352-Project-Report-Anthony-Brennan

Anthony Brennan

12/16/2019

Contents

1 Title

Predicting Soccer Game Outcomes Using FIFA Player Ratings

2 Author

Anthony Brennan

Undergraduate student at Case Western Reserve University

3 License

4 Abstract

Sports analytics has boomed in recent years with the explosion in available data to data. One of the biggest analytic interests is predicting the results of games. Soccer, the world's most popular sport, is full of data for this kind of analysis. This report will look into predicting various game outcomes for European soccer games using player rating data from EA Sport's FIFA video game series. The expected goal differential and total goals scored will be modeled using multiple linear regression and support vector machines. Win probabilities will be predicted using t-stat and poisson distributions. Finally, projected final standings will be created using these models. The goal for this report is to use these modelling methods to create reliable models for creating these outcomes. All of these game outcome predicitons are important to all kinds of people from front office executives to coaches to sports bettors. Given how much money is tied up into soccer, it is important for everyone involved to have the most accurate analytic information.

5 Question

Can I predict an expected goals scored, goal differential, and win probabilities for various European soccer matches using player data from FIFA video games?

6 Introduction

When I was thinking about ideas for my project I wanted to do it on sports data. There has been a large boom in the use of data analytics in the sports industry, which is what got me interested in data science in the first place. After finding some data sets on Kaggle, I decided to focus on predicting soccer match outcomes using soccer player data. I felt using soccer would be best (compared to baseball, football, etc) because there are many different major leagues, whereas most of the other sports essentially only have one major league with extensive data. Since soccer has so many leagues and teams, there is far more data and a much larger sample size.

For player data, I will be using player ratings from EA Sport's FIFA video games. While the ratings are somewhat subjective, they do a pretty decent job at assessing players talent level. They are also a simpler, more standardized way to compare talent on different teams, rather than having to create my own subjective rating system with the data I have available to me. I found a data set on Kaggle with match results from 2000-2019 across 30+ European leagues. I also found data sets with player ratings for the FIFA 17, FIFA 18, and FIFA 19 on Kaggle. These data sets represent the preseason ratings for the 2016-17, 2017-18, and 2018-19 seasons, respectively. There are no mid-season adjustments in the data sets (EA will sometimes update ratings throughout the season).

In the project I matched up the player rankings from the video games to their respective seasons to avoid adding bias to the models. With each new season, players have their ratings adjusted and some players will change teams or retire. So it is important to make sure that the team rosters I use have the same real life rosters. So all games during the 2016-2017 season will have player data from the FIFA 17 video game, 2017-2018 will be from Fifa 18, and 2018-2019 will be from Fifa 19. As a result I will only be looking at games from these three seasons and will not use match data from any other season.

The ultimate goal of this project is to look at predicting a few different types of outcomes. The first is to come up with a prediction for how many goals each team would score and the goal differential for every game. Goal differential is the difference between the amount of goals the home team scores minus the amount of goals the away team scores. The next thing I want to predict is the probabilities of each team winning plus the probability that the game ends in a draw. Finally I want to come up with a prediction for the final standings for each league. These predictions can be important for many reasons. It could help a club's front office determine how much extra value a new player would bring to a club, and whether its worth it to sign him. Win probabilities can help managers make strategic decisions on what games to rest players. Goal differential and win probabilities are also important in the European sports betting industry as many bets revolve around these metrics.

7 Data Science Methods

I had to go through a few steps to take the raw data from Kaggle and format it into a format I could use for the project. All of the data cleaning and formating is done in the Data_Cleaning.rmd file. All 3 fifa video game data sets (“./Datasets/FullData_2017.csv”, “.Datasets/fifa-18-demo-player-dataset/CompleteDataset.csv”, “./Datasets/data_2019.csv”) were in csv format, so they were loaded into R easily using read.csv. The data set with all of the match data, however, was in a SQLite file (“./Datasets/database.sqlite”). In order to load this file, I used the RSQLite package to pull the files from the database and then save the file I needed, which was the second data set.

Since the soccer leagues in the Fifa games and match data are not the same, the next step was to filter for all the leagues that exist in all 4 data sets. This came out to a total of 17 european leagues in all of the data sets. These include: the top 4 tiers in England (Premier League, Championship, League 1, League 2); the top 2 tiers in Germany (Bundesliga 1, Bundesliga 2), France (Le Championnat, Division 2), Italy (Serie A, Serie B), and Spain (La Liga Primera Division, La Liga Segunda Division); and the top tier in Belgium (Jupiler League), Netherlands (Eredivisie), Portugal (Liga 1), Scotland (Premier League), and Turkey (Futbol Ligi 1). Using the dplyr package I removed all of the leagues that were not included in this list. I also created a conversion array in order to convert all of the special characters from other languages into simple English letters.

The next set of steps were applied to the Fifa 17, Fifa 18, and Fifa 19 data sets individually. I did these seperately as I am applying the Fifa data to its corresponding season to avoid bias. So Fifa 17 player data is applied to games during the 2016-2017 season, Fifa 18 to 2017-2018, and Fifa 19 to 2018-2019. Many of the teams in all of the data sets had slight variations in the names of the teams. For example, the name for one Portuguese team was listed as SP Gijon in matches, Sporting Gijon in Fifa 17, and Real Sporting de Gijon in Fifa 18 & 19. So I pulled all of the unique team names from both Fifa and matches, and manually matched up all of the team names in excel so that I could make sure all of the team names were correctly converted and reduce the amount of code it would’ve taken to convert 200+ teams. I then reloaded this new conversion data set converted all of the Fifa team names so that they were the same as the team names in the matches data. Finally, I removed all of the teams that weren’t from the leagues I am analyzing.

After the team names were converted I needed to create a sample lineup of players for each team for each year. For each team I picked the best goalie (GK), the 2 best forwards (F1, F2), 4 best midfielders (M1, M2, M3, M4), 4 best defenders (D1, D2, D3, D4), and the best 7 remaining players for the bench (B1-B7). An example of these lineups can be seen below.

```r
str(all[,c(2:21)])
```

```
## 'data.frame':    999 obs. of  20 variables:
##  $ Club: Factor w/ 380 levels "Aberdeen","Accrington",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Div : Factor w/ 17 levels "B1","D1","D2",..: 14 7 17 6 9 9 12 17 17 15 ...
##  $ GK  : int  69 59 73 63 66 74 76 69 67 77 ...
##  $ F1  : int  70 64 74 67 73 71 76 71 78 76 ...
##  $ F2  : int  62 61 70 65 64 68 76 70 74 76 ...
##  $ M1  : int  72 64 73 65 71 69 80 73 74 77 ...
##  $ M2  : int  71 63 69 65 69 69 80 73 72 77 ...
##  $ M3  : int  71 62 68 64 66 68 78 72 71 76 ...
##  $ M4  : int  70 61 67 64 66 68 74 71 70 76 ...
##  $ D1  : int  70 62 73 65 69 71 78 78 73 77 ...
##  $ D2  : int  68 62 70 65 66 69 77 72 72 76 ...
##  $ D3  : int  66 62 70 64 66 67 76 69 70 76 ...
##  $ D4  : int  66 62 69 64 65 66 75 69 69 75 ...
##  $ B1  : int  70 60 68 64 65 71 75 70 69 75 ...
##  $ B2  : int  67 60 67 64 65 68 72 70 68 74 ...
##  $ B3  : int  66 60 67 63 64 67 72 70 68 74 ...
##  $ B4  : int  66 59 67 63 63 66 72 70 67 74 ...
##  $ B5  : int  64 59 67 63 62 66 72 68 67 74 ...
##  $ B6  : int  61 58 66 62 62 66 72 68 67 73 ...
##  $ B7  : int  56 56 66 60 61 65 71 68 66 72 ...
```

In order to determine which players go in which position I looked at the preferred positions in each of the Fifa data sets. For the forward position, I used players whose preferred position is Striker (ST), Left Striker (LS), Right Striker (RS), Center Forward (CF), Left Forward (LF), Right Forward (RF), Left Wing (LW), Right Wing (RW), Center Attacking Midfield (CAM), Left Attacking Midfield (LAM), or Right Attacking Midfield (RAM). For the midfield position, I used players whose preferred position is Center Attacking Midfield (CAM), Left Attacking Midfield (LAM), Right Attacking Midfield (RAM), Center Midfield (CM), Left Midfield (LM), Right Midfield (RM), Left Center Midfield (LCM), Right Center Midfield (RCM), Center Defending Midfield (CDM), Left Defending Midfield (LDM), or Right Defending Midfield (RDM). For the defending position, I used players whose preferred position is Center Back (CB), Left Back (LB), Right Back (RB), Left Center Back (LCB), Right Center Back (RCB), Left Wing Back (LWB), or Right Wing Back (RWB).

I ran a for loop through each of teams in order to pull all of the players on that team. From there I ordered each player on the team in descending order based on their ranking. From there, I went through each player and put them in the unfilled position that matched their preferred position. For example, in Fifa 18 Lionel Messi is the best player on FC Barcelona with a rating of 93. Since his preferred position is RW he will be inserted at F1, as seen below. Luis Suarez, the second best player with a rating of 92, will be placed next. Since his preferred position is ST, he will be placed in F2 since F1 has already been filled. This process will continue for all players on the team. If a player's preferred position is filled on the lineup, then the loop will look at their secondary position, if listed. If positions are filled for that as well, then they will be placed on the next available bench position. For example, Rafinha is a player on Barcelona with a rating of 81. Since preferred position is RW but both F1 and F2 are full. His secondary position is RM but M1-M4 are full as well so he is placed at B6, the first open bench position. An example of the 2018 Barcelona team can be seen below

```
##   year      Club GK F1 F2 M1 M2 M3 M4 D1 D2 D3 D4 B1 B2 B3 B4 B5 B6 B7
## 1 2018 Barcelona 85 93 92 87 87 86 83 87 85 83 83 82 82 82 81 81 81 81
```

After the lineups were set, I created variables for the aggregate overall and position ratings. For the overall rating I took the average of all 11 starting positions. The position aggregate rankings were calculated as follows: goalkeeper - goalie rating, attacking = average of F1 and F2, midfield = average of M1-M4, defense = average of D1-D4, and bench = average of B1-B7. These are the variables I will be using later in creating models. This can be seen below.

```
##   year      Club  Overall Goalkeeper Attacking Midfield Defense    Bench
## 1 2018 Barcelona 86.45455         85      92.5    85.75    84.5 81.42857
```

Once these variables were created, the final bit of variables created were summary statistics on various things about the teams. For some of the visualizations I will explain later in the report I needed to create average rankings for each league and then average team ratings - the average league ratings. In 2018, Barcelona played in La Liga Primera Division, which had an average overall rating of 79.27 (LGmeanOvr). Barcelona had a rating of 86.45, so its overall rating minus the league overall is 7.18 (OvrvsMean), meaning it has a rating 7.18 higher than the league average. I did this only for overall, attacking, and defensive ratings.

```
##   year      Club Div LGmeanOvr LGmeanAtt LGmeanDef OvrvsMean AttvsMean
## 1 2018 Barcelona SP1  79.27273      80.2    78.525  7.181818      12.3
##   DefvsMean
## 1     5.975
```

I then went into the games data set and used dplyr to find all of the games each team played by year. I then took the average goals the scored and the average goals they conceded in each game and then the difference between the 2. Like with the ratings I calculated the league average goals for, goals against, and goal differential and average team - the average league.

```
##   year      Club Div       GF        GA     Diff LGmeanGoals GFvsMean  GAvsMean
## 1 2018 Barcelona SP1 2.605263 0.7631579 1.842105    1.347368 1.257895 0.5842105
```

The final step for data wrangling was to pair up the rating variables with the games data. This was done by merging the club in the all data set with the home team in the games set and creating the variables HomeOverall, HomeAttacking, etc. Then doing the same with the away team.

