GROUP 3.5 ÜSTÜ

Ogün Gürcan
Öykü Selen Uysal
Enes Özeren
Süheyla Şeker
Musab Emir Baş

IE 256 PROJECT PART III

I. Part

In this part of the project, we determined some features (predictors) to estimate the total number of goals scored in a game (Home Goals + Away Goals) for each game. First of all, we calculated average goals of each team for their home and away matches separately. Then we sum the average home goal of home team and average away goal of away team and we used this sum as a predictor. Other predictors are the match time, the part of the season (beginning of season, middle of season and end of season) and the odds of 'ou' type.

We took average number of home goals and away goals for every team because for instance if home team is Arsenal and away team is Newcastle United we can predict total goals in that match by adding Arsenal's average number of home goals and Newcastle United's average number of away goals. We created a "meantotal" column in matches table and that column stores sum of avarage of home goals of home team and avarage of away goals of away team.

We wanted to consider the the match time, because the time of the day reflects the importance of the match and this may affect the strategy of teams maybe in defense or opposite. Since we evaluate the time of the match as morning, evening or else, they are not numerical variables but categorical variable.

About the part of season, we considered that after transfer periods, newcomers of the team may have difficulties in adapting the environment and this could have a little effect on the score of matches. In addition, the beginning of the season is more tolerable period for the teams but the end of the season is more stressful and critique. Similar to the match time they are categorical variables. (beginning, end and mid of season)

Lastly, we took the over type odds because it directly reflects the probability of the total score being more than 2.5. By taking the over odds, we actually consider the under ending possibility since higher odds for over means that it is more possible to end under while lower over odds means it is more possible to end over.

II. Part

In this part, we used dummy encoding method to determine codes of categorical variables. Dummy encoding makes categorical varaibles available for linear regression analaysis. To do that we assign values 0 or 1 with dummy row and normal columns. We assign 1 to interception and 0 to other. So we have spesific 0 and 1 series for all categorical variable such as Morning Match, Evening Match, Beginning of the Season, Middle of the Season, End of the season,

Table1: Dummy Encoding Model

EveningMatch	MorningMatch	BegOfSeason	MidOfSeason	EndOfSeason
1	0	0	0	1
0	0	0	0	1
0	0	0	0	1
0	0	0	1	0
1	0	0	1	0
1	0	0	0	1

III. Part

In this part, we obtain a model for Total Goals and obtained these plots. Residuals should ensure that all four condition of lineer regression method. We analized these conditions in third part of project.

