Predicting Baseball Hall of Fame Induction

Michael Hirsch

ILLC, University of Amsterdam

michaelahirsch@gmail.com

Abstract

Every year, the Baseball Writers Association of America votes on a new Hall of Fame class. Each ballot consists of 10 votes, and players need to appear on 75% of ballots in order to be inducted into the hall. Players have 15 years to get inducted, and are no longer eligible if that time period has passed. Here we attempt to classify Hall of Fame batters based on their career statistics using decision trees and an ensemble method, random forests.

1. Introduction

Major League Baseball (MLB) has been keeping thorough records of batting, pitching, and fielding statistics since its inaugural season in 1869. Recently, with the advent of sabrmetrics by the Society for American Baseball Research (SABR), many new metrics building on traditional statistics have been created. Such metrics caught the eye of statisticians, as these new metrics allowed for the creation of even more powerful predictive models for nearly every aspect of the game.

In this paper, we are investigating what makes a Hall of Fame (HOF) batter. There have been over 20,000 Major League baseball batters in the history of the organization, and only 211 of them have been inducted into the HOF. There are several ways in which a player can be inducted, and we only concern ourselves with players inducted from Baseball Writers Association of America (BBWAA) ballots, as this process is regulated and receives the most publicity. Often, fans and players feel that a worthy candidate is unfairly denied entry into the HOF because of voter bias, stacked ballots, or negative associations with performance enhancing drugs. The models proposed in this paper should be to able account for these cases.

We will investigate predictions based on both classification trees (aka decision trees) and random forests. Classification trees classify observations by means of successive binary splitting on feature variables. Random forests, introduced by Leo Breiman of UC Berkeley in 2001, is an ensemble learning technique that can be used for both classification and regression and improves on decision trees by correcting for overfitting. A random forest for classification consists of a collection of decision trees and outputs a prediction representing the most commonly occurring prediction of the decision trees.

2. Data

Data was collected from the Lahman Baseball Database, a freely available database that has been continuously updated since 1994 with the help of SABR and many individual researchers. Batting statistics were supplied on a by-year basis, so a player's career statistics were computed by aggregating his yearly results. The data was then merged with awards, all star appearances, and Hall of Fame voting results using the player's ID string. We will only be predicting whether or not players will be inducted by the BBWAA, since this is the current method of voting. All data preparation was done within R, the environment where we also build our learning models.

Attention is paid to 11 different statistics and a prediction classifier over the course of the player's career:

statistic	meaning
numSeasons	number of years in MLB
tAB	at-bats
tR	runs
tH	hits
tHR	homeruns
tBA	batting average
tRBI	runs batted in
tSB	stolen bases
mvp	number of mvp awards
gg	number of gold glove awards
Allstar	number of all star games played
inducted	classifier

Table 1: Hitting Statistics

We will be using several packages for our learning techniques, indicated in the sections below. Before training our model, it is important to have a quick look at our data.

