Predicting Premier League Win Using Fifa Ratings

Sam Assaf

Garrett Atkinson

Carlo Lopez Hernandez

University of Notre Dame - Mendoza School of Business

MSBA - Sports Analytics

MSSA-60220 Predictive Analytics

2023 Fall

Data Description

The data utilized for this project was sourced from two Kaggle datasets. The first dataset, titled *"FIFA 20 Complete Player Dataset,"* encompasses player data spanning FIFA 15 to FIFA 20, providing information across the last six versions of the FIFA video game. This dataset comprises 13,605 distinct player entries, each featuring attributes such as:

- The URL of the scraped player
- Player positions
- Player attribute ratings; numerical ratings (0-100) for skill level in categories like shooting, endurance rate, ball handling, etc
    - Work rate was only a categorical player attribute
- Roles in both the club and the national team
- Personal details like nationality, club, date of birth, wage, and salary

The attributes further include statistics related to attacking, skills, defense, mentality, goalkeeper skills, and more. In an analytical context, we leveraged this data to assign values to individual players, considering their team and position, as a preparatory step for our logistic regression analysis.

The second dataset was also obtained from Kaggle, known as the *"All Premier League Matches 2010-2021"* dataset, marketed as one of the most comprehensive datasets for the English Premier League. It encompasses information from 4,070 matches, featuring an array of 113 distinct features. Collected through web scraping, this dataset includes statistical features for each match, spanning over an 11-year period. This extensive coverage makes the dataset well-suited for applications in prediction models, such as forecasting the winner of a match. Noteworthy features in this dataset include:

- Date and Scores
- Home and away club
- Home and away team statistics from the respective match

When combined with the player and position information from the previous dataset, these factors serve as inputs for our logistic regression model.

The Question We Are Addressing

The question addressed in this project is: How reliable are FIFA ratings in predicting a Premier League team's outcome of winning (1) vs. losing or tying (0) based on their matchup with the

difference in aggregated FIFA measures of a team vs. their opponent? This inquiry holds significance due to the widespread engagement in sports betting. Various types of bets are available, and one prominent category centers on predicting whether a team will emerge victorious or not, known as moneyline betting. Oddsmakers determine payouts based on the teams involved, offering positive or negative values. Positive values indicate the team is an underdog. For instance, a moneyline value of +145 means a bettor, wagering $100 and winning, receives a payout of $145. Conversely, negative values denote the favored team. For example, a moneyline value of -300 implies a bet of $300 is required for a $100 payout upon winning. This methodology extends to preseason bets, influencing predictions on Premier League winners or relegated teams, as a team's win count impacts the final standings. Developing a predictive model for team outcomes, and considering matchups, aids individuals in strategic decision-making for sports betting, allowing for the maximization of winnings or the minimization of losses.

Data Exploration

In this phase of the project, we focused on accomplishing two crucial tasks: merging our two datasets into a cohesive whole and refining the final dataset for upcoming tests. The initial step involved categorizing players from the FIFA dataset into four main groups: Attack, Midfield, Defense, and Goalkeeper. Each player had a list of positions they play, in order from primary position to secondary and so on, we chose to only keep their primary position and assign the position to categories based on their role on a typical soccer team. This allowed us to aggregate by position groups within each club.

1. Attack - Left Forward, Striker, Center Forward, Right Forward, Left Wing, Right Wing
2. Midfield - Left Midfielder, Center Midfielder, Central Attacking Midfielder, Central Defensive Midfielder, Right Midfielder
3. Defense- Left Wing-Back, Left Back, Center Back, Right Back, Right Wing-Back
4. Goalkeeper- Goalkeeper

After categorization, we filtered out substitutes for each position to get a more accurate description of the talent and skills that would likely be on the field for the matchup. The subsequent step involved creating a comprehensive table. For this, we implemented a for loop that traversed through all our FIFA 15-20 datasets, extracting key properties for each grouping and position. This included the variance of each position group, the maximum and median work ratings for each group, and the maximum goalkeeper rating. We thought the maximum attribute for position groups (excluding goalkeeper) would capture any star players a team might have. In contrast, the median rating would capture the middle of the position group and not be pulled one way or another by outliers. Lastly, the variance would capture whether a whole position group is relatively similar in terms of rating or not. We chose to only keep the max goalkeeper in our final

