Soccer Match Results Prediction: Final Report

Ningning Long

May 4, 2017

1 Introduction

Soccer is an appealing sport all around the world, and it has many fans. The popularity of it generates an activity, which is considered as soccer betting. The soccer gambling industry contains many bookmakers, who set soccer odds for each match and make profit from the soccer betting activity. Thus, in this project, we are trying to find a statistical model that can achieve the soccer outcome prediction accuracy as high as possible. In this case, with repetively betting, betting via the model may allow law of large numbers to assure us profit.

The data for this project comes from Kaggle, and the dataset contains 25,000 European Professional Football matches information and around 10,000 players attributes. The matches come from 8 seasons from 2008 to 2016. More details about the data will be covered in next section.

Since we want to achieve the predication accuracy as high as possible, we want to know how much accuracy is considered as high with this dataset. According to the dataset provider on Kaggle, namely Hugo Mathien, the bookmakers get it right about 53% of the time and home team wins about 46% of the time among all the matches in the dataset. Thus, for every soccer match, if we always bet on the home team for winning, we can achieve around 46% accuracy. That is to say, the baseline of prediction accuracy with this dataset is 46%. However, we want to bet the bookmakers, so we will try to raise the accuracy score higher than 53%.

In order to find the right methods and features to start the project, we checked several previous studies. Generally, there are two types of modeling, which are goal modeling and result modeling.

Instead of predicting the result directly, goal modeling is often based on predicting the goals scored using estimates of team strength from prior games. For instance, one of the early study is created by Mehrez and Hu [1], who used current league ranking as an estimator of team strength and showed that the distribution of goals closely follows the Poisson. Later, a more comprehensive prediction study has been conducted by Rue and Salvesen [2], and they built a model with home and away teams as bivariate Poisson process.

Another popular method is result modeling. For example, Goddard and Asimakopoulos analyzed games from 15 seasons by ordered probit regression [3]. They used various features in the model, such as performance of a team in previous seasons, recent results including goals scored and conceded, the distance travelled by the away team, etc. These studies are valuable resources in our feature selection process.

Also, machine learning methods were used in previous studies for result prediction as well. For instance, Hucaljuk and Rakipovic evaluated an ensemble of methods, including Random Forests and k-Nearest Neighbor, on the 2008-2009 European Champions League [4]. Also, a neural network model was created by Huang and

Chang to predict the finals stage of the 2006 World Cup [5]. These studies provide us various approaches for modeling.

However, a previous research conducted by Goddard [6] shows there is an insignificant difference in predictive performance between goal modeling and result modeling. Thus, after read the previous studies and formatted the data, we chose to apply result modeling with multinomial logistic model, Random Forests and K-Nearest Neighbor. Also, we considered the naive bayes model as well.

2 Data

Hugo Mathien, the data provider in Kaggle, found the data of real soccer matches and betting odds from several different websites. However, the players and teams attributes are extracted from EA Sports FIFA games instead of real soccer matches. The final dataset provided on Kaggle is a SQLite dataset, containing 7 tables.

The Country table contains country name and id of the European Professional Football matches, and the League table contains league name and id. The Player and Player_attributes tables contain information over 10,000 players, including their id, name, birthday, height, weight, overall rating, crossing rate, heading accuracy, short passing accuracy, etc. The Team and Team_attributes tables contains information of 299 teams, including their id, team name, ability to create passing, ability to create shooting, etc. The most important table is the Match table, which contains information about 25,000 matches of 8 seasons from 2008 to 2016. The Match table contains data about goal of each team, the players id for each team, date of the match, id for each team, several bookmakers odds for each match, etc.

Our initial hypothesis is that the results of a soccer match can be predicted by the results of past three matches played by the home team and the away team in the same season. We believe these features can reflect the teams' performance in prior games, which is considered as an important factor for winning in the current game [3]. Also, following the previous study [1], we used league ranking from a week before the current game as an estimator of team strength in current match.

