Kevin Chonko

Predictive Modeling Final Project

An In-Depth Look at the History of Baseball

May 7, 2021

1. **Executive Summary**

As technology continues to advance and computers get more and more powerful, the applications of this computing power are virtually endless. One particular application that is becoming more prevalent in the modern day is machine learning. Machine learning is the process of developing algorithms to analyze data, draw inferences and establish relationships within the data, and use these inferences to solve problems. Machine learning is a very broad field, and there are a wide range of different applications that utilize machine learning to solve problems. One of these applications is predictive modeling. Predictive modeling is defined as a process that predicts future events or outcomes by analyzing patterns that can help to forecast future results. Over the past several weeks, many predictive modeling techniques have been learned and applied to a variety of data sets to analyze and predict outcomes within the data. After learning all of these techniques, the next step was to utilize these techniques to analyze a large data set, and answer a series of questions. For this process, my analysis was focused on the Lahman Baseball Database, created by baseball author/reporter/journalist Sean Lahman. This database contains extensive information on every player that has ever appeared in a professional baseball game in the United States, from 1870 to 2020. The original data set was contained within a series of CSV files that were downloaded from the internet. For this analysis, I wanted to focus on players that appeared during the “Live Ball Era”, which started in the 1920 season. Before the 1920 season, baseballs were often sloppily made, and did not fly nearly as far as modern baseballs. This era of Major League Baseball (MLB) was dominated by pitchers, and including data from the “Dead Ball Era” would greatly skew the analysis. For players that debuted during the Dead Ball Era whose careers extended into the Live Ball Era, any seasons that they played before the 1920 season were not considered in this analysis (for example, Babe Ruth played from 1914 to 1935. However, his 1914-1919 seasons were neglected within this analysis, and only his statistics from 1920-1935 were included). This data set was used to analyze several different questions. Namely, can Hall of Fame induction be accurately predicted based on players’ statistics, and can player salaries be accurately determined based on their season-by-season statistics?

1. **Data and Approach**

As previously mentioned, the data set being analyzed for this Predictive Modeling application is the Lahman Baseball Database. This database contains statistics for every player to ever appear in a professional baseball game in the USA, from Alexander Cartwright (born in 1820) to Luis Garcia (born in 2000). Before this data set could be adequately analyzed, extensive data manipulation and engineering had to be conducted. The original data set included players from several other, now-defunct leagues outside of the MLB, such as the American Association, Federal League, Union Association, etc. The focus of this investigation was strictly on MLB players, so all players that appeared in these other professional leagues were removed from the data set. Major League Baseball historians often divide the league into two separate eras: the “Dead Ball Era” and “Live Ball Era”. The Dead Ball Era existed from 1869, the first year of the MLB, until 1919, and was noted for being dominated by pitchers due to sloppily made, yarn-centered baseballs that easily turned mushy and “dead” after extensive use, hence the name of the era. After extensive complaints from fans and players alike, at the start of the 1920 season the MLB replaced the old baseballs with a newer, livelier cork-centered baseball to boost offense leaguewide. The Live Ball Era was then born, and still exists today in the modern game. This analysis only considered players that played in the Live Ball Era, therefore any players that were born before 1894 were deleted from the original data set, as a player born in 1895 or later would be debuting right around the beginning of the Live Ball Era.

