IE 582 Project Report

GROUP ID: 11

IE 582 Project Report

1. Introduction: Problem description, summary of the proposed approach, descriptive

Football, one of the most popular sports on the planet, has always been intensely followed by lots of people. Through technological improvements, very detailed and informative data on football can be collected for many matches in various countries such as time, number of yellow cards, number of passes or corners etc. for each match. With the increased accessibility to matches live and the widespread using of internet, betting market on this oft-followed sport has rapidly grown, recently. Since the accurate prediction on this high-valued market is significantly important, the number of machine learning studies on betting system is getting increased day by day.

This report covers our work regarding the forecasts on football betting. After a brief introduction to the problem description, proposed approach to the problem, and thereafter, analysis of the data and the model follows.

Main goal in our project was to examine different methods and to provide a better forecasting approach to guide how to place money on bets than the bookmakers have. The betting type focused on in this project is the three-way betting which is mainly to wager on one of three outcomes, win-tie-loss. The objective in the project was attained by building mathematical models and using different machine learning algorithms.

The problem in the project is modeled as a classification problem with three labels. The problem has ordinal nature since the result of the game cannot change without being tie at least for a while during the match. The data analyzed in this project consists of only Premier League matches. The league consists of 20 clubs and 10 matches are played in each round. All the matches from the beginning of 2018 season until December 2019 are used.

The bookmakers put some profit margin on their odds. To compare the probabilistic results found with the bookmakers' odds, this margin needs to be removed. Hence, getting the reciprocal of the odds and applying normalization on them is preferred in this comparison.

Decision tree algorithm and gradient boosting algorithm are used to generate predictions for the model. As a performance measure, “Ranked Probability Score” (RPS) is used in the project and it takes the ordinal nature of the problem into account. It expresses the forecasts as a distribution and considers matching them with the observed results. The lower the RPS score is, the better the prediction is. At the end, the model giving the better results was the gradient boosting which was improved by parameter tuning to find the best shrinkage and depth through cross validation technique.

2. Approach: Explain your approach to this problem.

Previous bets on finished games are taken into consideration while building the model. With the main focus on the bets (1X2 Bets) but also including goals and booking data various models are built, among all models best performing ones are reported.

2.1 Data Preprocessing

In order to prepare the data to be used in the model, some preprocessing was done. Namely,

- Only PL games are included, since model is built for PL games
- Last 2 season games are filtered among all matches
- It is considered that matches with early bookings and too much late goals, goals scored in extra time, can harm the model so these matches also removed
- For a given match and bookmaker, there are lots of different timed bets on the data, only the latest bets are taken into consideration.
- There are lots of bookmakers whose all bets are not available in the datasets. In order to reduce the unavailable bets number, bookmakers only less than 10 unavailable bets are included in the model. These bookmakers referred in the model as “trusted_bookers”.
- Data split into 2 sets, train and test. 67% of data randomly selected and used as Train and the remaining 33% of data used as Test

2.2 Model Building

As already mentioned, models is mainly built on the bets data. Tree based approaches are the best for the project. So the two best performing methods are reported: Decision Tree, Gradient Boosting.

To find best the model, overfitting is tried to be reduced and overall test performance was tried to be optimized, to do so below methods implemented:

1. 1st, each model was built using 10-fold validation, so the train data separated on 10-fold and model was build using these folds.
2. 2nd, to find best parameter settings, different parameters used when model building and best set of parameters selected which minimizes RPS of test data.

So tuning efforts were made for minimizing RPS. RPS was main performance metric of the project.