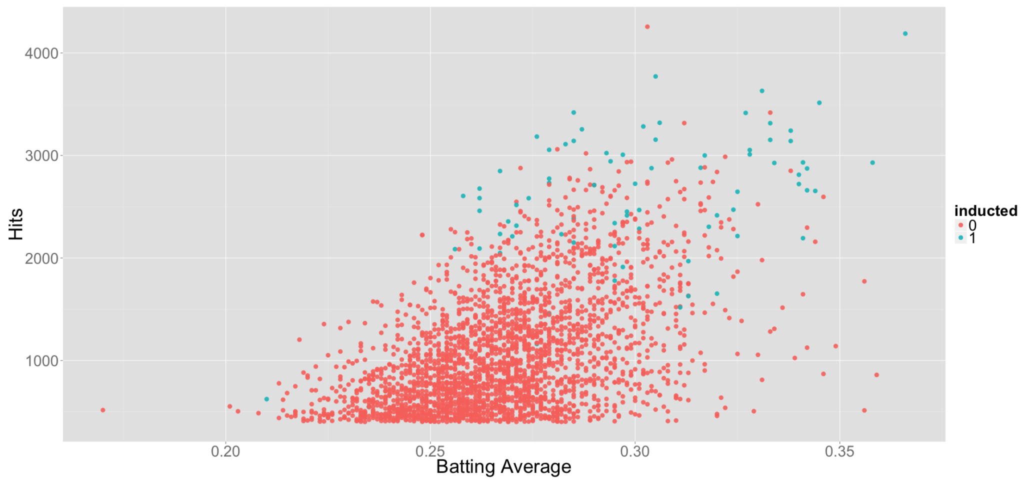

Figure 1: Batters with more hits and a higher batting average are inducted

We can see that more hits and a higher batting average seem to correlate with induction. This makes sense, as one often judges the quality of a batter by his batting average, the number of hits a player has per at-bat. The interplay between all of these statistics will be elucidated in more detail in the sections to follow. Keep in mind that a lot of the players with many hits and a high batting average that weren't inducted are those players whose induction we seek to predict. Also, the player with over 4000 hits who is not in the Hall of Fame is Pete Rose, who received a lifetime ban from baseball activity for gambling against his own team.

3. Tree-Based Methods for Classification

In this section, we will give an account of two learning methods that we will use for the basis of prediction: classification trees, and an ensemble method, random forests. We will also introduce the concept of bootstrap aggregating (bagging) which is a fundamental ingredient of random forests. The information here is theoretical, and will be motivated by a simple example. We will encounter applications of these ideas to our HOF data in the next section.

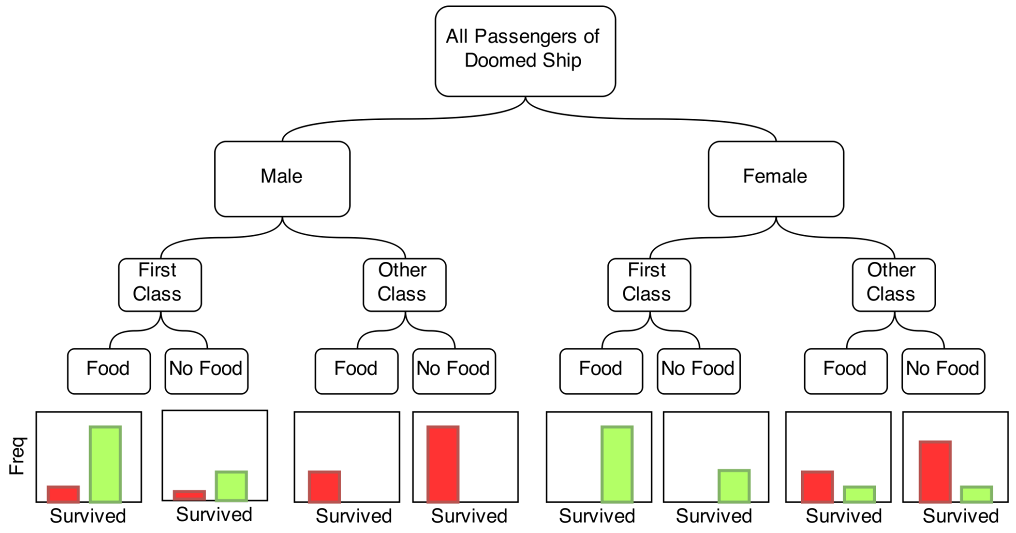

Figure 2: Decision Tree for Survival of Boat Disaster

3.1. Classification Trees

Classification trees are a fairly effective predictive tool that are lauded for their high degree of interpretability. When constructing a classification tree, we begin with the full set of observations and begin dividing the feature space into non-overlapping regions by means of binary splitting. In prediction, we assign each observation in a given region of the feature space to the most commonly occurring class of the training observations in that region.

When choosing splits in our tree, we are concerned with the *Gini Coefficient* of the nodes resulting from each split. We want to choose the feature to split on based on which split will give us the greatest decrease in this value. The Gini Coefficient is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it were randomly labeled according to the distribution of labels in the subset. The Gini Coefficient can be computed by summing the probability of each item being chosen times the probability of a mistake in categorizing that item. It reaches its minimum, 0, when all cases in the node fall into a single target category. This is defined as:

$$G = \sum_{i=1}^{m} p_i(1 - p_i) = 1 - \sum_{i=1}^{m} p_i^2$$

where i takes on values in $\{1, 2, \ldots, m\}$, the number of different categorical labels, and p_i is the fraction of items labeled with value i in the node. So G can be seen as a measure of total variance across the m classes. Since this paper will only be discussing a classification variable that takes on two values, inducted or not-inducted, in our case $m = 2$. Note that $G = 0$ when all observations in that node fall into one class. In splitting at nodes, we look to minimize this value.