All 21 original Lahman Baseball Database (Lahman) files were downloaded from seanlahman.com and read into Python to begin the data manipulation process. I used Python to manipulate the data instead of R strictly because I prefer Python’s data manipulation capabilities over those of R. The only Lahman files considered for this analysis were files pertaining to individual player statistics; any Lahman files related to team stats, college info, ballpark dimensions, etc. were deleted. Also, any postseason stats or fielding stats were disregarded. After deleting these files, I was left with 13 separate data frames to analyze. Any variables within these data frames deemed by me to be irrelevant to the analysis were deleted in Python. The data frames were cleaned, manipulated as necessary, and then uploaded into R for analysis. After loading the data frames into R, several different data frames were joined or merged for easier analysis. For example, the data frames “ASG” and “CareerAwards”, which contained info about the number of All-Star Games per player and the awards won by each player, respectively, were merged into a new data frame, “ASGAwards”. Two separate master data frames were created via the joining of several different data frames. The first master data frame, “MasterData”, contained the career batting and pitching statistics, awards won, All-Star Games played, Hall of Fame voting, and salary info for every MLB player since the start of the Live Ball Era. The second master data frame, “HOFMaster”, contained the same info as the MasterData frame, but only for the players that have ever been eligible for Hall of Fame consideration. The MLB Hall of Fame has two main criteria that players must meet to be placed on the ballot: players must have played in at least 10 MLB seasons in their career, and players must be retired for five full seasons before being eligible for the ballot. Due to these criteria, the number of players considered eligible for the Hall is quite small. Of the 15,427 players included in the MasterData data frame, only 1,203 players (7.80%) ever gained Hall of Fame eligibility. Creating the HOFMaster data frame presented some challenges that had not originally been considered. For example, according to MLB Hall of Fame rules, a player that garners 75% of the votes in a given year earns induction into the Hall of Fame. However, if a player garners more than 5% , but less than 75%, of votes in a given year, that player will be included on the ballot for the next year. A player can be included on the ballot for a maximum of ten years. Due to these rules, there were many repeated instances of players in the Hall of Fame Lahman file. To avoid including the same player more than once in the data frame, the result from their last year on the ballot was the only one considered (for example, Trevor Hoffman did not get inducted his first two years on the ballot, but gained induction in his third year of eligibility. The result from his third and final year on the ballot was the only one considered).

After creating the MasterData and HOFMaster files, further data manipulation had to be conducted. All data frames had been joined and merged by each player’s playerID. However, playerID was a unique variable that was different for each player, and this variable had to be removed before analysis could be conducted. After removal of the playerID variable, the HallOfFame data frame was ready to be analyzed by predictive models. A subset of the MasterData data frame was then created that included season-by-season statistics for every player from 1920 onward. This file was then merged with player salary info to create the new data frame SeasonStats. The SeasonStats data frame was the only data frame that any machine learning models to predict player salaries were trained on.

The first analysis conducted on the data was a linear regression model to predict player salaries. First, a correlation matrix was created to see which variables were most correlated with salary. According to the matrix, the five variables with the strongest correlation were yearID, HR, RBI, BB, and X2B. A linear regression model was then created to analyze the blind SeasonStats data set, with salary as the dependent variable and the five aforementioned variables as the independent variables.

Next, a logistic regression model was created to predict Hall of Fame induction results for past players. A correlation matrix was once again created on the Hall of Fame dataset to see which variables had the strongest correlation with the target variable, induction. The five variables in this data frame with the strongest correlation to the target variable were ASG, TotalAwards, BBMagAllStar, X3B, and RegMVP. This makes sense, as players nominated for many All-Star Games and winning many awards will likely have stronger Hall of Fame credentials. A logistic regression model was then created to analyze the blind Hall of Fame data set. In this model, induction was the dependent variable, and the five variables with the strongest correlation coefficients were the independent variables. Training and testing subsets were created from the original file so the regression model could train itself on the first subset to draw inferences and establish relationships in the data, and apply these inferences to the second subset. A 3:1 train/test ratio was utilized, and the model was then trained and fit on the data. Once the regression model was created, a confusion matrix was created to review the results. The confusion matrix is a useful tool that gives a detailed breakdown of how the model predicted the target variable compared to the actual results. A sample confusion matrix is shown below.

HOFpred2 0 1

0 214 26

1. 5 21

The accuracy and error rate of the model can be interpreted easily from the confusion matrix. In the sample matrix above, the total accuracy of the model can be calculated to be 235/266, or 88.3%, since the model correctly predicted 214 players that didn’t make the Hall of Fame as well as 21 players that did make the Hall of Fame, out of the 266 players contained in the test set. Since the accuracy was 88.3%, the error rate of the model is 11.7%.

Another classification model was created next, this time a quadratic discriminant analysis (QDA) model to once again analyze Hall of Fame induction results. Just as was done for the logistic regression model, the five variables with the highest correlation coefficients relative to the target variable were used as the independent variables. The QDA model was trained on the Hall of Fame training set and then tested on the testing set, and a confusion matrix was once again utilized to interpret the results.