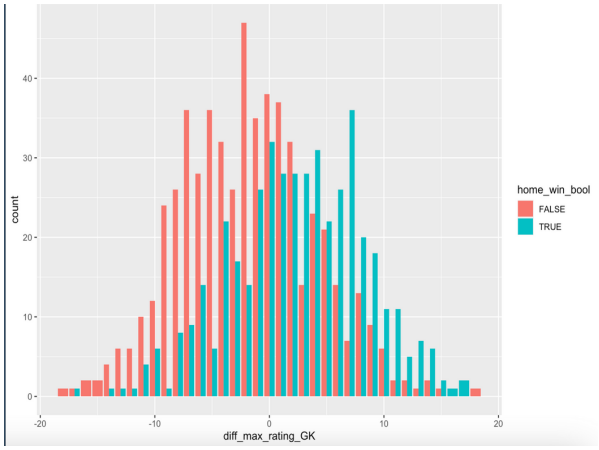
model due to the nature of one goalkeeper playing for a club assuming everyone is healthy. As for the categorical variable of work rate, we chose to assign values of -1 for Low, 0 for Medium, and 1 for High and sum it across the position group to get a position group work rate.

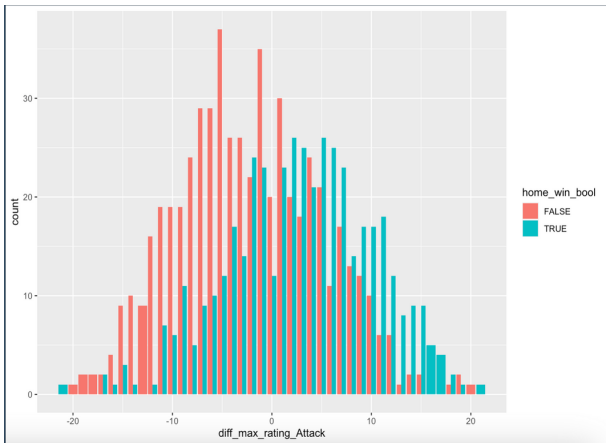
An integral part of our process was establishing a "home win boolean" column. Originally, our data featured a variable called "result_full," which was in the format "1-2" (home-away) for a match score. We created new columns for each team's individual score, and then, through an ifelse statement, assigned a TRUE or FALSE value to matches based on whether the home team won. This new column was labeled "home_win_bool."

Moving forward, we merged all FIFA datasets, and after some changing of the data format through pivoting and similar actions, we created a home and away data frame containing all the premier league clubs and their according ratings for each season. We then merged home premier league ratings and away premier league ratings with the premier league results data frame. We addressed the few missing values by dropping them due to the minimal impact on the sample size, checked and displayed the dimensions of the dataset after dropping rows with missing values, and stored the result in a new dataset named "premier_league_results_merged_dropped_nas."

Lastly, we converted our column "home_win_1_0" into a factor, representing whether the home team won (1) or not (0). Further, we iterated through the columns, calculating differences between corresponding 'home' and 'away' aggregated position rating measures and work rates, and generated new columns prefixed with 'diff_'. This left us with a home minus away column for each aggregated rating measure. With these preparations, our final dataset, named "premier_league_results_merged_fifa_rating_summary_stats," was ready for analysis.

Data Summaries





The graph on the left above shows the distribution of wins vs non-wins depending on the difference between the max ratings for each team's goalkeeper. As the difference between the two teams' goalkeepers becomes more positive, that increases the frequency of the home team winning; the reverse is also true, as more negative values decrease the frequency of the home team winning. This plot supported our expectation of the goalkeeper being a very important factor to a team's success.

The graph on the right above shows the distribution of wins vs non-wins depending on the difference between the max ratings for each team's attack position group (this would mean taking the best individual attacking player). As the difference between the two teams' attackers becomes more positive, that increases the frequency of the home team winning; the reverse is also true, as more negative values decrease the frequency of the home team winning. Again, this did not come as a surprise as we assumed when a team's best attacker is better than the other team's best attacker, that they would win more; both because of their attacker's skill and great players tend to be on / make great teams.