- **Normal Q-Q Plot**

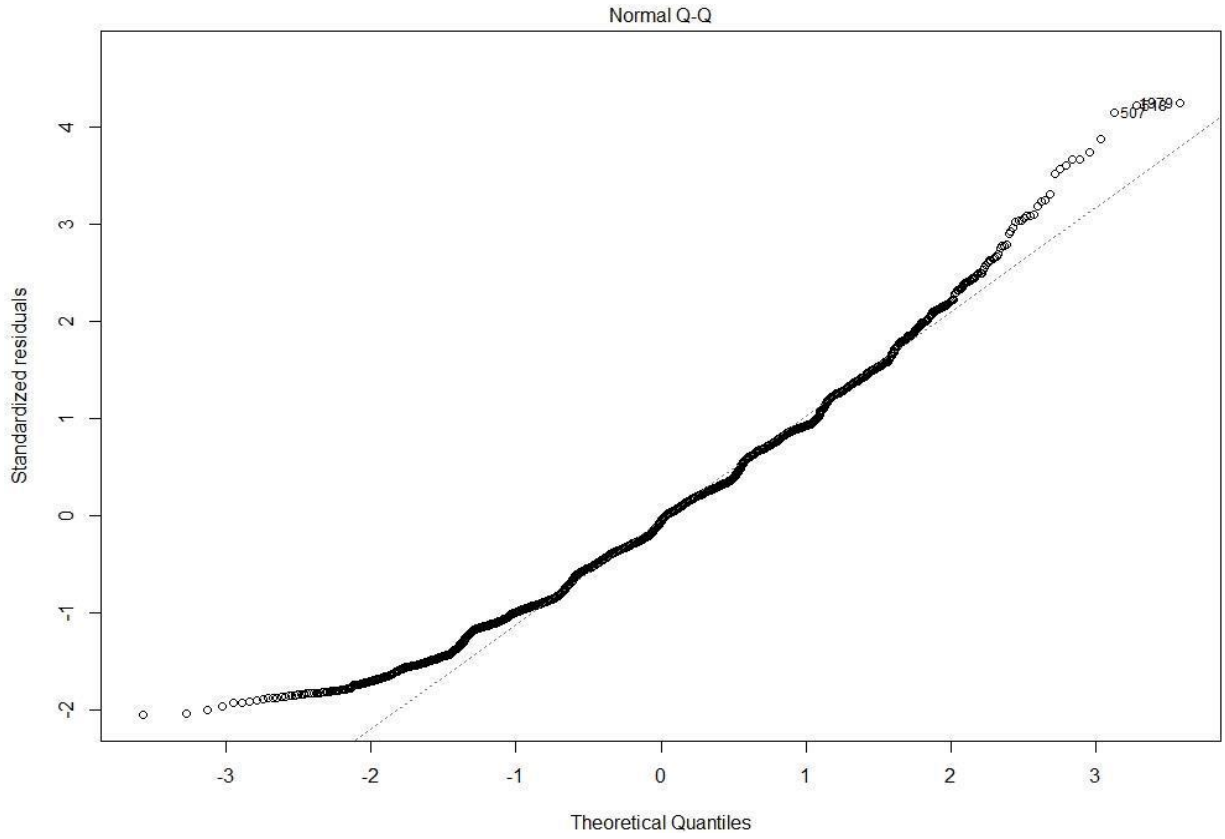

Figure 1: Normal Q-Q Plot for model.1 of Train Data

The normal Q-Q plot supports our model in that it mostly fits a straight line. There are deviations and outliers from the path of course, however they are not severe to danger our model's validity. The linearity assumption of multivariate normality is not severely violated. There are deviations but considering the overall picture, we decided that we can stick with this model.

One of the conditions of lineer regression test check is mean of residuals is approxiametly zero. In this part we can see if it satisfies the condition.