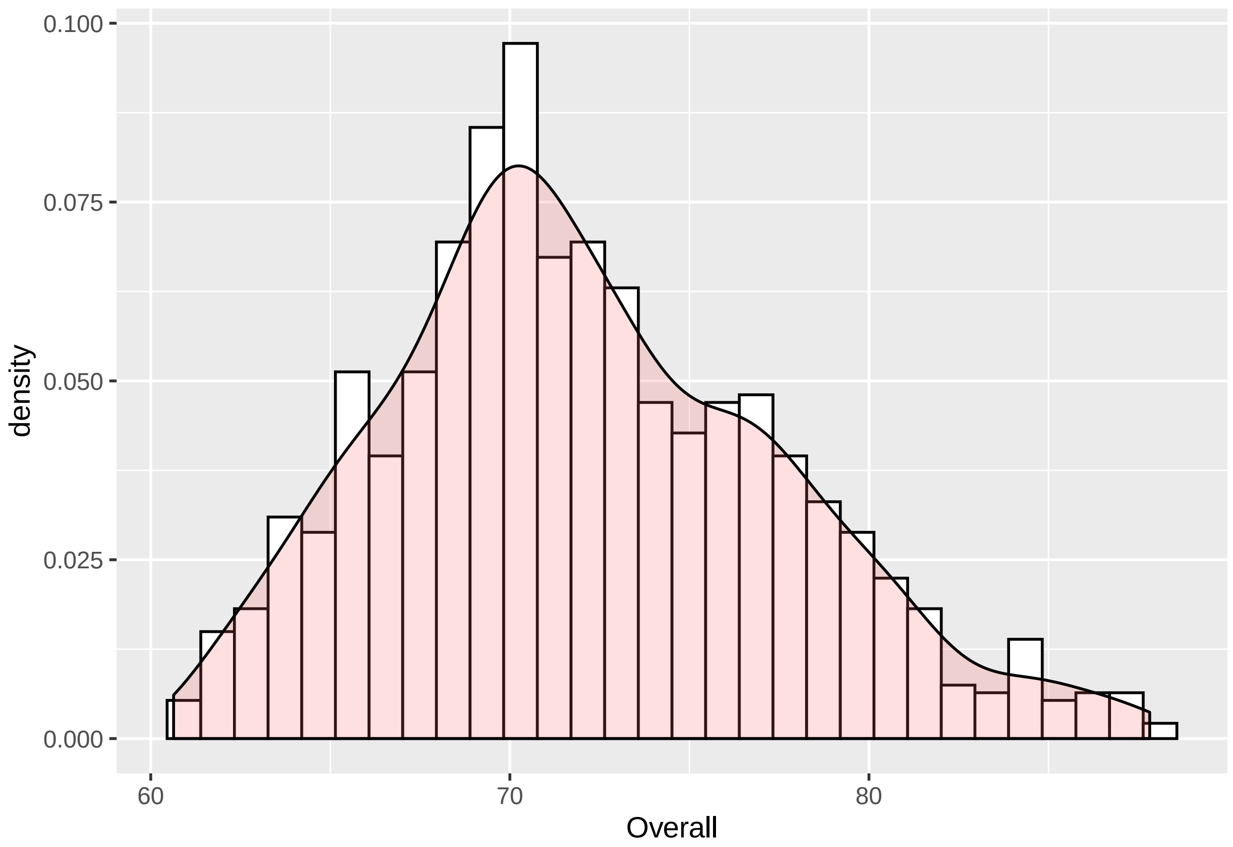
8 Exploratory Data Analysis

There were a few things I wanted to look at before I started modeling my data. The first was to look at some summary statistics for some of the main variables I would be using. I also included a ggplot histogram of the overall ratings as well to look at the shape of the data.

```
##        variable   min first_quantile median  mean third_quantile   max   sd
## 1       Overall 60.64          68.36  71.36 72.07          75.82 87.82 5.48
## 2    Goalkeeper 53.00          68.00  72.00 72.33          77.00 92.00 6.67
## 3     Attacking 58.50          68.50  72.00 72.41          76.00 92.50 5.96
## 4      Midfield 60.00          68.25  71.75 72.28          76.00 87.75 5.49
## 5       Defense 58.25          68.00  71.25 71.64          75.25 87.75 5.29
## 6     Goals For  0.38           1.05   1.26  1.33           1.50  3.50 0.41
## 7 Goals Against  0.53           1.12   1.32  1.33           1.55  2.53 0.33
## 8     Goal Diff -1.94          -0.39  -0.09  0.00           0.33  2.56 0.61
```

```
ggplot(all, aes(Overall)) +
  geom_histogram(aes(y=..density..), colour="black", fill="white")+
  geom_density(alpha=.2, fill="#FF6666")
```

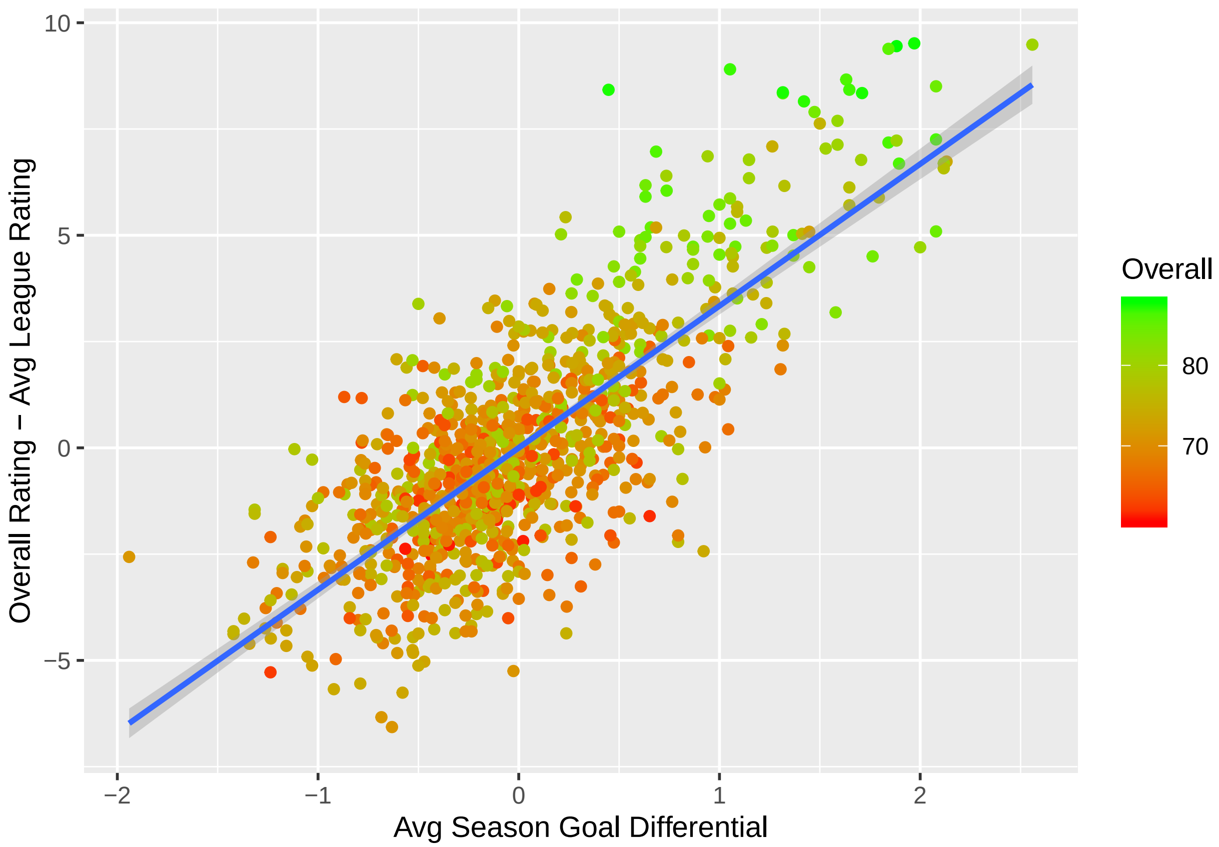
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Based on the histogram, the overall ratings for all of the teams is a mostly normal bell shaped distribution. There is a little bit of a skew to the right, this is mostly attributable to some of the European super clubs like FC Barcelona, Bayern Munich, Liverpool, etc, who are much better than many of the other teams.

I also wanted to look at the trends in the data. After adjusting ratings for the average league rating, the first plot is the adjusted overall rating vs average goal differential.

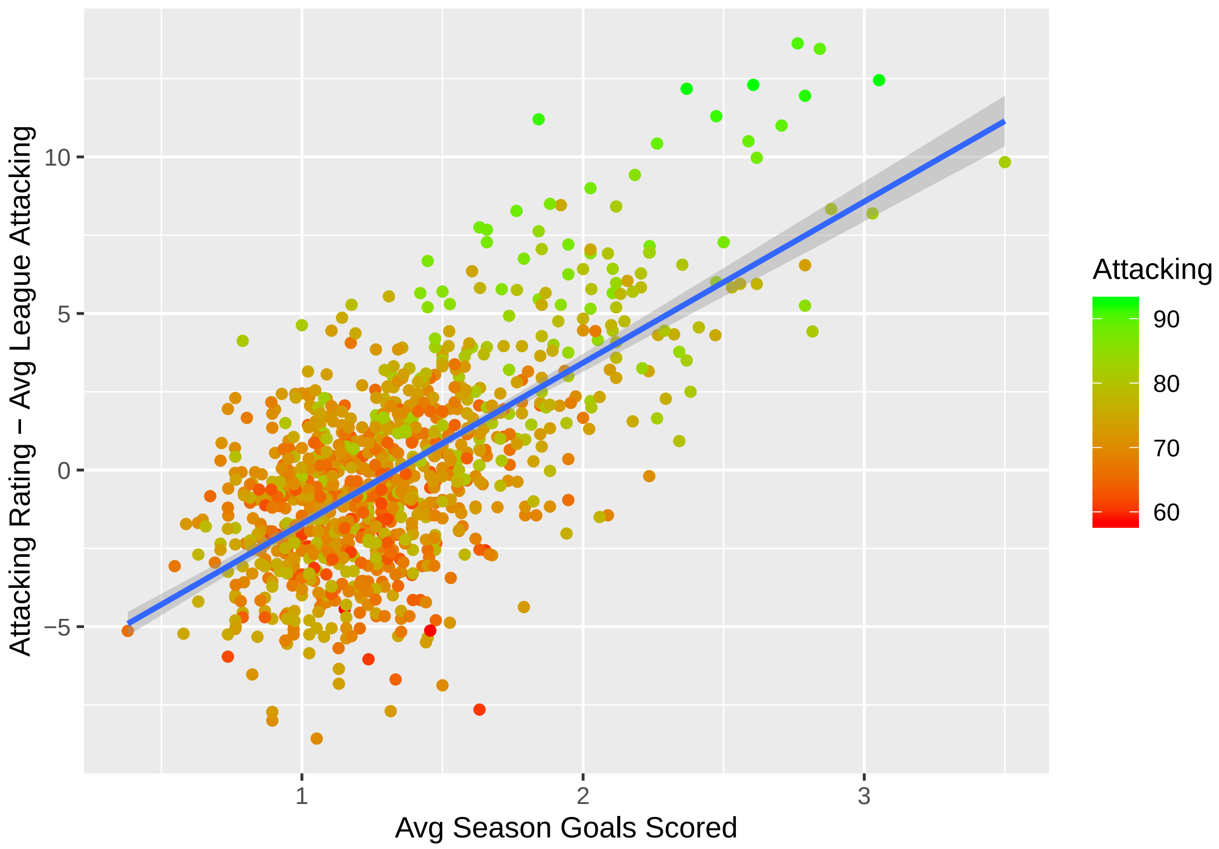
```
ggplot(all, aes(Diff, OvrvsMean, color = Overall)) +
  geom_point() +
  scale_color_gradient(low = "red", high = "green") +
  xlab("Avg Season Goal Differential") +
  ylab("Overall Rating - Avg League Rating") +
  geom_smooth(method = "lm")
```



There is a clear upward trend in the plot. This follows the logic that the better the team is compared to the league the more it can expect to win by.

The next plot is adjusted attacking vs average goals scored.

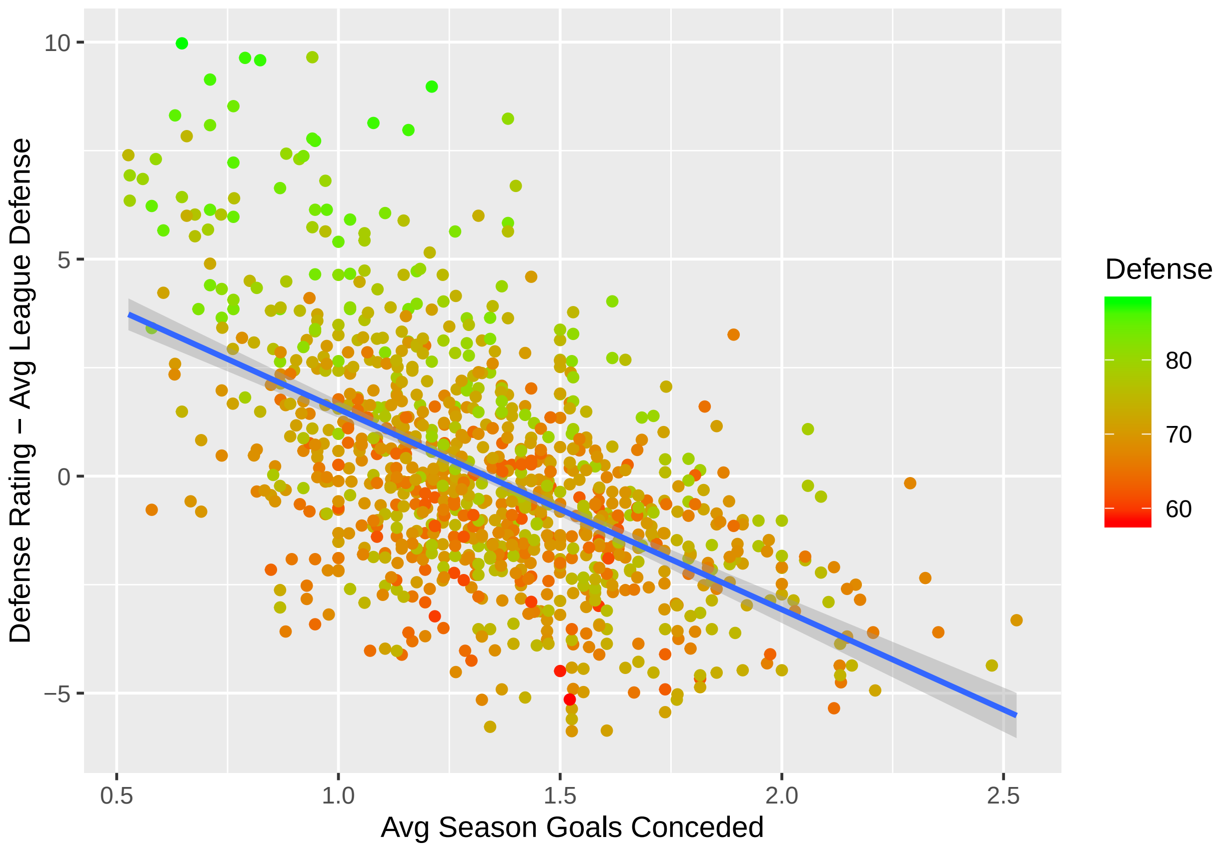
```
ggplot(all, aes(GF, AttvsMean, color = Attacking)) +
  geom_point() +
  scale_color_gradient(low = "red", high = "green") +
  xlab("Avg Season Goals Scored") +
  ylab("Attacking Rating - Avg League Attacking") +
  geom_smooth(method = "lm")
```



This upward trend follows the logic that the better the forwards are, the more goals the team will score.

