Predictive Modeling Applied to the MLB Hall of Fame

Justin Carter

STAT 4893W

Introduction:

The goal of this research project is to create predictive models for the MLB (Major League Baseball) Hall of Fame. Specifically, we create one model to predict if a player will be in the Hall of Fame and another model to predict how long it will take a player to be inducted to the Hall of Fame after they retire given that we already know they are in it. The predictive models will be based on a variety of in-game statistics that are indicative of a player's performance. Some examples of these attributes include Homeruns, Hits, Games Played, Runs Batted In.

Predicting whether a player will be inducted into the MLB Hall of Fame is a complex problem as there are a variety of attributes that qualify one to be inducted. For example, there are players that show strong hitting attributes (lots of Homeruns, hits, etc.) that are inducted. Others are strong pitchers who may have poor hitting attributes but have a low earned run average, few hits against them, etc. There are also managers and other baseball personalities that are in the Hall of Fame, however, these inductees will be ignored for the purpose of this research. Because players are voted in by a committee of writers, the decisions are highly variable, and sometimes hard to quantify. Additionally, a player may be on the Hall of Fame ballot for many years before being inducted. Generally, players with stronger careers are inducted into the Hall of Fame more quickly. The primary goal of this research is to determine which attributes lead to a player's induction into the Hall of Fame. Similarity, another goal of the research is to determine whether these same attributes are associated with less time to Hall of Fame induction.

Methods:

The data used in this research was collected from Kaggle.com. There are 20 files in the dataset, including team records, salary information, and player records. In total, there are 101,332 seasons of hitting data, and 44,139 seasons of pitching data. In this research, we focus on the player records. There are four files within this database that we choose to utilize: fielding, pitching, hitting, and Hall of Fame records. Fielding records include 18 attributes, but we utilize errors, assists, and putouts. Pitching data includes 30 attributes, we utilize wins, losses, games, earned run average, strikeouts, opponent homeruns, walks, and walks + hits/innings pitched (WHIP). Hitting data includes 22 attributes, and we utilize games, runs, hits, extra base hits, homeruns, RBIs, stolen bases, strikeouts, putouts, assists, and errors. For the Hall of Fame file, we utilize the list of players to assign a HOF attribute to each player (a 1 if they are found in the HOF data, a 0 if they are not).

The advanced method used for this research is the Random Forest. This method is an ensemble method, meaning it combines other classifiers to create a majority decision. In this case, we build an ensemble of decision trees. To do this, we create n random samples from our original set of data. For each random sample that is selected, we then create a decision tree fit to this data. We choose d features to train each decision tree and select the feature split at each node that optimizes our objective function (gini, information gain). After training n trees, we classify new data by giving the data to each tree, obtaining the bootstrapped tree predictions, and classifying based on what the majority decision is from all the trees. For regression, we perform a similar method, choosing the feature split that minimizes RSS at each split instead of gini index. Instead of reaching a classification at the terminal node of each tree, we determine the average value for our response variable in a particular subregion created by the tree. For the Random Forest, it is important to optimize the parameters n and d. For n, the error or RSS decreases as n increases to a certain point, and then "even out". We want to choose an n right after this point, so we minimize error while keeping n from being too large (as n increases, so does the runtime of our algorithm). Additionally, we want to optimize the d parameter (number of features to be chosen from at each node). As d increases, the accuracy of each individual tree increases, however, correlation between trees increases. We wish to find a value d that minimizes testing error by maximizing accuracy while minimizing correlation between trees. Higher correlation between trees leads to reduced benefit of using an ensemble method, as many of the trees are too similar. As it relates to this research, we utilize the Random Forest to create our classification and regression models.

The simple methods used in this analysis are logistic and linear regression. These methods will be used to act as a basic model from which Random Forest hopefully will improve. The accuracies between these models will be compared.

Much data manipulation was required for the creation of the models in this research. The fielding and the batting data were combined and merged for each player. We will refer to this combined data as the position player data. Pitching data was handled separately as the statistics used to evaluate pitcher effectiveness is very different than the features found in the hitting and fielding data. Additionally, the Hall of Fame records were merged with the pitching and position player data to update our HOF factor described above. Data was presented on a season-to-season basis in the original set. We aggregate the data to sum the total of each of our features (except

WHIP and ERA as these are averages and should not be summed). The reason we aggregate the data is many seasons contain incomplete data due to injuries, league shutdowns, or other unpredictable factors. Therefore, building a model based on career statistics is more indicative of Hall of Fame induction than the averaged values of features on a season-to-season basis.

Results:

We tune our Random Forest on each type of model; pitcher classification, position player classification, pitcher HOF regression, and position player HOF regression. We use OOB error to evaluate the performance of our models. The plots are shown below:

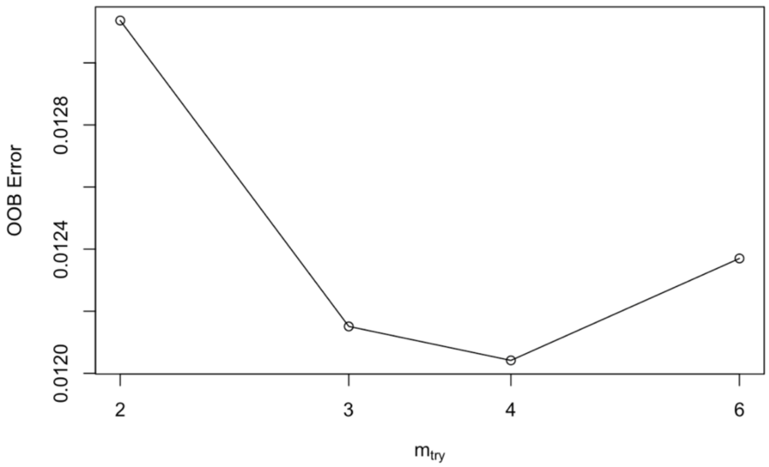


Figure 1, Position player classification

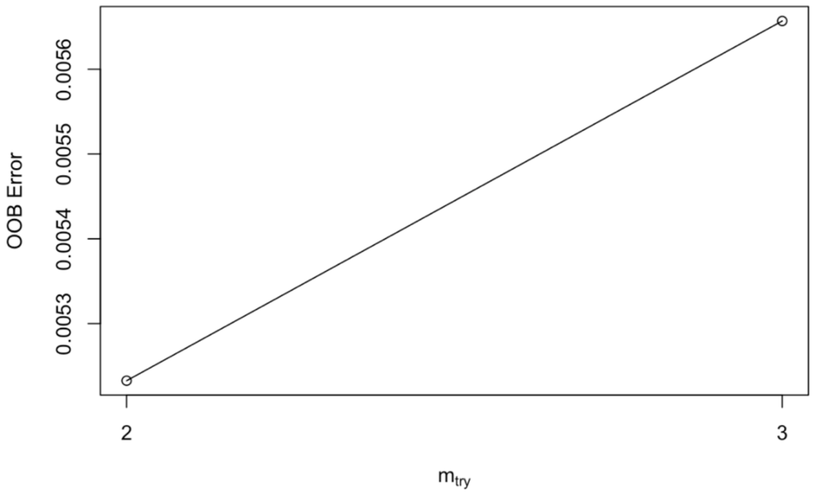


Figure 2, Pitcher classification

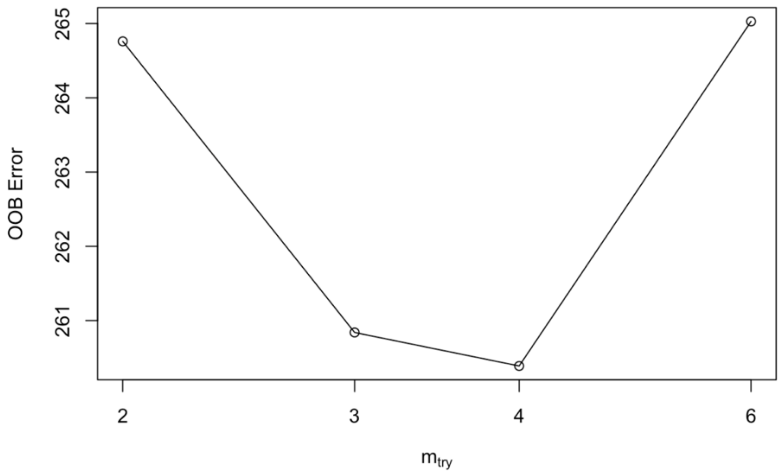


Figure 3, Position player regression

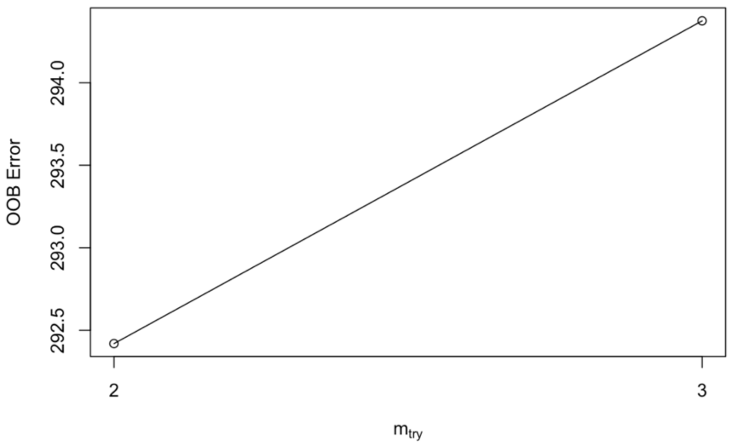
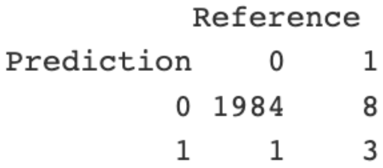


Figure 4, Pitcher regression

From these results, we choose mtry = 4 for both position player HOF regression and classification. We choose mtry = 2 for pitcher HOF regression and classification.

Additionally, we create plots of the error rate as ntree increases. The error converges around 100 trees for pitcher data and around 250 for position player data. Because the computational resources used to create each model are not too great, we keep ntree at 500 for all models, as there is no harm to model accuracy by having an ntree that is too large.

We run Random Forest on the combined batting and fielding data to obtain a model for prediction if a player has been inducted into the Hall of Fame. We first start by randomly splitting our data to into a training and testing set of data. The training set contains 75% of the data while the testing set contains the other 25%. We initialize the model with ntree = 500 and mtry = 4 (this represents our n and d values respectively). Based on the fitted model on the training data, the predicted class accuracy is 0.997 for the non-Hall-of-Fame class and 0.5267 for the Hall of Fame class. We then analyze the model using the testing set of data and report the results in table 1.



```
          Reference
Prediction    0    1
         0 1984    8
         1    1    3
```

Table 1

0 represents the non-Hall-of-Fame class and 1 represents the Hall of Fame class. Our prediction label indicates our model predictions of class, and the reference label indicates the true class. From the table, we can see that for our non-Hall-of-Fame class, we have an accuracy of 0.996. For our Hall of Fame class, we have an accuracy of 0.75. Overall, our accuracy is 0.996 (total accuracy of all predictions), sensitivity is 0.999 (True non-Hall-of-Fame / (True non-Hall-of-Fame + False Hall-of-Fame)), and specificity is 0.273 (True Hall-of-Fame / (True Hall-of-Fame + False Hall-of-Fame)). We can see that the ability to predict the positive class is much weaker than the negative class. There are also many more instances of non-Hall-of-Fame players, and this class has very high predictive accuracy.