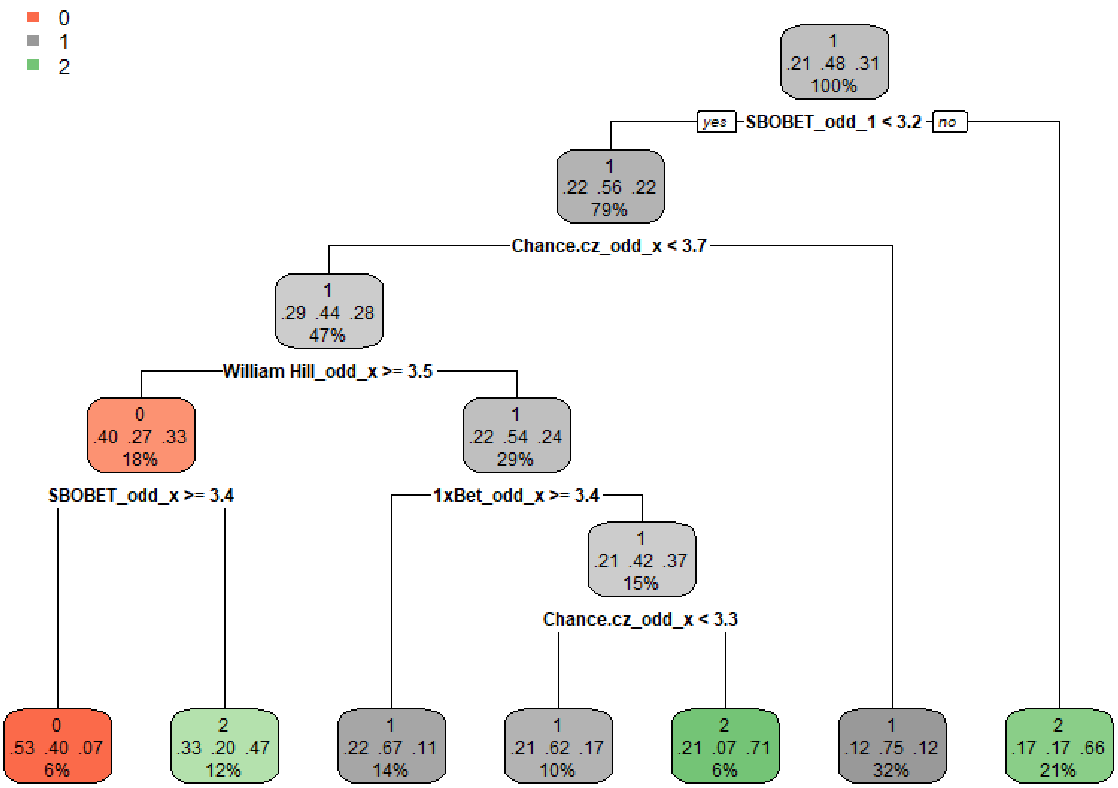
Below parameters were tuned for RPS:

- Decision Tree: Complexity parameter, “cp” of the rpart function
- Gradient Boosting: Shrinkage, depth.

3. Results: Provide your results and discussion.

In order to find the best resulting algorithm with best parameters, the performance metrics should be compared for each approach inevitably. Since two different approaches considered in this report for given problem, results and performances of these approaches according to RPS value on test data will be discussed.