Thus, our first task is preprocessing the data and obtaining the last three games for home team and away team in each soccer match, and their league positions in the previous week. The following table is the first 5 rows of original Match table with extracted related columns:

season	stage	date	home_team_api_id	away_team_api_id	home_team_goal	away_team_goal
2008/2009	2	2008-08-16 00:00:00	1601	8030	2	1
2008/2009	3	2008-08-22 00:00:00	1601	8031	2	1
2008/2009	5	2008-09-12 00:00:00	1601	2186	2	0
2008/2009	7	2008-09-27 00:00:00	1601	1957	0	0
2008/2009	9	2008-10-19 00:00:00	1601	2182	2	0

Figure 1: original data that only has the current match goals for away and home team

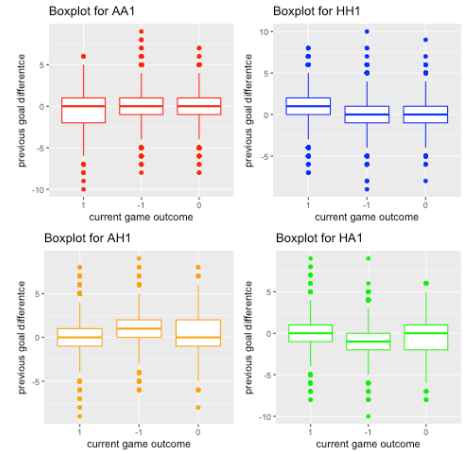
After the preprocessing, figure 2 shows the first few rows of the final formatted table with key features.

AA1	AA2	AA3	AH1	AH2	AH3	HH1	HH2	HH3	HA1	HA2	HA3	home_pos	away_pos	result
-1	-2	0	-2	2	-3	2	1	1	-3	-3	0	7	16	0
3	3	-1	0	1	3	0	2	1	-1	-3	-3	8	4	1
-1	-4	-2	-1	4	1	2	0	2	-1	-3	-3	7	9	0
0	0	0	-1	1	0	0	2	0	-2	0	-1	7	6	-1
-1	-1	0	4	3	1	-1	0	2	1	-2	0	7	4	-1
-1	2	-1	3	0	2	-1	-1	0	1	-2	0	8	4	1

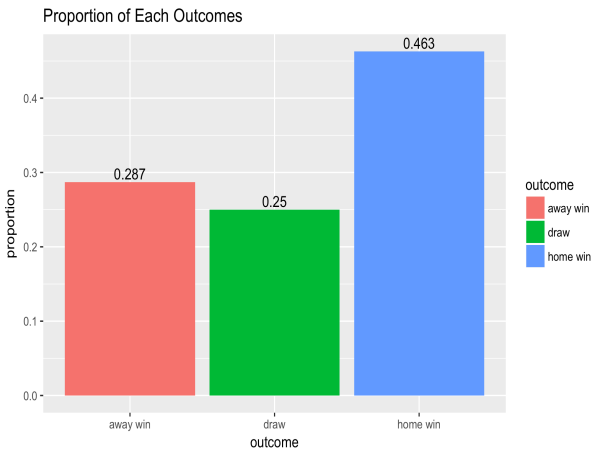
Figure 2: formatted data with past 3 games and league positions

In figure 2, AA1, AA2, AA3 are the score differences for away team's past three games when it played away (away difference = away_team_goal - home_team_goal). AH1, AH2, AH3 are the score differences for away team's past three games when it played home (away difference = home_team_goal - away_team_goal). Similarly, HH1 to HH3 are the score difference for home team's past three games when it played at home, and HA1 to HA3 are the goal difference for home team's past three games when it played away. The home_pos and away_pos are the two teams' weekly league standings before the match. Based on the score difference of current game, we generated the result column. For the "result" column, 1 denotes home team winning, 0 indicates draw, and -1 represents away team winning.