- **Residual vs. Fitted Values Graph**

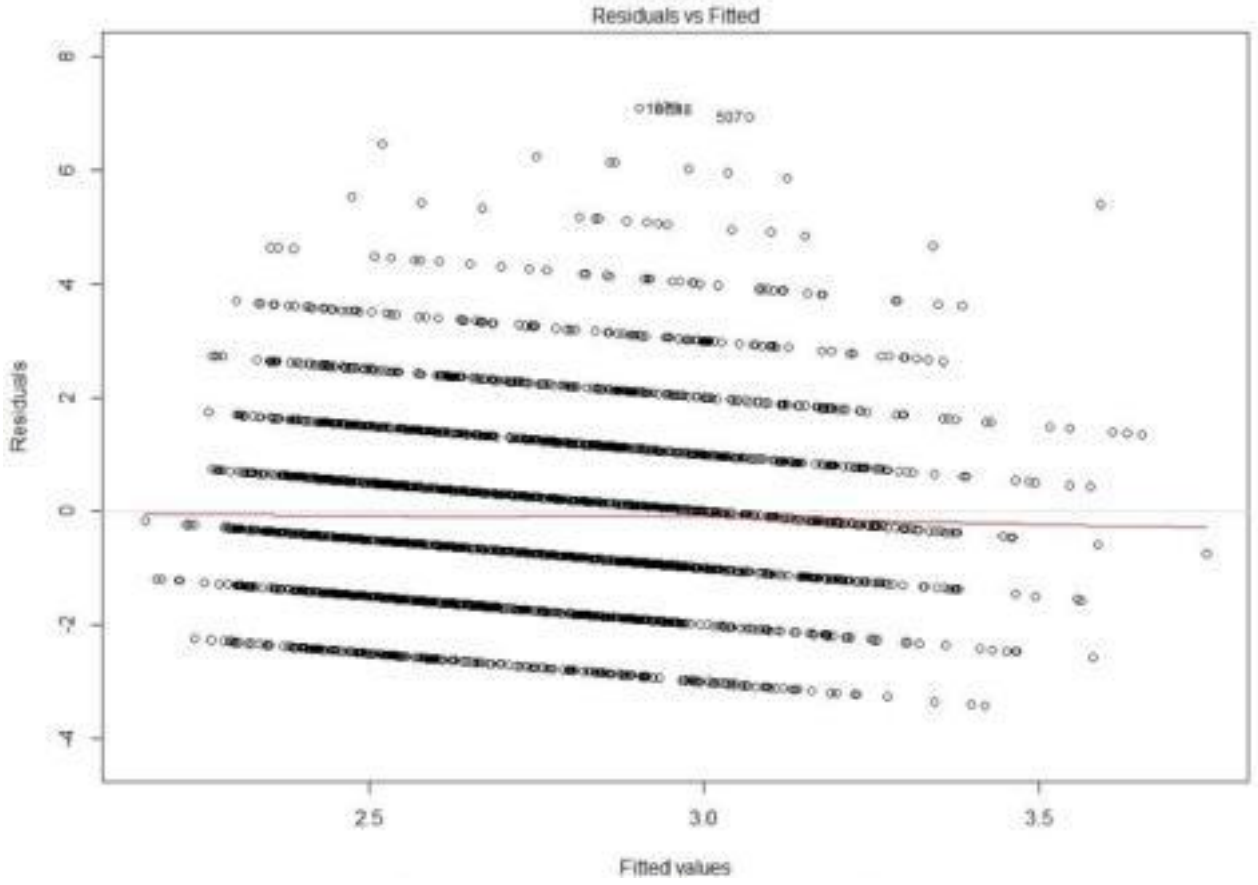

Figure 2: Residuals vs. Fitted Values for model.1 of Train Data

The Residuals vs Fitted Values plot is also supportive for model since the it lays around the zero line kind of symmetrically. Of course the relation is not perfect, however it does not violate linearity assumptions severely. There is not a distinctive pattern on residuals, so we can say that they are very close to a random.

IV. Part

The Mean Total is the most significant parameter in our stepwise regression applied model according to the p-value being the smallest. It is reasonable because we expect a team to score at home approximately close to their average number of home goals and we also expect the same for the away team since we expect that they score similarly to their average number of away goals. The BegOfSeason, MidOfSeason and EndOfSeason parameters are also significant since their p-values are obviously smaller than 0.05 and over parameter is also significant parameter. By predicting this way, we can estimate that total number of goals in a match close to sum of teams' average number of home goals and away goals and the season.

Table 2: The summary of model.1.

```
Call:
lm(formula = Total ~ -1 + EveningMatch + MorningMatch + over + MeanTotal +
    BegOfSeason + MidOfSeason + EndOfSeason, data = matches)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4270 -1.2972 -0.0961  1.1160  7.0757

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
EveningMatch -0.074883   0.080262  -0.933   0.3509
MorningMatch  0.110488   0.090740   1.218   0.2235
over         -0.012998   0.005711  -2.276   0.0229 *
MeanTotal     0.461537   0.060866   7.583 4.56e-14 ***
BegOfSeason   1.815018   0.240119   7.559 5.46e-11 ***
MidOfSeason   1.788782   0.232365   7.698 1.89e-11 ***
EndOfSeason   1.818811   0.235804   7.713 1.69e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.672 on 2825 degrees of freedom
Multiple R-squared:  0.7315,   Adjusted R-squared:  0.7309
F-statistic:  1100 on 7 and 2825 DF,  p-value: < 2.2e-16
```

According to the table, EveningMatch and MorningMatch is not significant because their p-values are larger than 0.05. Considering the Estimate column of EveningMatch we can say that total goal is negatively related with matches being played in the evening. Also, the over parameter is negatively related with the total goal. This is a logical result because as we stated in the beginning, as the odd for over oddtype increases the probability of that match ending over decreases thus decreasing the expected total goal in the match.

Table 3: Summary of stepwise regression applied model.1

```
Call:
lm(formula = Total ~ MorningMatch + over + MeanTotal + BegOfSeason
    + MidOfSeason + EndOfSeason - 1, data = matches)

Residuals:
   Min     1Q Median     3Q    Max
-3.418 -1.303 -0.098  1.124  7.097

Coefficients:
             Estimate Std. Error t value  Pr(>|t|)
MorningMatch  0.13147    0.08791   1.496    0.1349
over         -0.01307    0.00571  -2.290    0.0221 *
MeanTotal     0.45114    0.05983   7.540 6.31e-14 ***
BegOfSeason   1.82990    0.23958   7.638 3.00e-11 ***
MidOfSeason   1.79852    0.23212   7.748 1.29e-11 ***
EndOfSeason   1.82942    0.23553   7.767 1.11e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.672 on 2826 degrees of freedom
Multiple R-squared:  0.7315,   Adjusted R-squared:  0.7309
F-statistic:  1283 on 6 and 2826 DF,  p-value: < 2.2e-16
```

Considering this table, we can conclude that backward regression eliminated the EveningMatch parameter due to high p-value thus being insignificant.

The Adjusted R-squared is 0.7309 and this means that we can explain 73.09 % of the total variance in the data with our model.