The last plot is adjusted defense vs average goals conceded

```
ggplot(all, aes(GA, DefvsMean, color = Defense)) +
  geom_point() +
  scale_color_gradient(low = "red", high = "green") +
  xlab("Avg Season Goals Conceded") +
  ylab("Defense Rating - Avg League Defense") +
  geom_smooth(method = "lm")
```



This downward trend follows the logic that the better the defenders are, the less goals the team will concede.

All of this shows that the data meets some common statistical assumptions. It is mostly normal distribution, has a large sample size of 19,314, and the data is linear.

9 Statistical Learning: Modeling & Prediction

9.1 Goal Differential Prediction

All of the data for the statistical models came from the game data set. The data was split into training and testing data using a 25/75 sample split with a set.seed value so that the smaple was reproduceable

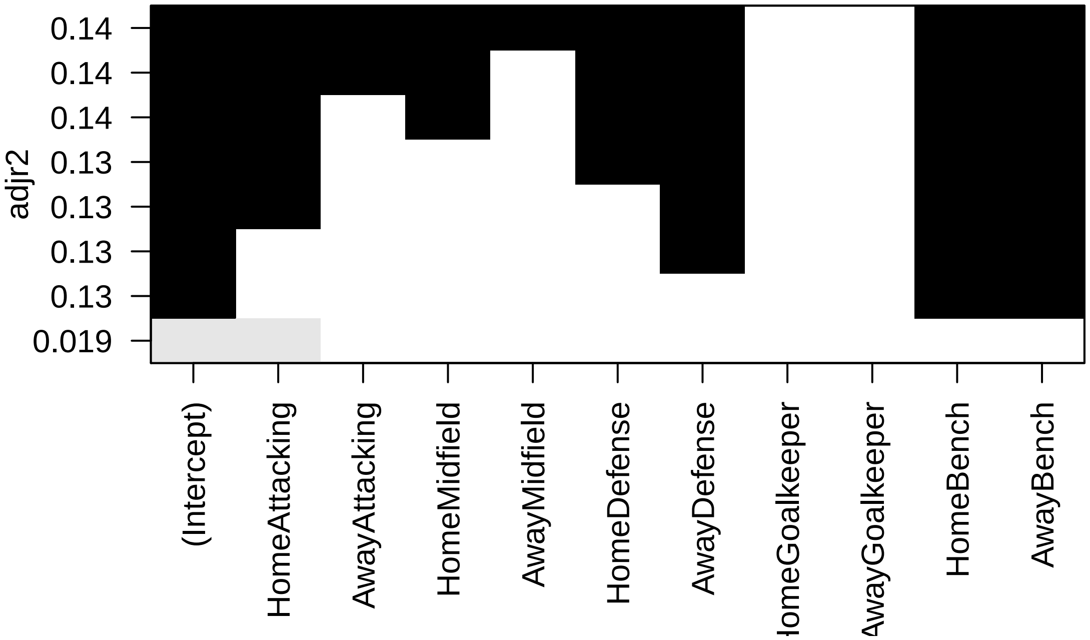
```
set.seed(100)
train.sample <- sample(1:nrow(games), (nrow(games)/4), replace=FALSE)
training <- games[train.sample,]
testing <- games[-train.sample,]
```

Since I have known outputs, goal differential and home and away goals score, I will be using supervised machine learning algorithms. I chose to use multiple linear regression and support vector machine models since I have real numerical values, so there is no classification involved.

9.1.1 Regression Model

The first thing to be predicted is the goal differential in the games, starting with the regression model. I ran a variable selection analysis using the regsubsets function from the leaps package, so that I could determine which variables to use.

```
w0 <- regsubsets(GoalDiff ~ HomeAttacking + AwayAttacking + HomeMidfield + AwayMidfield + HomeDefense +
# + HomeRest + AwayRest + RestDiff
```



Based on the plot above, most of the best selection of models includes all of the variables except for the Home and Away Goalkeepers. So I used all of these variables to run the regression model.

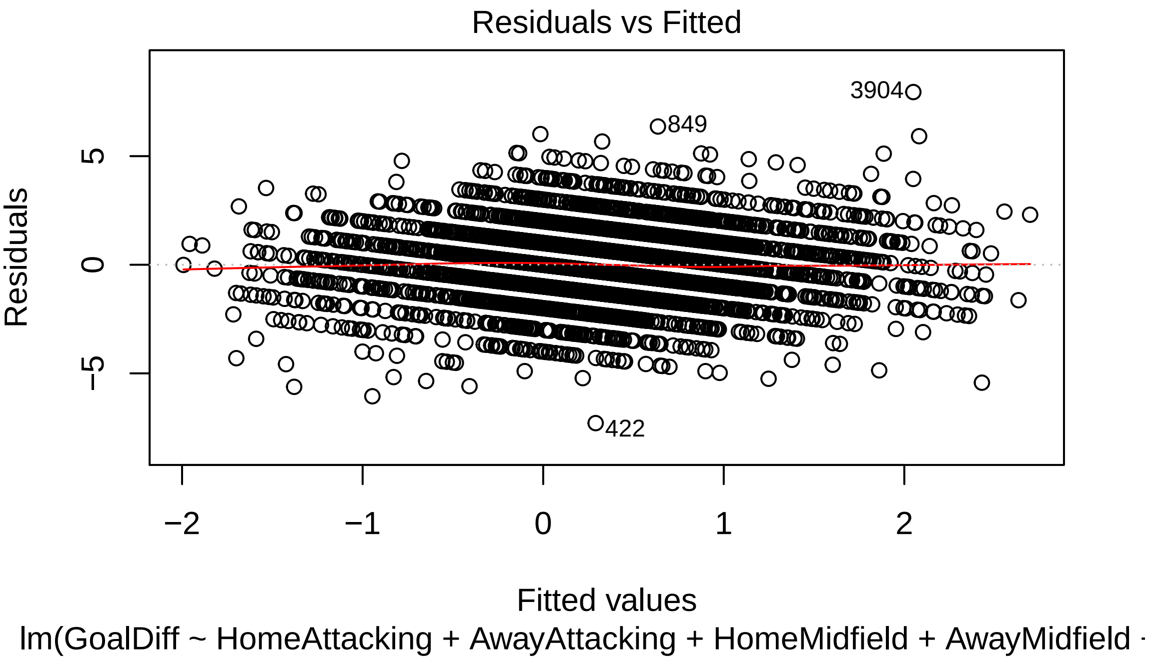
```
gf1 <- lm(GoalDiff ~ HomeAttacking + AwayAttacking + HomeMidfield + AwayMidfield + HomeDefense + AwayDe
```

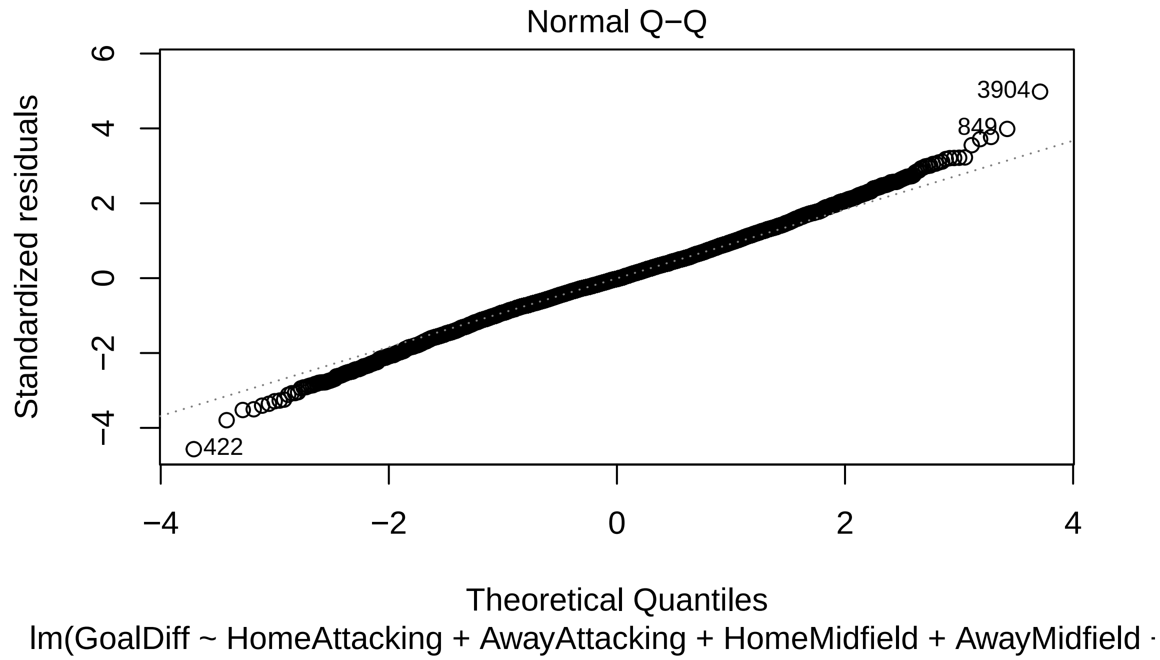
```
##
## Call:
## lm(formula = GoalDiff ~ HomeAttacking + AwayAttacking + HomeMidfield +
##     AwayMidfield + HomeDefense + AwayDefense + HomeBench + AwayBench,
##     data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.2905 -0.9961 -0.0334  0.9830  7.9501
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.05391    0.34228   0.158 0.874848
## HomeAttacking   0.03987    0.01333   2.991 0.002790 **
## AwayAttacking -0.02331    0.01341  -1.738 0.082224 .
## HomeMidfield    0.04280    0.01886   2.270 0.023272 *
## AwayMidfield  -0.02224    0.01945  -1.143 0.252966
## HomeDefense     0.04579    0.01864   2.457 0.014058 *
## AwayDefense   -0.06303    0.01876  -3.360 0.000784 ***
## HomeBench      0.04477    0.02331   1.921 0.054851 .
## AwayBench     -0.06185    0.02361  -2.620 0.008819 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.598 on 4819 degrees of freedom
## Multiple R-squared:  0.1372, Adjusted R-squared:  0.1357
## F-statistic: 95.75 on 8 and 4819 DF,  p-value: < 2.2e-16
```

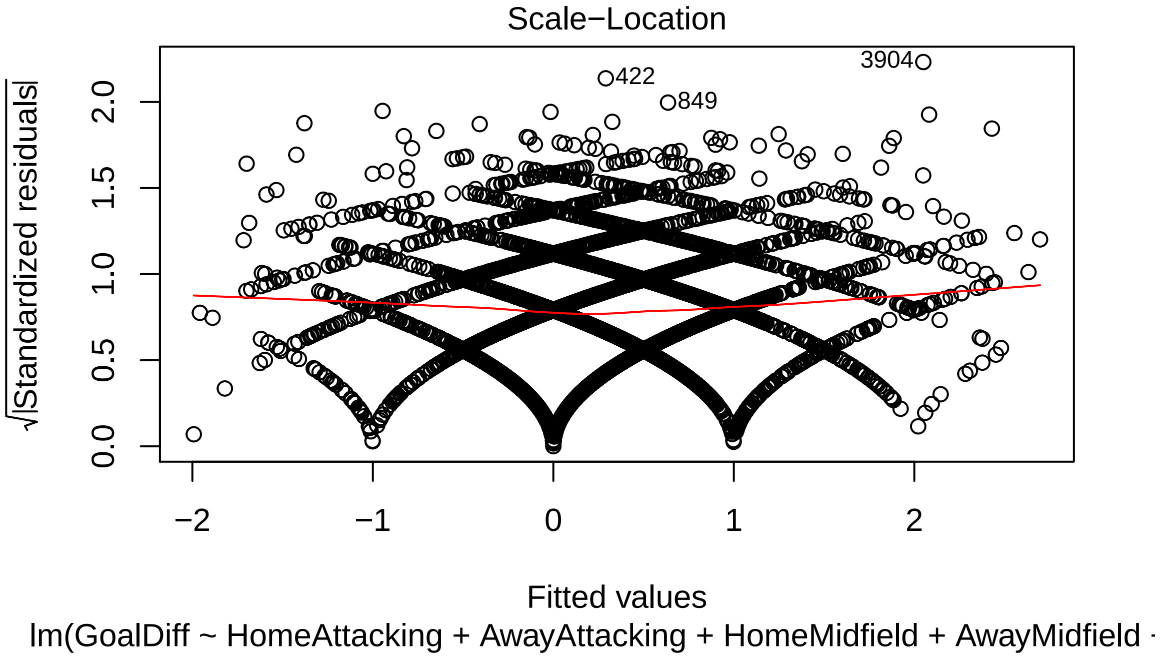
```
## [1] "Residual Standard Deviation"
```