Let us illustrate this with a very simple example of a decision tree created to predict the survivors of a boat disaster, show in Figure 2. The root node contains all passengers of the ship. We then split the full set of observations according to their sex, followed by a split on what type of ticket they had, followed lastly by a split on whether or not they chose to order a meal with their ticket. Below each terminal node is a distribution of those passengers who survived and those who didn't.

Table 2 shows the fates of all male passengers based on their ticket type as well as their food option. Now, we can consider the Gini Coefficient associated with splitting on ticket class or on food option.

Male	Survived	Died	total
First	122	19	141
Other	12	1113	1125
total	134	1132	1266

Male	Survived	Died	total
Food	60	500	560
No Food	74	632	706
total	134	1132	1266

Table 2: Aftermath of a Tragic Boat Disaster

If we split this subset of the observations on ticket type, we can calculate the Gini Coefficients for each of the two successive nodes as follows, with $i = 0$ indicating survival and $i = 1$ indicating death:

$$G(\text{male, first}) = 1 - \sum_{i=0}^{1} p_i^2 = 1 - ((122/141)^2 + (19/141)^2) \approx 0.233$$

$$G(\text{male, other}) = 1 - \sum_{i=0}^{1} p_i^2 = 1 - ((12/1125)^2 + (1113/1125)^2) \approx 0.021$$

$$G(\text{male, food}) = 1 - \sum_{i=0}^{1} p_i^2 = 1 - ((60/560)^2 + (500/560)^2) \approx 0.191$$

$$G(\text{male, no food}) = 1 - \sum_{i=0}^{1} p_i^2 = 1 - ((74/706)^2 + (632/706)^2) \approx 0.188$$

Averaging the decrease in the Gini Coefficients based on splitting on ticket class gives us a decrease of $G(male) - (0.233 + 0.021)/2 \approx G(male) - 0.127$ while splitting on food option gives us a decrease of $G(male) - (0.191 + 0.188)/2 \approx G(male) - 0.190$, meaning that it is better to split by ticket type, as this leads us to purer nodes. The Gini Coefficient is related to the notion of entropy and can be seen as a measure of how informative the answer to a "yes/no" question is at the split in a tree. Entropy can also be used in determining splits, and can be calculated using the following equation, where p_i is the same quantity as defined above:

$$-\sum_{i=0}^{m} p_i log_2 p_i$$

Decision trees, while useful, have their limitations. Firstly, it is possible to grow overly complex trees that overfit the training data [1]. This can happen if we grow our tree in such a way to have only a single observation at each terminal node. The misclassification rate in our training sample would be 0, but we would be very likely to encounter an error in our training set. In such cases, a method known as pruning is required to combat this problem. *Cost complexity pruning* is a method for considering subtrees of our initial tree model that minimize the cost-complexity criterion, which is defined as follows[2]:

Let T_0 be our unpruned decision tree, and define a subtree $T \subset T_0$ to be any tree that can be obtained by collapsing any number of internal (non-terminal) nodes. Let $|T| = |\{R_1, R_2, \ldots, R_m\}|$ denote the number of terminal nodes of our subtree. Define the number of observations at each terminal node as:

$$N_m = |\{x_i \in R_m\}|$$

And the cost-complexity criterion:

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m G_m(T) + \alpha |T|$$

For each α, there is a unique smallest subtree $T_\alpha \subseteq T_0$ that minimizes $C_\alpha(T)$. An estimation $\hat{\alpha}$ is done by using tenfold cross-validation, and the final pruned tree is $T_{\hat{\alpha}}$. In tenfold cross validation, the original training set is partitioned into ten equal subsamples, nine of which are used to train the model, and one used for validation. This process is repeated 10 times, each time selecting a different subsample as the test data. The 10 results are then averaged to produce a single estimation. More details can be found in Breiman et al.[3]

Using our Boat example, we note that that our tree has 8 terminal nodes. If we take a look at the distributions below the nodes, we note that the split on food option is not that meaningful in both the case of non-first class males and first class females. So let us consider the subtree T show in Figure 3 formed by removing these splits. We note that the Gini Coefficients of the nodes that precede the removed nodes are already equal to 1, indicating that we already have perfect information about the survival of these passengers. Splitting these nodes further by food option does not decrease the Gini Coefficient and therefore brings us no new information and can be pruned from our tree. Note that $C_\alpha(T) \leq C_\alpha(T_0)$.

