IE 582

Statistical Learning for Data Mining

Term Project Report

2021402000 Doğa Erçin

2020402126 Ahmet Çeliker

2020402000 Ender Purcu

1) **Introduction**

Motivation

Sports and betting have been accompanying each other since the beginning of sports events, dating back to 4000 BC to Ancient Egypt, where they bet on board games, fencing and even dies. (Bulski,2020) With the advancement of sports analyses, increasing number and type of sports, and increasingly fanatical nature of fans, it is now crucial to determine the "odds" as good as possible so that bookmakers can profit, while satisfying the competitive needs of the population.

In this project, we aim to identify the point in the match where it makes the most sense to bet on one of the 3 options "home win", "away win" and "tie", or not bet at all. This analysis will be based on soccer match data, and statistics of minute granularity.

The particularly challenging part of this project is the fact that we a re developing a "live betting" strategy, meaning that we have to make our predictions based on previous data only, and we are not to revise the bet afterwards. This is a complicated tradeoff between more profit and more risk.

This is a simply a multi-class classification problem where the classes are "1", "2" and "X", indicating home-win, away win and tie respectively.

2) **Descriptive Analysis**

The provided dataset contains detailed information about football matches, including both real-time statistics and final outcomes. It consists of 63,944 records and 106 features. Betting market dynamics are clear in the very frequent updates to bookmaker odds, especially with major shifts often caused by key events such as goals, red cards etc. Some features were engineered by analyzing the relations between existing features and using our own knowledge about the football dynamics. These new features aim to capture additional interactions that may improve the performance of the model. Interactions between these features were also calculated and incorporated into the dataset, increasing its ability to reflect complex relationships within the data. A detailed explanation of these features, along with their purpose and significance, is provided in the Feature Architecture section.

Match and Timing Information

- **fixture_id**: Unique identifier for each match.
- **halftime**: Current half of the game (e.g., "1st-half", "2nd-half").
- **current_time**: Timestamp of the current game state.
- **half_start_datetime**: Timestamp marking the start of the current half.
- **match_start_datetime**: Timestamp marking the start of the match.
- **minute**: Current minute of the game.
- **second**: Current second of the game.

Bookmaker Information

- **latest_bookmaker_update**: Timestamp of the most recent odds update from the bookmaker.
- **1**, **X**, **2**: Odds for home win, draw, and away win, respectively.
- **name**: Name of the bookmaker.
- **ticking**: Indicates whether the bookmaker's odds are updating in real time.

Match State

- **suspended**: Whether the match is currently suspended (True/False).
- **stopped**: Whether the match is stopped (True/False).
- **current_state**: Real-time state of the match outcome (e.g., "1" for home win, "X" for draw).

Performance Metrics

General Team Statistics

- **Goals - home / Goals - away**: Current number of goals scored by each team.
- **Score Change - home / Score Change - away**: Tracks changes in the score.

Possession and Attacks

- **Ball Possession % - home / Ball Possession % - away**: Percentage of ball possession for each team.
- **Attacks - home / Attacks - away**: Number of total attacking plays by each team.
- **Dangerous Attacks - home / Dangerous Attacks - away**: Attacks classified as dangerous.

Shots

- **Shots On Target - home / Shots On Target - away**: Shots directed towards the goal and on target.
- **Shots Off Target - home / Shots Off Target - away**: Shots that missed the goal.
- **Shots Insidebox - home / Shots Insidebox - away**: Shots taken inside the penalty box.
- **Shots Outsidebox - home / Shots Outsidebox - away**: Shots taken from outside the penalty box.
- **Shots Total - home / Shots Total - away**: Total number of shots taken by each team.
- **Shots Blocked - home / Shots Blocked - away**: Shots blocked by the opposing team.

Passing

- **Passes - home / Passes - away**: Total number of passes attempted.
- **Successful Passes - home / Successful Passes - away**: Number of completed passes.

- **Successful Passes Percentage - home / Successful Passes Percentage - away**: Success rate of passing.

Crosses

- **Total Crosses - home / Total Crosses - away**: Total number of crosses.
- **Accurate Crosses - home / Accurate Crosses - away**: Number of successful crosses.

Dribbles and Headers

- **Dribble Attempts - home / Dribble Attempts - away**: Total number of dribble attempts.
- **Successful Dribbles - home / Successful Dribbles - away**: Dribble attempts that were successful.
- **Headers - home / Headers - away**: Total number of headers.
- **Successful Headers - home / Successful Headers - away**: Headers that were successfully directed.

Defensive Actions

- **Tackles - home / Tackles - away**: Total tackles made.
- **Interceptions - home / Interceptions - away**: Total interceptions.
- **Successful Interceptions - home / Successful Interceptions - away**: Interceptions successfully executed.

Other Actions

- **Corners - home / Corners - away**: Number of corner kicks.
- **Throwins - home / Throwins - away**: Number of throw-ins.
- **Goal Kicks - home / Goal Kicks - away**: Number of goal kicks.
- **Free Kicks - home / Free Kicks - away**: Number of free kicks.
- **Offsides - home / Offsides - away**: Number of offsides called.

Disciplinary Information

- **Fouls - home / Fouls - away**: Total fouls committed by each team.
- **Yellowcards - home / Yellowcards - away**: Number of yellow cards.
- **Redcards - home / Redcards - away**: Number of red cards.
- **Yellowred Cards - home / Yellowred Cards - away**: Cards that count as both yellow and red.

Injuries and Substitutions

- **Injuries - home / Injuries - away**: Number of injuries affecting each team.
- **Substitutions - home / Substitutions - away**: Number of player substitutions.

Match Outcome

- **final_score**: The final score of the match.
- **result**: Final match outcome ("1" for home win, "X" for draw, "2" for away win).

3) References

a. (Bulski,2020)
b. https://www.betamatics.com/strategies.html
c. https://us.humankinetics.com/blogs/excerpt/a-brief-history-of-sports-betting?srsltid=AfmBOormJKKFHO9jWcdVt01A7SPur8YnRIdtyspOjJM9HgI4kc26eJdI

4) Approach

Before starting our analysis, we dealt with the data, getting some help from HW 2. We first eliminated the matches which were suspended, or stopped.

Then we dealt with the missing data in critical columns, namely "1","2","X". We applied a forward fill for missing values no more than 10 consecutively, and if they were more than 10, we discard the match completely, since these data are critical for our analysis. For the other columns, we used other methods of imputations based on the nature of the statistics.

We separated the data into training and test splits, and the test matches started from "01.11.2024".

i. Elimination of suspended & stopped matches
ii. Missing match minutes – there were none
iii. Missing data- forward fill grouping by fixture id
    1. Matches with too many NA values - None
iv. Ensuring numerical data
v. Changing result column to categorical data 1,2,X to 1,2,0

4.1) Feature architecture

In order to construct variables that seemed to have a large impact on the outcome, correlations between variables were checked.