V. Part

- **Residuals of Test Data vs Prediction**

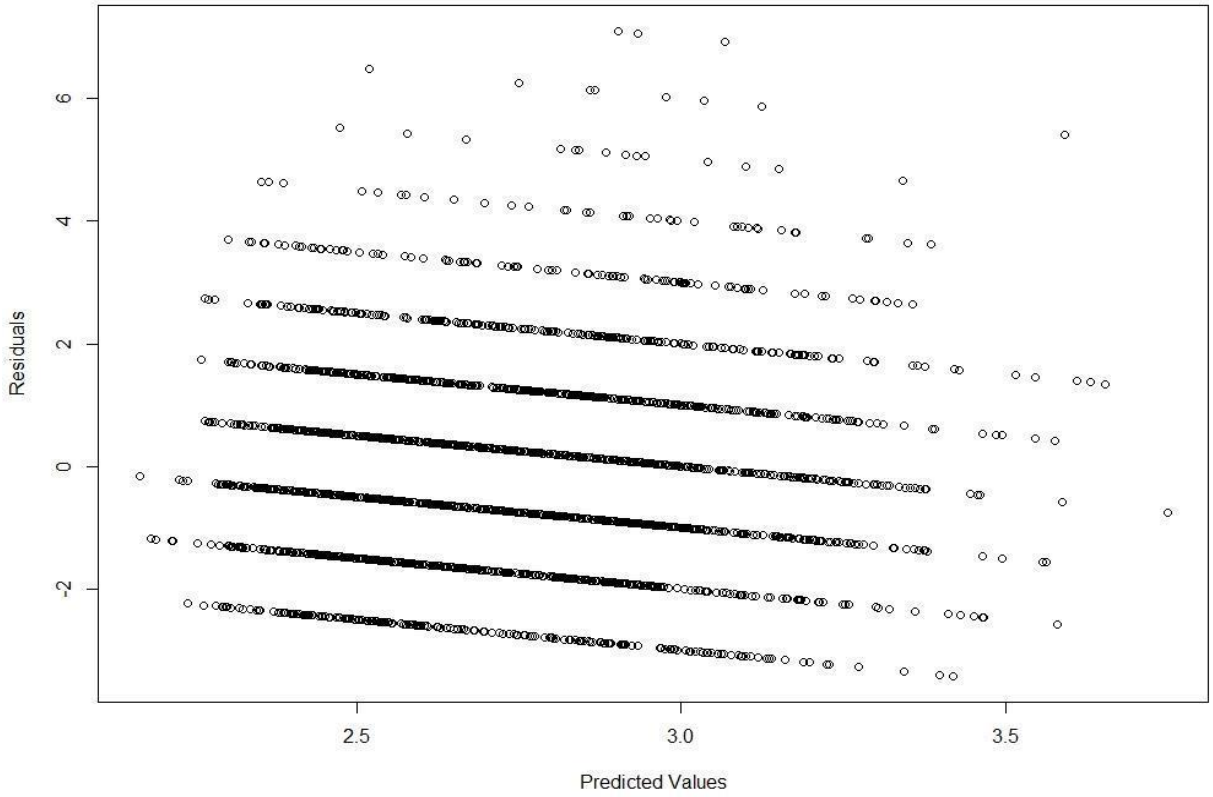

Figure 3: Residuals of Test Data vs Prediction

This plot represents the relation between the Test Data's residuals and the predicted values.In order to test the performance of our model on the test data, we evaluated the SSE as 763.7686 and Residual Standard Error as 1.672171. We calculated SSE by below codes.

Error_model_with_predicted = matches_2018_2019 – predicted

*SSE = sum(Error_model_with_predicted$Total**2)*

k=length(model.1_stepwise$coefficients)-1 #Subtract one to ignore intercept

*SS2E=sum(model.1_stepwise$residuals**2)*

SSE measures how far the data are from the model's predicted values. Since sample size is very large SSE seems large, however; for such a big data 763 is a tolerable SSE value.

VI. Part

We used data of matches between 2010-2018 in other words our train data for this part.

Based on our model obtained with train data, we decided a bet rule on predicted total goals. According to that bet rule, we bet on the predicted values of 3.0+ and 2.3- , doing so we are 73% successful with 3.0+ and 65% successful with 2.3- predicted values. There were regions that we were 100% successful however they were containing too little data and we ignored those regions. For instance, when we bet on 3.20+ we were successful 100% but there was only 1 match there.