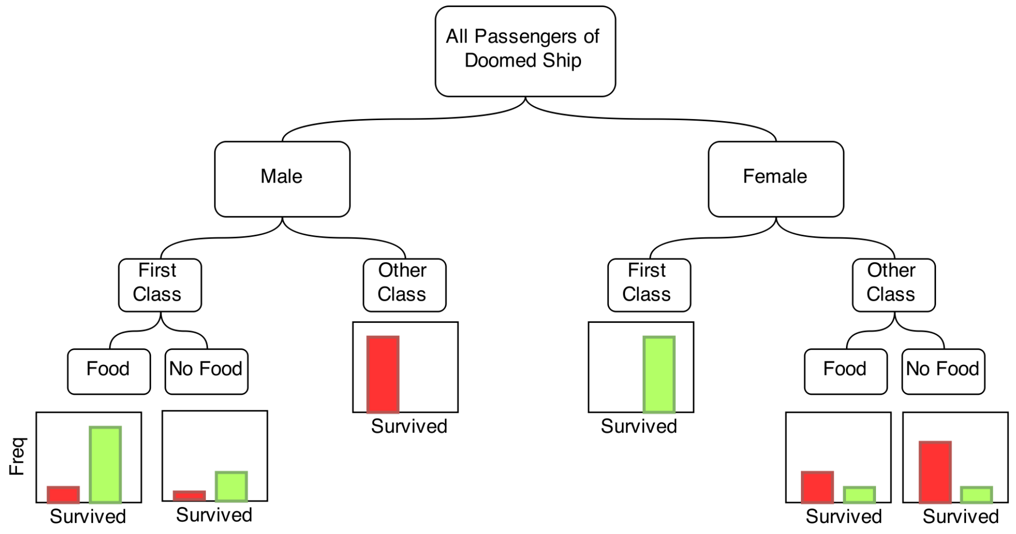

Figure 3: A less complex version of our tree

3.2. Ensemble Methods

Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the individual learning algorithms composing the ensemble. Here, we look at two such methods, bagging and random forests.

3.2.1. Tree Bagging

Decision trees as discussed in the previous section often suffer from high variance in the case of complex trees, meaning that if we fit a decision tree to two random samples from a set of training observations, we could end up with vastly different results[2]. *Bagging* (bootstrap aggregating) is a method for reducing the variance of any learning method by taking repeated samples from a single training set and taking, in the case of classification, the most commonly occurring class among the bootstrapped predictions.

On average, each of the B bootstrapped samples will take about two-thirds of the training observations. The remaining observations that were not used in fitting a bagged tree are referred to as the *out-of-bag* (OOB) observations, and we can predict the target variable for the ith observation using each of the trees in which that observation was OOB. This will yield, on average, $B/3$ predictions from which we can predict the classification for the observation by taking majority vote.[1] Using this method, we can get a single OOB prediction for each of our observations, obtaining an OOB error, which is a valid estimate of the test error for the entirety of our bagged model.

Bagging also gives us insight into which of our predictors were the most important in creating our model by looking at the Gini Coefficient. To do so, we take the sum of the amount that the Gini Coefficient has decreased by splitting on a given predictor and take the average over our B trees.

3.2.2. Random Forest

The random forest technique is quite similar to taking bagged samples of trees, with the main difference being that at each of the splits only a random sample of size $m < p$ of the set of all features p is considered. Typically, $m \approx \sqrt{p}$ predictors suffice[4]. This technique helps to decorrelate the tree, as there may be one predictor that has a very high influence in classification. In the case of bagging, this predictor will most likely be used as the first split in a majority of the trees, causing most bagged trees to be quite similar and correlated. By forcing each split to consider only a subset of the features, on average $(p-m)/p$ of the splits will not consider this a strong predictor[4].In this paper, we will be utilizing the *randomForest* package in R.

4. Prediction Models

In this section, we apply the techniques introduced in the last section to our HOF data. We will start by creating a single classification tree and then predict the induction classifier of our training data based on our model. We will also look at some metrics that will allow us to test the effectiveness of our classification model in order to see what parameters would be best to use.