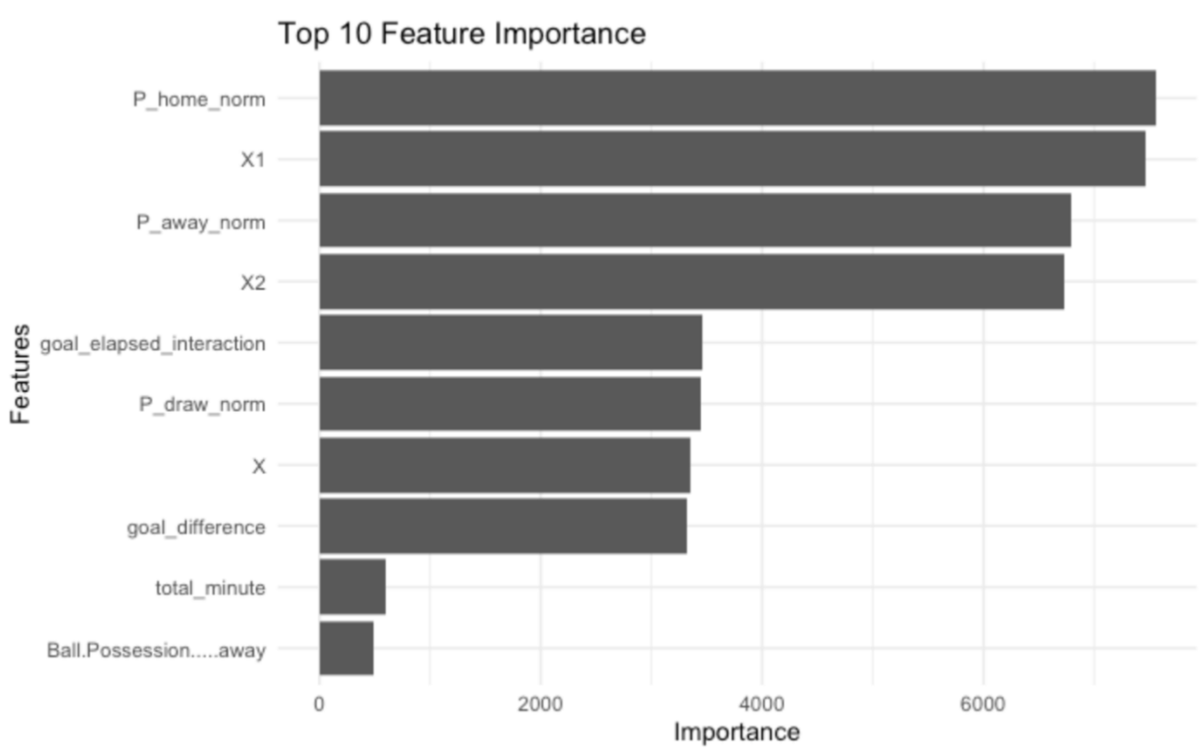
Then variables with high correlation coefficients were manually inspected, and according to the footbal Dynamics, heuristic approches were developed.

Interaction terms of some variables were added. These terms are:

Name	Interaction term 1	Interaction Term 2
Goal_elapsed_interaction	Goal_difference (Goals_home-goals_away)	Total_minute
goal_yellowcards_home_interaction	goal_difference	Yellowcards...home
goal_yellowcards_away_interaction	goal_difference	Yellowcards...away
attack_efficiency_home	Total.Crosses...home	`Goals...home

attack_efficiency_away	Total.Crosses...away	`Goals...away
interaction_passes_attacks_home	Passes...home	Attacks...home
interaction_passes_attacks_away	Passes...away	Attacks...away

For experimental purposes, an initial decision tree was fit including these variables, and they proved to be very high in feature importance, as can be observed from the plot below. As can be seen from these results, adding the interaction terms goal_yellowcards_away_interaction attack_efficiency_away oal_yellowcards_home_interaction yellow_card_difference to the model does not leave room for improvement in the model and increases the risk of overfitting. After careful consideration of these results, some of the built features were used in further analysis.

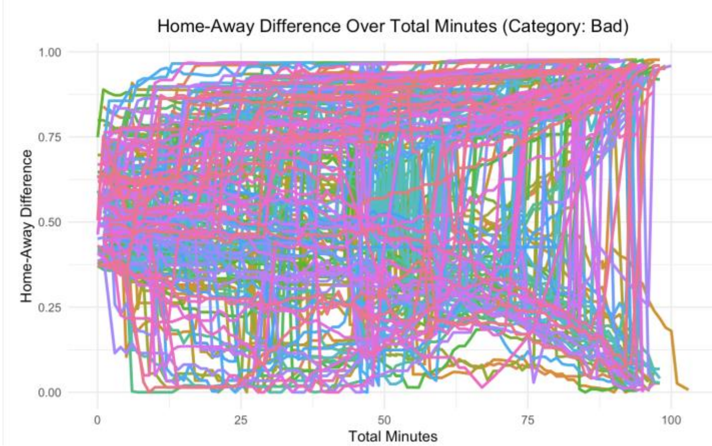


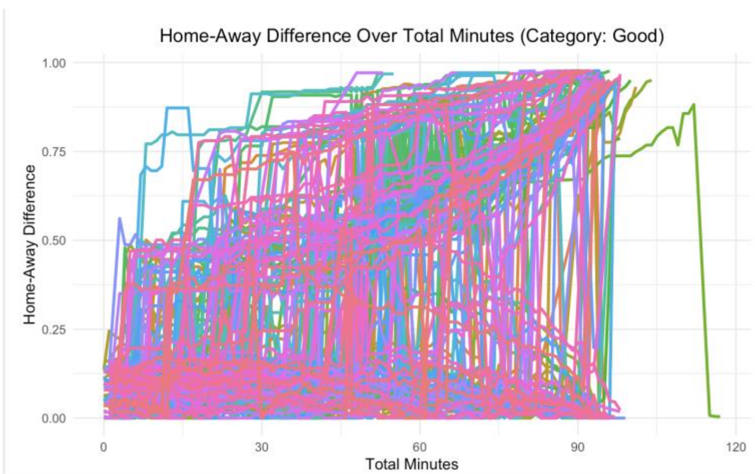
4.2) Minute selection and Time Series Analysis

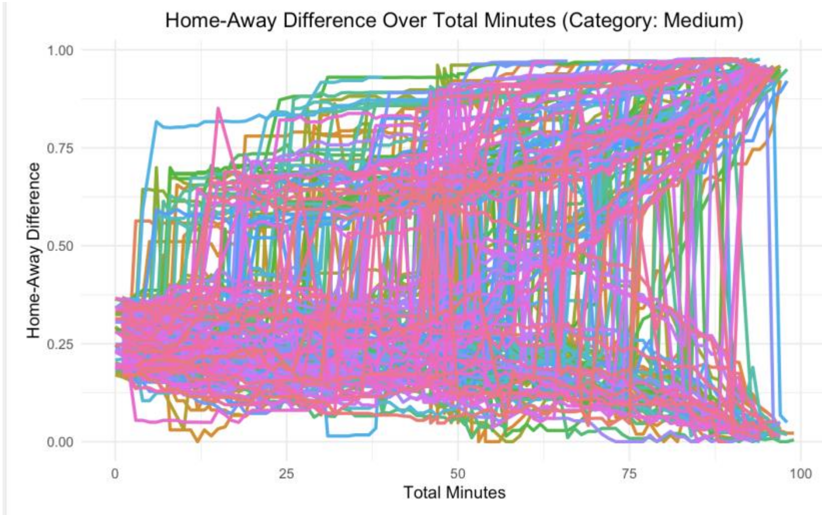
Using time series analysis to determine the optimal time for predictions is a logical approach because it leverages the temporal dynamics inherent in the data. Football match statistics, such as betting odds, ball possession, and goals, evolve over time, often following specific patterns, trends, or abrupt changes. Time series analysis helps identify these behaviors and pinpoint critical moments when decisions are most effective. This ensures decisions are made at stable or optimal points, improving accuracy and profitability. Time series analysis also mitigates risks by highlighting volatile periods and guiding predictions toward more consistent intervals. The metric analyzed using time series analysis was the difference in probabilities between the home and away teams. Examining this difference reveals the moments when the gap widens (as indicated by increasing odds differences set by the bookmaker). Making predictions slightly before these moments can both increase profit and maintain model accuracy. However, there is a trade-off: selecting later time points leads to higher accuracy but lower returns, while choosing earlier moments increases revenue potential but risks reduced accuracy due to limited data availability. Striking the right balance between these factors is crucial for an effective and profitable strategy.

To perform this analysis, matches were first categorized into "Good," "Medium," and "Bad." The motivation behind this was that the strength difference between the teams introduces bias, and the odds provided by the bookmaker vary accordingly. Since team-specific information such as recent performance, league rankings, team quality, or winning streaks was not available, these factors were inherently captured in the odds set by the bookmaker. For categorization, the normalized probabilities used in HW2 were also applied here. The difference between the probabilities of the home and away teams was calculated, and their absolute values were sorted. Subsequently, the 33.33rd and 66.67th percentiles were calculated, with matches having the smallest differences categorized as "Good," the largest as "Bad," and those in between as "Medium."