Analytics Task: Prediction

Model 1

In the logistic regression model (log_model_v1) conducted on the Premier League dataset, the aim was to assess the influence of various FIFA player rating differentials on the likelihood of the home team winning a match. The selected variables compare the home and away position groups to each other (like in a game). These measures representing differences in attack, defense, midfield, and goalkeeping ratings were used to predict the binary outcome of home team victory (home_win_1_0). We also used the median home attack minus median away defense, trying to account for the home offense vs. away defense measure (see the full list of measures in model 1 summary in the appendix). The results reveal intriguing insights: certain rating differences exhibit significance in predicting match outcomes. For instance, a higher median rating in midfield significantly increases the odds of a home team winning, as indicated by a positive coefficient and a low p-value. Conversely, a significant negative impact on home team victory is associated with a higher variance in attack ratings. The model's overall fit, assessed through deviance and AIC values, indicates a reasonable fit to the data. These findings provide valuable information for understanding the impact of FIFA player ratings on Premier League match outcomes. A slight drawback of this model is leaving in the NA values for the "diff_median_rating_home_defense_away_attack" and "diff_max_rating_home_gk_away_attack variables

Model 2

Our second model (log_model_v2) was constructed by specifically focusing on key position group matchups between home and away teams, and this kept home minus away but changed up what position groups we were comparing. The model includes variables related to the difference in median ratings for home attack versus away defense, home defense versus away attack, maximum ratings for home attack versus away goalkeeper, maximum ratings for home goalkeeper versus away attack, and median ratings for the midfield. Notably, the model identifies that a higher difference in maximum ratings for home attack versus away goalkeeper positively influences the likelihood of a home team victory, with a p-value of 0.00427 suggesting statistical significance. This was an interesting subset of predictor variables to look at as it compared the position groups that typically go against each other in a match, for example, home attack vs. away defense, rather than home attack vs. away attack. Although this model slightly beats model 1 in terms of AIC, in reference to the confusion matrix it was slightly edged out by model 1.

Final Model

Our final model (log_model_v3) directly reflects our first model, however, it was fine-tuned to drop “diff_median_rating_home_defense_away_attack” (home defense minus away attack) and “diff_max_rating_home_gk_away_attack” (home goalkeeper minus away attack) as they both added “NA” values to the model, which was likely due to the covariance between those predictors and other predictors that were in the model as well. The exclusion of these variables changed our predictive accuracy (compared the cutoff of 0.6) very slightly, less than 1%. As expected, the final model kept the AIC unchanged due to the fact that these predictors' effects weren’t in the model, to begin with as their effects were NA; along with that, all p-values and significance values remained the same for the rest of the predictors from model 1.

Predictive Accuracy

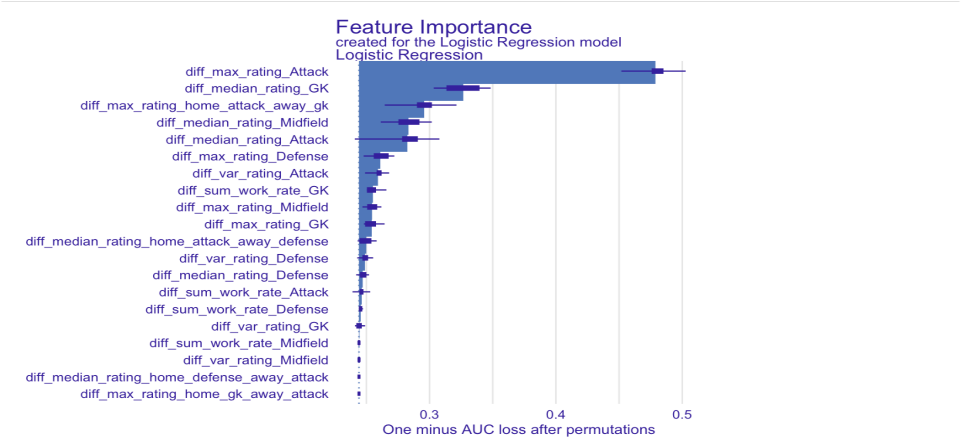
Initially, for the first two models, we aimed to assess basic predictive accuracy using a confusion matrix with a cutoff of 0.5. The confusion matrix output reveals an accuracy of 70.12%, indicating the model's reasonable ability to correctly classify match outcomes. Sensitivity, gauging the model's effectiveness in identifying true positives, stands at 66.67%, signifying a fair ability to detect home team wins. Specificity, measuring the capacity to identify true negatives correctly, is at 72.37%. The positive predictive value (PPV) is 61.11%, indicating that when the model predicts a home team win, it is correct 61.11% of the time. Overall, the model demonstrates promising predictive performance, capturing important nuances in Premier League match outcomes. The confusion matrix for the second model shows an accuracy of 68.13%, sensitivity of 63.64%, and specificity of 71.05%. This suggests the model was moderately successful in correctly identifying instances where the home team won. The positive predictive value was 58.88%, indicating the proportion of accurately predicted home team victories. While the model showed reasonable predictive capabilities, further refinement is needed to enhance

overall performance and reliability in forecasting Premier League match outcomes. The final model contained very similar measures to the first model due to it being a more refined version of the first model, but without the NA predictors; the combination of these factors made it the best fit of the data.

