Data Mining: Final Project Write-up

Misclassification-Terminator(Yuxi Chang, Mengran He, Alexander Lam, Xiaowen Yin)

May 10, 2017

Introduction

Every year, about 20% flights gets delayed or canceled across U.S., which could cause tremendous loss in money and time not only for travellers but also for airports or flight carriers. Pittsburgh as one of the main international airports in the United States, takes an important role in the transportation network. In this report, our goal is to predict whether a flight departing from Pittsburgh International Airport would be delayed at least 15 minutes using known flight information, along with weather information of both Pittsburgh International Airports and the destination airports.

Data

There are three major data sources used for our prediction:

1. An extract from the Airline On-Time Performance Data made from the Bureau of Transportation Statistics of the U.S. Department of Transportation which reflects commercial flight activity to and from Pittsburgh International Airport.
2. Hourly weather data scraped from http://www.wunderground.com, which was merged into the Airline On-Time Performance Data based on the CRS departure time. The closest Pittsburgh hourly weather record pervious to the CRS departure time was used. All missing values for hourly data were imputed with the previous hour data.
3. Daily weather data for the arrival airports was achieved from https://www7.ncdc.noaa.gov/. This was also merged into the first dataset based on the flight date and arrival destination. All missing values for daily data were impyted with the previous day data.

Exploratory Data Analysis

Our training dataset are flights and weather data from the year 2015 and there are 25870 records in our dataset in total, with 36 features. Among these features, 17 are features about flight activities, including departure time in different scales(hour, day, quarter, month, holiday), flight destination, flight carrier, CRS time, same hour flights in total, same day flights in total and whether the actual arrival time of the previous flight is already behind the CRS departure time of the next same flight. The features left are temperature features both at Pittsburgh International Airport and at the destination airport, including temperature, humidity, pressure, visibility, wind speed, snow, rain, fog, hail, thunder, or tornado conditions. Some of these features like snow, rain, fog, hail, thunder, and tornado were factorized into levels according to their occurance severities.

We split our 2015 dataset into 70% training and 30% for model validation in later stages.

Flight Activities

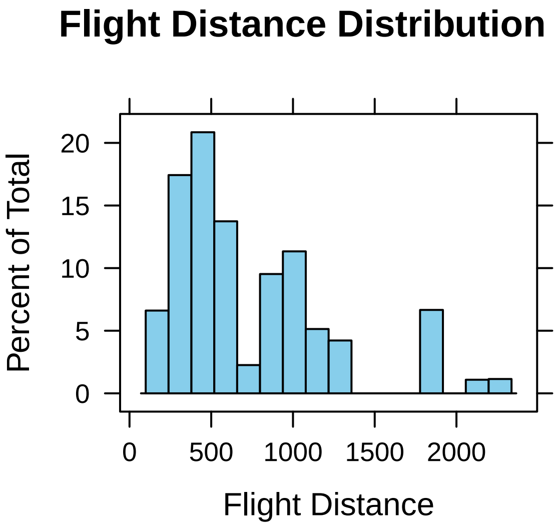


Figure 1

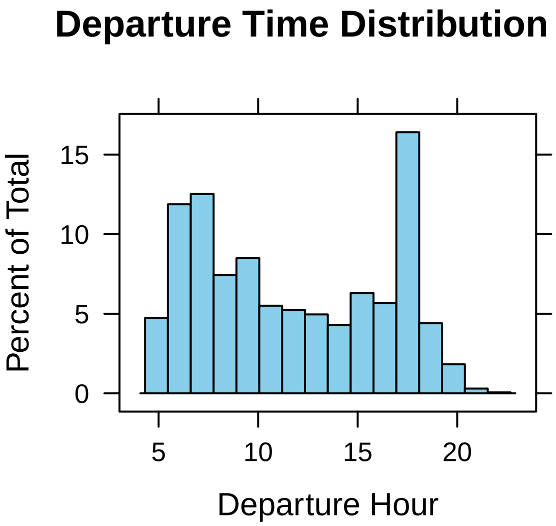


Figure 2

Figure 1 and Figure 2 above show the flight distance and departure time distribution across the year 2015. Most of the flight distance departing from Pittsburgh were in a range of 0-1000 miles. Morning (5-8am) is the busiest time for Pittsburgh International Airport during a day, and 4-7pm is the second busiest time. There is no departure flight after 11pm. In addition, weekdays are busier than weekends, and Saturdays have the fewest departure flights.