4.1. Classification Tree

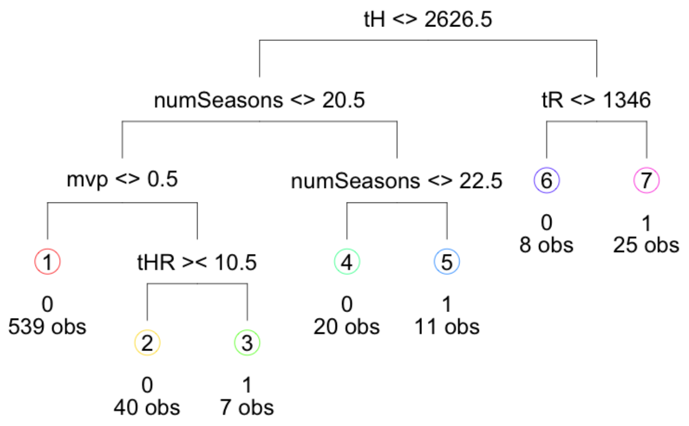

Figure 4: Batting

Using R's rpart package, we can fit a classification tree to our data. To do so, we will split the entire data set into a training and a test set and build our tree using the training data. As our training set, we will take a random sample of about 2/3 of our data. This amounts to ≈ 650 batters. The tree shown in figure 4 was obtained by predicting induction based on all of our features mentioned in section 2. Our model produced a tree with 8 splits and 9 terminal nodes and we notice that only 5 variables were used in the creation of the tree: tH, numSeasons, tR, mvp, tHR. The misclassification error of this tree on the training data was : $62/650 \approx 9.5\%$. Let's use this model on our test data to see how it performs, shown on the left in table 3.

unpruned	actual	
prediction	0	1
0	356	23
1	23	27

pruned	actual	
prediction	0	1
0	369	35
1	10	15

Table 3: Confusion Matrices for unpruned and pruned decision trees

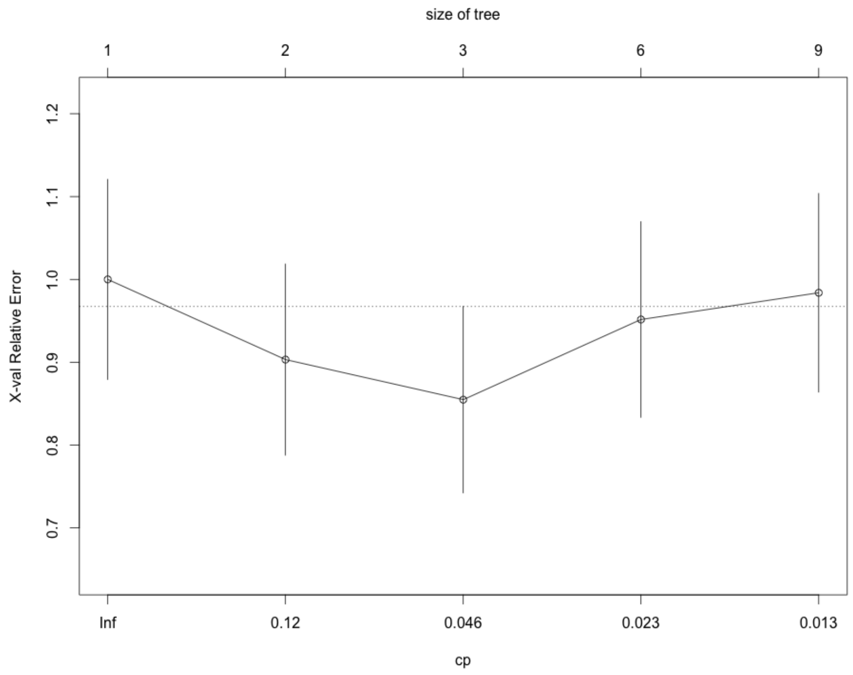

Figure 5: Complexity Parameter of Various Trees

Our model predicted the induction of our test players with a misclassification rate of $1 - 383/429 \approx 10.7\%$. We can perform cross validation on the tree to see if there is a more ideal pruned tree size. Having a look at figure 5, we notice that the cross validation error is minimized around cost complexity parameter (cp) 0.046. We can use this value to prune our original tree in hopes of obtaining a more stable model with only 3 terminal nodes. Generating a new tree model having only 3 terminal nodes actual performs slightly better than our original tree, with a misclassification rate of $1 - 384/429 \approx 10.4\%$ on the test data. This data is also summarized in table 3.