Due to the superior overall measurements of the final model, we decided to refine it to achieve the best possible accuracy. Testing the final model with different cutoffs revealed that a cutoff of 0.6 provides the highest accuracy at 71.31%, predicting the home team correctly winning 71.31% of the time. It also boasts the best specificity at 89.47%, indicating a few incorrectly picked wins for the home team. The PPV for this model increases from 61.11% to 72.88%, which marks a notable improvement. In the context of sports betting, particularly money-line betting, PPV and specificity are the most important. PPV would be the measure used if a person was an aggressive bettor and wanted to maximize their winnings; specificity is the opposite, as it would result in fewer incorrect bets, minimizing their losses. This combination of the high PPV and specificity present in the model makes this model ideal for sports betting, as it allows bettors to maximize their wins and minimize their losses.

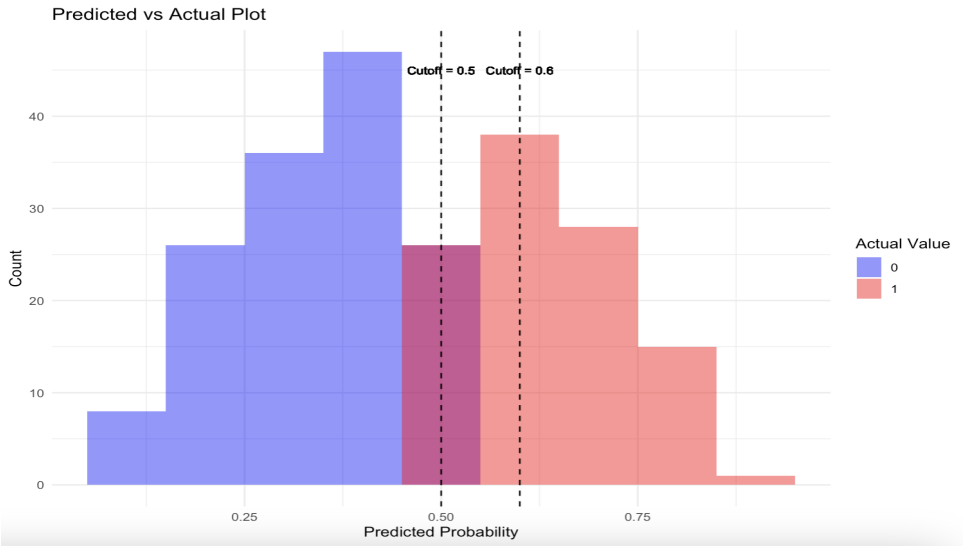
Once the ideal model and cutoff point were determined, we sought to identify the most important features influencing the outcomes of the various models. The output indicates that the difference between the max rating of attacking players between the home and away team is the most influential factor. Other important factors include the difference between median midfield ratings and the difference between starting goalkeeper ratings. The output underscores the importance of player ratings in Attack, Midfield, or Goalkeeper positions, as they consistently rank high in importance regardless of the measurement taken. Regarding specific categories, work rate was not a strong indicator of the model's predictive ability, as, aside from goalkeeper work rate, it did not significantly influence other position groups.

Data Visualizations



(Figure 1)

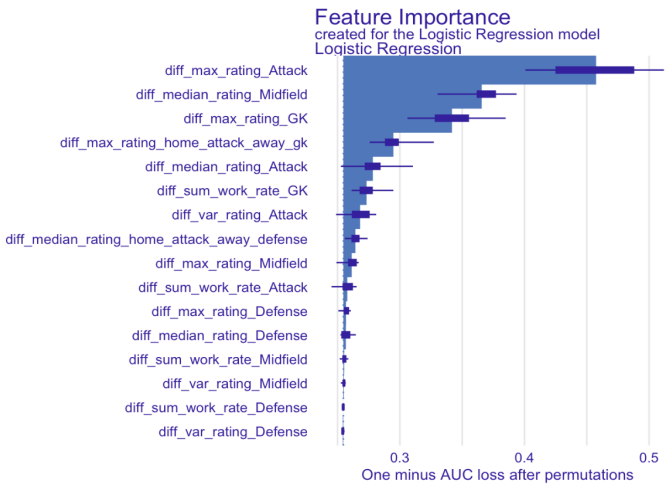
This graph shows the importance of each variable for the first model; as can be seen, the difference between the max attack ratings is the most important variable, with the difference between the median midfield rating and the difference between the max rating for goalkeepers being second and third respectively.