A k-nearest neighbors model was the final classification model to be created to analyze Hall of Fame induction. Before creating the model, all quantitative variables in the Hall of Fame dataset had to be normalized to ensure accurate results. Normalizing the quantitative values is imperative for a k-nearest neighbors model because the nearness of data points is calculated simply based on Euclidean distance from the closest central cluster. If values weren’t normalized, variables greater in magnitude would disproportionately skew these Euclidean distances compared to variables which are lesser in magnitude. Normalizing variables ensures that each variable affects the Euclidean distance equation equally. Many different k values were tried for this model, but the best results occurred when k equaled 2. After training and testing the model, a confusion matrix was used and results were interpreted.

Several different resampling approaches were the next models to be built and applied to the data. First, a Leave One Out Cross Validation (LOOCV) approach was applied to the Season Stats data to predict salaries. A LOOCV approach splits the observations into two groups, but does not split the data into the traditional training and test sets. Instead, a single observation is used for the test set, and all other observations are included in the training set. This approach is often advantageous since the test error is only based on a single observation, which all but eliminates any bias within the test error. However, since only a single observation is used to make a prediction, LOOCV models are often highly variable and vastly different results can occur based on which observation is chosen for the test set. Also, if a data set is sufficiently large with many observations and variables , this model can be expensive and time consuming to run, since the model has to be fit on all but one observation in the data set. A LOOCV model was created and fit on the Season Stats data, but for this approach, only one independent variable (yearID) was included. A polynomial function was included in the model to see if results could be improved if the function was fit to a regression line of a higher power. The model was trained on the data and then analyzed.

A k-fold cross validation model is often used as an alternative to the LOOCV model. In this approach, the data set is randomly divided into a set number, *k,* of distinct groups, or folds. The first fold is used as the test set for this model, and the remaining folds are used to train the model. Mean squared error is calculated on the observations in the test set, and then the process is repeated *k* more times, each time using a different fold as the test set. The mean squared error for all folds is determined, and all mean squared errors are added and divided by *k* to determine the average mean squared error for the data set. K-fold cross validation is preferable to LOOCV for large data sets since the k-fold model only has to be fit on the data *k* times, whereas the LOOCV method must be fit *n-1* times. This both significantly improves computation time and reduces computational cost. A k-value of 10 was used to analyze the Season Stats data set to predict salaries. Once again, a polynomial function was fit on the single independent variable, yearID, to try and improve the R2 value of the model. This model was fit on the data and the error values were then analyzed.

The bootstrap method was then utilized on the Season Stats data set to predict salaries. The bootstrap method is very versatile and can be used for countless different predictive modeling methods to help improve accuracy. This method allows one to simulate the process of procuring new data samples to calculate variability within the model without actually gathering new samples. The unique data sets are created by sampling observations from the original data many different times. Bootstrapping is done with replacement, meaning observations can appear more than once in a specific data set. For this model, the bootstrap method was used to help create a new linear regression model on the Season Stats data. The five independent variables used in the first linear regression model were once again used in this model. A function was created to obtain the coefficients for the linear regression model, and then the standard error for each variable was calculated.

A principal components analysis (PCA) was the next model to be built and fit on the Season Stats data. PCA is a useful technique for taking a large data set with numerous variables and turning it into a data set with much fewer variables. The PCA model was applied to the SeasonStats data frame. After running the PCA model on the data, a validation plot was constructed to determine which model size is optimal for this particular problem. After determining the optimal model size, the mean squared error for the optimal PCA regression was calculated. Finally, the R2 value of the model could be found to explain the variability within the data.

Best subset selection, ridge regression, and the lasso method were three different linear model methods deployed on the Season Stats data. Best subset selection is done by fitting a least squares regression for every possible combination of independent variables. All models are then analyzed, and the best one is then selected. Ridge regression is a method of estimating the coefficients of multiple regression models where the variables have strong correlation. Ridge regression helps to limit the impact of collinearity between independent variables. The lasso method is a regression analysis method that performs variable selection and normalization to enhance linear regression models. All three of these linear regression techniques utilize different functions to help limit bias and reduce test error within the data set.

1. **Detailed Findings**

As previously stated, the first model created was a linear regression analysis to predict player salaries. The residuals vs. fitted values plot for this linear regression is shown below.