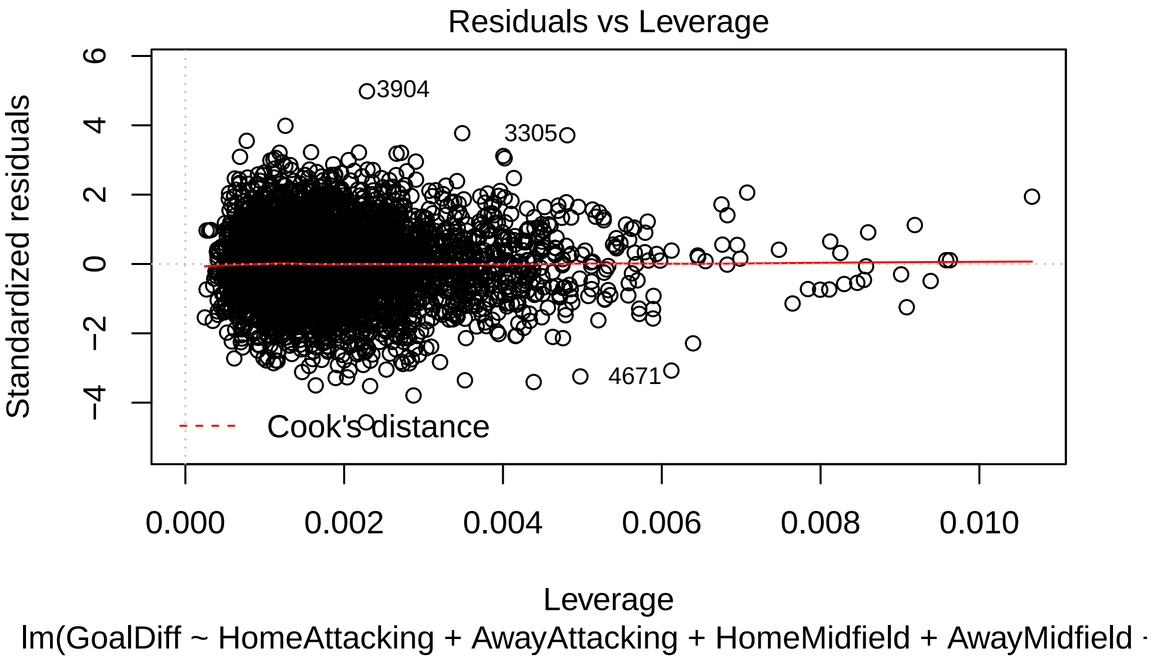
```
## [1] 1.596496
```

Nearly all of the predictors came out statistically significant, as well as the F-statistic. The R-squared value is a bit low, however, at 13.72%. In order to interpret the predictors, I'll use the HomeAttacking predictor as an example. According to the model, for every 1 increase in the home team's attacking rating, they can expect to win by 0.03987 more goals.




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(GoalDiff ~ HomeAttacking + AwayAttacking + HomeMidfield + AwayMidfield ·
6
4
2
0
−2
−4
−4
−2
0
2
4
3904
849
422


Scale−Location
√|Standardized residuals|
Fitted values
lm(GoalDiff ~ HomeAttacking + AwayAttacking + HomeMidfield + AwayMidfield ·
0.0
0.5
1.0
1.5
2.0
−2
−1
0
1
2
3904
422
849



The plots for the model don't appear to have any unusual trends, meaning there appears to be minimal bias in the fitted values.

I predicted the goal differential of the games in the testing data. I will be using the following 5 games from the 2018-2019 season as examples for comparisons: Sampdoria vs Juventus (Italy Div 1), Extremadura UD vs Lugo (Spain Div 2), Caen vs Reims (France Div 1), Sp Lisbon vs Tondela (Portugal Div 1), Rotherham vs Middlesbrough (England Div 2).

```
pred_gd <- predict(gf1, testing)
gd1 <- data.frame(GameNum,HomeTeam,AwayTeam,pred_gd)
gd1[c(20,25,176,188,234),c(2:4)]
```

```
##            HomeTeam      AwayTeam       pred_gd
## 20        Sampdoria      Juventus -1.3168560233
## 25   Extremadura UD          Lugo  0.0002996172
## 176            Caen         Reims  0.5253253610
## 188       Sp Lisbon       Tondela  2.1056185634
## 234       Rotherham Middlesbrough -0.6261657190
```

As you can see above, Juventus is expected to win by a total of 1.32 goals and Caen by 0.53 goals.

9.1.2 SVM Model

I used the same variables as the regression when making my svm model

```
svm1 <- svm(GoalDiff ~ HomeAttacking + AwayAttacking + HomeMidfield + AwayMidfield +
            HomeDefense + AwayDefense + HomeBench + AwayBench, data = testing,
            kernel = "linear", cost = 10, scale = FALSE)
print("Residual Standard Deviation")
```

```
## [1] "Residual Standard Deviation"
```

```
sd2 <- sd(resid(svm1))
sd2
```

```
## [1] 1.650754
```

The residual standard deviation on the svm model is a little bit higher than in the regression model, indicating a bit more variance in the model.

```
Fit <- fitted(svm1)
fit_svm1 <- data.frame(GameNum,HomeTeam,AwayTeam,Fit)
fit_svm1[c(20,25,176,188,234),c(2:4)]
```

```
##           HomeTeam      AwayTeam         Fit
## 20       Sampdoria      Juventus  0.08008844
## 25  Extremadura UD          Lugo  0.15123100
## 176           Caen         Reims  0.55089184
## 188      Sp Lisbon       Tondela  1.49713883
## 234      Rotherham Middlesbrough -0.84744266
```

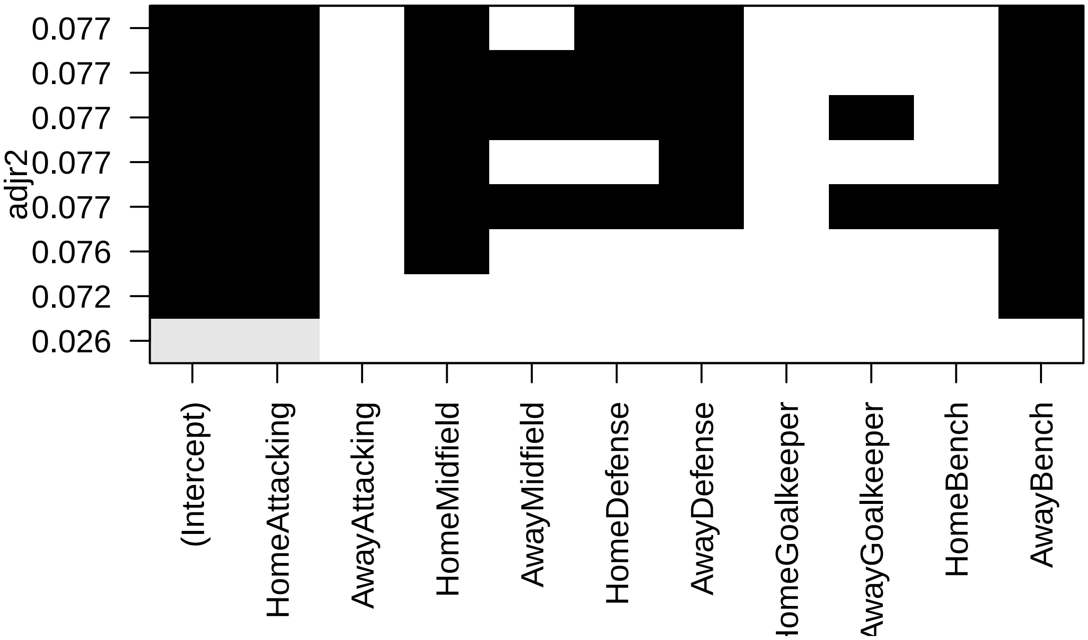
The svm expected goal differential came out for the most part, pretty close to the regression model. There were some games with big differences, such as the Sampdoria vs Juventus. Juventus was a heavy favorite in the regression model but is a slight underdog in the svm model.

9.2 Team Goals Prediction

For this prediction I looked at predicting the amount of goals each team will score. So I will be using to models, one for home goals and one for away goals.

9.2.1 Regression Model

```
w1 <- regsubsets(FTHG ~ HomeAttacking + AwayAttacking + HomeMidfield + AwayMidfield + HomeDefense + Awa
```



When looking at the subset selection above, there was not really a logical mixture of variables to use. So for the home team goals I ended up choosing each the home attacking and midfield, and the away defense and goalkeeper. For the away goals I did the reverse choosing the away attacking and midfield, and the home defense and goalkeeper. I chose these variables because the attackers/midfielders are usually responsible for most of the goals scored and the other team's defense/goalie are responsible for stopping goals.

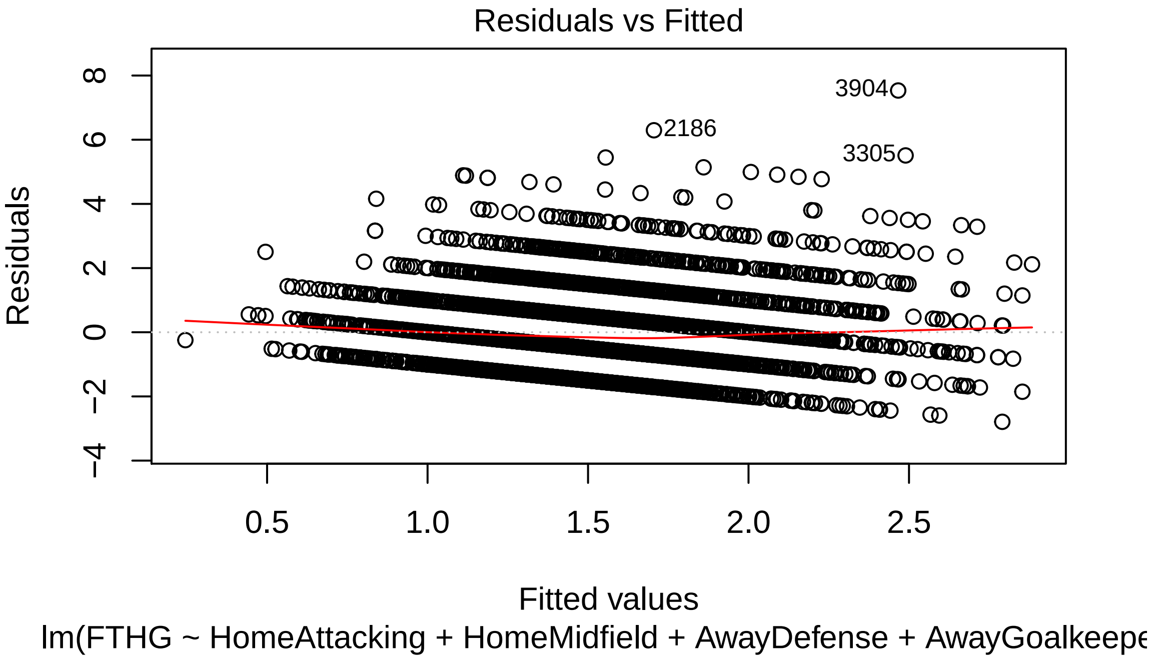
```
hgr1 <- lm(FTHG ~ HomeAttacking + HomeMidfield + AwayDefense + AwayGoalkeeper, training)
agr1 <- lm(FTAG ~ AwayAttacking + AwayMidfield + HomeDefense + HomeGoalkeeper, training)
```