We had assigned predicted total goal values from our model for matches between 2010-2018. We determined that if data which we assigned more than 3.0 it ends over, if data is less than 2.3 it ends under. Then we determined profit by subtracting -1 for every bet we play since we put money and we add income as latest odds values for the matches our prediction is true, otherwise we just subtract -1 for matches which our predictions are false. Then we calculated overall profit and found below results. We assigned our loss and profit for every match to y axis and matches according to match times to x axis. By doing this, we organized our matches based on chronology.

- **Profit-Loss graph for matches 2010-2018**

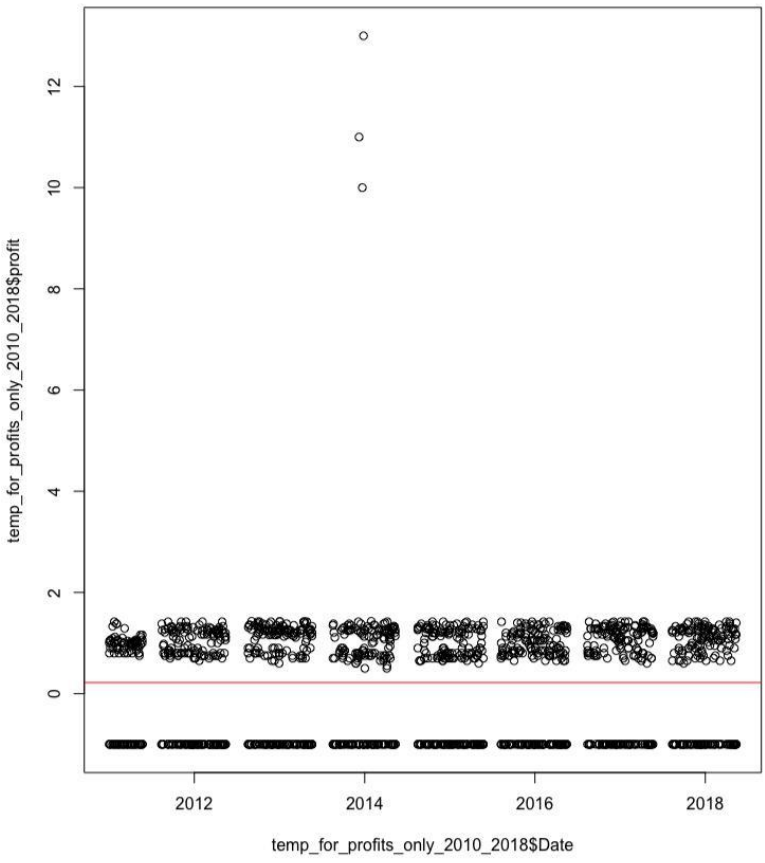

Figure4: Profit-Loss graph for matches 2010-2018

- **Average Profit Overtime graph for matches 2010-2018**

To calculate profit overtime, we analyzed average profit overtime cumulatively. We calculated total profit by using Cumsum function.

After we calculate total profit by cumsum function we divide it with number of matches we bet till that time and calculated average profit overtime.Then we analresults.

At first, even if there are some fluctuations in profit, as data grows average profit overtime viewed linearly.

