**Predicting Match Result**

**Introduction**

Bundesliga, the top soccer league in Germany, is gaining popularity worldwide during recent years. The increasing popularity has brought extra rewards for the top teams: being upper in the rank means higher brand endorsement revenue, higher broadcasting revenue, higher sales of fan products, and simply more rewards of money for winning matches. In contrast, the teams won least numbers of matches get none of the rewards above.

There are 18 competitive teams in this league striving to be the best each season. In order to win over these many rivals, a team needs to make as much money as it can to raise capital for building squad, brand marketing etc. Therefore, being able to understand the key factors influencing match result is crucial. Ability to predict a match’s result can not only help teams to prepare for future matches, but also can be used by people who do betting on match result to make profit.

This project builds model based on match results and statistics of every match in Bundesliga season 2015 to 2016 to predict future match result. The result will be in the form of whether the home team will win or not win.

**Methodology**

The project was made up of 4 phases: 1) Data Collection; 2) Data Preparation 3) Feature Selection and 4) Classifiers Evaluation.

**Data Collection**

The dataset was mostly collected from website: football-data.co.uk. There is also an extra factor, stadium attendance, collected from the official website of Bundesliga. This feature was collected because we want to predict whether the home team wins the match or not, and the size of the home stadium and its attendance have been widely discussed as important to the home team’s performance, since the majorities in the stadium are the home team’s fans.

The original dataset includes features listed below:

FTHG = Full Time Home Team Goals

FTAG = Full Time Away Team Goals

FTR = Full Time Result (H=Home Win, D=Draw, A=Away Win)

HS = Home Team Shots

AS = Away Team Shots

HST = Home Team Shots on Target

AST = Away Team Shots on Target

HC = Home Team Corners

AC = Away Team Corners

HF = Home Team Fouls Committed

AF = Away Team Fouls Committed

HY = Home Team Yellow Cards

AY = Away Team Yellow Cards

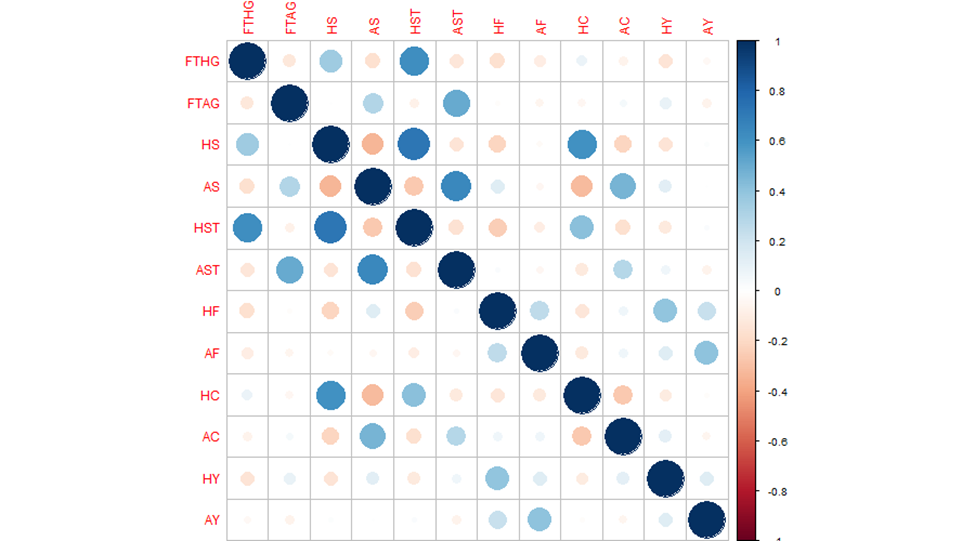
HR = Home Team Red Cards

AR = Away Team Red Cards

HT = Home Stadium Attendance

**Data Preparation**

Considered the fact that the dataset seemed to have more than necessary number of features, which indicates potential dependent features, we ran correlation test in between all the features except the target feature and generated the plot below:

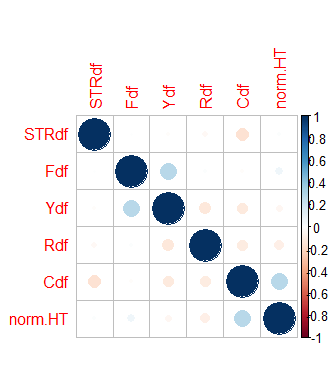
****

As we can observe from the plot, HS with HST, and AS with AST have strong correlation, which is not a surprising result because the number of shots on target is depend on the total shots. There are many other pairs of correlated features such as FTHG with HST, FTAG with AST, HS with HC, AS with AC, HST with HC etc. Some of them are not obviously depend on each other, so it is very hard to tell whether those features are meaningful or should be eliminated. Therefore, we modified and transformed features as described below:

* Calculated HSTR/ASTR (Home/Away team shots accuracy) by dividing HS/AS (Home/Away shots) by HST/AST (Home/Away shots on target).
* Calculated STRdf (Shots on target accuracy difference) by subtracting ASTR from HSTR.
* Calculated Fdf (Fouls committed difference) by subtracting AF (away fouls) from HF (home fouls).
* Calculated Ydf (Yellow Cards difference) by subtracting AF (away yellow cards) from HY (home yellow cards).
* Calculated Rdf (Red Cards difference) by subtracting AR (away Red cards) from HR (home Red cards).
* Calculated Cdf (Corners difference) by subtracting from AC (away Corners) from HC (home Corners).
* Normalized HT (Home Stadium Attendance), because of the big differences of attendances among these stadiums.
* Remove all the features used to generate the new ones.

The reason for subtracting away team features from home team features is that we want to focus more on the factors that can influence home team’s chance to win, and simply using the performance differences between home and away team should be enough and more efficient to realize our purpose.

The new correlation plot of non-target features is attached below:

****

From the new plot, it is easy to conclude that there is no more obvious correlation between any non-target features. Accompanied by the fact that there is no missing value or clear error in the dataset, we can carry on to the next step of the project using the current features.

**Feature Selection**

Correlation Test Between the DFs and the Target

FTR ~ HT (Full time result ~ Home Stadium Attendance)

p-value = 3.888e-05

cor: 0.2328954

FTR ~ STRdf (Full time result ~ Shot accuracy difference between home and away team)

p-value = 4.581e-13

cor: 0.3981229

FTR ~ Fdf (x) (Full time result ~ Fouls committed difference between home and away team)

p-value = 0.3923 (x)

cor -0.04907837 (x)

FTR ~ Ydf (x) (Full time result ~ Yellow Card gained difference between home and away team)

p-value = 0.1374 (x)

cor: -0.08511796 (x)

FTR ~ Rdf (Full time result ~ Red card gained difference between home and away team)

p-value = 0.02448

cor: -0.1285846

FTR ~ Cdf (Full time result ~ Connors difference between home and away team)

p-value = 0.3299 (x)

cor: 0.05587837

After conducting correlation tests between different features and full time match result, some features, including fouls committed, yellow card gained, corners played are seemed not significantly correlated with the match result: they have p-values much higher than the significance level of 0.05 and very low correlation rate. Home stadium attendance, shots on target rate and red card gained could be considered statistically significant predictors of match result. However, we did not remove the uncorrelated features at this point, because they may be helpful for building model when combine with other features. In order to select or potentially eliminate features, the following test was conducted:

The classifier Random Forest was used in the feature selection process, and we generated the attribute importance ranking below:

attr\_importance  
STRdf 36.3399895

norm.HT 25.8640686  
Fdf -0.9601667  
Ydf 8.6205440

Cdf 7.3069773  
Rdf 6.3807209

Backward Greedy Search

We chose to rely on the result of backward but not forward greedy search in this case, because after ran through forward search, there were only one or two features left, and to build an effective model, we expected more features than that.

The result of backward greedy search varies almost every time; therefore 10 sets of results were produced to find the features that were chosen most frequently.