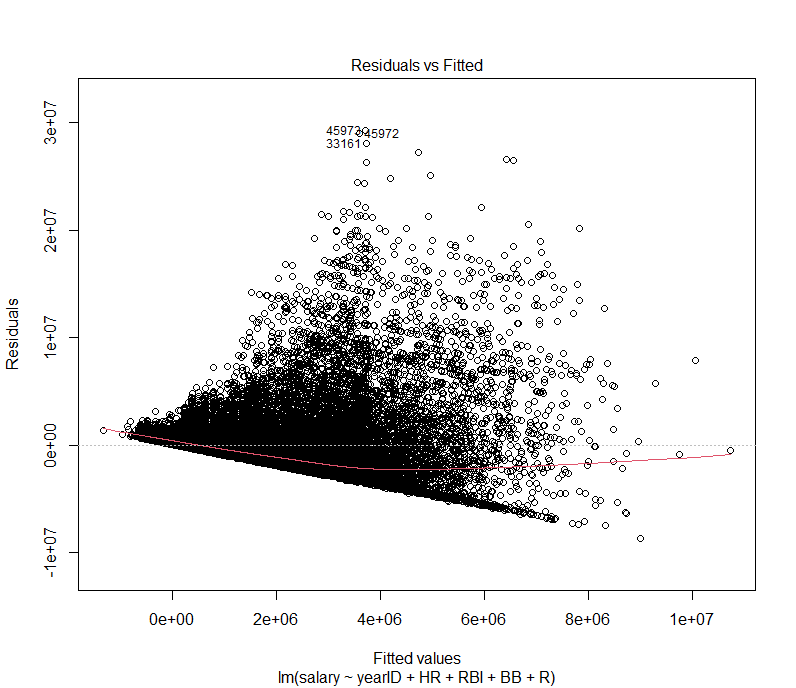


Figure 1: residuals vs fitted values plot for lin. regression

It is apparent that this linear regression model systematically underestimated the values of player salaries, since most of the dots on the plot lie above the regression line. The R2 value for this regression line is only 0.2042, indicating that the regression model explains little of the variability of the response values. However, the p-values of all five independent variables are well under 0.05, indicating that all five variables have a significant, non-zero impact on predicting player salaries.

Training and fitting the logistic regression model allowed for the construction of a confusion matrix to analyze the model’s accuracy. The confusion matrix is shown below.

HOFpred2 0 1

No 214 26

Yes 5 21

The model predicted 235 of 266 players correctly, or 88.3%. The model correctly predicted 214/219 (97.7%) of players that did not make the Hall of Fame. It correctly predicted 21/47 (44.7%) of players that made the Hall of Fame. The high accuracy for players that didn’t make the Hall is initially encouraging, but since a vast majority of players in the Hall of Fame data set did not earn Hall of Fame induction, these results can be a little misleading. Since the model sees many more “No” values in the target variable in the testing set as opposed to “Yes” values, this can lead to the model predicting “No” for future values at a disproportionate rate. Some techniques will be introduced to this model to help combat this bias.

The confusion matrices for the QDA and k-nearest neighbors analyses on the Hall of Fame data are shown below on the left and right, respectively.

qda1class 0 1 HOFknn 0 1

0 198 19 0 201 19

1 21 28 1 18 28

Analyzing the results from these two different models shows that the k-nearest neighbors model was just slightly better than the QDA model. The knn model predicted 229/266 (86.1%) of players correctly, including 201/219 (91.7%) of players that didn’t make it, and 28/47 (59.6%) of players that did make it. Both of these models are a marked improvement over the logistic regression model in terms of accuracy of players that made the Hall of Fame. The accuracy of players that didn’t make the Hall decreased slightly, but the knn and QDA models were able to increase accuracy of predicting players that did gain induction by almost 15%. The knn model is therefore the preferred method for predicting Hall of Fame induction so far.

After constructing the LOOCV model, problems arose when running the model. Runtime was super slow at just over 28 minutes, and the cross-validated error values were upwards of a billion dollars for all powers of the polynomial function. This model is obviously not a good approach for this data set, and was not considered any further. The k-fold cross validation model had a much quicker runtime than the LOOCV model, but the cross-validated error values were still in the billions of dollars, so this model was also disregarded.

The bootstrap method was able to calculate the intercepts for every variable in the linear regression model, as well as the standard error values for the variables. The results from the bootstrap method are shown below.

(Intercept) yearID HR RBI R

-2.747170e+08 1.381014e+05 8.723582e+04 1.008082e+04 -4.366789e+03

Bootstrap Statistics :

original bias std. error

t1\* -2.747170e+08 -462324.8072 4116104.807

t2\* 1.381014e+05 230.9212 2062.147

t3\* 8.723582e+04 -109.2804 7472.061

t4\* 1.008082e+04 -447.6888 3101.347

t5\* -4.366789e+03 546.8894 2093.107

Salary had the highest standard error by far, which is to be expected, but the standard errors for yearID, HR, RBI, and R all being in the thousands is very good considering the large range of salaries being dealt with in this analysis.