We use a Random Forest on the combined batting and fielding data for Hall of Fame players only to predict the number of years it took them to be inducted. We randomly split our data to into a training and testing set of data. The training set contains 75% of the data while the testing set contains the other 25%. We initialize the model with ntree = 500 and mtry = 4. We obtain an explained variance of 40.17%. We report the results of the mean increase in gini index (an increase in gini index indicates increased homogeneity in the subgroups split by the given feature) in table 2.

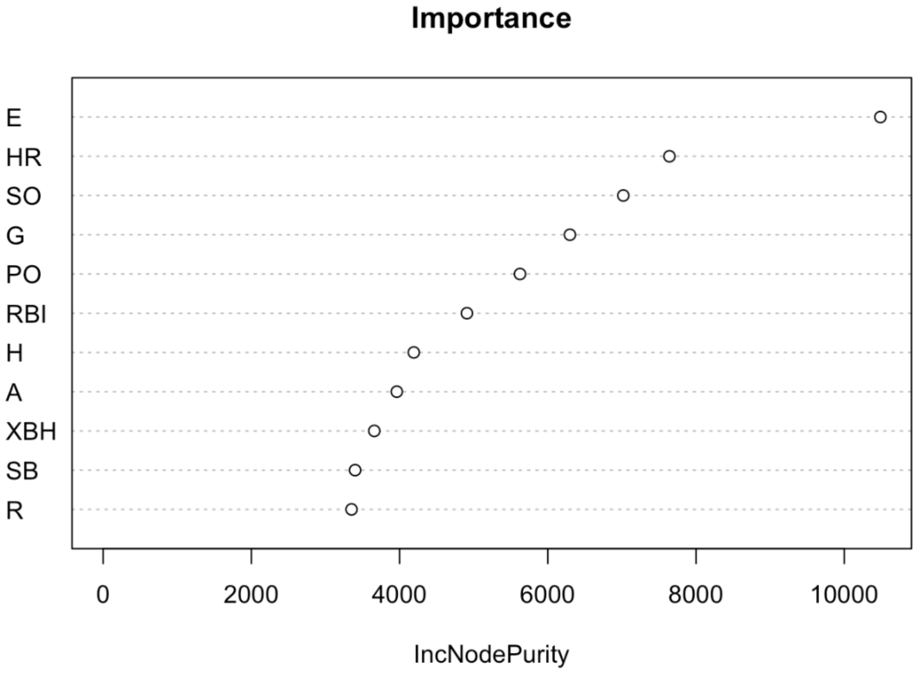
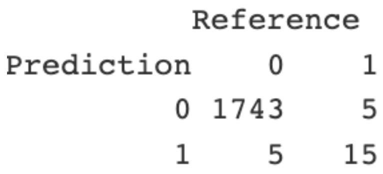


Figure 5

A larger increase in node purity indicates higher importance for the feature (a feature key is available in the appendix). As we can see from the plot, errors, homeruns, and strikeouts are the three most important features in our model. All the features increase node purity to some extent, indicating that they all have some usefulness in creating the predictive model.

We use Random Forest on the pitching data to obtain a model for prediction if a position player has been inducted into the Hall of Fame. We start by randomly splitting our data to into a training and testing set of data. The training set contains 75% of the data while the testing set contains the other 25%. We initialize the model with ntree = 500 and mtry = 2. The predicted accuracy using the training set of data is 0.9983 for the non-Hall-of-Fame class and 0.608 for the Hall of Fame class. We then analyze the model using the testing set of data and report the results in table 3.



```
          Reference
Prediction    0    1
         0 1743    5
         1    5   15
```

Table 3

Again, 0 represents the non-Hall-of-Fame class and 1 represents the Hall of Fame class. Our prediction label indicates our model predictions of class, and the reference label indicates the true class. From the table, we can see that for our non-Hall-of-Fame class, we have an accuracy of 0.997. For our Hall of Fame class, we have an accuracy of 0.75. Overall, our accuracy is 0.9943, sensitivity is 0.9971, and specificity is 0.75. This is a much higher specificity than for our position player class. This may be due to better features that are more indicative of Hall of Fame success than the features used in our position player data.

We use a Random Forest on the pitching data for Hall of Fame players only to predict the number of years it took them to be inducted. We randomly split our data to into a training and testing set of data. The training set contains 75% of the data while the testing set contains the other 25%. We initialize the model with ntree = 500 and mtry = 2. We obtain an explained variance of 38%. We report the results of the mean increase in gini index in table 4.