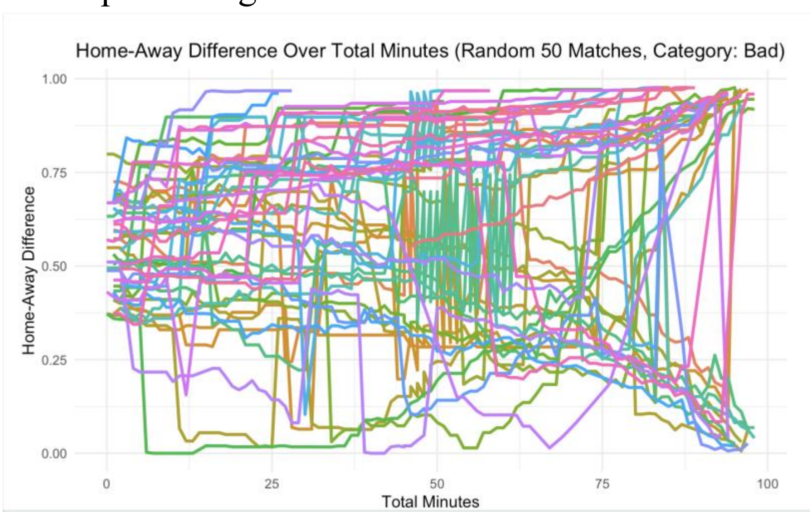
Time series analysis was conducted for each category, resulting in the following plots.

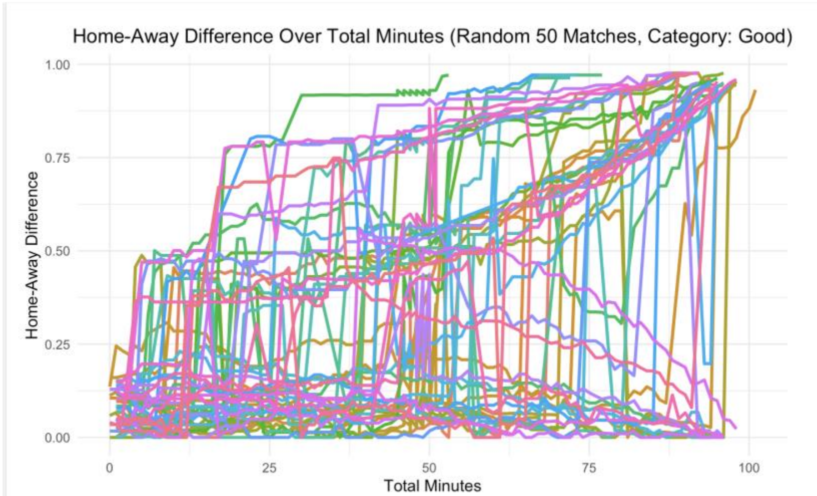


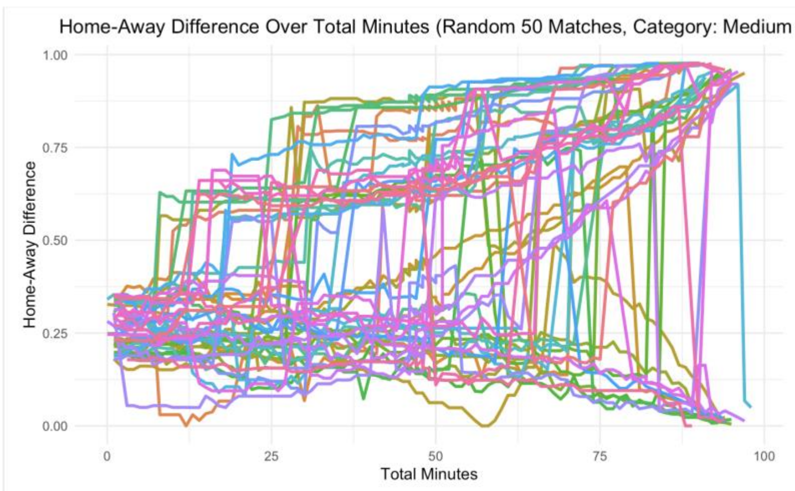




These plots did not yield meaningful insights due to the large number of matches. To address this, a random selection of 50 matches from each category was made, but the results remained unchanged.
These plots are given below.



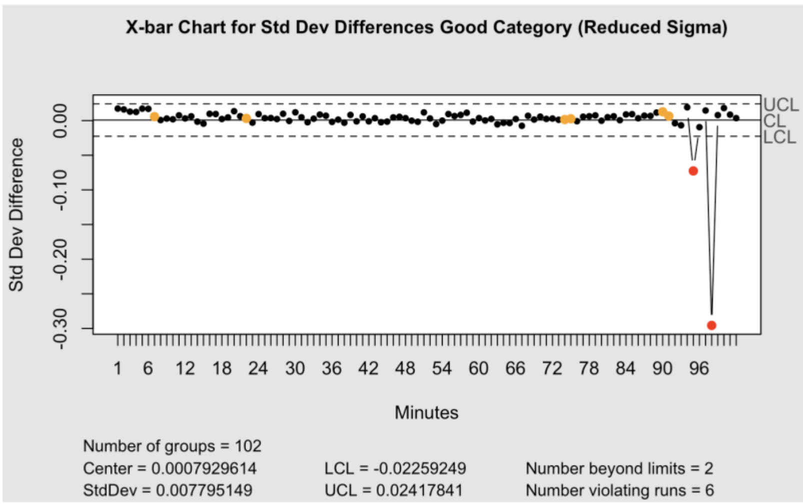


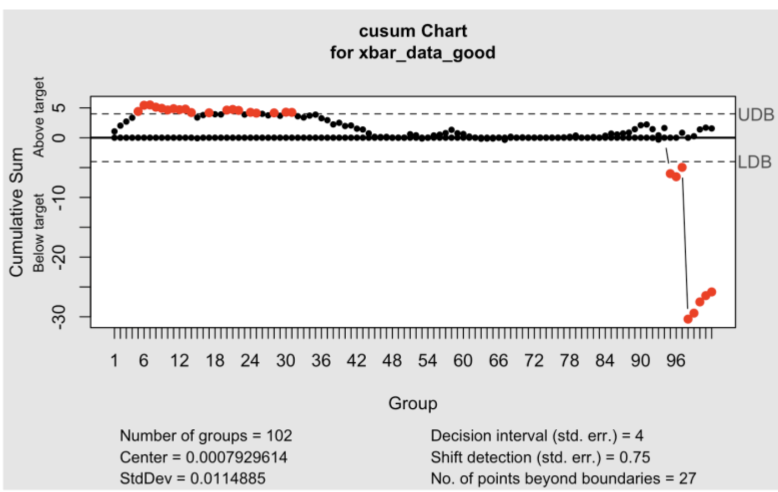


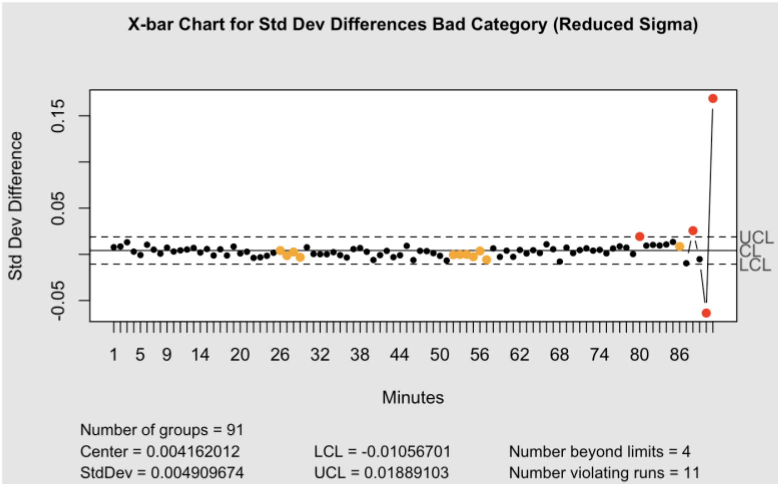
For a more precise approach, quality control charts were utilized, and predictions were made just before points identified as out-of-control (OOC).