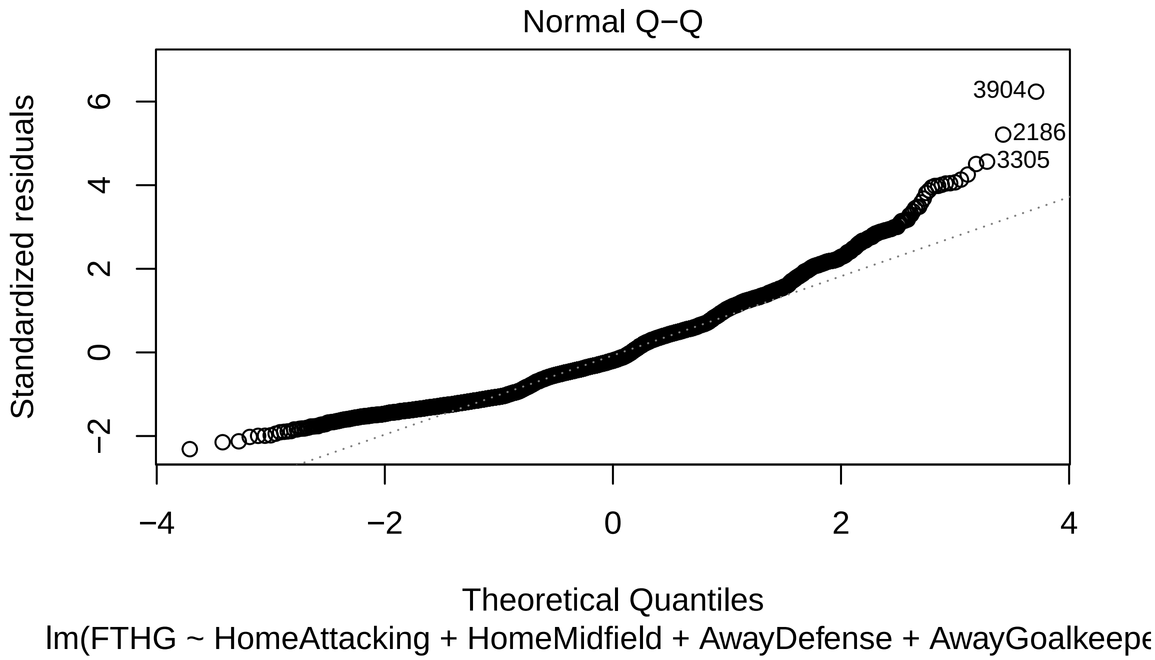
```
##
## Call:
## lm(formula = FTHG ~ HomeAttacking + HomeMidfield + AwayDefense +
##     AwayGoalkeeper, data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7906 -0.8585 -0.2376  0.6845  7.5337
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.464745   0.254929   1.823   0.0684 .
## HomeAttacking   0.048980   0.008725   5.614 2.09e-08 ***
## HomeMidfield    0.041998   0.009724   4.319 1.60e-05 ***
## AwayDefense    -0.065934   0.007798  -8.455  < 2e-16 ***
## AwayGoalkeeper -0.011342   0.005926  -1.914   0.0557 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

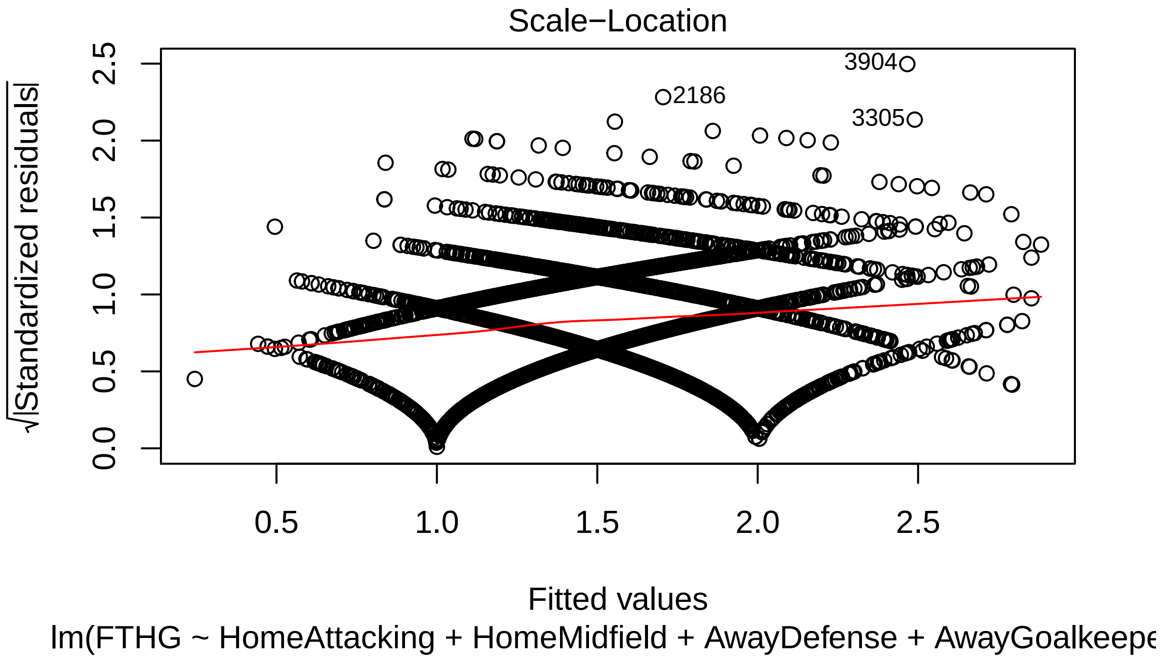
```
##
## Residual standard error: 1.209 on 4823 degrees of freedom
## Multiple R-squared:  0.07626,    Adjusted R-squared:  0.07549
## F-statistic: 99.54 on 4 and 4823 DF,  p-value: < 2.2e-16
```

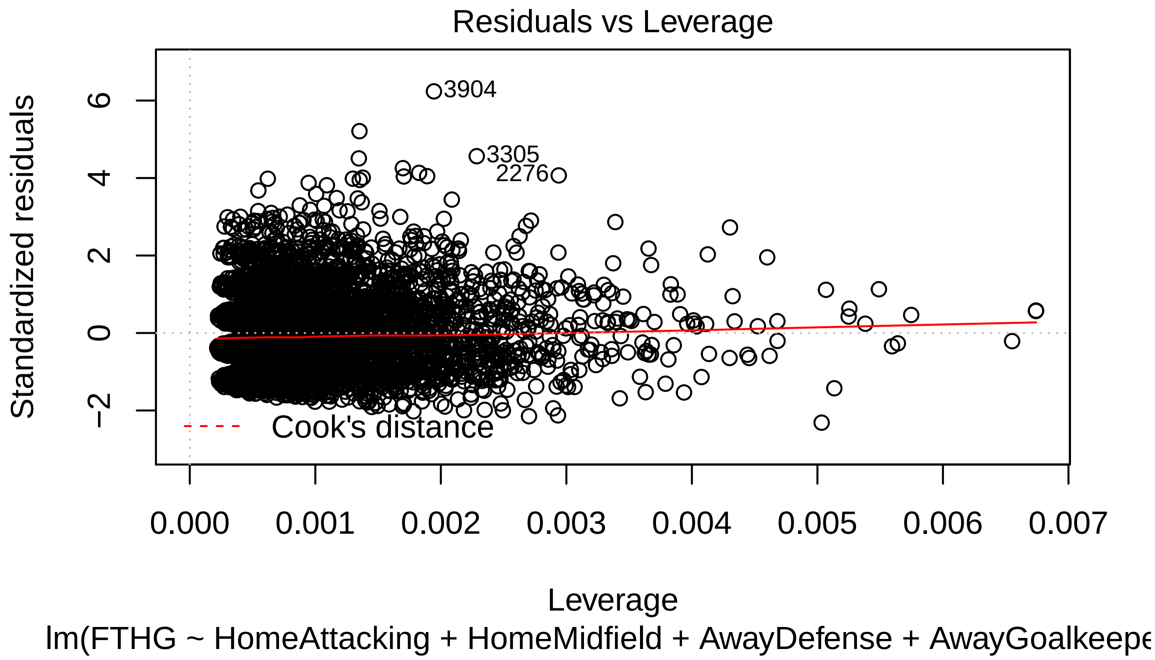
```
##
## Call:
## lm(formula = FTAG ~ AwayAttacking + AwayMidfield + HomeDefense +
##     HomeGoalkeeper, data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1034 -0.9264 -0.1458  0.7127  6.0758
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.473423   0.228960   2.068   0.0387 *
## AwayAttacking   0.041199   0.007909   5.209 1.97e-07 ***
## AwayMidfield    0.037685   0.008949   4.211 2.59e-05 ***
## HomeDefense    -0.058474   0.007024  -8.325  < 2e-16 ***
## HomeGoalkeeper -0.011223   0.005388  -2.083   0.0373 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.085 on 4823 degrees of freedom
## Multiple R-squared:  0.07201,    Adjusted R-squared:  0.07124
## F-statistic: 93.56 on 4 and 4823 DF,  p-value: < 2.2e-16
```

```
## [1] 1.208603
```




Normal Q−Q
Standardized residuals
−2
0
2
4
6
3904
2186
3305
−4
−2
0
2
4
Theoretical Quantiles
lm(FTHG ~ HomeAttacking + HomeMidfield + AwayDefense + AwayGoalkeepe


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3904
2186
3305
0.5
1.0
1.5
2.0
2.5
Fitted values
lm(FTHG ~ HomeAttacking + HomeMidfield + AwayDefense + AwayGoalkeepe



```
HomePred <- predict(hgr1, testing)
hg1 <- data.frame(GameNum,HomeTeam,AwayTeam,HomePred)
AwayPred <- predict(agr1, testing)
ag1 <- data.frame(GameNum,HomeTeam,AwayTeam,AwayPred)

result <- data.frame(GameNum,HomeTeam,AwayTeam,HomePred,AwayPred)
result[c(20,25,176,188,234),c(2:5)]
```

```
##            HomeTeam      AwayTeam  HomePred  AwayPred
## 20        Sampdoria      Juventus 0.9435511 2.2345122
## 25   Extremadura UD          Lugo 1.3179662 1.3718418
## 176            Caen         Reims 1.5204225 1.0230505
## 188       Sp Lisbon       Tondela 2.2663298 0.3370443
## 234       Rotherham Middlesbrough 0.9470495 1.5577845
```

Based on the model, in the Rotherham vs Middlesbrough, Rotherham has an expected goals of 0.95, while Middlesbrough has an expected goals of 1.56.

9.2.2 SVM Model

Again, I used the same variables as the regression when making my svm model

```
svmhg <- svm(FTHG ~ HomeAttacking + HomeMidfield + AwayDefense + AwayGoalkeeper,
             data = testing, kernel = "linear", cost = 10, scale = FALSE)
svmag <- svm(FTAG ~ AwayAttacking + AwayMidfield + HomeDefense + HomeGoalkeeper,
             data = testing, kernel = "linear", cost = 10, scale = FALSE)

HomePred <- fitted(svmhg)
fit_svmhg <- data.frame(GameNum,HomeTeam,AwayTeam,HomePred)
AwayPred <- fitted(svmag)
fit_svmag <- data.frame(GameNum,HomeTeam,AwayTeam,AwayPred)
```

```
result2 <- data.frame(GameNum,HomeTeam,AwayTeam,HomePred,AwayPred)
result2[c(20,25,176,188,234),c(2:5)]
```

```
##            HomeTeam      AwayTeam  HomePred  AwayPred
## 20        Sampdoria      Juventus 0.8182112 1.6126796
## 25   Extremadura UD          Lugo 1.0723554 1.0801967
## 176            Caen         Reims 1.3471580 0.8533573
## 188       Sp Lisbon       Tondela 2.0723432 0.4116691
## 234       Rotherham Middlesbrough 0.7482564 1.1940604
```

Similiarly to the goal differential models, the svm expected total goals came out pretty similar to the regression model.

9.3 Win Probabilities

The next thing I wanted to model was the probabilities of each possible match result. During league matches in soccer, there is no extra time. This means each match can end in either a home win, away win, or a draw. Since there are three possible results, a binary 0 or 1 (loss or win) can't be used to predict the probabilities. This limits the amount of methods that can be used to predict outcome probabilities, like logistic regression. I tried out 2 different statistical modeling techniques, one using t-stat distributions and the other using poisson distributions.

9.3.1 Goal Differential - t-stat

For this method I predicted the probability that the previously prdicted goal differential will result in a home win, away win, and draw based on the t distribution of the data. To predict the chance of a home win, I found the p-value of the t-score that the goal differential will be above 0.5 (rounds up to 1 or greater). To predict the chance of an away win, I found the p-value of the t-score that the goal differential will be below -0.5 (rounds down to -1 or less). To predict the chance of an draw, I plugged in the following equation: 1 - Home Win - Away Win to find the p-value that the goal differential will be between -0.5 and 0.5.

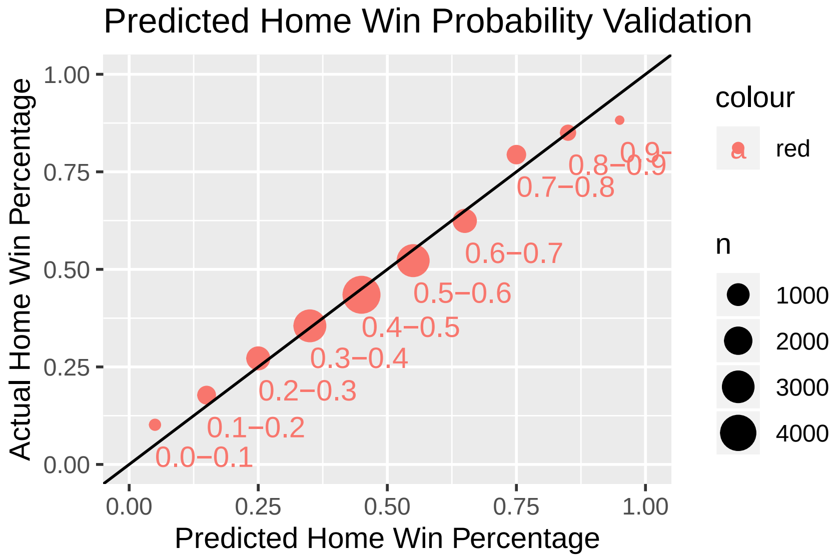
9.3.1.1 Regression Goal Differential Model Using the predicted goal differential from the regression model I got the following win probabilities.