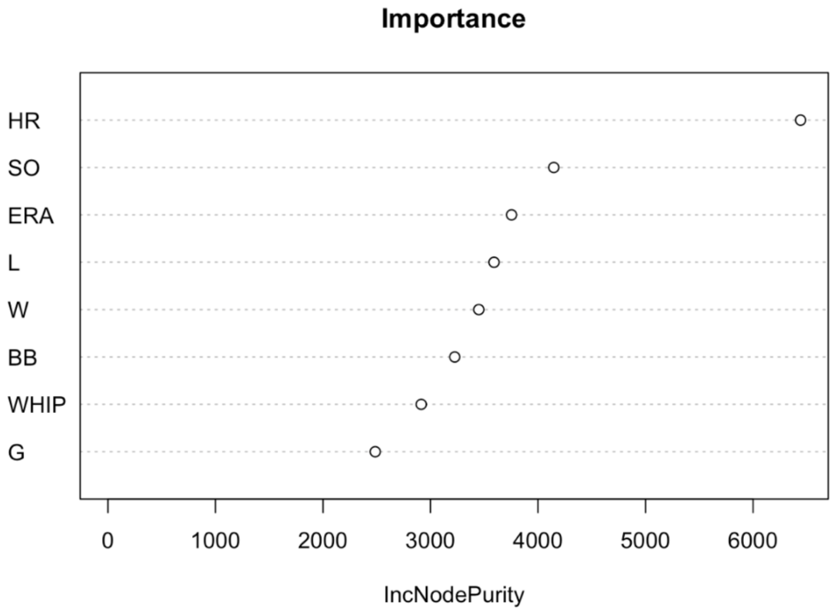
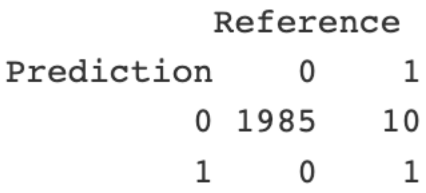


Figure 6

As we can see from table 4, the three most important features in our model are homeruns against, strikeouts, and ERA. Homeruns against is significantly higher in the average increase in node purity. Like the batting data, all the features used lead to increased node purity on average and are useful to the model as a result.

We use logistic regression on the combined batting and fielding data to obtain a model for prediction if a player has been inducted into the Hall of Fame.

We use our testing set of data to analyze our model in table 5.



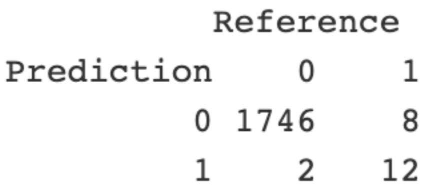
```
          Reference
Prediction    0    1
         0 1985   10
         1    0    1
```

Table 5

From the table, we can see that for our non-Hall-of-Fame class, we have an accuracy of 0.995. For our Hall of Fame class, we have an accuracy of 1. Overall, our accuracy is 0.995, sensitivity is 1, and specificity is 0.091. Specificity is much lower than for Random Forest, indicating issues in predicting the positive class in our logistic model.

We use linear regression on the combined batting and fielding data for Hall of Fame players only to predict the number of years it took them to be inducted. We obtain a proportion of variance explained equal to 44.55%. Our statistically significant features at a 0.05 level are errors and assists. This value is higher than for our random forest model, indicating a stronger model and potentially linear relationships between variables.

We use logistic regression on the pitching data to obtain a model for prediction if a player has been inducted into the Hall of Fame. We use our testing set of data to analyze our model in table 6.



```
          Reference
Prediction    0    1
         0 1746    8
         1    2   12
```

Table 6

From the table, we can see that for our non-Hall-of-Fame class, we have an accuracy of 0.995. For our Hall of Fame class, we have an accuracy of 0.857. Overall, our accuracy is 0.994, sensitivity is 0.9989, and specificity is 0.6. This model performs very similar to our Random Forest classifier, with slightly lower specificity indicating issues with predicting the positive class.

We use linear regression on the pitching data for Hall of Fame players only to predict the number of years it took them to be inducted. We obtain a proportion of variance explained equal to 54.99%. This value is significantly greater than our value obtained in our Random Forest model (38%). This indicates that linear regression may be better suited to model the regression problem than Random Forest, especially for pitchers. Our statistically significant features at a 0.05 level are losses and homeruns.

Discussion:

In total, we created eight models for our data, one of the following types for both position players and pitchers: random forest classification, random forest regression, logistic regression, and linear regression. Overall, for our classification problem, random forest performs better than logistic regression. It predicts each class (Hall of Fame and non-Hall-of-Fame) better than logistic regression for pitchers and position players. However, both models suffer from an imbalanced class problem. The accuracy scores for the non-Hall-of-Fame class are much higher than the Hall-of-Fame class. This is because there is far fewer data for the Hall-of-Fame class (hundreds of instances versus thousands). This means that for both methods, we have strong confidence for predicting a player that is not in the Hall of Fame. When predicting Hall of Fame players, the model is far less accurate. This presents problems as the Hall-of-Fame class is the more interesting of the two and is the goal of our predictions. However, for both logistic regression and random forest classification, the models created for pitchers were far more accurate than the models for position players. This could be for many reasons; we can infer that strong pitching performance has less variation amongst Hall of Fame players and is easier to quantify. In contrast, hitting and fielding data encompasses a wider range of skills, indicating more variation in each feature for our Hall of Fame class. In future studies, we wish to explore this further and test this inference.

For our regression models, linear regression has a higher proportion of explained variance for our data. For the position player data, we have Errors as both our most important factor and one of two statistically significant factors in our linear regression model. This indicates that the number of errors a player makes over their career is likely a very important factor for predicting the length of time it takes to be inducted into the Hall of Fame for position players. For the pitching data, our random forest model indicates that two of the top three most important factors include losses and homeruns. Again, we reach a similar conclusion from our linear regression model, where we have homeruns and losses as our two statistically significant factors. This indicates that the number of homeruns hit against a pitcher and the number of losses they have in their career is very important for determining the length of time waited before being inducted into the Hall of Fame.