(Figure 2)

This graph shows the distribution of how frequently a positive or negative classification is chosen along with misclassification at a 0.5 cut-off. This plot does a fantastic job of showing the benefit of moving the cutoff to 0.6. The red represents the true positive class (win or 1), while the blue represents the true negative class (loss/tie or 0), and the overlap to the left of 0.5 represents the false negative predictions and the overlap to the right of 0.5 represents the false positive predictions. It is easy to visualize that as we moved the cutoff (visually the dotted line) to the right to 0.6, we would miss out on a few actual wins (red shade) but we would also correctly classify more losses or ties (blue shade).

Final Model Variable importance



(Figure 3)

Graph reflects the same insights and feature importance of the first model *(Figure 1)*, excluding "diff_median_rating_home_defense_away_attack" and "diff_max_rating_home_gk_away_attack" to reflect our final model.

Conclusion and Findings

Finding out these specific measurements is crucial, as they represent different scenarios when betting. Specificity primarily highlights false positives, which in this case would mean people incorrectly predicting a certain team to win their match. Higher values indicate fewer incorrectly predicted wins, which would make the version of the model with the 0.6 cutoff more desirable. PPV as previously indicated means how accurate predictions of the home team winning are, which would make the 0.6 cutoff once again more desirable due to the increase in that value. In the context of sports betting, the measurement that should be paid attention to depends on the type of bettor, which is one that tries to maximize winnings and one that tries to minimize losses. If a bettor is trying to maximize their win total, they will look at PPV and primarily use the Final Model with a 0.6 cutoff, as it has the highest measurement of correctly predicting wins. If a bettor is trying to minimize their losses, they will look at specificity and also use the Final Model with a 0.6 cutoff due to that having the highest value.

The model combined with the breakdowns to determine which factors are the most important show that when choosing to bet on Premier League matches using FIFA data, the differences between attack, midfield, and goalkeeper ratings are the primary ones to watch, while work rate can be primarily ignored. While FIFA ratings are not real life and are subject to change by game developers, they can certainly be a useful tool and proxy variable for figuring out how good teams can be if they are used creatively.

References

Dataset 1 “FIFA 20 complete player dataset.”
Source: https://www.kaggle.com/datasets/stefanoleone992/fifa-20-complete-player-dataset

Dataset 2: “Premier League matches 2010-2021”
Source: https://www.kaggle.com/datasets/pablohfreitas/all-premier-league-matches-20102021

Appendix:

Summary of Model 1:

```
##
## Call:
## glm(formula = home_win_1_0 ~ ., family = binomial(link = "logit"),
##     data = premier_league_results_merged_fifa_rating_summary_stats[,
##         c("home_win_1_0", colnames_holder_filt)])
##
## Coefficients: (2 not defined because of singularities)
##                                          Estimate Std. Error z value
## (Intercept)                             -0.230548   0.200547  -1.150
## diff_max_rating_Defense                  0.006635   0.108690   0.061
## diff_max_rating_GK                       0.062816   0.035631   1.763
## diff_max_rating_Midfield                -0.047616   0.038311  -1.243
## diff_max_rating_Attack                   0.092200   0.081020   1.138
## diff_median_rating_Defense              -0.036475   0.110795  -0.329
## diff_median_rating_Midfield              0.091401   0.043416   2.105
## diff_median_rating_Attack               -0.106982   0.080096  -1.336
## diff_var_rating_Defense                 -0.003235   0.010343  -0.313
## diff_var_rating_Midfield                 0.001344   0.007897   0.170
## diff_var_rating_Attack                  -0.045337   0.020061  -2.260
## diff_sum_work_rate_Defense               0.001802   0.076106   0.024
## diff_sum_work_rate_GK                    0.057999   0.023069   2.514
## diff_sum_work_rate_Midfield              0.021932   0.065295   0.336
## diff_sum_work_rate_Attack                0.077553   0.081822   0.948
## diff_median_rating_home_attack_away_defense  0.010364   0.032316   0.321
## diff_median_rating_home_defense_away_attack        NA         NA      NA
## diff_max_rating_home_attack_away_gk      0.032468   0.042087   0.771
## diff_max_rating_home_gk_away_attack            NA         NA      NA
##                                         Pr(>|z|)
## (Intercept)                               0.2503
## diff_max_rating_Defense                   0.9513
## diff_max_rating_GK                        0.0779 .
## diff_max_rating_Midfield                  0.2139
## diff_max_rating_Attack                    0.2551
## diff_median_rating_Defense                0.7420
## diff_median_rating_Midfield               0.0353 *
## diff_median_rating_Attack                 0.1817
## diff_var_rating_Defense                   0.7545
## diff_var_rating_Midfield                  0.8648
## diff_var_rating_Attack                    0.0238 *
## diff_sum_work_rate_Defense                0.9811
## diff_sum_work_rate_GK                     0.0119 *
## diff_sum_work_rate_Midfield               0.7370
## diff_sum_work_rate_Attack                 0.3432
## diff_median_rating_home_attack_away_defense   0.7484
## diff_median_rating_home_defense_away_attack       NA
## diff_max_rating_home_attack_away_gk       0.4404
## diff_max_rating_home_gk_away_attack           NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1376.5  on 1001  degrees of freedom
## Residual deviance: 1220.7  on  985  degrees of freedom
## AIC: 1254.7
##
## Number of Fisher Scoring iterations: 4
```