Building the PCA model allowed a validation plot to be constructed that shows the mean squared error of the data as a function of the number of components considered in the analysis. This validation plot is included below.

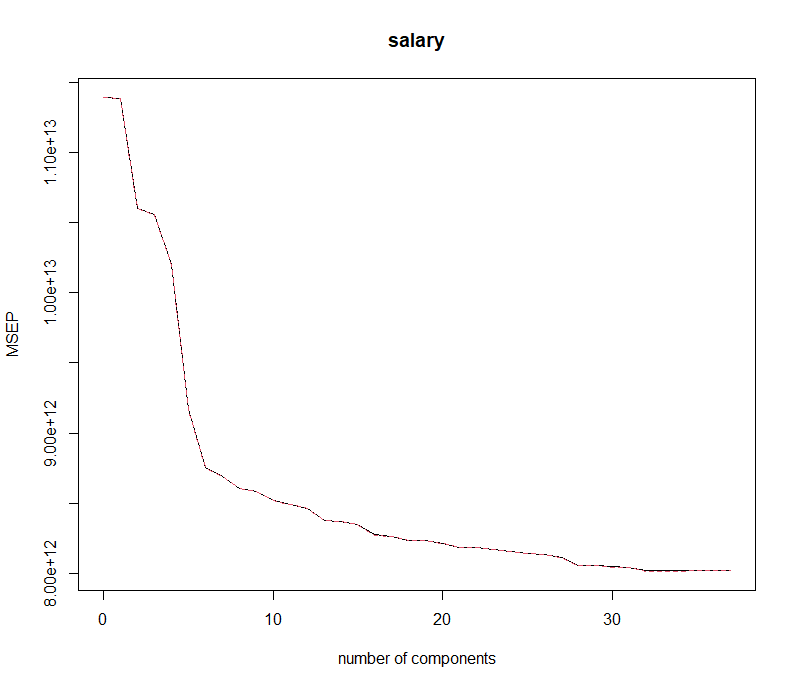


Figure 2: validation plot of PCA model

It was determined from this validation plot that the optimal model occurs when the number of components included, *m*, equals 20. The mean squared error may be lower for models where *m* is greater than 20, but only marginally. Therefore, a model with 20 components is sufficient for this analysis. The mean squared error when *m* = 20 is still over 8.0 e12, however, so a PCA analysis may not be ideal for this data set. However, the R2 value for this regression was determined to be 0.283, which is a decent improvement over the previously-calculated linear regression model.

The best subset selection model was trained on the Season Stats data frame and once again a validation plot was made to determine the optimal number of components in the model. This validation plot is below.