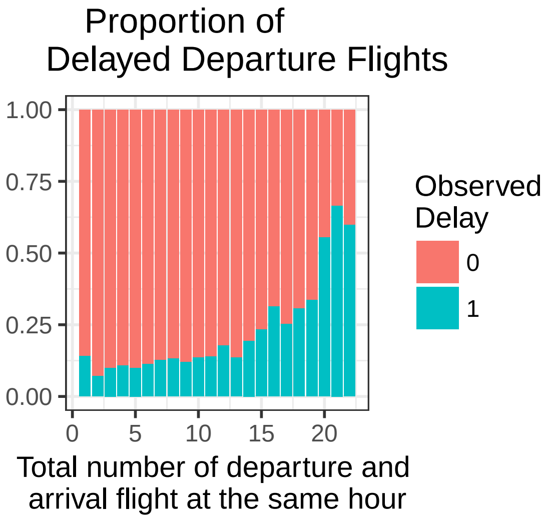


Figure 3

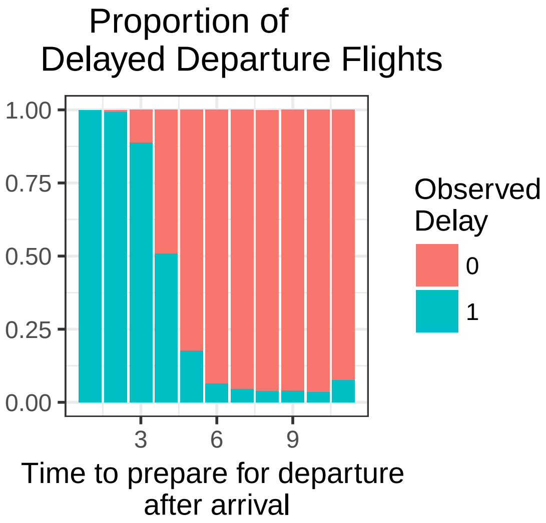


Figure 4

Figure 3 above shows the proportion of delayed departure flights versus the total number of departure and arrival flight at the same hour. From the plot, the proportion of PIT departure delayed flights increase with the total number of flights departing and arriving at the PIT.

Based on the plane registered tail number (TAIL_NUM), we was able to identify if a planned departure plane was just arrived from another airport to PIT or not. We calculated time between the plane actual arrival plus taxing time (ARR_TIME + TAXI_IN) and the plane planned departure time (CRS_DEP_TIME). If a scheduled departure plane wasn't from another city, time left for plane departure preparation was not applicable. In order to still use this variable, we discretized this time into 12 levels, with 1 meaning less than 5 minutes left, and 12 meaning more than 90 minutes left.Figure 4 shows the proportion of delayed departure flights versus the actual time left for departure preparation after the plane arrived at PIT.

Feature of Departure Airport (Pittsburgh International Airport)

After some exploration in the weather information we scrped, we found that Pittsburgh is more humid from Dec to January and from June to August. We also found that the temperature follows a clear seasonal pattern (cold during winter months, hot during summer months). The visibility is very low during January through March. Wind speed is high during winter months. Heavy snows may occur from January to March. Visibility decreases along with the increase in rain and snow severity, so does pressure.

Feature of Destination Airport

After some exploration in the weather information we scraped, we found that certain airports with more rains and snows have relatively low visibility and high wind speed, such airports include but limited to: Lambert St Louis International Airport, Boston Logan International Airport, Atlanta International Airport, Charlotte International Airport.

Delay vs. Non-delay

The base rate of our dataset (i.e. proportion of delayed flights) is 13.77%. An interesting finding by comparing delay and non-delay flights is that there are more delay flights during evenings than in the mornings even though mornings are normally busier, as shown in figure 5 below. Moreover, temperatures at destination airports for non-delayed flights are much higher than those for delayed flights, as shown in figure 7.