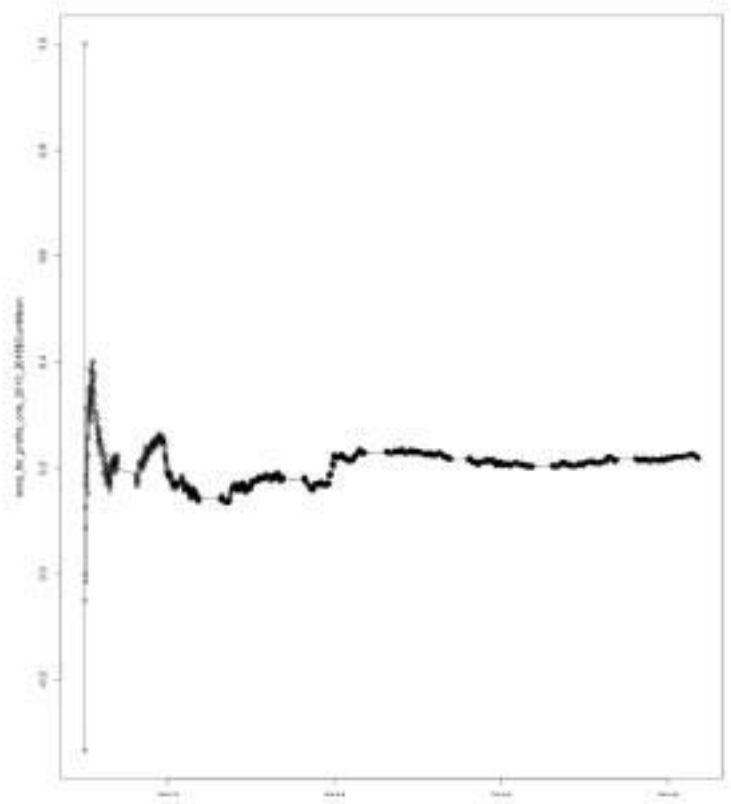

Figure5: Average Profit Overtime for matches 2010-2018

VII. Part

We calculated data for 2018-2019 for the same variables' columns and calculated MeanTotal. Then we assigned predicted total goal values to matches by using data from our model for matches between 2010-2018. If the value we assigned is more than 3.0, we bet for match ends over. If the value we assigned is less than 2.3, we bet for match ends under. Then we determined profit by subtracting -1 for every bet we play since we put money and we add income as latest odds values, and we calculated overall profit and found below results.

All in all, considering the number of matches we are successful on betting and the precision, the most efficient region is 3.0+ and 2.3- region. In that region we obtained a total profit of 105.6097 for 1 $ for every match we bet on.

- **Profit-Loss Graph for matches 2018-2019**

In the same way as we did in part 6, we assigned our loss and profit for every match to y axis and matches according to match times to x axis. By doing this, we organized our matches based on chronology.

We observed outlier odd values in our data.

Odd value of some matches is seemed 100 in data which is unrealistic. Unfortunately, it lowers our model's ability to measure consistency.

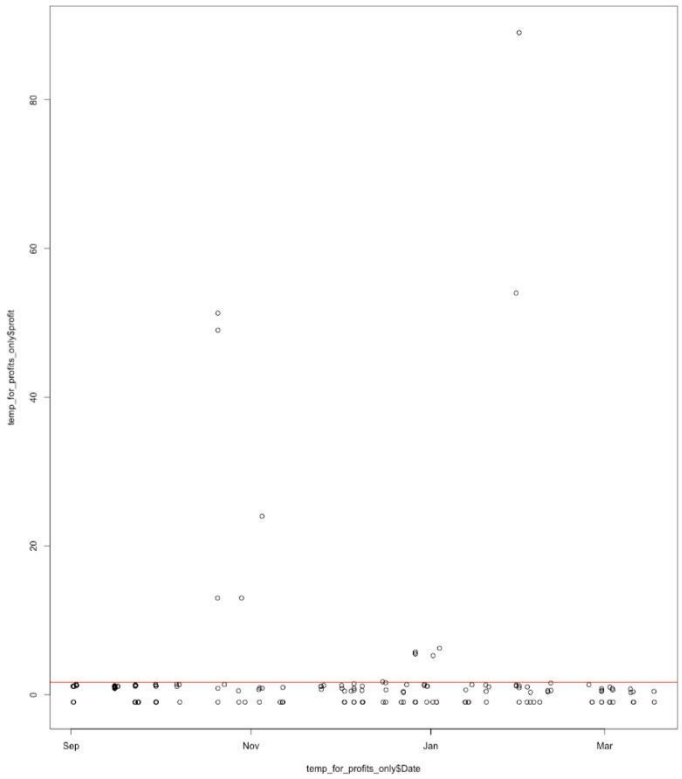

Figure6: Profit-Loss graph for matches 2018-2019

In the same way in part 6, we analyzed average profit overtime for season 2018-2019 cumulatively. We did this by dividing total profit with number of matches we bet till that time, and observed the change in average profit over time. We observed that average profit increased over time.