Confusion Matrix for Model 1:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 110  33
##          1  42  66
##
##                Accuracy : 0.7012
##                  95% CI : (0.6404, 0.7571)
##     No Information Rate : 0.6056
##     P-Value [Acc > NIR] : 0.001025
##
##                   Kappa : 0.3843
##
##  Mcnemar's Test P-Value : 0.355611
##
##             Sensitivity : 0.6667
##             Specificity : 0.7237
##          Pos Pred Value : 0.6111
##          Neg Pred Value : 0.7692
##              Prevalence : 0.3944
##          Detection Rate : 0.2629
##    Detection Prevalence : 0.4303
##       Balanced Accuracy : 0.6952
##
##        'Positive' Class : 1
##
```

Summary of Model 2:

```
##
## Call:
## glm(formula = home_win_1_0 ~ diff_median_rating_home_attack_away_defense +
##     diff_median_rating_home_defense_away_attack + diff_max_rating_home_attack_away_gk +
##     diff_max_rating_home_gk_away_attack + diff_median_rating_Midfield,
##     family = binomial(link = "logit"), data = premier_league_results_merged_fifa_rating_summary_stats[,
##         c("home_win_1_0", colnames_holder_filt)])
##
## Coefficients:
##                                          Estimate Std. Error z value
## (Intercept)                             -0.22333    0.19866  -1.124
## diff_median_rating_home_attack_away_defense -0.02088    0.02464  -0.847
## diff_median_rating_home_defense_away_attack -0.03094    0.02453  -1.261
## diff_max_rating_home_attack_away_gk      0.07511    0.02629   2.857
## diff_max_rating_home_gk_away_attack      0.04274    0.02630   1.625
## diff_median_rating_Midfield              0.05405    0.02315   2.335
##                                         Pr(>|z|)
## (Intercept)                              0.26094
## diff_median_rating_home_attack_away_defense  0.39673
## diff_median_rating_home_defense_away_attack  0.20719
## diff_max_rating_home_attack_away_gk          0.00427 **
## diff_max_rating_home_gk_away_attack          0.10417
## diff_median_rating_Midfield                  0.01954 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1376.5  on 1001  degrees of freedom
## Residual deviance: 1241.7  on  996  degrees of freedom
## AIC: 1253.7
##
## Number of Fisher Scoring iterations: 4
```

Confusion matrix for Model 2 with 0.5 cutoff point:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 108  36
##          1  44  63
##
##                Accuracy : 0.6813
##                  95% CI : (0.6197, 0.7385)
##     No Information Rate : 0.6056
##     P-Value [Acc > NIR] : 0.007858
##
##                   Kappa : 0.3421
##
##  Mcnemar's Test P-Value : 0.433848
##
##             Sensitivity : 0.6364
##             Specificity : 0.7105
##          Pos Pred Value : 0.5888
##          Neg Pred Value : 0.7500
##              Prevalence : 0.3944
##          Detection Rate : 0.2510
##    Detection Prevalence : 0.4263
##       Balanced Accuracy : 0.6734
##
##        'Positive' Class : 1
##
```