```
gd1$HomeWin <- pt((gd1$pred_gd - .5)/(sd1), nrow(testing))
gd1$AwayWin <-  pt(-(gd1$pred_gd + .5)/(sd1), nrow(testing))
gd1$Draw <- 1 - gd1$HomeWin - gd1$AwayWin

gd1[c(20,25,176,188,234),c(2:7)]
```

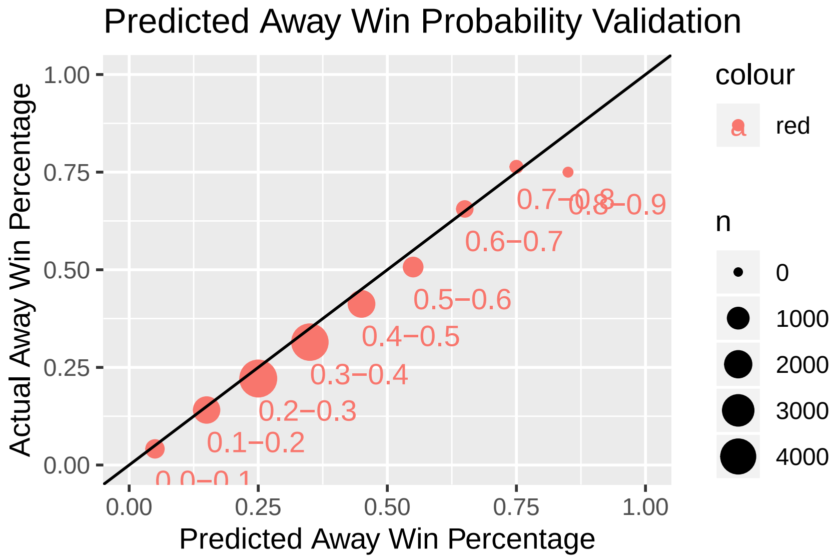
```
##            HomeTeam      AwayTeam       pred_gd   HomeWin    AwayWin      Draw
## 20        Sampdoria      Juventus -1.3168560233 0.1275639 0.69555006 0.1768860
## 25   Extremadura UD          Lugo  0.0002996172 0.3771433 0.37700069 0.2458561
## 176            Caen         Reims  0.5253253610 0.5063281 0.26036542 0.2333065
## 188       Sp Lisbon       Tondela  2.1056185634 0.8427151 0.05134155 0.1059434
## 234       Rotherham Middlesbrough -0.6261657190 0.2402868 0.53149374 0.2282194
```

These probabilities look pretty consistant with what you would expect from the corresponding goal differentials. Juventus is expected to beat Sampdoria by 1.32 goals, this means that Juventus is expected to win 69.6% of the time, Sampdoria is expected to win 12.8%, and they are expected to draw 17.9% of the time. The Caen vs Reims game is predicted to be closer with Caen winning by an expected 0.53 goals. As a result, Caen's win probability is a bit lower than Juventus's at 50.6%, and Reims is a bit higher than Sampdoria's at 26.0%. Since the expected goals is closer to 0, the probability of a draw is higher at 23.3%.

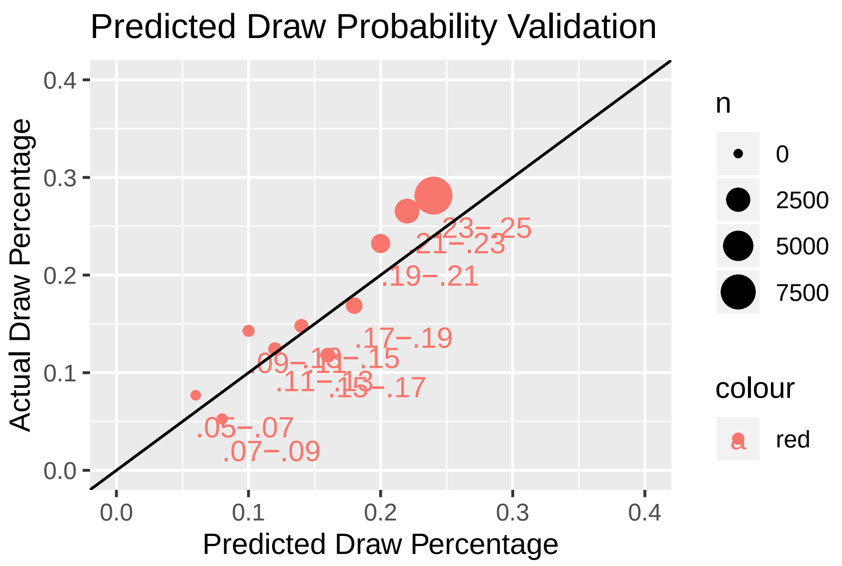
When the model predicts that Juventus has a win probability of 69%, it is predicting that they will actually win 69% of the time in real life. Same thing goes for Sampdoria winning and both of them drawing. In order to validate how well the model worked, you need to compare the predicted results to the actual results. So by grouping all of the games where the home team is expected to win between 40%-50% of the time, you'd expect the home team to win on average 45% of the real life games. This can be visualized by grouping all of the predicted values for every game onto the graphs below.



```
## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_text).
```



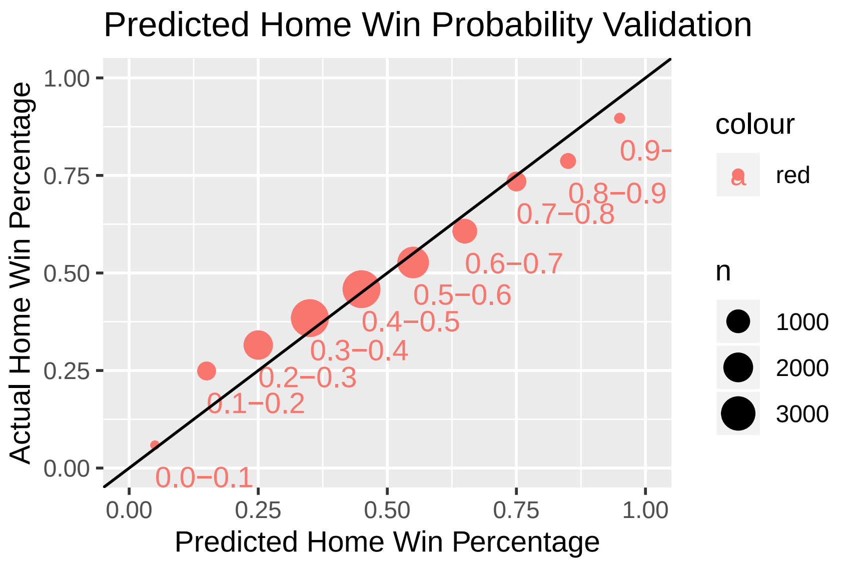
```
## Warning: Removed 5 rows containing missing values (geom_point).

## Warning: Removed 5 rows containing missing values (geom_text).
```



As the graph shows, the home win predictions were pretty accurate. For all the games with a home win probability between 40%-50%, the home team won those games on average nearly 45% of the time. The away probabilities were also pretty accurate. Most of the groups were a little below the line, indicating that there may be a little bit of an over prediction of away wins. For the draw probabilities, the majority of the games are above the line, indicating there is some noticable underprediction.

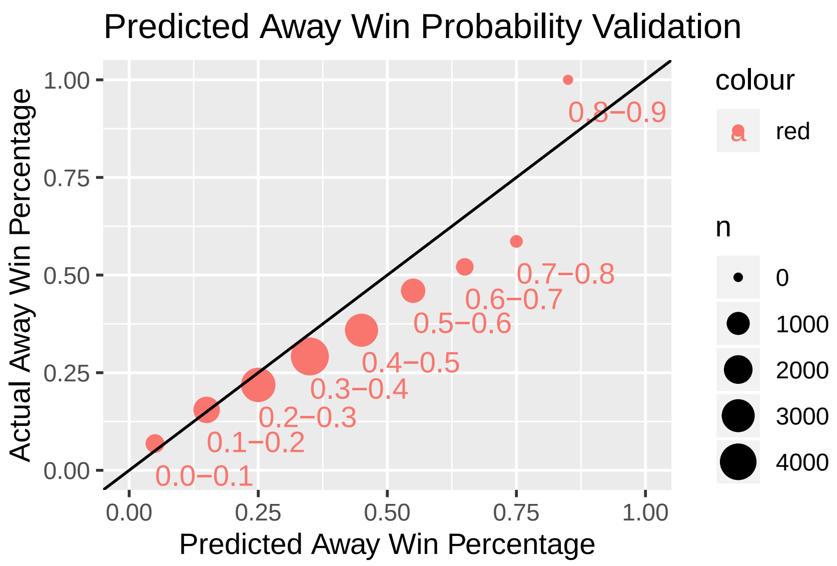
9.3.1.2 SVM Goal Differential Model The same steps were repeated for the svm model.

```
##             HomeTeam       AwayTeam         Fit   HomeWin   AwayWin      Draw
## 20         Sampdoria       Juventus  0.08008844 0.3996045 0.3626436 0.2377519
## 25   Extremadura UD           Lugo  0.15123100 0.4163364 0.3466069 0.2370566
## 176             Caen          Reims  0.55089184 0.5122970 0.2621934 0.2255096
## 188        Sp Lisbon        Tondela  1.49713883 0.7270903 0.1131811 0.1597286
## 234        Rotherham Middlesbrough -0.84744266 0.2071827 0.5833501 0.2094672
```

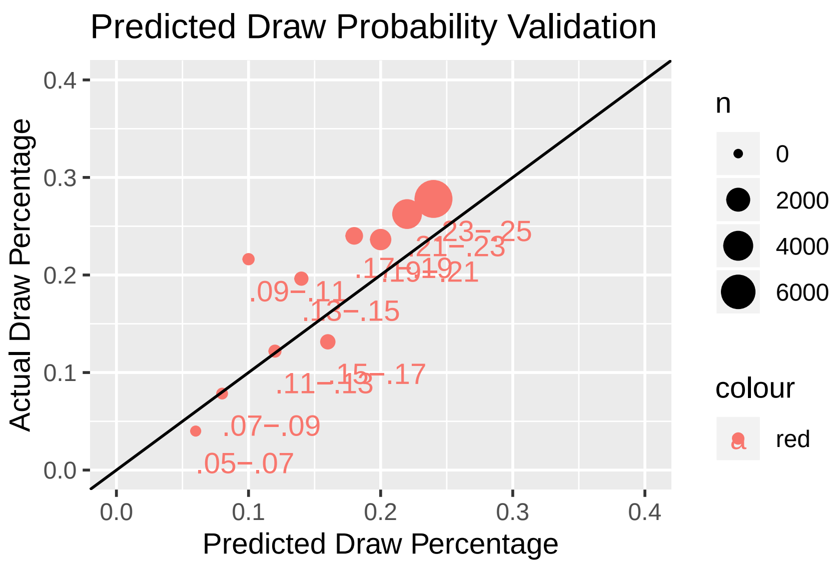


```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Warning: Removed 1 rows containing missing values (geom_text).
```



```
## Warning: Removed 5 rows containing missing values (geom_point).

## Warning: Removed 5 rows containing missing values (geom_text).
```



There are some much more noticeable issues with the svm goal differential prediction model. For home win probabilities, the model is not good at predicting the extremes. There is a pretty big over prediction in the away team's win probabilities. There is a similar but larger under prediction of the draw probabilities.

9.3.2 Team Goals Scored - Poisson

Another way of going about predicting win percentages is to use poisson distributions. In real life, soccer scores are integers, not decimals. Poisson distributions would predict the probability a team would score 0, 1, 2, etc goals in a game. Since poisson distributions can't use negative numbers, I will be using the total goals scored models.

Below is an example of how the predictions are formulated. Caen is expected to score 1.52 goals in this game. By plugging this number into the dpois function, they are expected to score 0 goals 21.9% of the time, 1 goal 33.2% of the time, and so on. Reims has a lower expected goals so their chance of scoring 0 & 1 goals is higher than Caen and the chance of scoring 2 or more goals is much lower. (Goal5 represents the probability of scoring 5 or more goals, not just 5 goals)

```
##   X  Team      Pred      Goal0     Goal1      Goal2     Goal3       Goal4
## 1 1  Roma 2.4379152 0.08734275 0.2129342 0.25955779 0.2109266 0.128555312
## 2 2 Parma 0.6015173 0.54797956 0.3296192 0.09913582 0.0198773 0.002989136
##          Goal5
## 1 0.0626813904
## 2 0.0003596033
```

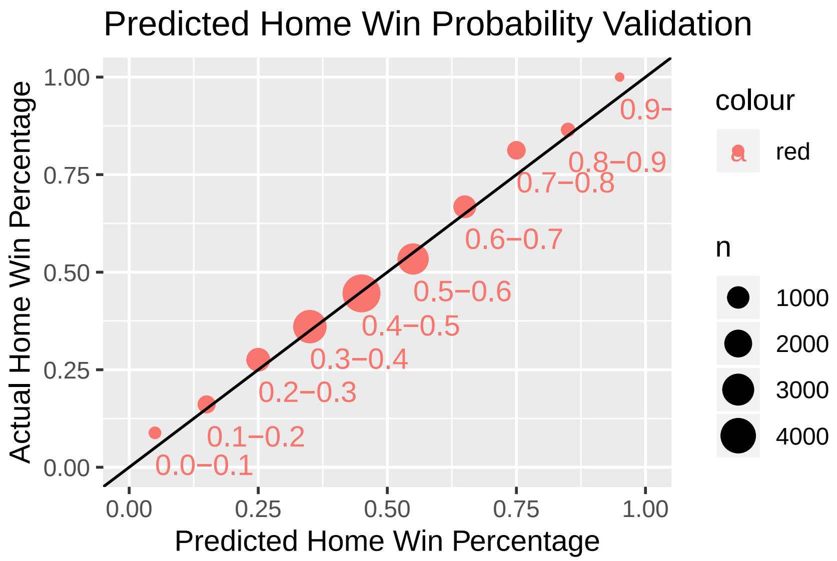
After these goal probabilities are calculated, they are all multiplied together to find the probabilities of every possible result up to 5 goals to create the data frame below. There is a 7.85% chance the game ends 0-0 or a 9.29% chance Caen, the home team, will win 2-1. In order to calculate the win probabilities you need to add up all of the results that result in a home win, away win, or draw. So all of the results where the home team wins (bottom left side) are added up together. All of the results where the away team wins (top right side) are added up together. All of the results in the middle, [Home0,Away0], [Home1,Away1], etc are added up together.