With the formatted dataset, we tried to look the relationship between the goal difference in a past game with the current match outcome and confirm the baseline of prediction accuracy. The following two graphs provide a basic analysis of the data:



(a) box plot: the previous game's goal difference in each match outcome category



(b) proportion of each match outcome

Figure 3: data description

From figure 3(a), we can see that the influence of previous game's goal difference on current match outcome is not very obvious. However, there are still some important trends showed here. For instance, from box plot for AA1, if the goal difference of previous game played away by away team is negative, the probability of home team winning is the highest in the current match among all the outcomes. Also, from box plot for HH1, if the the goal difference of previous game played home by home team is positive, the probability of

home team winning is the highest in the current match among all the outcomes. Similar findings can be found from the box plots for AH1 and HA1.

According to figure 3(b), we confirmed that 46.3% of the games are won by home team, 25% of the games are draws, and 28.7% of the games are won by away team. Thus, the baseline is about 46%.

3 Methods

3.1 Model Setup

Before fitting different models, we did some preparations with the formatted data. First, in every model, we partitioned 20% of the data set as a test set and used rest of the data as training set. In order to compare the prediction accuracy of different models, we used set.seed function before subsetting the data.

Our feature set has changed several times. First, we used past three games for the home and away team. Then, we tried past four games, to see if adding one more past game can improve the prediction accuracy. Finally, we added league position features in three game and four game data set respectively. The results indicate that adding league position features improve the accuracy of all models. Hence, we disregarded the earlier models and only compared the models that include league position variables.

We did classification with multinomial logistic model, Radom Forest, K-Nearest Neighbor and considered Naive Bayes model.

3.2 Multinomial Logistic

Let "result" be the current match result. As mentioned in the Data Section, $result \in \{-1, 0, 1\}$, and the design matrix with past three games is

$$X = [AA_1, AA_2, AA_3, AH_1, AH_2, AH_3, HH_1, HH_2, HH_3, HA_1, HA_2, HA_3, home\_pos, away\_pos]$$

The design matrix with past four games is very similar, except for adding features AA_4, AH_4, HH_4, HA_4. Here, we use design matrix with past three games in the formula. Let $\beta_i = [\beta_{1i}, \beta_{2i}, \ldots, \beta_{14i}]^\top$ and β_{0i} as the intercept for the result i, where $i \in \{-1, 0, 1\}$, the mutinomial logistic regression model can be expressed as following for each match:

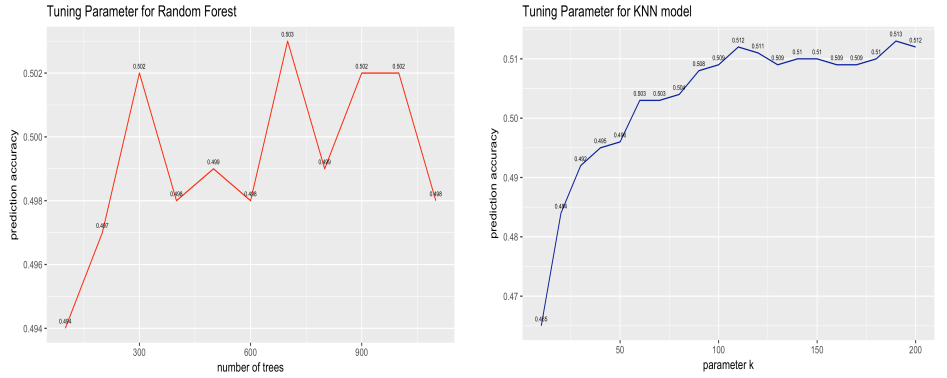
$$Pr(result = i|X = x) = \frac{e^{\beta_{0i}+x\beta_i}}{\sum_{I\in\{-1,0,1\}} e^{\beta_{0I}+x\beta_I}}$$

According to the multinomial logistic regression model, the result will be predicted as the one with the highest probability. The model is first trained with training set and minimizes the cost function, and then the model generates the optimal weights β_i and the intercept β_{0i} for each result i. With the optimal weights and intercept, we can use the model to do the prediction on our test set.

