors but together be great predictors in a model. In statistics this is called moderation. For example, when predicting if someone has a recessive trait, if the mother has it and the father doesn't or vice versa, the child will not have that trait, but if both have it then so will the child. Therefore it is important to test all combinations and not just the features individually.

The matches that contained missing data were removed from the data set in order to be able to make the predictions. One could wonder whether there is selection bias in this, are the matches that were removed significantly different from those that were not? This isn't a problem if the test set has the same bias as that in the training set since it will not affect the accuracy of the predictions. It is a problem if this is somehow different in the test set, for example if there was more missing data in the past than there was now. In order to check this we compared the percentage of deleted data of the training set with the test set. These were both around 10% so there is no evidence of such an effect.

4 Evaluation

We have found that the variables we introduced have significant effect on match outcome: the win and loss ratio's (total win ratio and win ratio of last 5 matches) and goals scored and received (average per match total and average of last five matches) all influence the model. Furthermore, player id's have significant influence on the outcome, which seemed strange until we realized what they (indirectly) represented: player experience. The individual player ratings from computergame FIFA have no significant influence of the outcome, probably because that is already reflected in the 'team strength' represented in both recent and long term win ratio's. The results of our models can be found in table 3. Contradictory to what we expected, the Neural Network did not outperform the base case of predicting 'Home Wins'. In fact, the Neural Network and Linear Regression were the only two models that perform significantly worse than the other models. We think the similarity in accuracy is caused by the difference in difficulty of predicting an outcome: Ajax - Feyenoord is more difficult to predict than Feyenoord - Go Ahead Eagles.
Ensemble models did outperform all base models except the SVM's, which is due to the fact that the ensemble relies heavily on the SVM for its prediction. Therefore we see that the confusion matrices of the ensembles and the R SVM are almost equal (table 4).
The prediction of our own accuracy was very accurate. We correctly predicted approximately 53% of the matches in the test set, which is equal to the score most bookmakers achieve.

Since the ensemble logistic regression has the highest accuracy on the test set this is the model that will be selected. Although its accuracy is practically the same as the ensemble decision tree, it is preferred since it gives weights to the different models while the ensemble tree does not.

Model	Accuracy	Lower bound 95% CI	Upper bound 95% CI
Ensemble Logistic Regression	0.535	0.509	0.561
Ensemble Decision Tree	0.534	0.508	0.561
SVM (R)	0.534	0.507	0.560
Ensemble Majority Vote	0.527	0.500	0.553
Ensemble Random Forest	0.526	0.500	0.552
SVM (Python)	0.523	0.497	0.549
Logistic regression	0.518	0.491	0.544
Random Forest	0.510	0.484	0.536
Linear Regression	0.462	0.436	0.488
Neural Network	0.459	0.433	0.486

Table 3: Test set accuracy of the models

Table 4: Confusion matrices of featured models

n = 1422	**Reference**		
Prediction	away	draw	home
away	190	77	68
draw	0	0	0
home	245	271	571

Ensemble using Logistic Regression

n = 1422	**Reference**		
Prediction	away	draw	home
away	197	84	88
draw	0	0	0
home	245	271	571

Ensemble using Random Forest

n = 1422	**Reference**		
Prediction	away	draw	home
away	188	77	68
draw	1	0	0
home	246	271	571

SVM (R)

n = 1422	**Reference**		
Prediction	away	draw	home
away	125	40	28
draw	228	172	252
home	81	136	359

Linear Regression

5 Conclusion

We have tested a lot of models, some of them more extensively than others. It is only a pity that our prediction accuracy did not break away from 53.5%. Someone on Kaggle had run a Gaussian Naive Bayes classifier with Principle Component Analysis, yielding a prediction accuracy of 55%.

In conclusion, the hypothesis that a neural network could best describe the data turned out not to be true. The neural network was the model with the lowest accuracy on the test set. A reason for this might be that although steps to avoid over-fitting were made, the network still over-fit on the training data. It did, however, turn out that the ensembles were better at predicting the outcomes when compared to the standalone models. The model that was selected for making the final predictions was the ensemble logistic regression. The final accuracy of the ensemble on the test set was 53.5% which is comparable to bookmaker's predictions. Things that went well were the use of ensembles in predicting the soccer outcomes. They tend to have higher accuracies than the standalone models. SVM, logistic regression and a random forest are good at predicting home wins with high accuracy. Things that could be improved upon are the ability to predict draws and away wins. It is also likely that the neural network over-fit on the training data so further steps to combat this should be taken in the future.

6 Further Research

Further research could be focused on increasing the accuracy of the models by testing to see the effect of different variables on the outcome of games. For example, it could be interesting to see what effect different referees have on the outcome of a game. Referee bias could have an influence on the outcome of a game by favoring the home/away team or even on the next game if he is more likely to give red/yellow cards. There are of course a lot more variables that can be taken into account, level of fatigue(could be measured in days since last match), the weather, relative importance of a match, etc.

Furthermore, the models in this research now solely focus on predicting the outcome of matches. The model can be, and has already been in other research, extended to predict the goal distributions of the teams and to predict a whole tournament or competition, for example the league table at the end of the season. This could be useful in designing a profitable betting strategy.

Further, FIFA ratings of individual players change, as do the clubs they play for. If one would be able to predict those changes or use previous changes one could make better predictions of the matches. Possibly even how those changes within the FIFA game relate to the form of the team. A Bayesian learner is a method that seems very suitable for this kind of modeling.

Lastly, as mentioned earlier the models used made relatively poor prediction of draws and away wins and the ensembles could therefore greatly benefit from a model that better fit this subset of the data.

Group Contribution

Table 5: Group contribution

Student	% of participation
Frank	25
Omar	25
Pim	25
Robert	12.5
Krijn	12,5

Source Files

The source files for this report can be found at the following location:
`https://www.dropbox.com/sh/x6o6j9bdhzwe9xb/AAAKwioSoPqF99WmjAsmDq_Ga?dl=0`

References

[1] D. Dyte and S. R. Clarke (2000), *A Ratings Based Poisson Model for World Cup Soccer Simulation*, The Journal of the Operational Research Society, volume 51, number 8: 993-998

[2] John Goddard and Ioannnis Asimakopoulos (2004), *Forecasting Football Results and the Efficiency of Fixed-odds Betting*, Journal of Forecasting, voume 23, number 1: 51-66

[3] Albert V. Carron and Todd M. Loughhead and Steven R. Bray (2005), *The home advantage in sport competitions: Courneya and Carron's (1992) conceptual framework a decade later*, Journal of Sports Sciences, volume 23, number 4: 395-407

[4] Gianluca Baio and Marta A. Blangiardo (2010), *Bayesian hierarchical model for the prediction of football results*, Journal of Applied Statistics, volume 37, number 2: 253-264

[5] Anthony C. Constantinou and Norman E. Fenton and Martin Neil (2012), *pi-football: A Bayesian network model for forecasting Association Football match outcomes*, working paper

[6] Matt Porter (2016), *How FIFA 17's ranking is calculated*, `http://www.ign.com/articles/2016/09/27/how-fifa-17s-rankings-are-calculated`

[7] Sci-kit Learn *Package used for Machine Learning Algortihms* `http://scikit-learn.org/stable/`