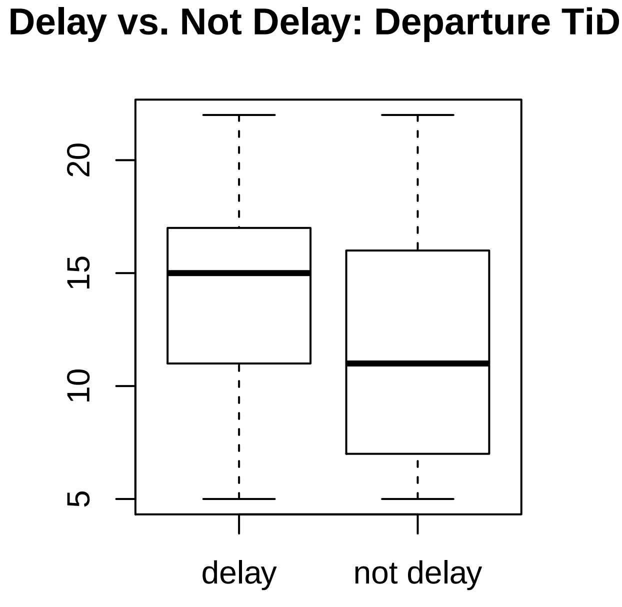


figure 5

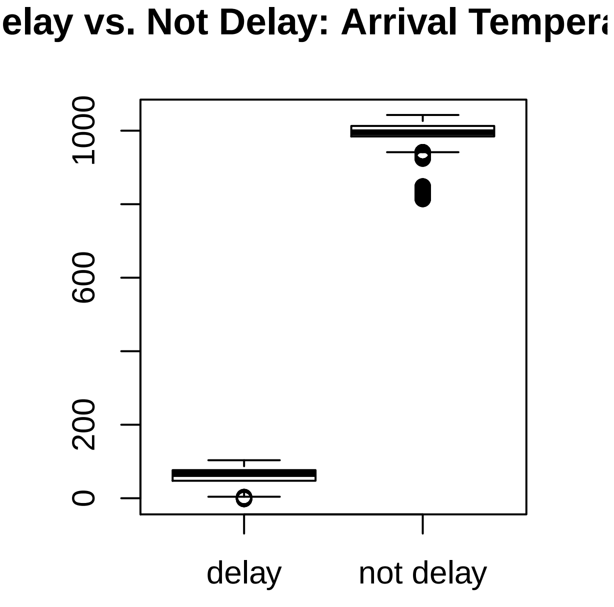


figure 6

Unsupervised Learning: Clustering

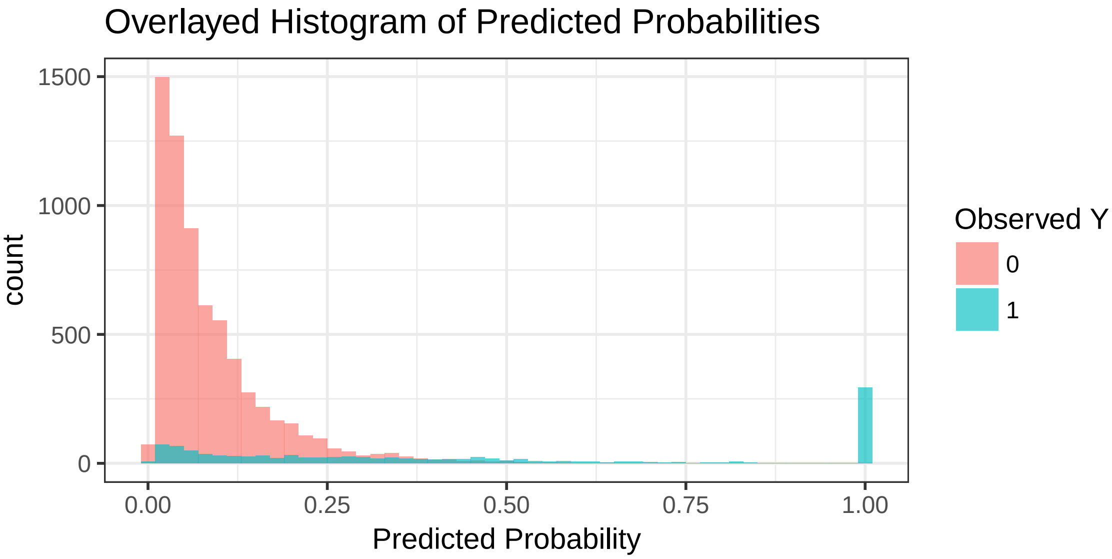
Clusetrings using k-means, k-metroids, compelete linkage, and minimax linkage were examined. All methods gave 2 clusters, with K-means and K-metroids presenting similar results while compelete linkage and minimax linkage giving slightly different results. The clustering results given by minimax linkage were the closest to the results we got from previous exploratory data analysis. As Figure 7 below shows, the average temperature of arrival destinations are higher for cluster 2 compared to that of cluster 1, which agrees with what we got from the previous section (figure 6).