There are a few issues with the data that may have impacted the accuracy of the modeling. One key issue is that the amount of time that a player must wait before being inducted into the Hall of Fame is currently 5 years after the end of their career. However, this was not always a rule, making wait times somewhat dependent on the era that they were inducted. Additionally, the data is strongly right skewed, as there are a few instances of players waiting over 100 years to be inducted into the Hall of Fame. This may have strong implications for the regression problem as this will skew the data towards higher number of years waited until induction. For the position player data, there are players that are inducted for historical significance rather than accolades alone. For example, some players may play most of their career in another league and have few seasons in the MLB. They may be inducted into the Hall of Fame without the typical feature values of other Hall of Fame players. This may impact the accuracy of the classification models.

Appendix A: Variable Names:

G - Games

R - Runs

XBH – Extra base hits

HR – Home Runs

RBI – Runs batted in

SB – Stolen Bases

SO – Strike outs

PO – Put outs

A - Assists

E – Errors

W – Wins

L – Losses

WHIP – Walks hits over innings pitched ((W+H)/IP)

ERA – Earned run average

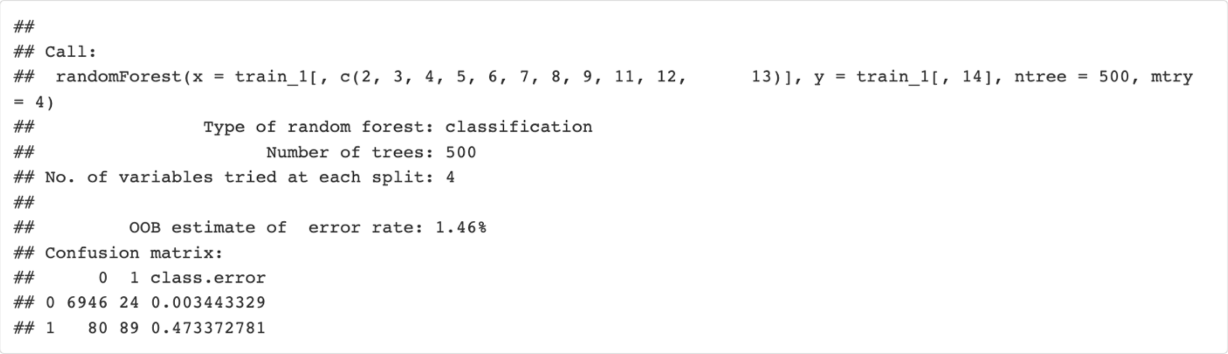
SO – Strikeouts

BB – Walks

HR – Homeruns

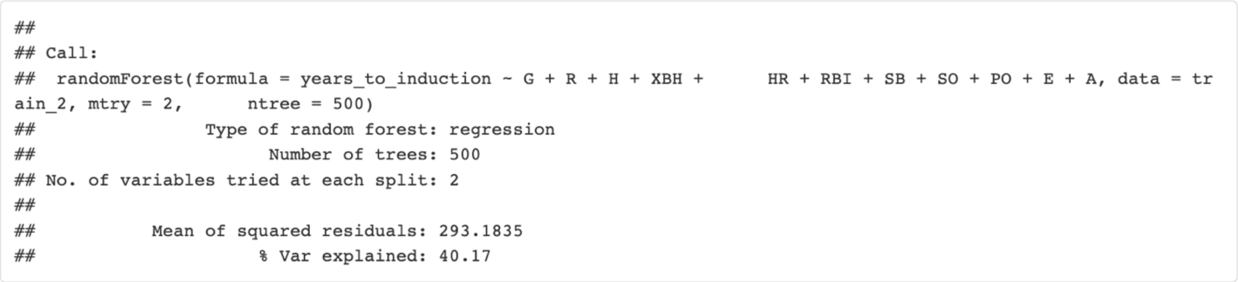
Appendix B: Model Outputs

RF Position Player Classification:



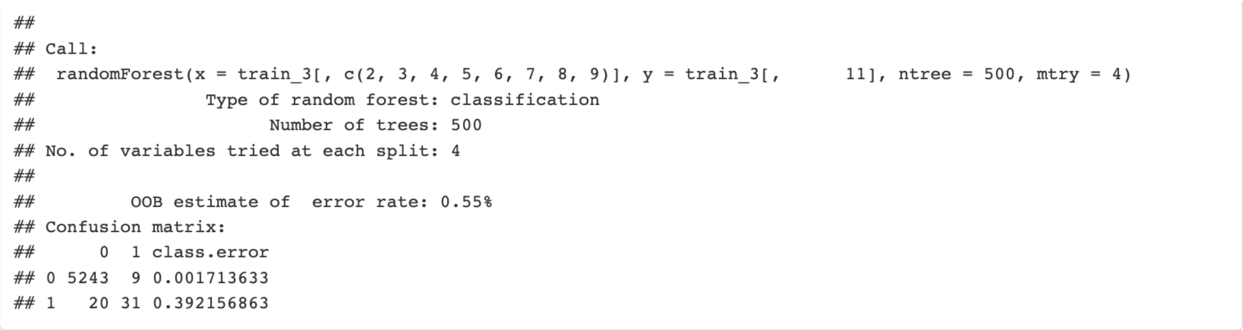
```
##
## Call:
##  randomForest(x = train_1[, c(2, 3, 4, 5, 6, 7, 8, 9, 11, 12,      13)], y = train_1[, 14], ntree = 500, mtry
= 4)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 1.46%
## Confusion matrix:
##      0  1 class.error
## 0 6946 24 0.003443329
## 1   80 89 0.473372781
```

RF HOF Position Player Regression:



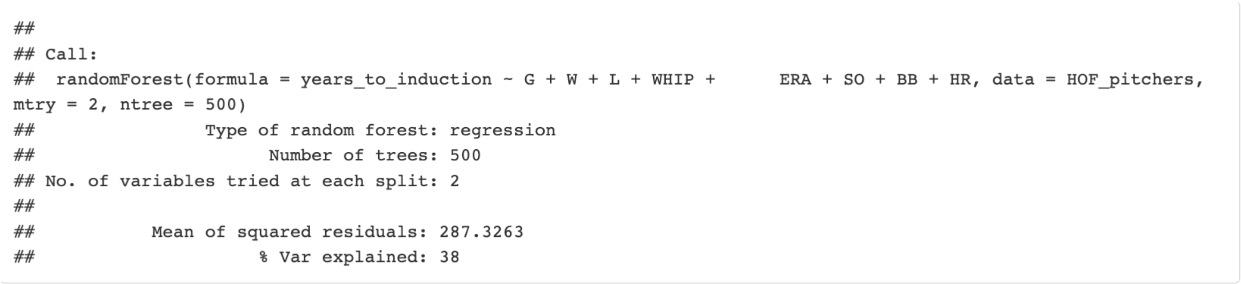
```
##
## Call:
##  randomForest(formula = years_to_induction ~ G + R + H + XBH +      HR + RBI + SB + SO + PO + E + A, data = tr
ain_2, mtry = 2,      ntree = 500)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 293.1835
##                     % Var explained: 40.17
```

RF Pitcher Classification:



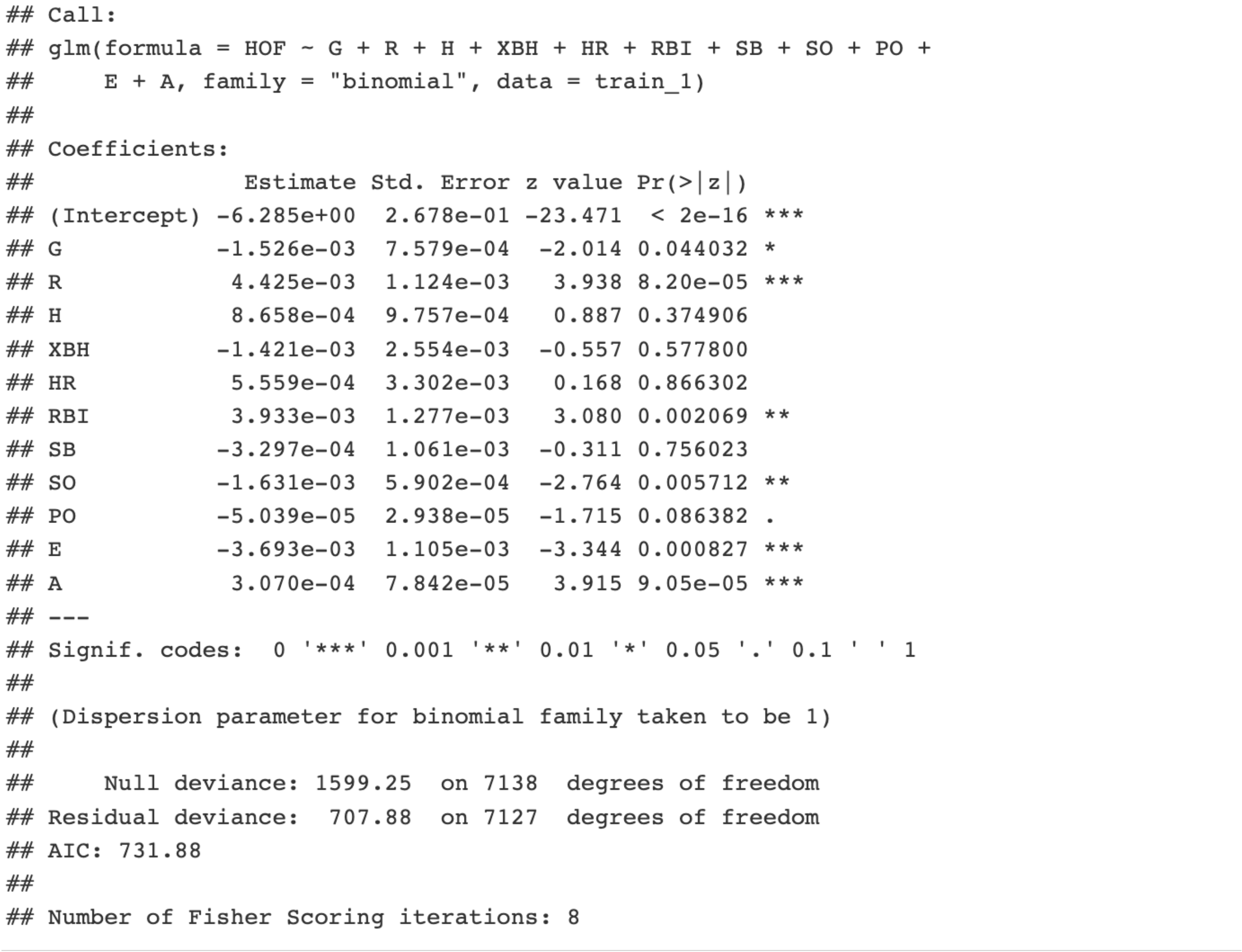
```
##
## Call:
##  randomForest(x = train_3[, c(2, 3, 4, 5, 6, 7, 8, 9)], y = train_3[,      11], ntree = 500, mtry = 4)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 0.55%
## Confusion matrix:
##      0  1 class.error
## 0 5243  9 0.001713633
## 1   20 31 0.392156863
```

RF HOF Pitcher Regression:



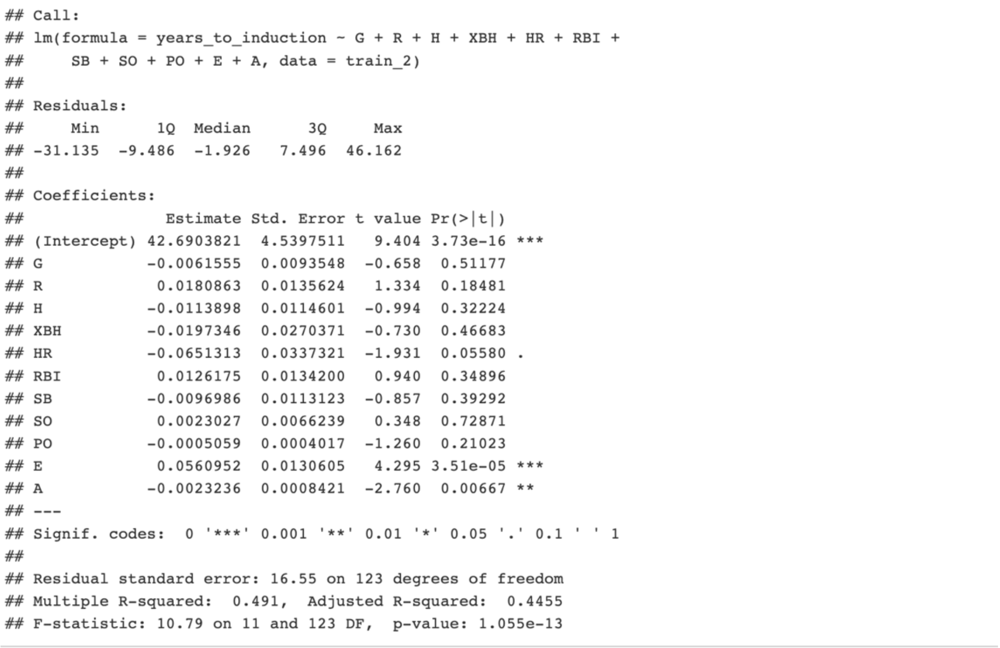
```
##
## Call:
##  randomForest(formula = years_to_induction ~ G + W + L + WHIP +      ERA + SO + BB + HR, data = HOF_pitchers,
mtry = 2, ntree = 500)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 287.3263
##                     % Var explained: 38
```

Logistic Regression Position Player Classification:



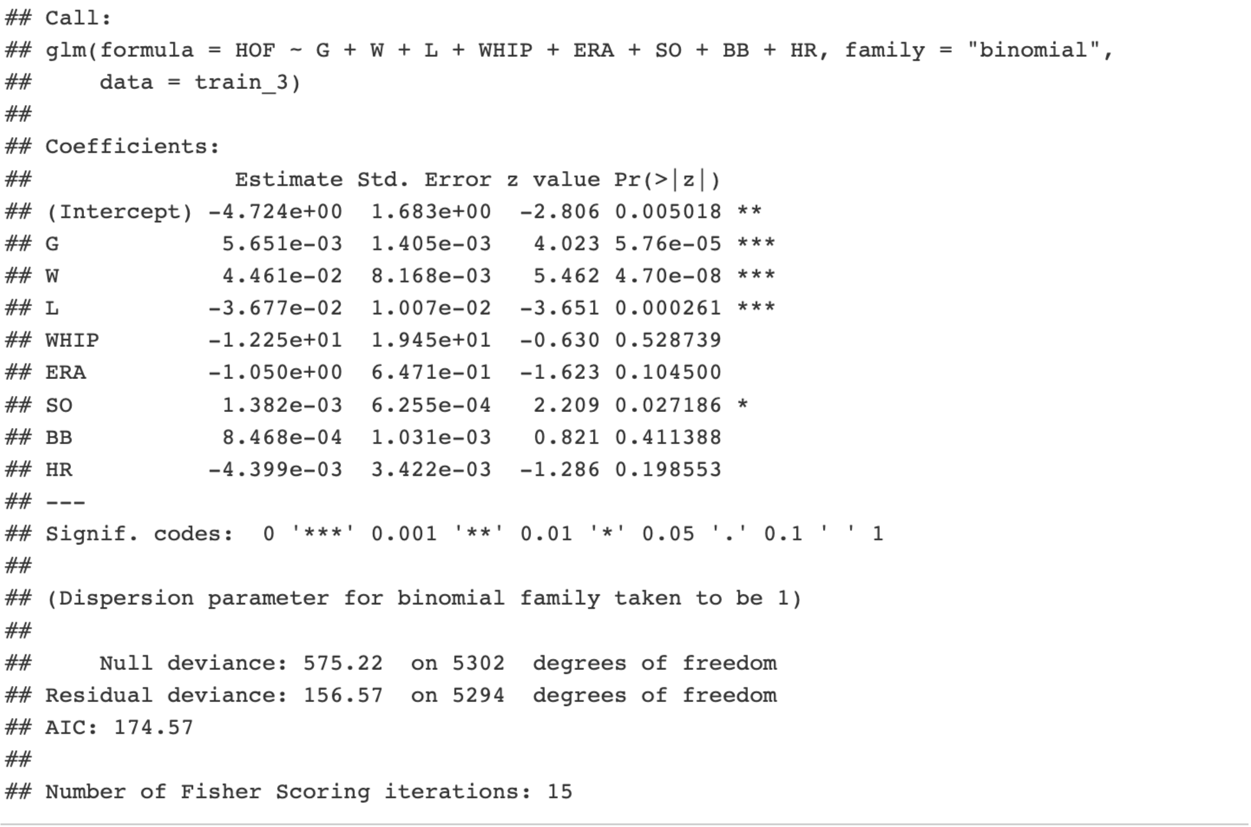
```
## Call:
## glm(formula = HOF ~ G + R + H + XBH + HR + RBI + SB + SO + PO +
##     E + A, family = "binomial", data = train_1)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.285e+00  2.678e-01 -23.471  < 2e-16 ***
## G           -1.526e-03  7.579e-04  -2.014 0.044032 *
## R            4.425e-03  1.124e-03   3.938 8.20e-05 ***
## H            8.658e-04  9.757e-04   0.887 0.374906
## XBH         -1.421e-03  2.554e-03  -0.557 0.577800
## HR           5.559e-04  3.302e-03   0.168 0.866302
## RBI          3.933e-03  1.277e-03   3.080 0.002069 **
## SB          -3.297e-04  1.061e-03  -0.311 0.756023
## SO          -1.631e-03  5.902e-04  -2.764 0.005712 **
## PO          -5.039e-05  2.938e-05  -1.715 0.086382 .
## E           -3.693e-03  1.105e-03  -3.344 0.000827 ***
## A            3.070e-04  7.842e-05   3.915 9.05e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1599.25  on 7138  degrees of freedom
## Residual deviance:  707.88  on 7127  degrees of freedom
## AIC: 731.88
##
## Number of Fisher Scoring iterations: 8
```

