The data that I used to create my model was from IMDBpie, which includes only the 250 movies with the best IMDB ratings. My goal was to use the features available to me to make a tree-based model to predict if movie will have a high or low rating. These movies ranged in ratings from 8.0 to 9.2. You can see how many movies received each rating in the histogram shown:



The x-axis shows the movie rating, and the y-axis show the number of movies which received a specific rating.

To be considered a “high-rated” movie, the movie needed to have a rating of 8.5 or higher. This would categorize the top 50 movies as “high-rated”, and the remaining 200 as “low-rated”. The features I culled together to create my model were the total number of votes the movie received, the year the movie was released, the number of alpha-numeric characters in the movie’s title, and three features related to awards – Oscars, other awards won, and other award nominations. The only two numeric features the data started out with were the number of votes and year. I wanted to make use of the title, and I thought it’s length could be an indicator because it varies wildly among titles (between 1 and 68 characters) and since people often “judge books by their cover,” the title could impact the rating it receives. Finally, to get the three awards features, I did a web scrape of the actual IMDB webpage, and scraped those 3 values for each movie on the top 250 list. Most of the movies had values for all 3 awards categories, but if values were missing I didn’t include any of their awards categories.

To create a model, I looked at a decision tree regression and a decision tree classifier. The classifier greatly outperformed the regression. To get a generalized model that can predict new data not previously seen, I used a random forest classifier, to create many random decision trees. After splitting the 250 movies into a training set and a testing set, my model consistently scores in the 80% range for mean accuracy on the test set, which is decent given the small sample size of 250 movies, and the small range in the ratings column.

To see visually the effectiveness of the model I created the scatter plot show:



The x-axis is the release year of each movie, and each movie is a point, with a y-value of 1 for those correctly predicted by my model and a y-value of 0 for those incorrectly predicted. Clearly there are many more points that are correctly predicted, and, as mentioned before, 80% of the points have a y-value of 1.

Some features to add in a future model are looking to see if specific key words in a title correlate with the rating, which I started but didn’t complete, adding an ensemble such as boosting and bagging, and increasing my data set to include bad movies as well. I found a list of 150 low rated movies that I wanted to append to create a large range of rating, but I couldn’t get that web scrape to work by the deadline.