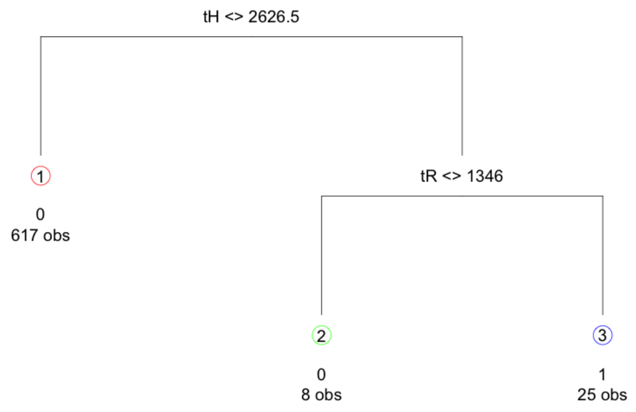

Figure 6: Our Pruned Tree

4.2. Random Forest

In this section, we will build a model for predicting Hall of Fame induction using an ensemble of decision trees. The method is similar to the one followed above, except in this case we will use bagging to ensure that we do not overfit the training data. Instead of building just a single tree in which all variables are considered at each binary split, we will build numerous decision trees in which a randomly selected variable is chosen at each split. As before, each tree is trained using a training set, and the trees are built so that each terminal node is *pure* in the sense that all of the observations in the node fall into the same class.

The test data will be referred to as *out-of-bag* sample, since it is not included in the bagging method used for training. The prediction for the out-of-bag samples will be the majority vote of the predictions from the individual trees. To change things up, in this section we will be considering the full set of batters rather than simply those that have appeared on at least one ballot. This will be useful in predicting induction for players that have not yet retired, or have retired in the last 5 years.

To do this, we will make our training set all players that have retired before 2009 and our test set all players who are still currently active, or have retired after 2009. We will arbitrarily choose to grow a forest of 1000 trees, and use a tuning function to determine how small of a subset of our variables should be chosen from at each split. Figure 7 indicates that 3 variables will suffice and result in the lowest OOB-error for 1000 trees. This low OOB-error for choosing 3 variables at each split indicates that OOB observations were misclassified most infrequently in such a scenario.

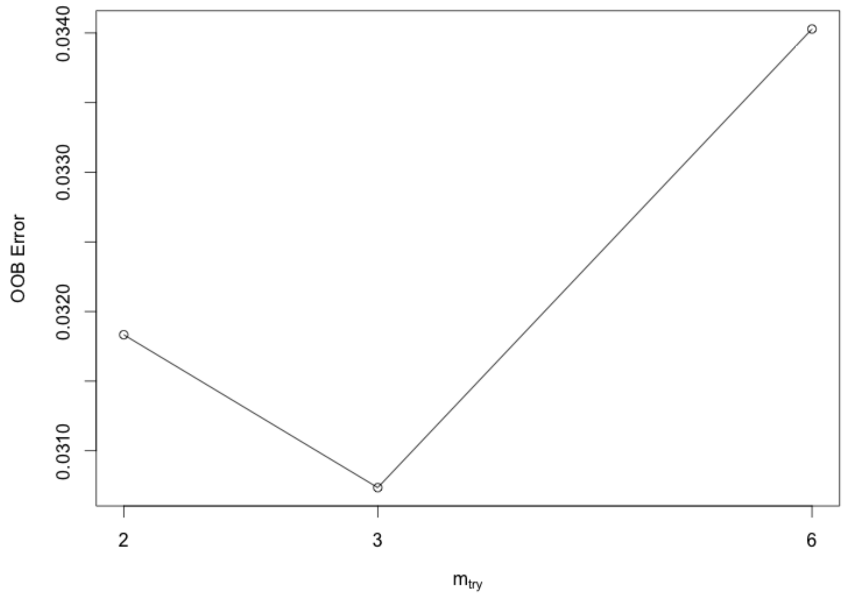

Figure 7: number of variables again OOB-error

After training our forest, the results in table 4 were obtained, indicating an OOB-error rate of ≈ 2.96% We see that the misclassification rate for non HOF players is much lower than that of players who were inducted. This is due to the fact that there is a much larger population of non HOF players, making them easier to classify by the model.