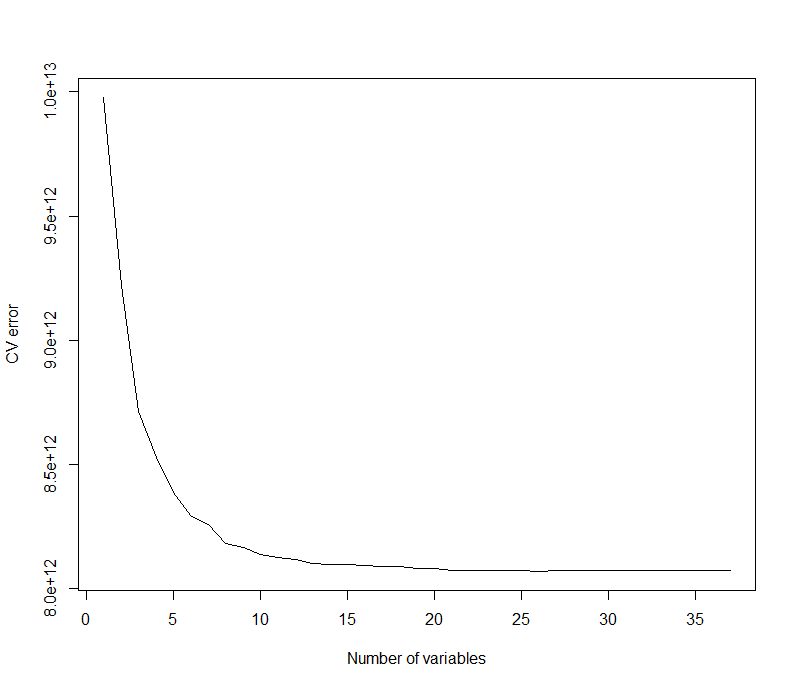


Figure 3:validation plot for best subset model

The ideal number of variables as determined by the best subset selection method was 12. However, the cross validated error for this model was over 8.0 e12, much like the PCA analysis. The high error values for the PCA analysis and best subset selection analysis means they will be disregarded as the optimal method for analyzing this data.

Analysis of the lasso method, ridge regression, and PLS models all resulted in mean squared error values for the salary variable that were over 8 e 12. These error values are too high for any of these models to be considered any further for this analysis. The lasso method concluded that there were 29 nonzero coefficients for the model, while the PLS model showed that included 10 components in the regression model was sufficient. The cross validation plots for the lasso and ridge regression methods, as well as the validation plot for the PLS model, are shown in the Appendix.

The decision tree model was by far the best at determining Hall of Fame induction for former players. The decision tree is shown below.

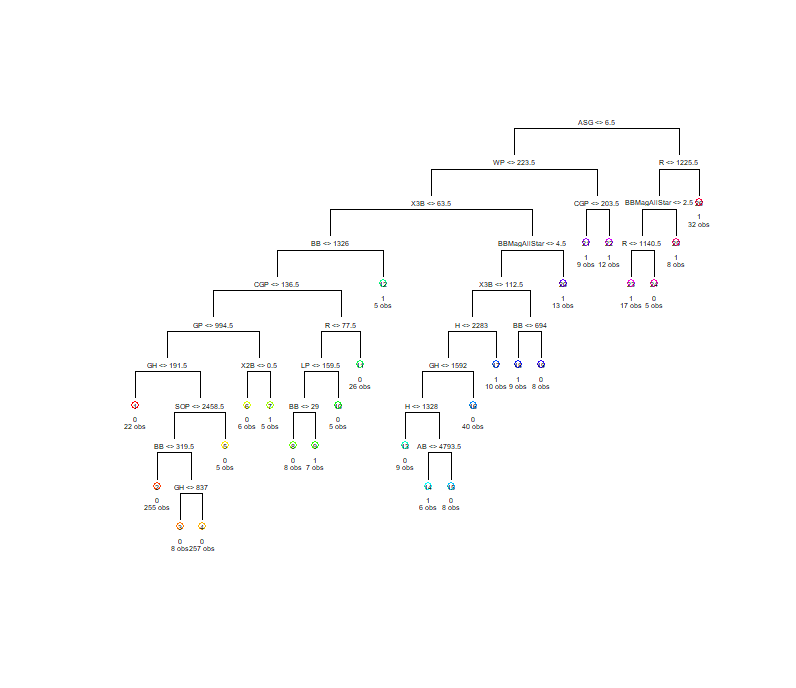


Figure 4: decision tree for Hall of Fame

The misclassification error rate for the decision tree was 43/795, or 5.4%. This makes the decision tree far and away the most accurate method for predicting Hall of Fame induction. The number of terminal nodes was found to be 26. The most important predictor seems to be number of All-Star Games played, since players over 6.5 All-Star Games played were almost exclusively Hall of Fame members. The test error rate of the tree was determined to be 13.5%, which is the lowest test error rate so far. The bagged random forest model was then built and deployed on the Hall of Fame data. The mean of squared residuals of the bagged model was found to be 0.0744, and the R2 value of the model was 0.4689, which was the highest R2 value of any of the models by far. An importance plot was created that ranked each variable by the impact they had on the random forest model. This importance plot is included below.