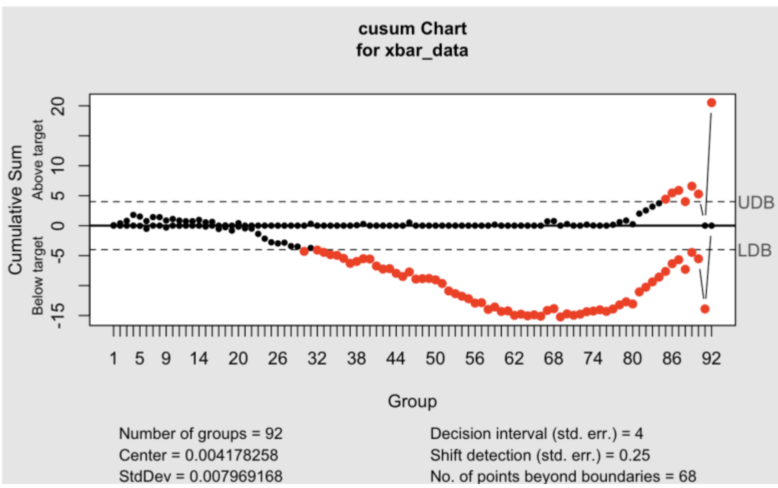
For each category, the standard deviation of the home-away probability differences was calculated for each minute across the matches. Subsequently, the differences between consecutive minutes were computed. This data was then analyzed using X-bar and CUSUM charts to identify patterns and potential out-of-control points. The CUSUM control chart was used because it is particularly effective in detecting small, sustained shifts in a process over time. Unlike X-bar charts, which focus on individual data points, the CUSUM chart accumulates deviations from the target value, making it sensitive to gradual changes that might not be immediately apparent.

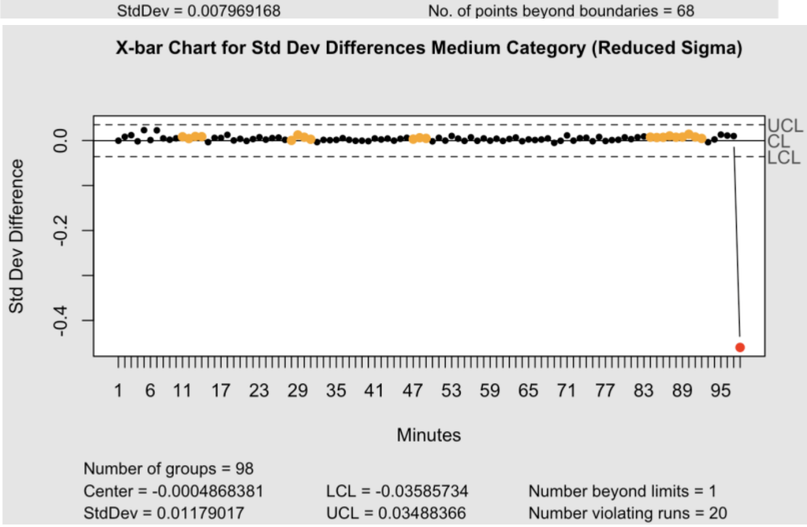
Initially, when the control charts were created without grouping the data, the following results were obtained.

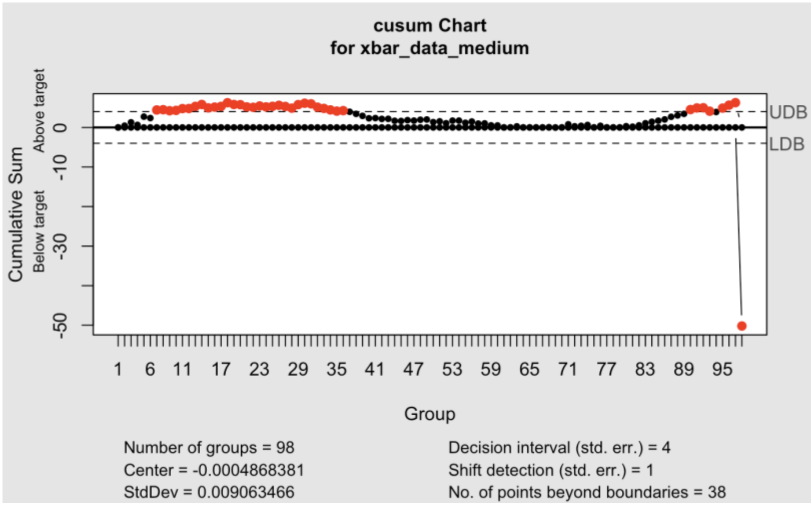





X-bar Chart for Std Dev Differences Bad Category (Reduced Sigma)
Std Dev Difference
Minutes
UCL
CL
LCL
Number of groups = 91
Center = 0.004162012
StdDev = 0.004909674
LCL = -0.01056701
UCL = 0.01889103
Number beyond limits = 4
Number violating runs = 11

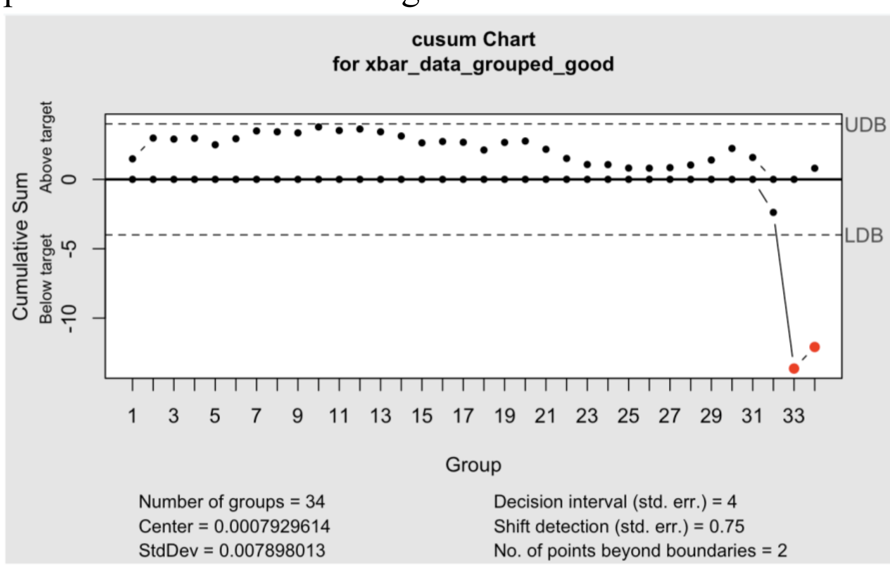

cusum Chart
for xbar_data
Cumulative Sum
Above target
Below target
Group
UDB
LDB
Number of groups = 92
Center = 0.004178258
StdDev = 0.007969168
Decision interval (std. err.) = 4
Shift detection (std. err.) = 0.25
No. of points beyond boundaries = 68

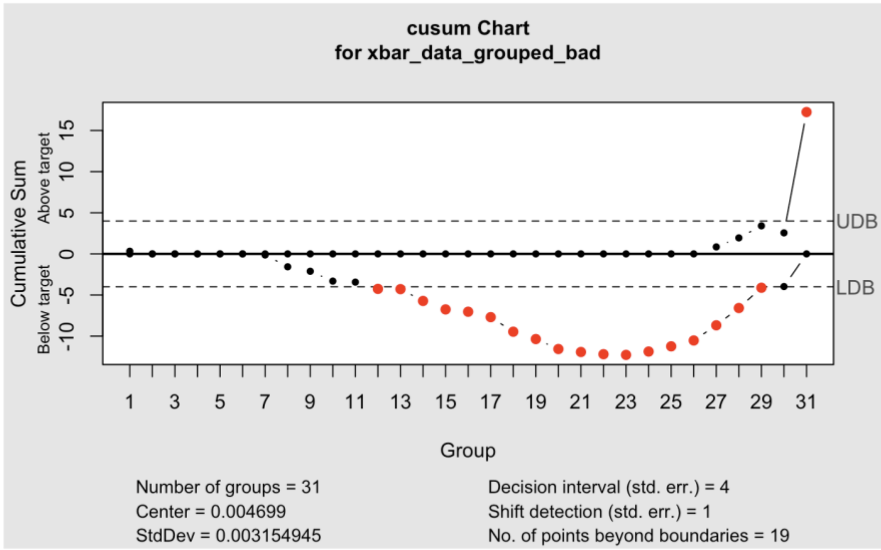

X-bar Chart for Std Dev Differences Medium Category (Reduced Sigma)
Std Dev Difference
Minutes
UCL
CL
LCL
Number of groups = 98
Center = -0.0004868381
StdDev = 0.01179017
LCL = -0.03585734
UCL = 0.03488366
Number beyond limits = 1
Number violating runs = 20