```
##   X  Home      Away0      Away1       Away2       Away3        Away4
## 1 1 Home0 0.04786204 0.02878985 0.008658795 0.001736138 0.0002610793
## 2 2 Home1 0.11668360 0.07018721 0.021109409 0.004232558 0.0006364893
## 3 3 Home2 0.14223236 0.08555523 0.025731475 0.005159309 0.0007758534
## 4 4 Home3 0.11558348 0.06952546 0.020910385 0.004192653 0.0006304883
## 5 5 Home4 0.07044568 0.04237430 0.012744436 0.002555333 0.0003842692
## 6 6 Home5 0.03434812 0.02066099 0.006213971 0.001245937 0.0001873632
##          Away5
## 1 3.140875e-05
## 2 7.657186e-05
## 3 9.333785e-05
## 4 7.584992e-05
## 5 4.622892e-05
## 6 2.254044e-05
```

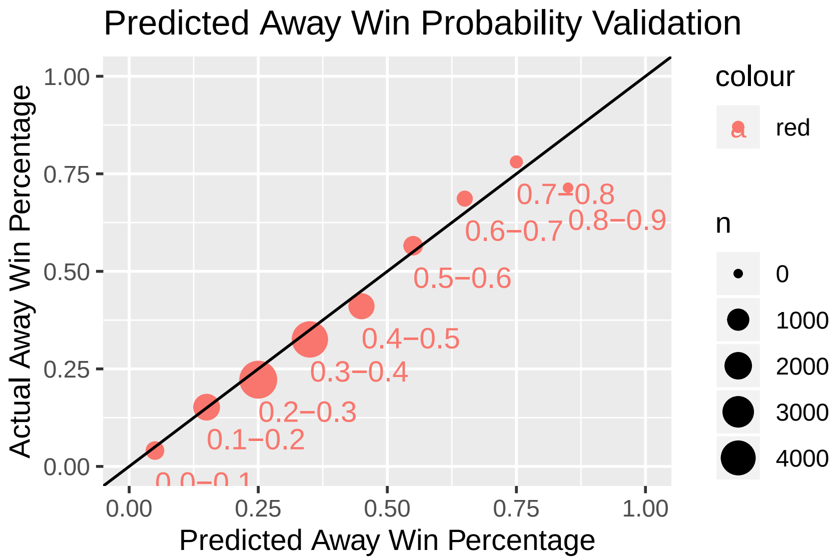
9.3.2.1 Regression Total Goals Scored Model Below are the predicted probabilities for all of the regression predicted goals scored.

```
hg1$Goal0 <- dpois(0, lambda = hg1$HomePred)
hg1$Goal1 <- dpois(1, lambda = hg1$HomePred)
hg1$Goal2 <- dpois(2, lambda = hg1$HomePred)
hg1$Goal3 <- dpois(3, lambda = hg1$HomePred)
hg1$Goal4 <- dpois(4, lambda = hg1$HomePred)
hg1$Goal5 <- 1-ppois(4, lambda = hg1$HomePred)

ag1$Goal0 <- dpois(0, lambda = ag1$AwayPred)
ag1$Goal1 <- dpois(1, lambda = ag1$AwayPred)
ag1$Goal2 <- dpois(2, lambda = ag1$AwayPred)
ag1$Goal3 <- dpois(3, lambda = ag1$AwayPred)
ag1$Goal4 <- dpois(4, lambda = ag1$AwayPred)
ag1$Goal5 <- 1-ppois(4, lambda = ag1$AwayPred)

result$HomeWin <- hg1$Goal1*ag1$Goal0 + hg1$Goal2*ag1$Goal0 + hg1$Goal2*ag1$Goal1 + hg1$Goal3*ag1$Goal0
result$AwayWin <- ag1$Goal1*hg1$Goal0 + ag1$Goal2*hg1$Goal0 + ag1$Goal2*hg1$Goal1 + ag1$Goal3*hg1$Goal0
result$Draw <- hg1$Goal0*ag1$Goal0 + hg1$Goal1*ag1$Goal1 + hg1$Goal2*ag1$Goal2 + hg1$Goal3*ag1$Goal3 +

result[c(20,25,176,188,234),c(2:8)]
```

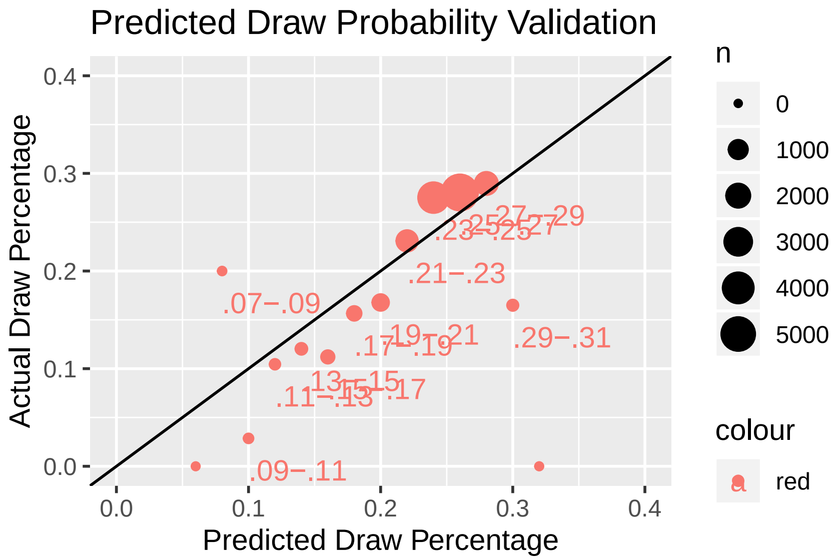
```
##          HomeTeam      AwayTeam  HomePred  AwayPred   HomeWin    AwayWin
## 20       Sampdoria      Juventus 0.9435511 2.2345122 0.1458304 0.66585551
## 25  Extremadura UD          Lugo 1.3179662 1.3718418 0.3582212 0.38308838
## 176           Caen         Reims 1.5204225 1.0230505 0.4876610 0.25486491
## 188      Sp Lisbon       Tondela 2.2663298 0.3370443 0.8159005 0.04177306
## 234      Rotherham Middlesbrough 0.9470495 1.5577845 0.2295754 0.51574987
##          Draw
## 20  0.1883141
## 25  0.2586904
## 176 0.2574741
## 188 0.1423264
## 234 0.2546748
```



```
## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_text).
```



```
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_text).
```



The home and win probabilities were pretty accurate and close to the line. The draw probabilities are a little better distributed than the t-stat goal differential regression model. Part of this is likely do to the fact that the highest draw probability for the t-stat model was around 24.5%. This model is able to predict higher and as a result removed some of that previous under prediction.

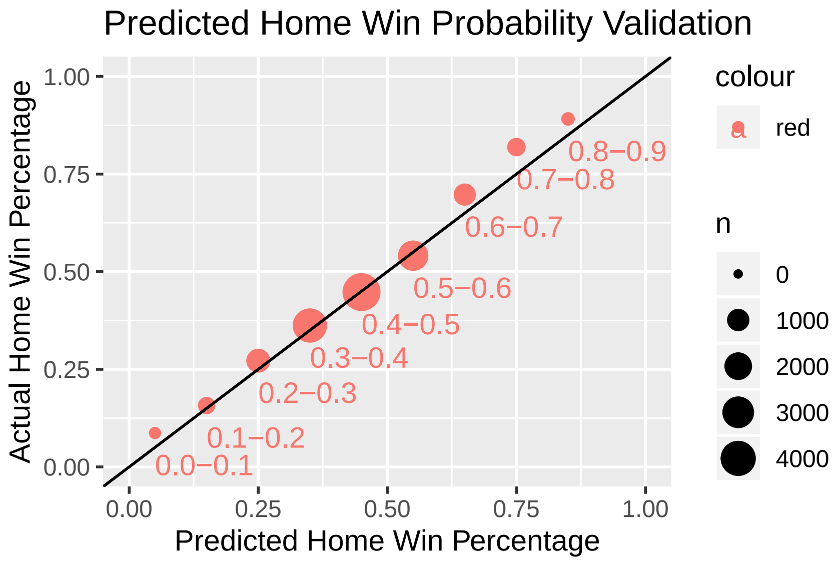
9.3.2.2 SVM Total Goals Scored Model The same steps were repeated for the svm model.

```
fit_svmhg$Goal0 <- dpois(0, lambda = fit_svmhg$HomePred)
fit_svmhg$Goal1 <- dpois(1, lambda = fit_svmhg$HomePred)
fit_svmhg$Goal2 <- dpois(2, lambda = fit_svmhg$HomePred)
fit_svmhg$Goal3 <- dpois(3, lambda = fit_svmhg$HomePred)
fit_svmhg$Goal4 <- dpois(4, lambda = fit_svmhg$HomePred)
fit_svmhg$Goal5 <- 1-ppois(4, lambda = fit_svmhg$HomePred)

fit_svmag$Goal0 <- dpois(0, lambda = fit_svmag$AwayPred)
fit_svmag$Goal1 <- dpois(1, lambda = fit_svmag$AwayPred)
fit_svmag$Goal2 <- dpois(2, lambda = fit_svmag$AwayPred)
fit_svmag$Goal3 <- dpois(3, lambda = fit_svmag$AwayPred)
fit_svmag$Goal4 <- dpois(4, lambda = fit_svmag$AwayPred)
fit_svmag$Goal5 <- 1-ppois(4, lambda = fit_svmag$AwayPred)

result2$HomeWin <- fit_svmhg$Goal1*fit_svmag$Goal0 + fit_svmhg$Goal2*fit_svmag$Goal0 + fit_svmhg$Goal2*
result2$AwayWin <- fit_svmag$Goal1*fit_svmhg$Goal0 + fit_svmag$Goal2*fit_svmhg$Goal0 + fit_svmag$Goal2*
result2$Draw <- fit_svmhg$Goal0*fit_svmag$Goal0 + fit_svmhg$Goal1*fit_svmag$Goal1 + fit_svmhg$Goal2*fit_