Supervised Learning

Lasso logistic Regression

Lasso logistic regression as one of the most popular modern regularized regression models, could carry out the variable selection process automatically by shrinking the unnecessary variables to zero. We constructed model matrix including all features and fitted lasso logistic regression using 70% of 2015 dataset as training data. To be more precise, we decided to use the minimum lambda instead of the 1SE lambda. If a simpler model is expected in the future, we could switch to the 1SE lambda. In this model, 88 of all 104 feature combinations were selected.

The following distribution of the predicted probabilities from lasso logistic shows how well our model separates the zeros from the ones. As can be seen, a good portion of the true positives have high predicted probabilities.

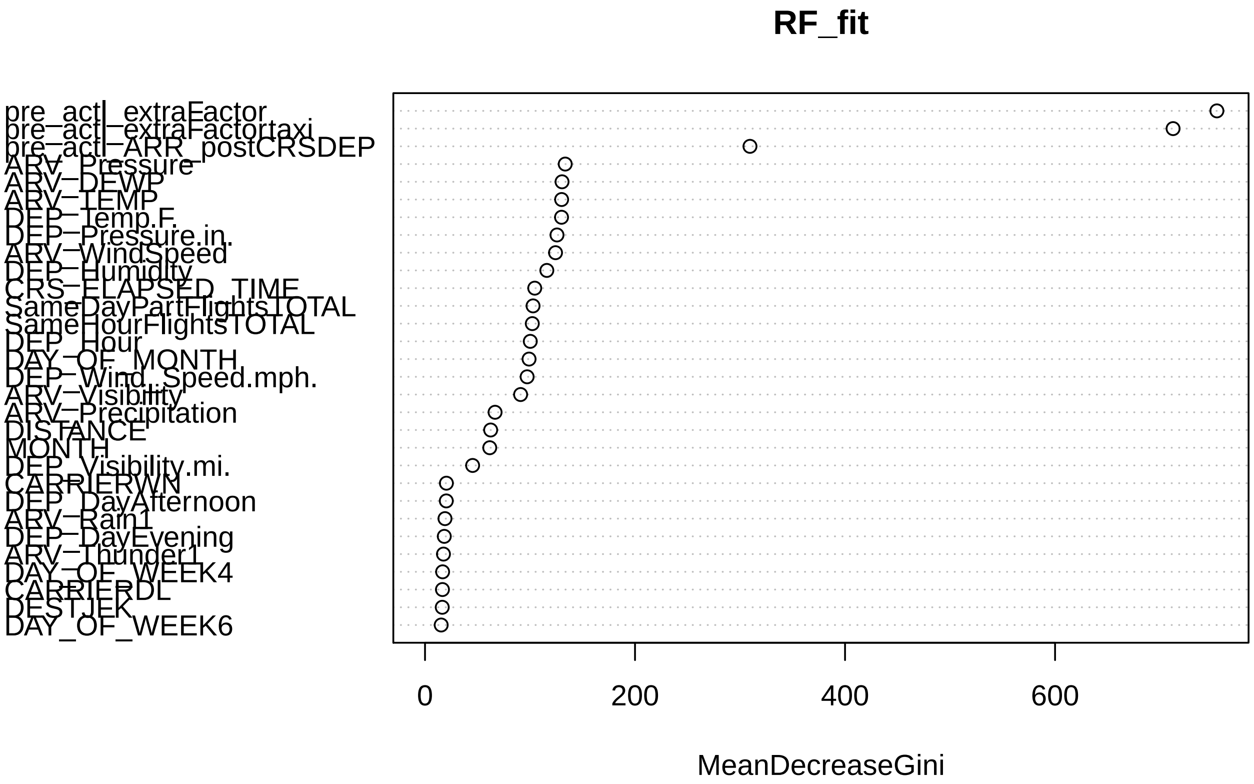


For prediction, we designed a function which allows for best threshold selection by minimizing the test error with the other 30% of 2015 flights data. In practice, the chosen threshold should be very close to 50% since the data generating process for the training and testing datasets (70/30% of 2015 flight data should be similar