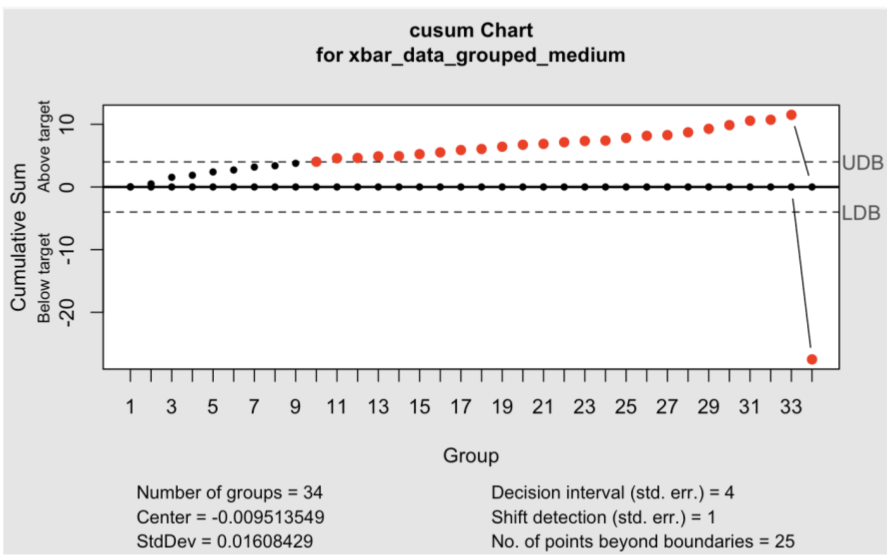


Grouping minutes into three-minute intervals reduces random noise and smoothens the data, making trends and sustained changes easier to identify. It increases stability by providing more reliable estimates. After dividing the minutes into groups of three, the following results were obtained.

No OOC points were observed in the X-bar charts; therefore, the OOC points were identified using the CUSUM charts.







The CUSUM charts obtained with the grouped data are provided below.

Upon examining the charts, an OOC point is observed around the 25th minute for matches in the Bad category. Therefore, this minute is selected as the prediction minute for the Bad category. For the Good category, the chart shows no OOC points, even when parameters are adjusted, but the UCL remains flat. The midpoint of this stability, the 36th minute, is chosen as the prediction minute. For the Medium category, an OOC point is detected at the 30th minute, which is selected as the prediction minute.

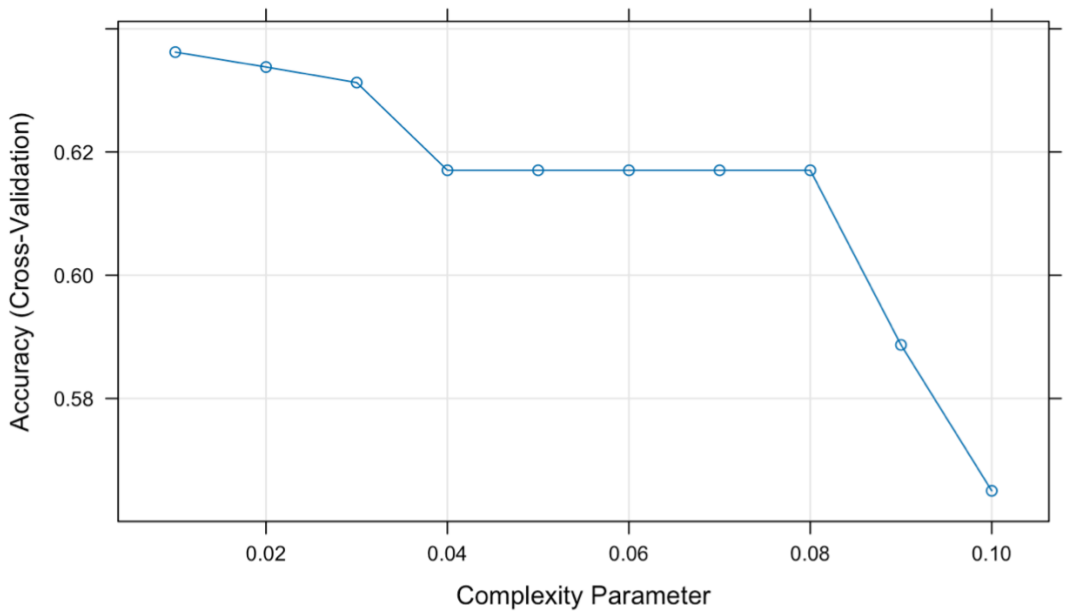
4.3) Decision Models

a) Base model: Decision Tree

As the basic model, we used a Decision Tree classifier. Decision trees were selected due to their simplicity of use, interpretability, and capacity to efficiently handle both numerical and categorical data. It is also a quick method.

As a result of the cross-validation, we ended up with the following complexity graph. We decided the best trade-off between accuracy and complexity would be with complexity parameter 0.08.

tune_grid <- expand.grid(cp = 0.08) # Complexity parameter



Output:

```
Overall Statistics

               Accuracy : 0.5807
                 95% CI : (0.5709, 0.5905)
    No Information Rate : 0.4965
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3534

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity           0.42136   0.6075   0.6623
Specificity           0.77366   0.8011   0.8008
Pos Pred Value        0.35164   0.7508   0.5612
Neg Pred Value        0.82110   0.6742   0.8604
Prevalence            0.22560   0.4965   0.2779
Detection Rate        0.09506   0.3016   0.1840
Detection Prevalence  0.27033   0.4018   0.3279
Balanced Accuracy     0.59751   0.7043   0.7315
Decision Tree Accuracy: 0.5807239
```

---

b) Random Forest

In this project, we used the Random Forest algorithm to predict football match outcomes and optimize our decision-making process for live betting. The Random Forest approach was chosen for its robustness, ability to handle non-linear relationships, and inherent feature importance evaluation.

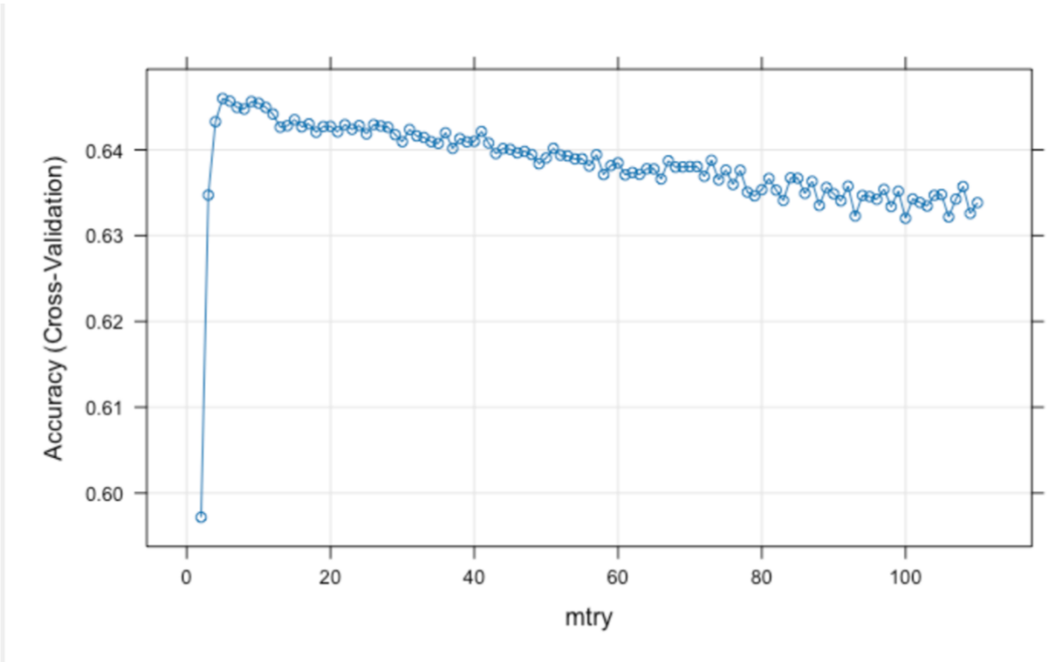
Cross-Validation and Model Accuracy

Random Forest inherently uses the concept of **out-of-bag (OOB) instances** for internal validation. With the help of this, we performed a **10-fold cross-validation**, providing the model's accuracy without requiring additional data splitting.