A benefit of random forests is that they can indicate which variables were most important in prediction. This is a measure of the total amount that the Gini Coefficient has decreased by splitting over a given variable and then averaged over all the trees in the forest. In our case, the results in table 5 show that All Star Game appearances followed by the total number of runs, RBIs, and hits are the factors that best classify a HOF batter.

rf	actual	
prediction	0	1
0	851	10
1	17	3

Table 4: Confusion Matrix of training data in random forest model

variable	MeanDecreaseGini
ASgame	25.511962
tR	18.343892
tRBI	12.891316
tH	12.077022
tHR	7.500770
mvp	5.923212
tBA	5.071497
tSB	4.261315
gg	2.661624

Table 5: Variable Importance

Now we are in a position to use our random forest model to predict Hall of Fame induction on the test set, shown in table 6. Since our test set contained players not yet eligible for induction, these predictions can best be tested against actual data once these players appear on ballots. It should be noted that these predictions were made as if all of the players in our test set had retired *today.* Since nothing in our model projects players' performance in the future, there are definitely some players that are worthy, but just haven't had enough playing time to accumulate the required statistics. For example, in table 6, we see that Miguel Cabrera is not HOF worthy. However it is incredibly impressive that he still has around 10 years left in him and he is already fairly close to being at a point where his career statistics get him into the hall.

player	inducted	percent of ballots
Rodriguez, Ivan	1	0.842
Sheffield, Gary	1	0.810
Ramirez, Manny	1	0.800
Pujols, Albert	1	0.798
Griffey, Ken	1	0.781
Guerrero, Vladimir	1	0.733
Suzuki, Ichiro	1	0.728
Jeter, Derek	1	0.558
Rodriguez, Alex	1	0.553
Jones, Chipper	1	0.517
Mauer, Joe	0	0.421
Thome, Jim	0	0.410
Delgado, Carlos	0	0.373
Cabrera, Miguel	0	0.368
Ortiz, David	0	0.336

Table 6: A few predictions from our random forest

5. Conclusion

Decision trees and random forests are just 2 learning methods that can be used for classification or regression. There are numerous other models including linear regression, support vector machines, nearest neighbors, neural networks, and logistic regression, just to name a few. The methods chosen here were done so to illustrate their effectiveness and interpretability.

As we have seen, decision trees can suffer from overfitting the training data, and appear to perform worse in comparison to random forests. Decision trees are advantageous since they appear to mimic the human decision making process for classifying observations as seen with our description of the Gini Coefficient. They can also be neatly depicted in a graph, aiding in presentation. Also, as we have seen, the downsides of decision tree learning can be remedied by aggregating them in a random forest.

With regards to the Hall of Fame Data, we have shown that our decision tree model was able to predict induction with about 90% accuracy. The difference between an urpuned and a pruned model was quite minimal with respect to accuracy, but quite substantial for interpretability. We can think of the outcome of the pruned tree in the following way:

1. Did the player have at least 2327 hits in his career?
2. If yes, did the player score more than 1346 runs?
3. If yes, this player will be inducted with 90% certainty.

Random forests performed much better in prediction, with a misclassification error rate near 3%. The model we created with the random forest technique was also more informative, as we took advantage of the power of the method and used the full set of batters. We were also able to view the importance of each variable in deciding whether or not a player would be inducted. This can prove to be useful in evaluating the patterns of voters and indicating periods of time where they too heavily favor one statistic. For example, all star game selection is done by the general public, and players from higher market teams often make the all star team not because of their talent, but because of the higher population of voters in their region.

Going along with this, some interesting extensions of this model could be formed by doing time series analysis on voting patterns. This could indicate certain eras in which BBWAA voters were more interested in one statistic over another, and in classifying the types of batters that were most lauded during that era. For example, in the early 2000s during the height of steroid use, power hitters were held in higher regard than "scrappy" players, while now we are in an offensive lull, where high contact batters are more vital to a teams' success.

6. References

[1] Hastie, James, Tibshirani, Witten, An Introduction to Statistical Learning, Springer, 1st edition, 2014.

[2] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer Series in Statistics, Springer New York Inc., New York, NY, USA, 2001.

[3] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, Classification and Regression Trees, Wadsworth International Group, Belmont, CA, 1984.

[4] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32.