result2[c(20,25,176,188,234),c(2:8)]
```

```
##           HomeTeam      AwayTeam  HomePred  AwayPred   HomeWin    AwayWin
## 20       Sampdoria      Juventus 0.8182112 1.6126796 0.1894092 0.56216951
## 25   Extremadura UD         Lugo 1.0723554 1.0801967 0.3504321 0.35441582
## 176           Caen         Reims 1.3471580 0.8533573 0.4830296 0.23728689
## 188      Sp Lisbon       Tondela 2.0723432 0.4116691 0.7668916 0.06184778
## 234      Rotherham Middlesbrough 0.7482564 1.1940604 0.2314606 0.46545130
##          Draw
## 20  0.2484213
## 25  0.2951520
## 176 0.2796835
## 188 0.1712606
## 234 0.3030881
```

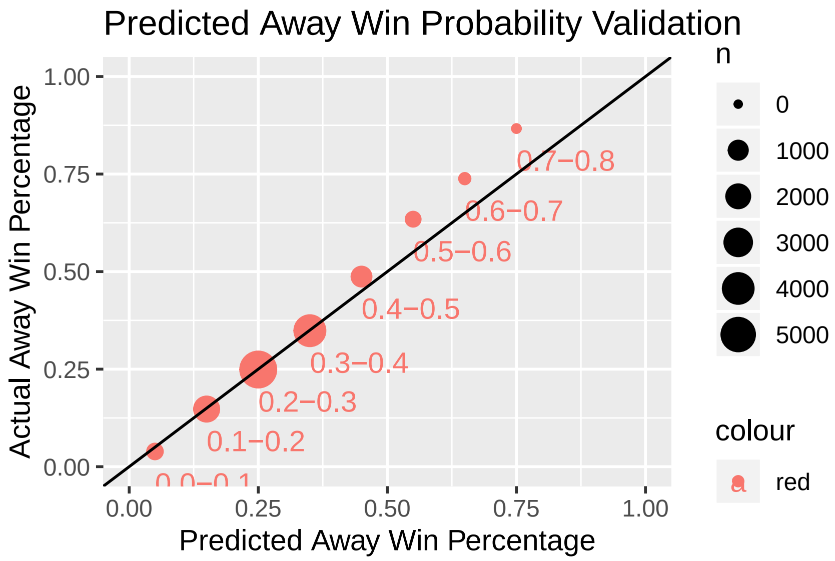
```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Warning: Removed 1 rows containing missing values (geom_text).
```



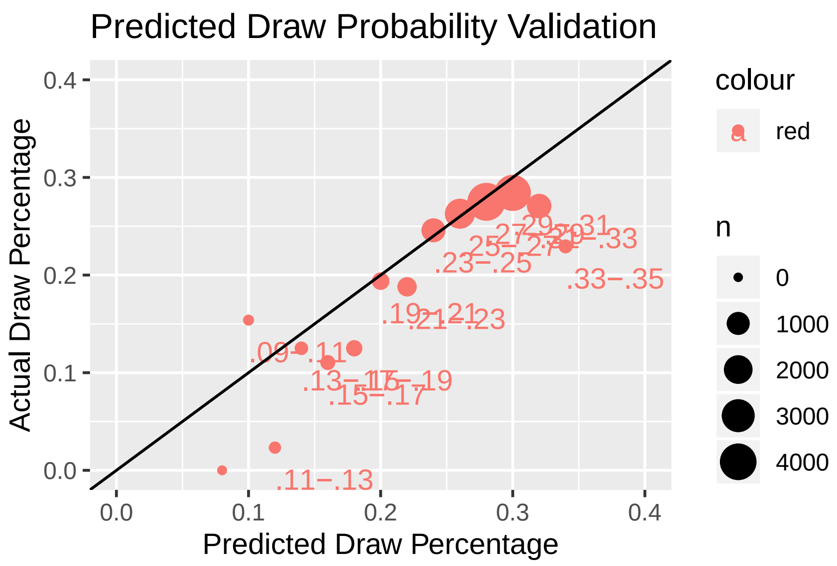
```
## Warning: Removed 2 rows containing missing values (geom_point).
```

```
## Warning: Removed 2 rows containing missing values (geom_text).
```



```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Warning: Removed 1 rows containing missing values (geom_text).
```



This svm model was fairly accurate for the home team, and had a slight under prediction of the away wins. This model was overall the best at predicting draw probabilities.

9.4 Final Standings Predictions

One last interesting thing I wanted to create using the models was a full season standings projection.For this process, I pulled all of the games that were played for a specific league during a specific season. Using each teams poisson regression win probabilities, I was able to tally up each team's expected wins, draws, and losses. In soccer they use a point system to determine the final standings. Each team gets 3 points for a win, 1 point for a draw, and 0 points for a loss. So I took the expected wins and draws and created an expected points variable. Below are 3 final standings examples for 3 leagues with different variations of team talent levels.

Italy Division 1 - Serie A 2018/2019 season

```
italyclubs <- arrange(italyclubs, desc(pred_points))
italyclubs
```

```
##    italyclubs   rating      wins    draws    losses pred_points actual_points
## 1    Juventus 87.00000 27.061944 6.215209  4.722847    87.40104            90
## 2       Inter 84.27273 23.194113 7.539239  7.266648    77.12158            69
## 3      Napoli 83.81818 22.770585 7.393322  7.836094    75.70508            79
## 4        Roma 82.36364 20.579462 8.101916  9.318622    69.84030            66
## 5       Milan 82.00000 20.242265 7.943062  9.814673    68.66986            68
## 6       Lazio 81.72727 19.611988 7.966726 10.421286    66.80269            59
## 7      Torino 78.27273 15.152996 8.486235 14.360769    53.94522            63
## 8    Atalanta 77.36364 14.499538 8.251553 15.248909    51.75017            69
## 9   Sampdoria 76.72727 13.258013 8.378149 16.363838    48.15219            53
## 10 Fiorentina 76.54545 12.679989 8.724297 16.595714    46.76426            41
## 11     Chievo 76.54545 12.616819 8.457794 16.925387    46.30825            20
## 12    Bologna 76.54545 12.375096 8.739640 16.885264    45.86493            44
## 13   Cagliari 76.27273 12.141431 8.693897 17.164672    45.11819            41
## 14   Sassuolo 76.27273 11.992684 8.573914 17.433402    44.55197            43
## 15      Genoa 76.09091 11.687035 8.883807 17.429157    43.94491            38
## 16    Udinese 74.90909 10.727595 8.251244 19.021161    40.43403            43
## 17       Spal 75.00000 10.564597 8.313668 19.121735    40.00746            42
## 18     Empoli 73.72727  9.243776 8.339138 20.417086    36.07047            38
## 19      Parma 73.27273  9.226577 7.968842 20.804581    35.64857            41
## 20  Frosinone 73.18182  8.664515 8.196310 21.139175    34.18986            25
```

In Serie A, there is a very dominant team, Juventus, a strong 2nd tier, Inter-Lazio, and a 3rd tier of the rest, and the projections reflect that. There was a bit of variation and there were some teams that greatly overperformed projections, like Atalanta, and underperformed, like Chievo.

Turkey Division 1 - Futbol Ligi 1 2017/2018 season

```
turkclubs <- arrange(turkclubs, desc(pred_points))
turkclubs
```

```
##               turkclubs   rating      wins    draws    losses pred_points
## 1              Besiktas 80.18182 21.180752 7.119913  5.699335    70.66217
## 2            Fenerbahce 78.54545 18.652509 7.835103  7.512388    63.79263
## 3           Galatasaray 77.72727 18.169777 7.555790  8.274433    62.06512
## 4           Trabzonspor 76.36364 16.542276 7.964018  9.493706    57.59085
## 5            Buyuksehyr 76.36364 15.663742 8.241452 10.094807    55.23268
## 6           Antalyaspor 75.72727 15.652625 7.579078 10.768297    54.53695
## 7           Kayserispor 73.45455 12.934285 8.065422 13.000294    46.86828
## 8             Bursaspor 73.18182 12.059685 8.365460 13.574856    44.54451
## 9           Osmanlispor 73.18182 11.252824 8.478249 14.268927    42.23672
## 10            Konyaspor 72.54545 11.200910 8.367685 14.431406    41.97041
## 11               Goztep 72.54545 11.065382 8.453795 14.480823    41.64994
## 12            Sivasspor 72.18182 10.976762 8.101018 14.922220    41.03130
## 13           Alanyaspor 72.27273 10.926876 8.046819 15.026305    40.82745
## 14          Karabukspor 71.27273  9.894229 8.064995 16.040776    37.74768
## 15            Kasimpasa 71.18182  9.764967 8.239840 15.995193    37.53474
## 16 Akhisar Belediyespor 71.18182  9.510864 8.326747 16.162389    36.85934
## 17     Yeni Malatyaspor 70.45455  9.040053 8.011158 16.948790    35.13132
## 18       Genclerbirligi 70.72727  8.854866 8.496697 16.648437    35.06129
##    actual_points
## 1             71
## 2             72
## 3             75
## 4             55
## 5             72
## 6             38
## 7             44
## 8             39
## 9             33
## 10            36
## 11            49
## 12            49
## 13            40
## 14            12
## 15            46
## 16            42
## 17            43
## 18            33
```

The Turkish league is a little more balanced with a few stronger teams at the top. There was a tighter range in projected points than Serie A.

Spain Division 2 - La Liga Segunda Division 2016/2017 season

```
esp2clubs <- arrange(esp2clubs, desc(pred_points))
esp2clubs
```

```
##        esp2clubs   rating     wins     draws   losses pred_points actual_points
## 1        Levante 74.72727 21.71416  9.915493 10.37035    75.05798            84
## 2         Getafe 72.72727 18.59108 10.790570 12.61835    66.56380            68
## 3         Girona 72.09091 17.41670 11.095019 13.48828    63.34512            70
## 4      Vallecano 72.00000 17.43247 10.613498 13.95404    62.91089            53
## 5         Oviedo 71.63636 17.07022 10.492032 14.43775    61.70269            61
## 6        Almeria 71.36364 16.62870 11.082195 14.28911    60.96829            51
## 7       Zaragoza 71.63636 16.72396 10.695631 14.58041    60.86751            50
## 8     Valladolid 71.90909 16.64808 10.751523 14.60040    60.69576            63
## 9       Tenerife 71.54545 16.59834 10.653623 14.74803    60.44865            66
## 10         Cadiz 71.00000 15.91592 10.812226 15.27185    58.56000            64
## 11      Alcorcon 70.45455 15.04832 11.020607 15.93107    56.16558            50
## 12          Lugo 70.45455 15.15323 10.667726 16.17904    56.12742            55
## 13     Gimnastic 70.54545 15.03848 10.550380 16.41114    55.66583            52
## 14         Elche 70.54545 14.94839 10.689020 16.36259    55.53420            43
## 15       Cordoba 70.45455 14.58669 10.929589 16.48372    54.68965            55
## 16        Huesca 70.36364 14.68168 10.612483 16.70584    54.65752            63
## 17 Reus Deportiu 70.27273 14.58811 10.844314 16.56757    54.60865            55
## 18      Numancia 70.09091 14.60297 10.713170 16.68386    54.52208            50
## 19      Mallorca 69.72727 13.71463 10.757305 17.52807    51.90119            45
## 20     Sevilla B 69.72727 13.26680 10.985265 17.74793    50.78567            53
## 21      Mirandes 68.09091 11.95687 10.373925 19.66920    46.24454            41
## 22   UCAM Murcia 68.27273 11.75108 10.800621 19.44829    46.05388            48
```

The Spanish second division is very evenly balanced. Outside of the top team, Levante, the range between the 2nd best projection and worst team is only 20 points, whereas Serie A's range is 43. There is also a low amount of variation between the projections and the actual results.

10 Discussion

There are a few limitations and sources of bias in the data and model itself that should be addressed. As mentioned in the introduction, EA Sports Fifa ratings are subjective ratings made by talent scouts. While the ratings tend to be fairly good at rating a player's skill set, they are not perfect. The data sets used in the project are also only preseason rankings. EA Sports often puts out updates to the game where it will make adjustments to player ratings if a player is overperforming or underperforming their preseason ratings. Since I don't have access to any of those in season adjustments there may be some increasing bias as the season goes on because the models are only using the pre season rankings. One last major source of bias in the model is that it always uses the same lineup for each team for the whole season. In real life, game day lineups are always changing, whether its due to tactical strategies, injuries, or individual performance. Since I do not have data on game time lineups, I used the same lineups with the best players for each entire season. So there might be a bit of an over estimation for teams that lost a lot of players to injury.

In the introduction, I stated that my goals were to create models to predict goal differential, goals scored, win probabilities, and final standings. I was able to successfully create multiple models for all of these different predicitions. There is plenty of room for improvements in the models, but they seemed to do a good at making predictions I outlined previously, some better than others. The models are also able to serve the real world functions I previously outlined. A club's front office could insert potential free agents into a model to see how much of an impact they might have on the final standings. Managers can insert different lineups into a model and see how it affects their probability of winning. Sports bettors can look at the goal differential models to find inefficiencies in the betting market. There are also plenty of other models that could be created out of the data, like predicting a player's win shares or future player value.

11 Conclusions

Overall, I used 4 different models in predicting at goal differential, goals scored, win probabilities, and final standings. The regression model ended up being the strongest model for predicting the expected goal differential and goals scored. Based on the validation plots, the regression model using a poisson distribution was the most accurate model for predicting win probability. That is also why I chose to use it to predict the final standings. I believe that FIFA player data, while it does have its flaws, does a good job at making soccer predictions. There is likely room for improvement with more robust objective data, but the FIFA data is good for doing a cheap open sourced project. While these models did have fairly low R-squares, much of that can be attributed to the many unpredictable events at sporting events. This is esspecially true in a low scoring game like soccer, where one misstep could be all it takes to change the outcome.

12 Acknowledgments

I want to thank Prof. Laura Bruckman first for all of her support throughout the whole process of writing this report. Thank you to Prof. Roger French for teaching DSCI 351 & 353 where I learned many of the data science methods I used in this project. I also want to thank my Statistics advisor Prof. Paula Fitzgibbon for being an additional resource for the report. Thanks to Jiqi Liu and the DSCI 351 class for their valuable feedback. Finally I want to thank Case Western Reserve University for providing many resources for completing this report, especially the ODS-VDI.

13 References, Citations

Data set links:

https://www.kaggle.com/rovilayjnr/fifa-17-datasets

https://www.kaggle.com/thec03u5/fifa-18-demo-player-dataset

https://www.kaggle.com/karangadiya/fifa19

https://www.kaggle.com/sashchernuh/european-football

References

https://fivethirtyeight.com/methodology/how-our-club-soccer-predictions-work/

https://stackoverflow.com

https://ismayc.github.io/rbasics-book/4-rmarkdown.html

Packages used:

dplyr, ggplot2, jtools, leaps, e1071