The model's accuracy and generalizability are ensured by combining parameter tuning with Random Forest's OOB mechanism, which is essential given the dynamic and real-time nature of developing live betting strategies.
This methodology allowed us to create a model that balances prediction accuracy.

We also conducted parameter tuning with 4-fold cross validation for Max_nodes and number of trees to find a balance between complexity and accuracy. We ended up with the best tuned parameter values as number of trees= 300, max_nodes=10.

As a result of cross-validation, we ended up with the following mtry graph. We decided the best trade-off between accuracy and mtry complexity would be with complexity parameter 3. We commented the code on the script not to interfere with the flow of the document, since this output took approximately 12 hours ☺.



Output:

```
Cross-validated Accuracy: 0.6378563

Call:
 randomForest(x = x, y = y, ntree = 300, mtry = param$mtry, maxnodes = 10)
               Type of random forest: classification
                     Number of trees: 300
No. of variables tried at each split: 3

        OOB estimate of  error rate: 36.71%
Confusion matrix:
     0     1    2 class.error
0 2870  6990 3312   0.7821136
1  704 16996 1564   0.1177326
2  653  3771 9431   0.3193071
```

c) Multinomial logistics regression

We also used **Multinomial Logistic Regression** as part of our analysis methodology to predict football match outcomes.

A confusion matrix was computed to assess how well the model classified match outcomes. The overall accuracy of the model on the training data was calculated from the confusion matrix.

It is a simple and interpretable model, which also provides the probabilities of belonging to a specific class, which is essentially very useful for the analysis done in this project, since our aim is to maximize the expected profit on a bet placed at a certain minute. This transparency allows us to observe the trade-off between the placed (or not placed) bets.

```
Residual Deviance: 73133.64
AIC: 73565.64
[1] "Multinomial Logistic Regression Accuracy:"
 Accuracy
0.6557646
```

D) XgBoost

Extreme Gradient Boosting, or XGBoost, is also an ensemble learning technique that we used in this project to forecast football match results. For further analysis, we tried training 3 different models for the 3 different categories: good, medium and bad. This analysis was not done in other methods, since we wanted to compare the accuracies first to see the tradeoff between complexity of the overall model vs. the average return.

To optimize XGBoost, we conducted hyperparameter tuning using a cross-validation approach. The tuned parameters are learning_rate, max_depth. We observed, that this approach was more prone to overfitting, since we obtained training accuracy values of 99% and test accuracy values of 50%. After a reduction in max_depth and learning rates, we achieved optimal results.

```
params <- list(
  objective = "multi:softmax",
  num_class = 3,
  eval_metric = "merror",
  eta = 0.05,
  max_depth = 3,
  nthread = 2
)
```

Outputs for all categories:

```
-- Good Category --
Confusion Matrix and Statistics

          Reference
Prediction    0    1    2
         0 2979  305  215
         1  832 4165  565
         2  630  888 4821

Overall Statistics

               Accuracy : 0.7769
                 95% CI : (0.7703, 0.7835)
    No Information Rate : 0.3637
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6616

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            0.6708   0.7773   0.8607
Specificity            0.9526   0.8609   0.8451
Pos Pred Value         0.8514   0.7488   0.7605
Neg Pred Value         0.8772   0.8787   0.9139
Prevalence             0.2884   0.3479   0.3637
Detection Rate         0.1934   0.2705   0.3131
Detection Prevalence   0.2272   0.3612   0.4116
Balanced Accuracy      0.8117   0.8191   0.8529
```

```
-- medium Category --
Confusion Matrix and Statistics

          Reference
Prediction    0    1    2
         0 3613  421  319
         1  828 4814  532
         2  677  487 4281

Overall Statistics

               Accuracy : 0.7956
                 95% CI : (0.7893, 0.8019)
    No Information Rate : 0.3583
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6925

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            0.7059   0.8413   0.8342
Specificity            0.9318   0.8673   0.8926
Pos Pred Value         0.8300   0.7797   0.7862
Neg Pred Value         0.8705   0.9073   0.9192
Prevalence             0.3204   0.3583   0.3213
Detection Rate         0.2262   0.3014   0.2680
Detection Prevalence   0.2725   0.3866   0.3409
Balanced Accuracy      0.8189   0.8543   0.8634
```

```
-- bad Category --
Confusion Matrix and Statistics

          Reference
Prediction    0    1    2
         0 2061   43  126
         1 1331 8058  467
         2  221   83 2529

Overall Statistics

               Accuracy : 0.8478
                 95% CI : (0.8419, 0.8535)
    No Information Rate : 0.5486
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.729

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            0.5704   0.9846   0.8101
Specificity            0.9851   0.7330   0.9742
Pos Pred Value         0.9242   0.8176   0.8927
Neg Pred Value         0.8777   0.9751   0.9509
Prevalence             0.2422   0.5486   0.2093
Detection Rate         0.1381   0.5401   0.1695
Detection Prevalence   0.1495   0.6606   0.1899
Balanced Accuracy      0.7777   0.8588   0.8921
```

5) Results

After a quick comparison of the accuracy values, it is obvious that the maximum accuracy is reached with xgboost method. However, for this specific analysis, this is not sufficient, since the profit values (namely the corresponding bet values for each match) are what determines the goodness criteria of the models.

Therefore, we calculated the profit values for all of the approaches, and then proceeded to make conclusions using that data. The profit is calculated by the return of the model (that is the accuracy (as 0,1) * odd of the bookmaker for the prediction) – 1.

The advantage of using these approaches were that all of them were able to provide probability values for each prediction of the result belonging to the selected class. This probability helped us when dealing with "no bet" situations, and we could tune a threshold under which we did not bet.

We decided to consider 2 options: not allowing "no bet" scenario, meaning that whatever the model predicts we have to bet somehow, and allowing "no bet" scenario. When allowing a no bet scenario, we made use of the probability outcomes of the model. If the threshold (the probability that the match belonging to the predicted class) was under a value, we chose "no bet". For the tuning of the threshold, we developed an iterative approach to find the no-bet threshold to maximize out profit. This iteration tries values between 0.5 and 0.9 with 0.01 step length to come up with a value that maximizes the profit, not the accuracy.

We tested all these models using the provided test data, made predictions and calculated the expected profits.

The results are of both analyses are as follows, color coded blue for the "bet for all outcomes" scenario and green for "allowing no bet" scenario.

Decision Tree:

Category : GOOD

```
Confusion Matrix and Statistics

          Reference
Prediction 0 1 2
         0 7 5 2
         1 3 8 1
         2 3 3 9

Overall Statistics

               Accuracy : 0.5854
                 95% CI : (0.4211, 0.7368)
    No Information Rate : 0.3902
    P-Value [Acc > NIR] : 0.0089

                  Kappa : 0.3815

 Mcnemar's Test P-Value : 0.6369

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            0.5385   0.5000   0.7500
Specificity            0.7500   0.8400   0.7931
Pos Pred Value         0.5000   0.6667   0.6000
Neg Pred Value         0.7778   0.7241   0.8846
Prevalence             0.3171   0.3902   0.2927
Detection Rate         0.1707   0.1951   0.2195
Detection Prevalence   0.3415   0.2927   0.3659
Balanced Accuracy      0.6442   0.6700   0.7716
```

"Optimal Threshold for Good Category: 0.5"

[1] "Maximum Total Return for Good Category: 61.8"

[1] "Maximum Total Profit for Good Category: 44.8"

"Total Return for Good Category: 61.8"

"Average Return per Good Category Match: 1.50731707317073"

"Total Profit for Good Category: 44.8"

"Average Profit per Good Category Match: 1.09268292682927"