First, we used multinom() function in "nnet" package in R to fit the multinomial logistics regression model, and tested the model with the test set and calculated the prediction accuracy. Also, we tried cv.glmnet() in "glmnet" package to fit the multinomial logistic regression with cross validation, in order to prevent overfitting.

3.3 Random Forest, K-Nearest Neighbor and Naive Bayes

Since the match outcome is our dependent variable, which is categorical, we decided to try both random forest and k-nearest neighbor classification. For both random forest and k-nearest neighbor, we tuned the hyperparameters and chose the optimal one for our prediction models. The following two graphs indicate the tuning parameter process.



(a) tuning parameter "number of trees" in random forest model

(b) tuning parameter "k" in k-nearest neighbor model

Figure 4: tuning hyperparameters

Figure 4(a) shows the accuracy level on test set given a different number of trees in random forest model with past three games. We started with 100 trees and increased the number of trees by 100 each time. We can see that the highest accuracy achieved at 700 trees. Figure 4(b) is a graph about the change of prediction accuracy on test set with respect to different k. The model we used here is the k-nearest neighbor model with past 3 games. We can see after k=100, there are not many changes on accuracy. So, we chose k=190 in our knn model. Similarly, we did the hyperparameter tuning for both models with past four games as well.

Finally, we considered naive bayes model, because the highest prediction accuracy achieved on Kaggle so far is given by naive bayes. We tried this model with past three and four games respectively. However, the prediction accuracy of naive bayes is the lowest among all the models.

After understanding the theories behind the model, we realized that one of the assumption about naive bayes is mutually independence among features. However, our features are highly correlated, since we are using the past games results for prediction. The person on Kaggle used PCA before fitting the model, so the impact of dependence should be removed. However, we do not have as much features as him, and PCA may not be a good choice for us now, since the number of features is much smaller than our sample size. Thus, we will not report the results from naive bayes in the results section.

4 Results

4.1 Multinomial Logistic

We applied multinomial logistic model with past three games and past four games respectively. The prediction accuracy with past three games is 50.55% and 51.03% with past four games. The following tables are the confusion matrices with different feature sets:

Logistic Prediction with 3 Past Games	True Result 1	-1	0
1	0.381	0.162	0.185
-1	0.084	0.124	0.064
0	0.000	0.000	0.000

(a) confusion matrix with past 3 games

Logistic Prediction with Past 4 Games	True Result 1	-1	0
1	0.387	0.169	0.191
-1	0.065	0.123	0.065
0	0.000	0.000	0.000

(b) confusion matrix with past 4 games

Figure 5: confusion matrices of multinomial logistic model

From figure 5 (a) and (b), we can see that predicting the home winning game has the highest accuracy. However, a game ended in draw is the hardest one for prediction, since the incorrect prediction rate is the highest and none of the games is predicted as draw by the multinomial logistic model.

In order to comprehensively interpret the results, we also take a look at the coefficients and p values of the multinomial logistic model. Since the model with past four games achieves higher accuracy, we will just report the coefficients and p values with past four games here.

```
Coefficients:
   (Intercept)        AA1        AA2         AA3        AA4        AH1         AH2        AH3        AH4
-1  -0.4425321 0.03424391 0.04680738 0.025545864 0.04474844 0.06263106 0.031961172 0.06043622 0.03120300
0   -0.5943525 0.02303254 0.02342113 0.002366934 0.01827071 0.03098374 0.004310471 0.00713811 0.01064155
           HH1         HH2         HH3         HH4         HA1         HA2         HA3         HA4   home_pos
-1 -0.07262084 -0.04376516 -0.05995508 -0.05488443 -0.04505372 -0.03686643 -0.03675855 -0.03549050 0.05723074
0  -0.04886541 -0.04757975 -0.02162746 -0.04529118 -0.01822188 -0.01778476 -0.01265718 -0.04043534 0.03159544
      away_pos
-1 -0.06741755
0  -0.02733906
```