Results (1~10):

FTR ~ STRdf + Fdf + Ydf + Cdf + norm.HT

FTR ~ STRdf + Fdf + Ydf + norm.HT

FTR ~ STRdf + Fdf + Ydf + Rdf + Cdf + norm.HT

FTR ~ STRdf + Fdf + Ydf + norm.HT

FTR ~ STRdf + Fdf + Rdf + Cdf + norm.HT

FTR ~ STRdf + Fdf + Ydf + Rdf + norm.HT

FTR ~ STRdf + Fdf + Ydf + Rdf + norm.HT

FTR ~ STRdf + Fdf + Ydf + Rdf + Cdf

FTR ~ STRdf + Fdf + Ydf + Rdf + Cdf + norm.HT

FTR ~ STRdf + Fdf + Rdf + Cdf + norm.HT

Times each feature was selected:

STRdf = 10

Fdf = 10

Norm.HT = 9

Ydf = 8

Rdf = 7

Cdf = 6

Combining the result of backward greedy search and importance ranking, Cdf and Rdf seem like the most questionable features that might need to be eliminated. However, they were not deleted at this point because although comparing to other features, they seem less important, even Cdf had been chosen 6 out of 10 times by the backward greedy search. It is risky to drop a feature that still has a potential too easily. Those questionable features will be tested manually in the process of choosing classifier. Fdf will also be considered as questionable. Even though it had been chosen every time during the search, it also had an outstanding value of importance (-0.9601667).

**Classifiers Evaluation**

4 potential classifiers were chosen: One Rule, Naïve Bayes, Random Tree, PART and Random Forest. The results are shown below:

One Rule

STRdf

=== Evaluation result ===

Correctly Classified Instances 56 53.8462 %

Incorrectly Classified Instances 48 46.1538 %

Kappa statistic 0.2522

Mean absolute error 0.3077

Root mean squared error 0.5547

Relative absolute error 71.4753 %

Root relative squared error 119.6924 %

Total Number of Instances 104

Naïve Bayes

=== Evaluation result ===

Correctly Classified Instances 49 47.1154 %

Incorrectly Classified Instances 55 52.8846 %

Kappa statistic 0.1469

Mean absolute error 0.3905

Root mean squared error 0.4647

Relative absolute error 90.7057 %

Root relative squared error 100.2735 %

Total Number of Instances 104

Random Tree

=== Evaluation result ===

Correctly Classified Instances 47 45.1923 %

Incorrectly Classified Instances 57 54.8077 %

Kappa statistic 0.1607

Mean absolute error 0.3654

Root mean squared error 0.6045

Relative absolute error 84.8769 %

Root relative squared error 130.4318 %

Total Number of Instances 104

PART

=== Evaluation result ===

Correctly Classified Instances 49 47.1154 %

Incorrectly Classified Instances 55 52.8846 %

Kappa statistic 0.1097

Mean absolute error 0.3999

Root mean squared error 0.5095

Relative absolute error 92.8953 %

Root relative squared error 109.9447 %

Total Number of Instances 104

RandomForest

=== Evaluation result ===

Correctly Classified Instances 53 49.5327 %

Incorrectly Classified Instances 54 50.4673 %

Kappa statistic 0.1918

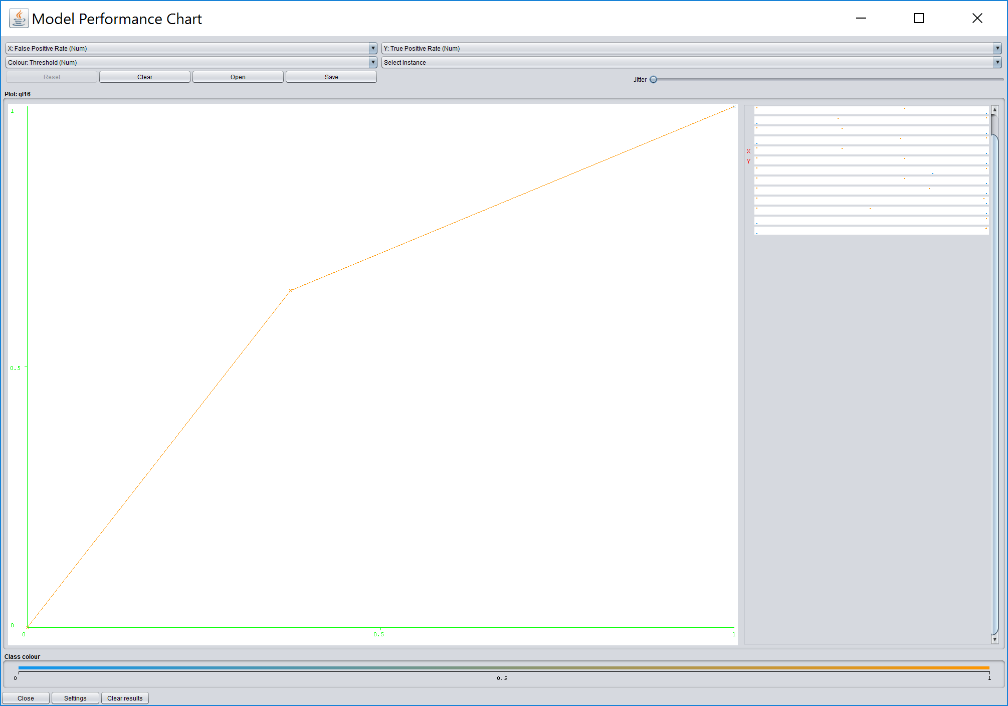
Mean absolute error 0.3869

Root mean squared error 0.4627

Relative absolute error 90.0243 %

Root relative squared error 100.2152 %

Total Number of Instances 107

The prediction was based on three classes: home team win, away team win and draw. Therefore, correct rates of around 50% did not mean the models were meaningless. 

However, One Rule was the best classifier in this case, and it only considers the feature of STRdf (shot accuracy difference), so we had to run the classifier test with some features removed to get a better result. The target feature was also changed from three classes into two classes: the home team wins and not win.

**\*Changed into two-classes (Cdf removed)**

OneR

=== Evaluation result ===

Correctly Classified Instances 68 63.5514 %

Incorrectly Classified Instances 39 36.4486 %

Kappa statistic 0.2589

Mean absolute error 0.3645

Root mean squared error 0.6037

Relative absolute error 73.8414 %

Root relative squared error 121.36 %

Total Number of Instances 107

NaiveBayes

=== Evaluation result ===

Correctly Classified Instances 72 **67.2897 %**

Incorrectly Classified Instances 35 32.7103 %

Kappa statistic 0.3079

Mean absolute error 0.4087

Root mean squared error 0.466

Relative absolute error 82.8079 %

Root relative squared error 93.6812 %

Total Number of Instances 107

RIPPER

**Confusion Matrix and Statistics  
  
 Reference  
Prediction NotWin Win  
 NotWin 41 17  
 Win 27 37  
   
 Accuracy : 0.6393**

RandomTree

=== Evaluation result ===

Correctly Classified Instances 70 65.4206 %

Incorrectly Classified Instances 37 34.5794 %

Kappa statistic 0.2914

Mean absolute error 0.3458

Root mean squared error 0.588

Relative absolute error 70.0546 %

Root relative squared error 118.2072 %

Total Number of Instances 107

PART

=== Evaluation result ===

Correctly Classified Instances 66 61.6822 %

Incorrectly Classified Instances 41 38.3178 %

Kappa statistic 0.199

Mean absolute error 0.4057

Root mean squared error 0.5046

Relative absolute error 82.1847 %

Root relative squared error 101.4391 %

Total Number of Instances 107

RandomForest

=== Evaluation result ===

Correctly Classified Instances 67 62.6168 %

Incorrectly Classified Instances 40 37.3832 %

Kappa statistic 0.2232

Mean absolute error 0.3991

Root mean squared error 0.4774

Relative absolute error 80.8468 %

Root relative squared error 95.973 %

Total Number of Instances 107

Cdf was the first feature to be removed considering the test results in the feature selection process, and we successfully got a much better correct prediction rate of 67.2897 % with Naïve Bayes. A few more manual backward search tests were conducted, and the best result is shown below:

**\*Eliminated both Cdf and Fdf**

**Naïve Bayes**

=== Evaluation result ===

Correctly Classified Instances 73 **68.2243 %**

Incorrectly Classified Instances 34 **31.7757 %**

Kappa statistic 0.329

Mean absolute error 0.4061

Root mean squared error 0.4626

Relative absolute error 82.2709 %

Root relative squared error 92.9888 %

Total Number of Instances 107

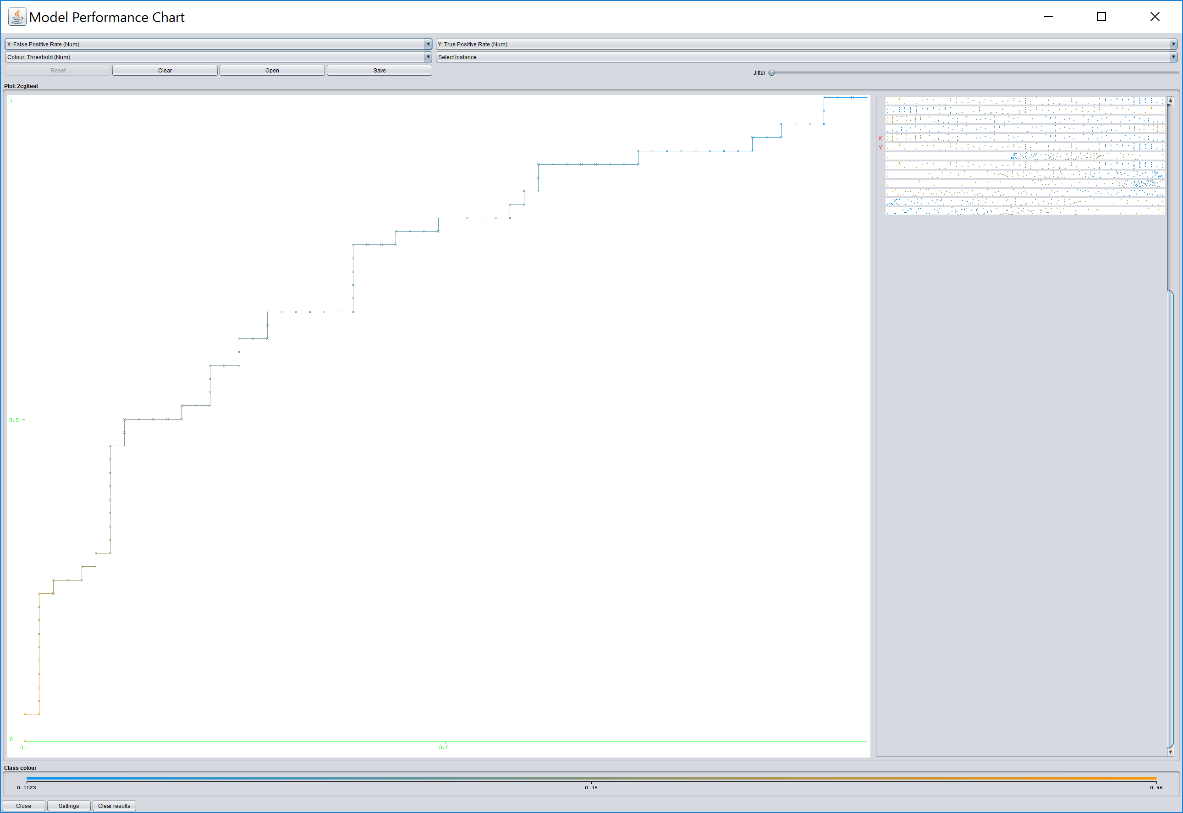
=== Confusion Matrix ===

a b <-- classified as

20 28 | a = Win

6 53 | b = Not Win

The best classifier in this case is Naïve Bayes. The model predicts 68.2243% correctly and 31.7757 % wrong.



**Conclusion**

The attributes in the dataset are proved able to predict the match result in a certain level. The error rate is still high, but is within in an acceptable range, especially considering the amount of other unpredictable factors related to a match that can influence its result, such as player’s injury conditions, weather and number of matches played by each team recently and many more.

This model can be used to predict a future match’s result based on home and away team’s recent statistics. A team can design its training plan by looking at the model. For instance, for team A to win over team B, team A may need to improve its shoot accuracy to a certain level to increase the shoot accuracy difference between it and team B, and the exact numbers of improvement needed can be found in the model. A possibly more effective way to use this model is to predict the match’s result upon betting. Although the highest correct rate is only about 68%, it is still better than guessing (which is 50%).

The future work that may help to improve this model is to find more features that are highly related to the match result, such as total running distance, ball possession rate, and the total market worth of the squad.