Category : MEDIUM

```
Confusion Matrix and Statistics

          Reference
Prediction 0 1 2
         0 1 3 4
         1 4 6 3
         2 2 5 5

Overall Statistics

               Accuracy : 0.3636
                 95% CI : (0.204, 0.5488)
    No Information Rate : 0.4242
    P-Value [Acc > NIR] : 0.8100

                  Kappa : 0.0198

 Mcnemar's Test P-Value : 0.7269

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            0.1429   0.4286   0.4167
Specificity            0.7308   0.6316   0.6667
Pos Pred Value         0.1250   0.4615   0.4167
Neg Pred Value         0.7600   0.6000   0.6667
Prevalence             0.2121   0.4242   0.3636
Detection Rate         0.0303   0.1818   0.1515
Detection Prevalence   0.2424   0.3939   0.3636
Balanced Accuracy      0.4368   0.5301   0.5417
```

"Optimal Threshold: 0.63"

"Maximum Total Return: 17.92"

"Maximum Total Profit: 2.92"

"Total Return for Medium Category: 23.04"

"Average Return per Medium Category Match: 0.698181818181818"

"Total Profit for Medium Category: 2.04"

"Average Profit per Medium Category Match: 0.0618181818181819"

Category : BAD

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1  2
         0  0  4  1
         1  3 19  3
         2  1  0  2

Overall Statistics

               Accuracy : 0.6364
                 95% CI : (0.4512, 0.796)
    No Information Rate : 0.697
    P-Value [Acc > NIR] : 0.8290

                  Kappa : 0.1681

 Mcnemar's Test P-Value : 0.3701

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            0.0000   0.8261  0.33333
Specificity            0.8276   0.4000  0.96296
Pos Pred Value         0.0000   0.7600  0.66667
Neg Pred Value         0.8571   0.5000  0.86667
Prevalence             0.1212   0.6970  0.18182
Detection Rate         0.0000   0.5758  0.06061
Detection Prevalence   0.1515   0.7576  0.09091
Balanced Accuracy      0.4138   0.6130  0.64815
```

[1] "Optimal Threshold for Bad Category: 0.68"

[1] "Maximum Total Return for Bad Category: 24.89"

[1] "Maximum Total Profit for Bad Category: 15.89"

[1] "Total Return for Bad Category: 26.5"

[1] "Average Return per Bad Category Match: 0.803030303030303"

[1] "Total Profit for Bad Category: 14.5"

[1] "Average Profit per Bad Category Match: 0.439393939393939"

Multinomial Logistics Regression

Category : GOOD

```
Confusion Matrix and Statistics

          Reference
Prediction 0 1 2
         0 1 7 5
         1 6 6 1
         2 6 3 6

Overall Statistics

               Accuracy : 0.3171
                 95% CI : (0.1808, 0.4809)
    No Information Rate : 0.3902
    P-Value [Acc > NIR] : 0.8696

                  Kappa : -0.0214

 Mcnemar's Test P-Value : 0.7607

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity           0.07692   0.3750   0.5000
Specificity           0.57143   0.7200   0.6897
Pos Pred Value        0.07692   0.4615   0.4000
Neg Pred Value        0.57143   0.6429   0.7692
Prevalence            0.31707   0.3902   0.2927
Detection Rate        0.02439   0.1463   0.1463
Detection Prevalence  0.31707   0.3171   0.3659
Balanced Accuracy     0.32418   0.5475   0.5948
```

[1] "Optimal Threshold for Good Category: 0.55"

[1] "Maximum Total Return for Good Category: 10.57"

[1] "Maximum Total Profit for Good Category: 5.57"

[1] "Total Return for Good Category: 20.63"

[1] "Average Return per Good Category Match: 0.503170731707317"

[1] "Total Profit for Good Category: -7.37"

[1] "Average Profit per Good Category Match: -0.179756097560976"

Category : MEDIUM

```
Confusion Matrix and Statistics

          Reference
Prediction 0 1 2
         0 6 5 3
         1 0 6 2
         2 1 3 7

Overall Statistics

               Accuracy : 0.5758
                 95% CI : (0.3922, 0.7452)
    No Information Rate : 0.4242
    P-Value [Acc > NIR] : 0.05732

                  Kappa : 0.3815

 Mcnemar's Test P-Value : 0.10228

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            0.8571   0.4286   0.5833
Specificity            0.6923   0.8947   0.8095
Pos Pred Value         0.4286   0.7500   0.6364
Neg Pred Value         0.9474   0.6800   0.7727
Prevalence             0.2121   0.4242   0.3636
Detection Rate         0.1818   0.1818   0.2121
Detection Prevalence   0.4242   0.2424   0.3333
Balanced Accuracy      0.7747   0.6617   0.6964
```

"Optimal Threshold for Medium Category: 0.58"

[1] "Maximum Total Return for Medium Category: 13.03"

[1] "Maximum Total Profit for Medium Category: 8.03"

[1] "Total Return for Medium Category: 40.38"

[1] "Average Return per Medium Category Match: 1.22363636363636"

[1] "Total Profit for Medium Category: 26.38"

[1] "Average Profit per Medium Category Match: 0.799393939393939"

Category : BAD

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1  2
         0  0  2  0
         1  3 21  3
         2  1  0  3

Overall Statistics

               Accuracy : 0.7273
                 95% CI : (0.5448, 0.867)
    No Information Rate : 0.697
    P-Value [Acc > NIR] : 0.4347

                  Kappa : 0.3188

 Mcnemar's Test P-Value : 0.2407

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity           0.00000   0.9130  0.50000
Specificity           0.93103   0.4000  0.96296
Pos Pred Value        0.00000   0.7778  0.75000
Neg Pred Value        0.87097   0.6667  0.89655
Prevalence            0.12121   0.6970  0.18182
Detection Rate        0.00000   0.6364  0.09091
Detection Prevalence  0.06061   0.8182  0.12121
Balanced Accuracy     0.46552   0.6565  0.73148
```

[1] "Optimal Threshold for Bad Category: 0.5"

[1] "Maximum Total Return for Bad Category: 31.35"

[1] "Maximum Total Profit for Bad Category: 25.35"

[1] "Total Return for Bad Category: 31.35"

[1] "Average Return per Bad Category Match: 0.95"

[1] "Total Profit for Bad Category: 22.35"

[1] "Average Profit per Bad Category Match: 0.677272727272727"

Random Forest

Category : GOOD

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1  2
         0  0  0  0
         1  5 11  4
         2  8  5  8

Overall Statistics

               Accuracy : 0.4634
                 95% CI : (0.3066, 0.6258)
    No Information Rate : 0.3902
    P-Value [Acc > NIR] : 0.210659

                  Kappa : 0.1867

 Mcnemar's Test P-Value : 0.004402

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            0.0000   0.6875   0.6667
Specificity            1.0000   0.6400   0.5517
Pos Pred Value            NaN   0.5500   0.3810
Neg Pred Value         0.6829   0.7619   0.8000
Prevalence             0.3171   0.3902   0.2927
Detection Rate         0.0000   0.2683   0.1951
Detection Prevalence   0.0000   0.4878   0.5122
Balanced Accuracy      0.5000   0.6638   0.6092
```

"Optimal Threshold for Good Category: 0.57"

[1] "Maximum Total Return for Good Category: 15.24"

[1] "Maximum Total Profit for Good Category: 11.24"

1] "Total Return for Good Category (RF): 39.55"

[1] "Average Return per Good Category Match (RF): 0.964634146341463"

[1] "Total Profit for Good Category (RF): 18.55"

[1] "Average Profit per Good Category Match (RF): 0.452439024390244"

Category : MEDIUM

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1  2
         0  0  0  0
         1  5 10  3
         2  2  4  9

Overall Statistics

               Accuracy : 0.5758
                 95% CI : (0.3922, 0.7452)
    No Information Rate : 0.4242
    P-Value [Acc > NIR] : 0.05732

                  Kappa : 0.2968

 Mcnemar's Test P-Value : 0.06748

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            0.0000   0.7143   0.7500
Specificity            1.0000   0.5789   0.7143
Pos Pred Value            NaN   0.5556   0.6000
Neg Pred Value         0.7879   0.7333   0.8333
Prevalence             0.2121   0.4242   0.3636
Detection Rate         0.0000   0.3030   0.2727
Detection Prevalence   0.0000   0.5455   0.4545
Balanced Accuracy      0.5000   0.6466   0.7321
```

[1] "Optimal Threshold for Medium Category: 0.5"

[1] "Maximum Total Return for Medium Category: 16.28"