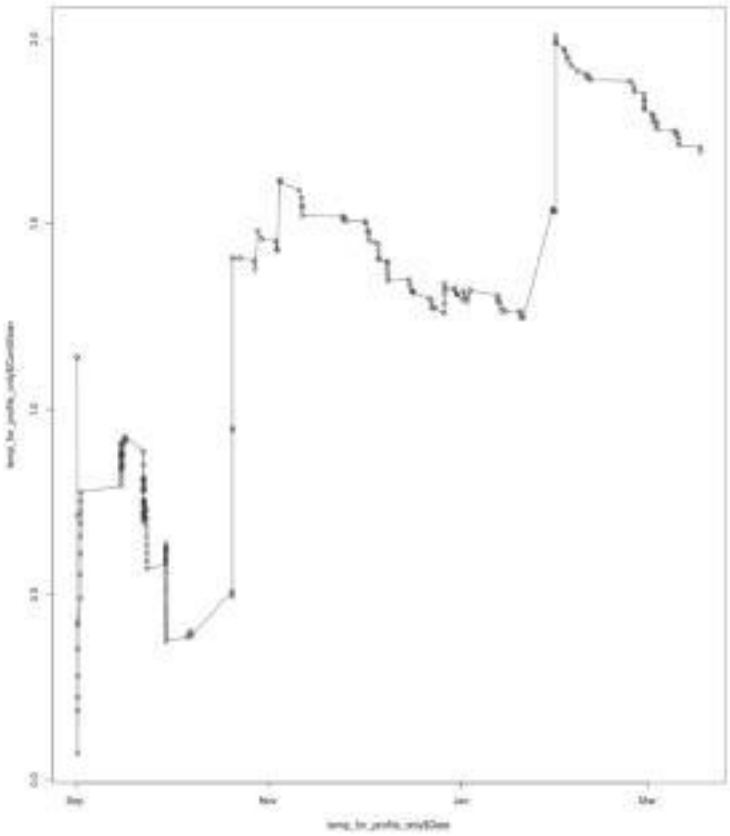

Figure7: Average Profit Overtime for season 2018-2019

R codes for Plotting Figures

i. Code for Figure 1

```
model.1_stepwise<-step(model.1,direction="backward")
model.1_stepwise
summary(model.1_stepwise)
plot(model.1_stepwise)
anova(model.1_stepwise)
```

ii. Code for Figure 2

```
model.1
summary(model.1)
plot(model.1)
anova(model.1)
```

```
model.1_stepwise<-step(model.1,direction="backward")

model.1_stepwise

summary(model.1_stepwise)

plot(model.1_stepwise)

anova(model.1_stepwise)
```

iii. Code for Figure 3

```
model.1

summary(model.1)

plot(model.1)

anova(model.1)

model.1_stepwise<-step(model.1,direction="backward")

model.1_stepwise

summary(model.1_stepwise)

plot(model.1_stepwise)

anova(model.1_stepwise)

predicted=predict(model.1_stepwise,matches_2018_2019)

summary(predicted)

plot(predicted, matches_2018_2019$Total)
```

iv. Code for Figure 4

```
###2010-2018 OU ODDS AND PROFIT

matches_for_2010_2018 <- copy(matches_for_ou_bets)

matches_for_2010_2018[, Predicted_Goals := predict(model.1_stepwise,
matches_for_2010_2018)]

matches_for_2010_2018[, Predicted_ou := ifelse(Predicted_Goals >=3.0, 1,
ifelse(Predicted_Goals <= 2.3, 0, 13))] #If our odds is over it is coded as 1, if it is under it
is coded as 0

matches_for_2010_2018[, real_ou:= ifelse(Total >=2.5, 1, 0)]
```

```
temp_2010_2018 <- copy(matches_for_2010_2018)

temp_overs_2010_2018 <- subset(temp_2010_2018, Predicted_ou==1)

total_exp_for_over_2010_2018 = nrow(temp_overs_2010_2018)

abcdef = subset(temp_overs_2010_2018, real_ou == 1)

total_income_for_over_2010_2018 = sum(abcdef$over)

total_profit_for_over_2010_2018 = total_income_for_over_2010_2018
- total_exp_for_over_2010_2018

#for under bets

temp_another <- copy(matches_for_2010_2018)

temp_another[, under:= 1/(1-1/over)]

temp_unders_2010_2018 <- subset(temp_another, Predicted_ou == 0)

total_exp_for_under_2010_2018 = nrow(temp_unders_2010_2018)

klmn = subset(temp_unders_2010_2018, real_ou == 0)


total_income_for_under_2010_2018 = sum(klmn$under)

total_profit_for_under_2010_2018 = total_income_for_under_2010_2018
- total_exp_for_under_2010_2018

##TOTALPROFIT

Total_profit_2010_2018 = total_profit_for_over_2010_2018 +
total_profit_for_under_2010_2018
```

v. Code for Figure 5

```
temp_2010_2018_predicted_ou <- copy(matches_for_2010_2018)

temp_2010_2018_predicted_ou = merge(temp_2010_2018_predicted_ou,
temp_matches_2010_2018, by = "matchId")

temp_2010_2018_predicted_ou[, profit:=ifelse(Predicted_ou==real_ou, over.x-1,

 ifelse(Predicted_ou==13, 0, -1))]

temp_for_profits_only_2010_2018 <- copy(temp_2010_2018_predicted_ou)

temp_for_profits_only_2010_2018 <- subset(temp_for_profits_only_2010_2018, profit!=0)

plot(temp_for_profits_only_2010_2018$Date, temp_for_profits_only_2010_2018$profit)

mean_profit_2010_2018 = mean(temp_for_profits_only_2010_2018$profit)
```

```
abline(h=mean_profit_2010_2018, col="red")

temp_for_profits_only_2010_2018 <- temp_for_profits_only_2010_2018[order(date),]

temp_for_profits_only_2010_2018[, CumMean:= cumsum(profit) / seq_along(profit)]

plot(temp_for_profits_only_2010_2018$Date,
temp_for_profits_only_2010_2018$CumMean,

    xlim=c(min(temp_for_profits_only_2010_2018$Date),
max(temp_for_profits_only_2010_2018$Date)), type = "o")
```