Figure 6: coefficients of multinomial logistic model with past four games

```
    (Intercept)         AA1          AA2        AA3          AA4          AH1         AH2          AH3
-1 1.296787e-10 0.003424648 7.305274e-05 0.03087914 0.0001420016 9.947728e-08 0.006764428 3.422269e-07
0  0.000000e+00 0.051628291 4.974253e-02 0.84330802 0.1240974663 9.184653e-03 0.718306460 5.485879e-01
          AH4          HH1          HH2          HH3          HH4          HA1          HA2          HA3
-1 0.009317494 1.175056e-09 2.750939e-04 5.025209e-07 4.837013e-06 0.0001090621 0.001904487 0.001929219
0  0.381211085 4.497121e-05 8.460511e-05 7.097737e-02 1.783738e-04 0.1189763100 0.135515077 0.288861747
           HA4     home_pos     away_pos
-1 0.0027917056 0.000000e+00 0.000000e+00
0  0.0006779465 8.248797e-10 1.195692e-07
```

Figure 7: p values of multinomial logistic model with past four games

We can interpret the coefficients from figure 6. For instance, in -1 category, one unit increase in the variable AA1 is associated with the increase in the log odds $log\frac{p(result=-1)}{p(result=1)}$ in the amount of 0.034. In other words, if the away team scored 1 more goal in the previous away game, then the probability of winning for the away team is increased by 3.4% with respect to home team winning. Similarly, for -1 category, which is away team win, we can easily understand that the coefficients of AA, AH, home_pos are positive and the coefficients of HH, HA, away_pos are negative.

According to figure 7, with $\alpha = 0.05$, all the coefficients are statistically significant with -1 category. However, only AA2, AH1, HH1, HH2, HH4, HA4, home_pos and away_pos are statistically significant in 0 category.

From figure 6 and figure 7, we realize that among all the significant coefficients, the coefficient with largest absolute value is HH1 in both -1 and 0 categories, which should be the most influential factor in the model.

4.2 Multinomial Logistic with Balanced Dataset

From figure 5, one interesting result is that none of the games is predicted as draw in our multinomial logistic model. We believe that none of the games is predicted as a draw in the original data set is due to small proportion of draw games. As showed in figure 3 (b), the match outcomes in the original data set are unevenly distributed with 46% of home team wins, 29% of away team wins, and 25% of draws. Thus, We randomly sampled the same amount of games in each category so that the result in the training set has $\frac{1}{3}$ of home team wins, $\frac{1}{3}$ of away team wins, and $\frac{1}{3}$ of draws. With rest of the data, we created a balanced test dataset as well. After the balanced sampling, the baseline for prediction is around 33%.

Prediction with Balanced Dataset	True Result 1	-1	0
1	0.192	0.090	0.122
-1	0.091	0.187	0.142
0	0.050	0.056	0.069

Figure 8: confusion matrices of multinomial logistic model with balanced dataset

The figure 8 shows that although only 18% of the games are predicted as draws, the model does give draw prediction. Thus, we believe with more data and balanced sampling, the prediction accuracy of multinomial logistic model can be improved.

4.3 Radom Forest and K-Nearest Neighbor

Since multinomial logistic model does not have any draw in its prediction, we want to see if random forest and k-nearest neighbor can generate draws in prediction and improve the prediction accuracy.