[1] "Maximum Total Profit for Medium Category: 14.28

[1] "Total Return for Medium Category (RF): 36.87"

[1] "Average Return per Medium Category Match (RF): 1.11727272727273"

[1] "Total Profit for Medium Category (RF): 21.87"

[1] "Average Profit per Medium Category Match (RF): 0.662727272727273"

Category : BAD

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1  2
         0  0  0  0
         1  3 23  3
         2  1  0  3

Overall Statistics

               Accuracy : 0.7879
                 95% CI : (0.6109, 0.9102)
    No Information Rate : 0.697
    P-Value [Acc > NIR] : 0.1725

                  Kappa : 0.4196

 Mcnemar's Test P-Value : 0.0719

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            0.0000   1.0000  0.50000
Specificity            1.0000   0.4000  0.96296
Pos Pred Value            NaN   0.7931  0.75000
Neg Pred Value         0.8788   1.0000  0.89655
Prevalence             0.1212   0.6970  0.18182
Detection Rate         0.0000   0.6970  0.09091
Detection Prevalence   0.0000   0.8788  0.12121
Balanced Accuracy      0.5000   0.7000  0.73148
```

"Optimal Threshold for Bad Category: 0.5"

"Maximum Total Return for Bad Category: 33.42"

"Maximum Total Profit for Bad Category: 26.42

"Total Return for Bad Category (RF): 35.52"

"Average Return per Bad Category Match (RF): 1.07636363636364"

"Total Profit for Bad Category (RF): 28.52"

"Average Profit per Bad Category Match (RF): 0.864242424242424"

XGboost

Category : GOOD

```
        0  3  3  1
        1  3 11  2
        2  7  2  9

Overall Statistics

               Accuracy : 0.561
                 95% CI : (0.3975, 0.7153)
    No Information Rate : 0.3902
    P-Value [Acc > NIR] : 0.01984

                  Kappa : 0.3399

 Mcnemar's Test P-Value : 0.21229

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity           0.23077   0.6875   0.7500
Specificity           0.85714   0.8000   0.6897
Pos Pred Value        0.42857   0.6875   0.5000
Neg Pred Value        0.70588   0.8000   0.8696
Prevalence            0.31707   0.3902   0.2927
Detection Rate        0.07317   0.2683   0.2195
Detection Prevalence  0.17073   0.3902   0.4390
Balanced Accuracy     0.54396   0.7438   0.7198
```

[1] "Optimal Threshold for Good Category

(XGBoost): 0.52"

[1] "Maximum Total Return for Good Category (XGBoost): 14.33"

[1] "Maximum Total Profit for Good Category (XGBoost): 7.33"

[1] "Total Return for Good Category (XG): 48.02"

[1] "Average Return per Good Category Match (XG): 1.17121951219512"

[1] "Total Profit for Good Category (XG): 30.02"

[1] "Average Profit per Good Category Match (XG): 0.732195121951219"

Category : MEDIUM

```
          Reference
Prediction 0 1 2
         0 1 4 1
         1 4 6 4
         2 2 4 7

Overall Statistics

               Accuracy : 0.4242
                 95% CI : (0.2548, 0.6078)
    No Information Rate : 0.4242
    P-Value [Acc > NIR] : 0.5663

                  Kappa : 0.0978

 Mcnemar's Test P-Value : 0.9536

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            0.1429   0.4286   0.5833
Specificity            0.8077   0.5789   0.7143
Pos Pred Value         0.1667   0.4286   0.5385
Neg Pred Value         0.7778   0.5789   0.7500
Prevalence             0.2121   0.4242   0.3636
Detection Rate         0.0303   0.1818   0.2121
Detection Prevalence   0.1818   0.4242   0.3939
Balanced Accuracy      0.4753   0.5038   0.6488
```

[1] "Optimal Threshold for Medium Category (XGBoost): 0.5"

[1] "Maximum Total Return for Medium Category (XGBoost): 13.76"

[1] "Maximum Total Profit for Medium Category (XGBoost): 9.76"

[1] "Total Return for Medium Category (XG): 25.16"

[1] "Average Return per Medium Category Match (XG): 0.762424242424242"

[1] "Total Profit for Medium Category (XG): 6.16"

[1] "Average Profit per Medium Category Match (XG): 0.186666666666667"

Category : BAD

```
          Reference
Prediction  0  1  2
         0  0  0  1
         1  3 23  3
         2  1  0  2

Overall Statistics

               Accuracy : 0.7576
                 95% CI : (0.5774, 0.8891)
    No Information Rate : 0.697
    P-Value [Acc > NIR] : 0.2912

                  Kappa : 0.34

 Mcnemar's Test P-Value : 0.1116

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            0.0000   1.0000  0.33333
Specificity            0.9655   0.4000  0.96296
Pos Pred Value         0.0000   0.7931  0.66667
Neg Pred Value         0.8750   1.0000  0.86667
Prevalence             0.1212   0.6970  0.18182
Detection Rate         0.0000   0.6970  0.06061
Detection Prevalence   0.0303   0.8788  0.09091
Balanced Accuracy      0.4828   0.7000  0.64815
```

[1] "Optimal Threshold for Bad Category (XGBoost): 0.51"

[1] "Maximum Total Return for Bad Category (XGBoost): 30.99"

[1] "Maximum Total Profit for Bad Category (XGBoost): 26.99"

[1] "Total Return for Bad Category (XG): 33.86"

[1] "Average Return per Bad Category Match (XG): 1.02606060606061"

[1] "Total Profit for Bad Category (XG): 25.86"

[1] "Average Profit per Bad Category Match (XG): 0.783636363636364"

Summary Table for Average Profit per Match:

Category	Good	Medium	Bad
Decision Tree (Base model)	1.09	0.09	0.48
Random Forest	0.13	0.24	0.768
Multinomial Logistics Regression	0.27	0.43	0.80
Xgboost	0.18	0.3	0.81

When we compare the training and test errors in xgboost, we observe a significant reduction in good and medium categories, which may indicate that the models overfit the training data and are not as robust as we would like. These two categories require more parameter tuning, or another model to consider in prediction. Although the accuracy levels were very high in xgboost, it is clear that the best profit is not given by xgboost. This may be due to the odds of the matches it detects correctly.

In the “Good” matches category, which are the matches that are more balanced in terms of team performance, we observe that our base model (decision tree) has the best profit returns, although the overall accuracy was only 58%. This indicates that our model predicts critical matches with high odds more accurately.

In Medium matches, the best performance is yielded by Multinomial Logistics regression. And in Bad matches the best returns are given by Multinomial Logistics and Xgboost.

Another point of analysis should be the determined thresholds that are yielded by the result of iterations. The thresholds lie very close to 50%, meaning that we earn more by betting more regardless of how sure we are of the result of the match. This is actually why we decided to conduct another analysis, which has no threshold, and thus no option for “no bet”. In that scenario, one can observe that the avergage profits are higher.

Another point of interest is that in Medium matches, both the accuracies and the expected profits are very low compared to the other categories. This can be due to an interaction term which is not very relevant in the two ends of the spectrum, but is effective at the middle. That should be analyzed in further research to come up with a better alternative model.

6) Conclusions and Future Work

Similarities with ELO Model

Similar to the ELO Model, we developed another approach to keep track of a team's success measure, which was to extract the first minutes of each match and get the winning probabilities as a measure of its own. This measure reflects the public opinion regarding the teams, and proved very useful in the analysis.

Opportunities for future work

Due to the limited resources and time constraints for this project, we could not develop separate approaches for Good, Medium and Bad matches, which would have been ideal. For future conclusions, an approach like that could be taken to ensure higher accuracy and return.

Usage of SVM as a prediction model: In this project, we could not use Support Vector Machines (SVMs) as a predictive model due to time constraints. SVMs in the future could bring several benefits. SVMs are particularly good when we are working with datasets that have many features. They're particularly good at tackling complex problems where there's a lot of interaction between variables. SVMs are also better at resisting overfitting, which is an issue we had to deal with a lot in this project. Also, usage of kernels (especially radial basis function) would be very good to detect non-linear relationships. It is also good in providing probabilistic approaches to calculate profits.

7) Code

The code is provided in different R Markdown and HTML Files.