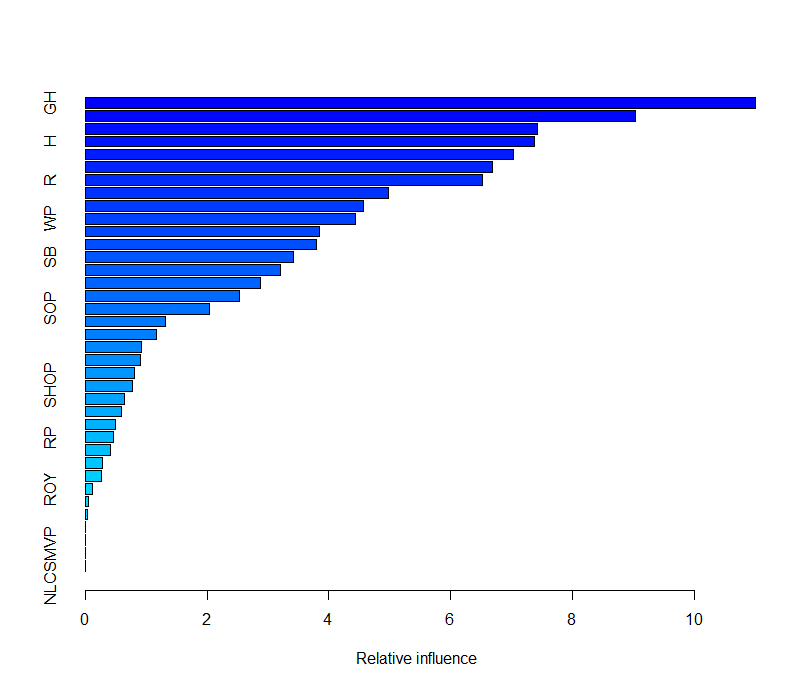


Figure 5: feature importance plot

From this importance plot, it was determined that the four most influential variables in the random forest model were GH, X3B, ASG, and AB. The test mean squared error of the boosted random forest model was calculated to be 0.1037, just slightly worse than the decision tree model.

1. **Validity and Reliability Assessment**

After analyzing all of the predictive models that were created, it was clear that the model best suited for use in predicting Hall of Fame induction rate was the decision tree model. The decision tree model had a misclassification error rate of just over 5%, and a test error rate of just over 13%. The R2 value for the decision tree was the highest R2 value of any of the models tested at 0.46, and the mean of squared residuals was also very low, at 0.0744. The original linear regression model was the model best suited for use in predicting player salaries. Although the R2 of this model was relatively low at 0.2042, every independent variable included in the linear regression model was shown to have a p-value much less than 0.05, which indicates that all of these variables are statistically significant. Furthermore, it shows that these variables have a nonzero impact on predicting player salaries.

Though this data set is very thorough and contains extensive data about every Major League Baseball player, more info would probably be needed to more accurately predict player salaries. MLB player salaries are dictated by many different variables that can’t be easily quantified, such as current market conditions, arbitration rules, service time add-ons, bonus incentives, etc. Hall of Fame induction seems to be pretty easily predicted, but more variables would be needed to accurately predict player salaries.

Inclusion of many different predictive models when analyzing a data set is a great way to ensure that accurate results are being obtained. While some models may not be well-suited to answer the particular question being proposed, other models may excel at solving these problems. In order to verify the accuracy of the findings within this report, the random forest model could be used in future years to predict new Hall of Fame classes, and the linear regression model could be tested over new data that includes future players’ stats and salaries. Based on the future findings of these models, one will be able to determine if these models are adequate in answering the proposed question, or if further revision is needed.