RF Prediction with 3 Past Games	True Result 1	-1	0
1	0.384	0.160	0.185
-1	0.069	0.116	0.058
0	0.012	0.010	0.007

(a) confusion matrix with past 3 games

RF Prediction with 4 Past Games	True Result 1	-1	0
1	0.388	0.179	0.195
-1	0.058	0.109	0.057
0	0.007	0.004	0.004

(b) confusion matrix with past 4 games

Figure 9: confusion matrices of random forest model

KNN Prediction with Past 3 Games	True Result 1	-1	0
1	0.398	0.172	0.194
-1	0.066	0.114	0.054
0	0.001	0.000	0.000

(a) confusion matrix with past 3 games

KNN Prediction with Past 4 Games	True Result 1	-1	0
1	0.397	0.183	0.200
-1	0.054	0.108	0.055
0	0.001	0.002	0.000

(b) confusion matrix with past 4 games

Figure 10: confusion matrices of k-nearest neighbor model

According to figure 9 and figure 10, we can see the highest rate of wrong prediction is predicting a draw game as a game won by home team, while the highest rate of correct prediction is predicting a home team winning game. Also, there are games predicted as draws in both models, although the proportion of draw in prediction is still very small. Another interesting result is that only 0.7% and 0.4% of draw games are correctly predicted by random forest model with past three and four games respectively, and none of the draw game is correctly predicted in k-nearest neighbor model. It shows a draw game is really hard to be predicted correctly.

5 Conclusions

In order to compare all the models we tried, we created figure 11 to give a conclusion about our prediction accuracy among all the models.

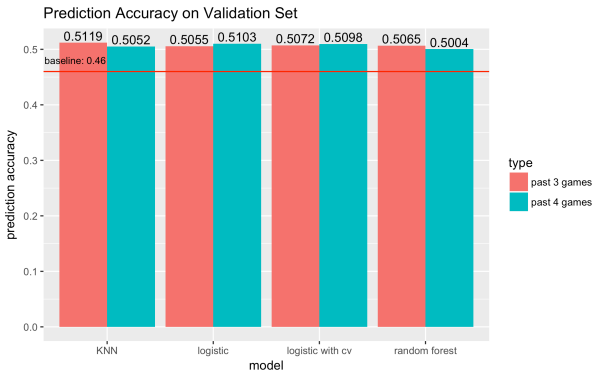


Figure 11: prediction accuracy of all models

According to figure 11, after adding the league position, the best prediction accuracy is achieved by k-nearest neighbor with past 3 games. we can see that all models give very similar prediction accuracy, although k-nearest neighbor is slightly better, it is hard to say which classification method is the best.

Another thing is that adding one additional past game does not significantly help improve the prediction accuracy. From the graph, we can see that some models perform better with past four games and some do not. For logistic model, we also tried cross validation, and the prediction accuracy is not changed much as showed in figure 11. So, overfitting is not a problem for us now.

For this project, the most effective way we found for prediction accuracy improvement is just adding good features. By feature engineering, our prediction accuracy is improved from 48%, to 50%, and finally to 51%.

If we have more time to do the project in the future, there are two important things we can try. One is definitely adding more features. For instance, considering adding the seasonality of the matches, the number of injured players for home team and away team, the distance travelled by away team, etc. The other possible way is to write our own multinomial logistic classifier. Since the number of wrong predictions is highest on predicting draw games, we can modify the objective function and give higher penalty on wrong prediction for draw games, not just using the common cross entropy loss function.

6 References

[1] Mehrez, A.; Hu, M. Y. Zeitschrift fur Operations Research 1995, 42, 361-372.

[2] Rue, H.; Salvesen, O. Journal of the Royal Statistical Society: Series D (The Statistician) 2000, 49, 399-418.

[3] Goddard, J.; Asimakopoulos, I. Modelling football match results and the efficiency of fixed-odds betting; 2003.

[4] Hucaljuk, J.; Rakipovic, A. Predicting football scores using machine learning techniques. 2011.

[5] Huang, K.-Y.; Chang, W.-L. A neural network method for prediction of 2006 World Cup Football Game. 2010.

[6] Goddard, J. International Journal of Forecasting 2005, 21, 331-340.