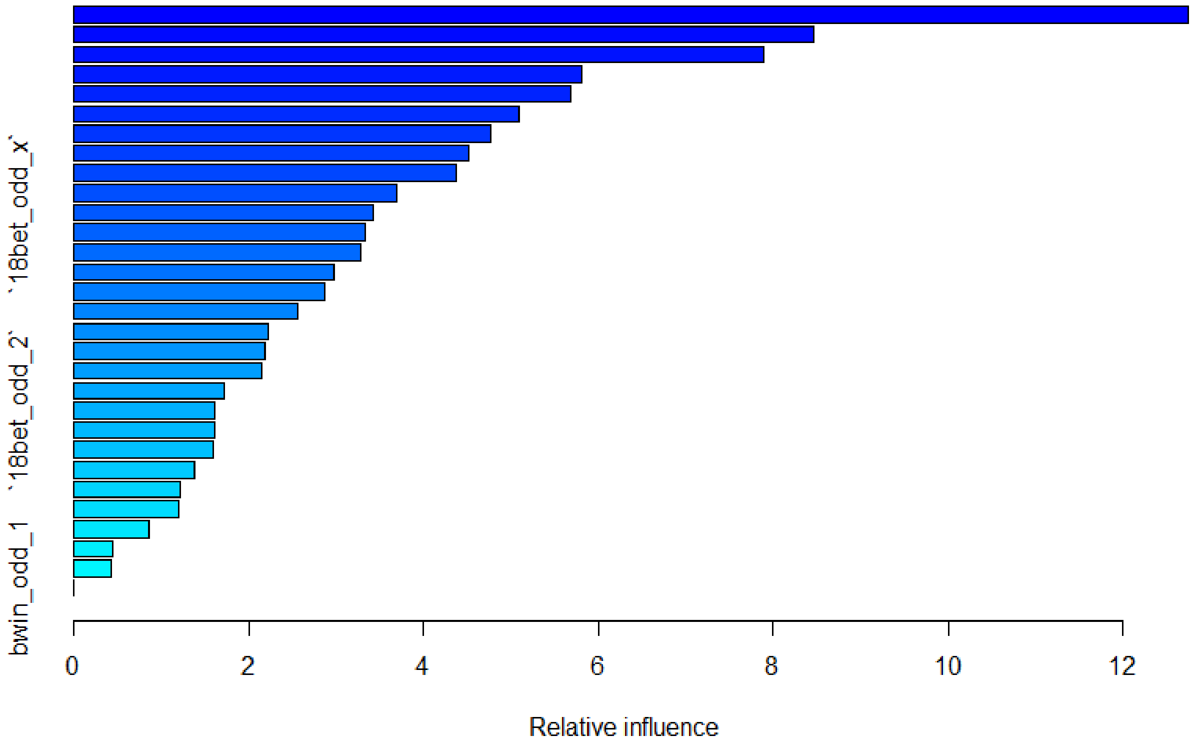
First approach was the decision tree it is implemented by using rpart and the first trial for decision tree ended up with the tree given below



Interestingly, rparts algorithm chooses bets of tie as an argument to predict match results. Even though it looks like a legitimate tree according to number of nodes and size of these nodes the performance of this model quite mediocre with a rps value 0.2366201.

So, using cross validation best cp value is seeked and after 50 trials cp value of 0.15 gives the minimum rps value 0.2233876. As it can be observed usual decision trees has very little space to improve for this problem.

Secondly, gradient boosted decision tree used for prediction. For this method as mentioned above 2 parameters needed to be identified. After the cross-validation algorithm ended up with depth 6 and shrinkage value 0.1 as the minimum RPS driver. With these parameters RPS value of 0.215892 on test data can be found which is a fair value considering the results of other decision tree models.



variable	**relative inf.**
Chance.cz_odd_x	12.7499
BetVictor_odd_1	8.4684
Unibet_odd_x	7.887777
BetVictor_odd_x	5.814025
Unibet_odd_2	5.682379
BetVictor_odd_2	5.084853
1xBet_odd_2	4.764107

1xBet_odd_x	4.507885
888sport_odd_2	4.377835
William Hill_odd_2	3.696481

In our best resulting GBM model, considering the relative influence of columns again Interestingly the odd values of ties have higher impact same as decision trees.

4. Conclusions and Future Work: Summarize your findings and comments regarding your approach. What are possible extensions to have a better approach?

Tree based model was fitted well for the project. Gradient boosting trees have the best performance. Decision trees are faster; however, performance of GBM is also competitive in terms of speed. Even loops trying different parameter combinations did not take too much time.

Unfortunately, datasets have lots of unavailable bets. Only 32 bookmakers' odds could be used, if the other bets also be used, better results would have been achieved. So, having more reliable data would be good. Investigating personal statistics of bookmakers like variance of bets, time series analysis of bets could be helpful.

Beside tree-based approaches, some regression models can also be used as future work. Logistic regression could be a good case as a future work.

5. Code: Provide the Github link for your codes at the end of your report.

https://bu-ie-582.github.io/fall19-fethiysmli/project/Project.html is the code for the project.