Linear Regression HOF Position Player:



```
## Call:
## lm(formula = years_to_induction ~ G + R + H + XBH + HR + RBI + 
##     SB + SO + PO + E + A, data = train_2)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -31.135  -9.486  -1.926   7.496  46.162 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 42.6903821  4.5397511   9.404 3.73e-16 ***
## G           -0.0061555  0.0093548  -0.658  0.51177    
## R            0.0180863  0.0135624   1.334  0.18481    
## H           -0.0113898  0.0114601  -0.994  0.32224    
## XBH         -0.0197346  0.0270371  -0.730  0.46683    
## HR          -0.0651313  0.0337321  -1.931  0.05580 .  
## RBI          0.0126175  0.0134200   0.940  0.34896    
## SB          -0.0096986  0.0113123  -0.857  0.39292    
## SO           0.0023027  0.0066239   0.348  0.72871    
## PO          -0.0005059  0.0004017  -1.260  0.21023    
## E            0.0560952  0.0130605   4.295 3.51e-05 ***
## A           -0.0023236  0.0008421  -2.760  0.00667 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 16.55 on 123 degrees of freedom
## Multiple R-squared:  0.491,  Adjusted R-squared:  0.4455 
## F-statistic: 10.79 on 11 and 123 DF,  p-value: 1.055e-13
```

Logistic Regression Pitcher Classification:



```
## Call:
## glm(formula = HOF ~ G + W + L + WHIP + ERA + SO + BB + HR, family = "binomial", 
##     data = train_3)
## 
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -4.724e+00  1.683e+00  -2.806 0.005018 ** 
## G            5.651e-03  1.405e-03   4.023 5.76e-05 ***
## W            4.461e-02  8.168e-03   5.462 4.70e-08 ***
## L           -3.677e-02  1.007e-02  -3.651 0.000261 ***
## WHIP        -1.225e+01  1.945e+01  -0.630 0.528739    
## ERA         -1.050e+00  6.471e-01  -1.623 0.104500    
## SO           1.382e-03  6.255e-04   2.209 0.027186 *  
## BB           8.468e-04  1.031e-03   0.821 0.411388    
## HR          -4.399e-03  3.422e-03  -1.286 0.198553    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 575.22  on 5302  degrees of freedom
## Residual deviance: 156.57  on 5294  degrees of freedom
## AIC: 174.57
## 
## Number of Fisher Scoring iterations: 15
```

Linear Regression HOF Pitcher:

```
## Call:
## lm(formula = years_to_induction ~ G + W + L + WHIP + ERA + SO +
##     BB + HR, data = train_4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.450  -7.331  -0.654   6.936  38.181
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.269e+01  6.632e+01   1.096 0.279014
## G           -2.980e-02  1.530e-02  -1.948 0.057850 .
## W           -9.125e-02  5.867e-02  -1.555 0.127042
## L            3.142e-01  8.327e-02   3.773 0.000478 ***
## WHIP        -1.064e+02  7.904e+02  -0.135 0.893575
## ERA         -9.852e+00  1.626e+01  -0.606 0.547792
## SO          -7.779e-03  6.477e-03  -1.201 0.236172
## BB           1.012e-02  1.560e-02   0.649 0.519770
## HR          -8.063e-02  3.547e-02  -2.273 0.027963 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.34 on 44 degrees of freedom
## Multiple R-squared:  0.5499, Adjusted R-squared:  0.4681
## F-statistic:  6.72 on 8 and 44 DF,  p-value: 1.004e-05
```

Appendix C: Rcodes

https://rpubs.com/cart0509/1170232

References:

GfG. (2020, June 5). Random Forest approach in R programming. GeeksforGeeks. https://www.geeksforgeeks.org/random-forest-approach-in-r-programming/

James, Gareth, et al. *An introduction to statistical learning: With applications in python*. Springer Nature, 2023.

King, Ed. (2020). Baseball Databank. Kaggle.com. https://www.kaggle.com/datasets/open-source-sports/baseball-databank/data?select=Fielding.csv

Pn, T., Steinbach, M., & Kumar, V. (2005). Introduction to data mining. Addison-Wesley.