Confusion matrix for Model 1 with different test and training data:

```
# ran into a splitting error, make sure enough variances in each group
init_split <- initial_split(premier_league_results_merged_fifa_rating_summary_stats)

training_data <- training(init_split)

testing_data <- testing(init_split)


log_model_v1_test <- glm(home_win_1_0 ~ ., data = training_data[,c('home_win_1_0', colnames_holder_filt)], family
= binomial(link = 'logit'))

predictions_log_model_v1<- predict(log_model_v1_test, newdata = testing_data, type = 'response')

cut_off <- 0.5

# make a vector for win vs. loss, 1 win, 0 loss
win_loss_bool_v1 <- ifelse(predictions_log_model_v1 > cut_off, 1, 0) %>% unname() %>% as.factor()

confusionMatrix(win_loss_bool_v1, testing_data$home_win_1_0, positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 111  35
##          1  41  64
##
##                Accuracy : 0.6972
##                  95% CI : (0.6363, 0.7534)
##     No Information Rate : 0.6056
##     P-Value [Acc > NIR] : 0.001594
##
##                   Kappa : 0.3728
##
##  Mcnemar's Test P-Value : 0.566280
##
##             Sensitivity : 0.6465
##             Specificity : 0.7303
##          Pos Pred Value : 0.6095
##          Neg Pred Value : 0.7603
##              Prevalence : 0.3944
##          Detection Rate : 0.2550
##    Detection Prevalence : 0.4183
##       Balanced Accuracy : 0.6884
##
##        'Positive' Class : 1
##
```

Confusion Matrix for Model 1 with Different Test/Training Data, 0.6 Cutoff Point:

```
cut_off <- 0.6

# make a vector for win vs. loss, 1 win, 0 loss
win_loss_bool_v1 <- ifelse(predictions_log_model_v1 > cut_off, 1, 0) %>% unname() %>% as.factor()


confusionMatrix(win_loss_bool_v1, testing_data$home_win_1_0, positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 135  53
##          1  17  46
##
##                Accuracy : 0.7211
##                  95% CI : (0.6612, 0.7757)
##     No Information Rate : 0.6056
##     P-Value [Acc > NIR] : 0.00008563
##
##                   Kappa : 0.3767
##
##  Mcnemar's Test P-Value : 0.00002873
##
##             Sensitivity : 0.4646
##             Specificity : 0.8882
##          Pos Pred Value : 0.7302
##          Neg Pred Value : 0.7181
##              Prevalence : 0.3944
##          Detection Rate : 0.1833
##    Detection Prevalence : 0.2510
##       Balanced Accuracy : 0.6764
##
##        'Positive' Class : 1
##
```

Summary for Model 3:

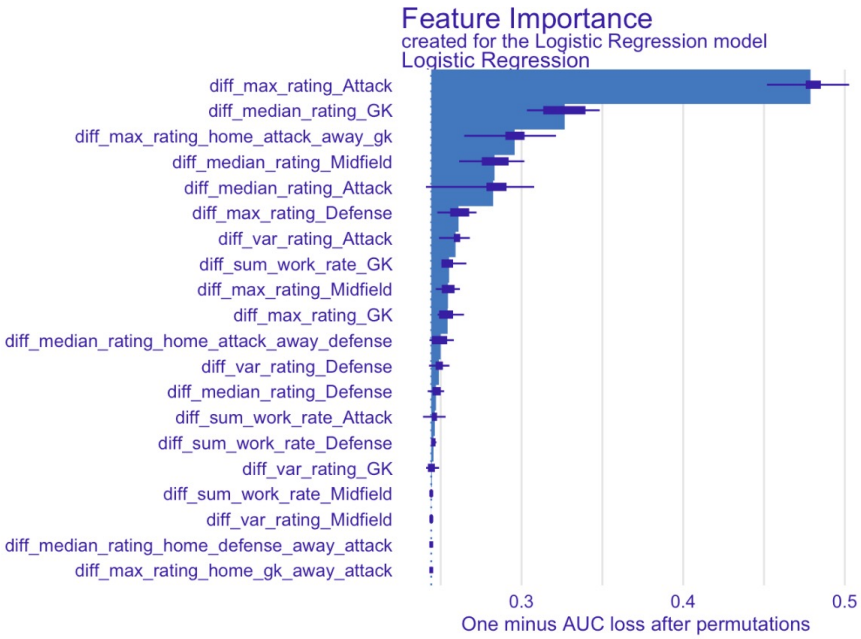
```
##
## Call:
## glm(formula = home_win_1_0 ~ ., family = binomial(link = "logit"),
##     data = premier_league_results_merged_fifa_rating_summary_stats[,
##         c("home_win_1_0", colnames_holder_filt)])
##
## Coefficients:
##                                          Estimate Std. Error z value
## (Intercept)                             -0.230548   0.200547  -1.150
## diff_max_rating_Defense                  0.006635   0.108690   0.061
## diff_max_rating_GK                       0.062816   0.035631   1.763
## diff_max_rating_Midfield                -0.047616   0.038311  -1.243
## diff_max_rating_Attack                   0.092200   0.081020   1.138
## diff_median_rating_Defense              -0.036475   0.110795  -0.329
## diff_median_rating_Midfield              0.091401   0.043416   2.105
## diff_median_rating_Attack               -0.106982   0.080096  -1.336
## diff_var_rating_Defense                 -0.003235   0.010343  -0.313
## diff_var_rating_Midfield                 0.001344   0.007897   0.170
## diff_var_rating_Attack                  -0.045337   0.020061  -2.260
## diff_sum_work_rate_Defense               0.001802   0.076106   0.024
## diff_sum_work_rate_GK                    0.057999   0.023069   2.514
## diff_sum_work_rate_Midfield              0.021932   0.065295   0.336
## diff_sum_work_rate_Attack                0.077553   0.081822   0.948
## diff_median_rating_home_attack_away_defense  0.010364   0.032316   0.321
## diff_max_rating_home_attack_away_gk      0.032468   0.042087   0.771
##                                         Pr(>|z|)
## (Intercept)                               0.2503
## diff_max_rating_Defense                   0.9513
## diff_max_rating_GK                        0.0779 .
## diff_max_rating_Midfield                  0.2139
## diff_max_rating_Attack                    0.2551
## diff_median_rating_Defense                0.7420
## diff_median_rating_Midfield               0.0353 *
## diff_median_rating_Attack                 0.1817
## diff_var_rating_Defense                   0.7545
## diff_var_rating_Midfield                  0.8648
## diff_var_rating_Attack                    0.0238 *
## diff_sum_work_rate_Defense                0.9811
## diff_sum_work_rate_GK                     0.0119 *
## diff_sum_work_rate_Midfield               0.7370
## diff_sum_work_rate_Attack                 0.3432
## diff_median_rating_home_attack_away_defense   0.7484
## diff_max_rating_home_attack_away_gk       0.4404
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1376.5  on 1001  degrees of freedom
## Residual deviance: 1220.7  on  985  degrees of freedom
## AIC: 1254.7
##
## Number of Fisher Scoring iterations: 4
```

Confusion Matrix for Model 3:

```
# ran into a splitting error, make sure enough variances in each group
init_split <- initial_split(premier_league_results_merged_fifa_rating_summary_stats)

training_data <- training(init_split)

testing_data <- testing(init_split)


log_model_v3_test <- glm(home_win_1_0 ~ ., data = training_data[,c('home_win_1_0', colnames_holder_filt)], family
= binomial(link = 'logit'))

predictions_log_model_v3<- predict(log_model_v3_test, newdata = testing_data, type = 'response')

cut_off <- 0.6

# make a vector for win vs. loss, 1 win, 0 loss
win_loss_bool_v3 <- ifelse(predictions_log_model_v3 > cut_off, 1, 0) %>% unname() %>% as.factor()


confusionMatrix(win_loss_bool_v3, testing_data$home_win_1_0, positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 136  56
##          1  16  43
##
##                Accuracy : 0.7131
##                  95% CI : (0.6529, 0.7683)
##     No Information Rate : 0.6056
##     P-Value [Acc > NIR] : 0.0002442
##
##                   Kappa : 0.354
##
##  Mcnemar's Test P-Value : 0.000004303
##
##             Sensitivity : 0.4343
##             Specificity : 0.8947
##          Pos Pred Value : 0.7288
##          Neg Pred Value : 0.7083
##              Prevalence : 0.3944
##          Detection Rate : 0.1713
##    Detection Prevalence : 0.2351
##       Balanced Accuracy : 0.6645
##
##        'Positive' Class : 1
##
```

Feature Importance for Model 1



Feature Importance for Model 3