vi. Code for Figure 6

```
##SEASON 2018-2019

##OVER UNDER ODDS FOR SEASON 2018-2019 WITH RESPECT TO PREDCITIONS

matches_2018_2019_for_ou_bets[, Predicted_total_goals := predict(model.1_stepwise,
matches_2018_2019_for_ou_bets)]

matches_2018_2019_for_ou_bets[, Predicted_ou_odds := ifelse(Predicted_total_goals >=3.0,
1, ifelse(Predicted_total_goals <= 2.3, 0, 13))] #If our odds is over it is coded as 1, if it is
under it is coded as 0

matches_2018_2019_for_ou_bets[, real_ou:= ifelse(Total >=2.5, 1, 0)]

##CALCULATION OF PROFIT

#for over bets

temp123 <- copy(matches_2018_2019_for_ou_bets)

temp_overs <- subset(temp123, Predicted_ou_odds==1)

total_exp_for_over = nrow(temp_overs)

abc = subset(temp_overs, real_ou == 1)

total_income_for_over = sum(abc$over)

total_profit_for_over = total_income_for_over - total_exp_for_over

#for under bets

temp456 <- copy(matches_2018_2019_for_ou_bets)
```

```
temp456[, under:= 1/(1-1/over)]

temp_unders <- subset(temp456, Predicted_ou_odds == 0)

total_exp_for_under = nrow(temp_unders)

def = subset(temp_unders, real_ou == 0)

total_income_for_under = sum(def$under)

total_profit_for_under = total_income_for_under - total_exp_for_under


Total_profit = total_profit_for_over + total_profit_for_under
```

vii. Code for Figure 7

```
temp_2018_2019_predicted_ou <- copy(matches_2018_2019_for_ou_bets)

temp_2018_2019_predicted_ou = merge(temp_2018_2019_predicted_ou, temp_match_list, by = "matchId")

temp_2018_2019_predicted_ou[, profit:=ifelse(Predicted_ou_odds==real_ou, over.x-1, ifelse(Predicted_ou_odds==13, 0, -1))]

temp_for_profits_only <- copy(temp_2018_2019_predicted_ou)

temp_for_profits_only <- subset(temp_for_profits_only, profit!=0)

plot(temp_for_profits_only$Date, temp_for_profits_only$profit)

mean_profit = mean(temp_for_profits_only$profit)

abline(h=mean_profit, col="red")

temp_for_profits_only <- temp_for_profits_only[order(date),]

temp_for_profits_only[, CumMean:= cumsum(profit) / seq_along(profit)]

plot(temp_for_profits_only$Date, temp_for_profits_only$CumMean,

    xlim=c(min(temp_for_profits_only$Date), max(temp_for_profits_only$Date)), type = "o")
```

viii. Code for Figure 8

```
temp_2018_2019_predicted_ou <- copy(matches_2018_2019_for_ou_bets)
```

```
temp_2018_2019_predicted_ou = merge(temp_2018_2019_predicted_ou, temp_match_list, by = "matchId")

temp_2018_2019_predicted_ou[, profit:=ifelse(Predicted_ou_odds==real_ou, over.x-1, ifelse(Predicted_ou_odds==13, 0, -1))]

temp_for_profits_only <- copy(temp_2018_2019_predicted_ou)

temp_for_profits_only <- subset(temp_for_profits_only, profit!=0)

plot(temp_for_profits_only$Date, temp_for_profits_only$profit)

mean_profit = mean(temp_for_profits_only$profit)

abline(h=mean_profit, col="red")

temp_for_profits_only <- temp_for_profits_only[order(date),]

temp_for_profits_only[, CumMean:= cumsum(profit) / seq_along(profit)]

plot(temp_for_profits_only$Date, temp_for_profits_only$CumMean,

xlim=c(min(temp_for_profits_only$Date), max(temp_for_profits_only$Date)), type = "o")
```