Appendix

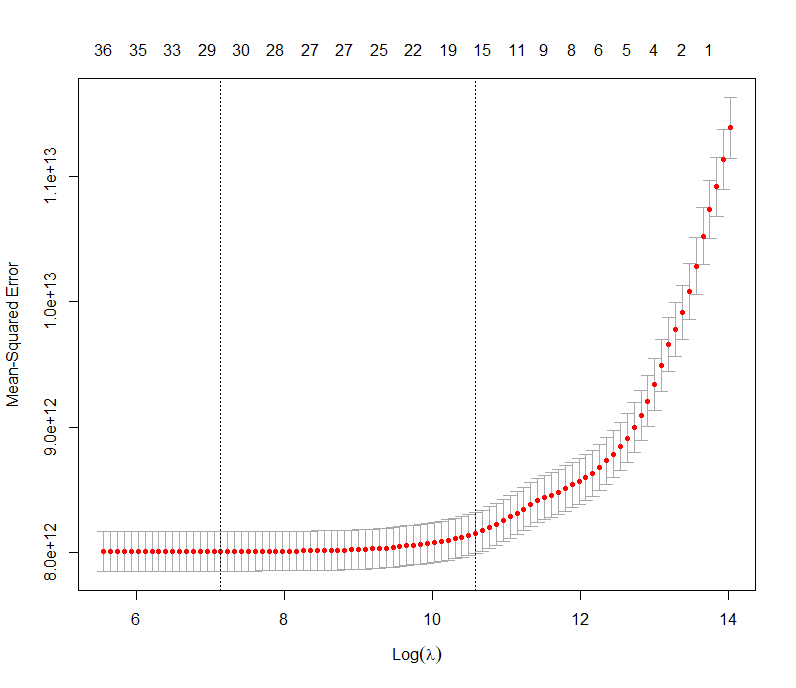


Figure 6: lasso cross validation plot

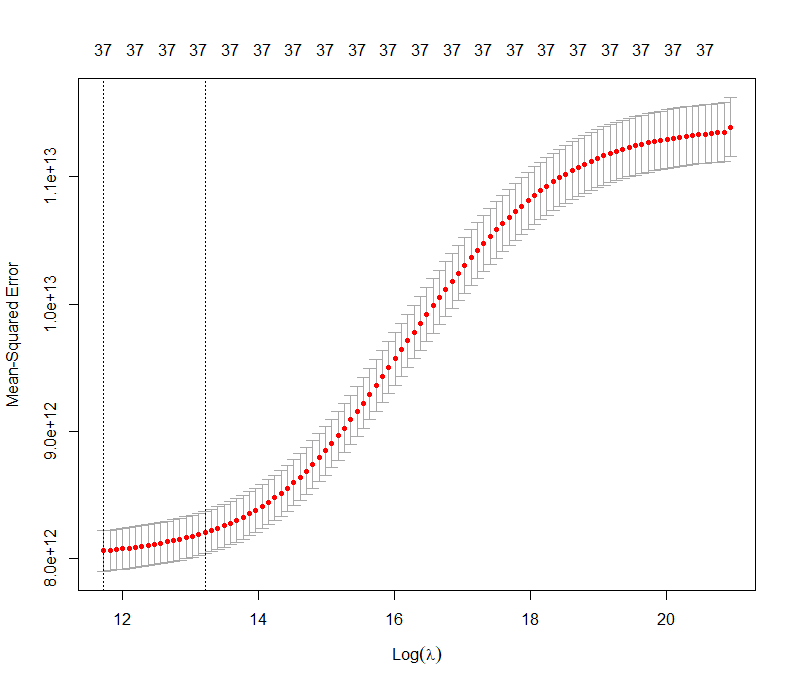


Figure 7: ridge cross validation plot

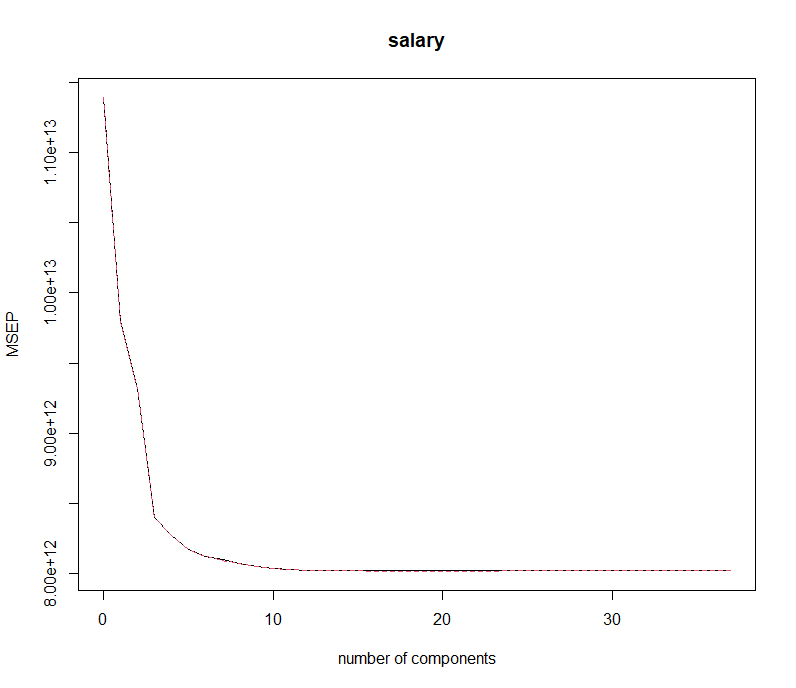


Figure 8: PLS validation plot