if not the same) As a result, a threshold of 0.4 was selected and the misclassification rate using the selected threshold was 8.6%. This is much lower than the 13.7% base rate in the 2015 dataset.

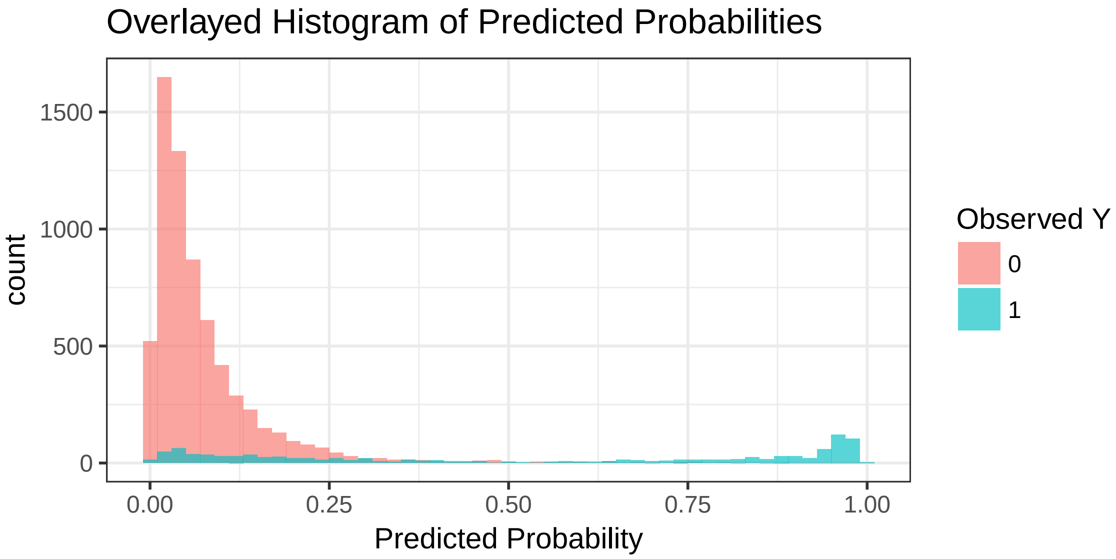
Random Forests

Random forests is another robust method for classification, and is operated by constructing a group of decision trees. As an improvement of "bagging" strategy, random forests could make more independent trees by only allowing a small random subset of variables to be considered for each split. We tried 500 trees which should give us a reasonable reduction in variance.

As we can see from the following variance improtance plot (Figure 8):



almost all of the arrival variables and some weather variables are important for the splits in random forest. This suggests that the features we created are useful for predicting flight delays.



The above histogram shows that random forest is doing an even better job in separating the two classes as a lot of true positives have high probabilities.

Again, we used the function designed for best threshold selection. As a result, a threshold of 0.363 was selected and the misclassifcation rate using this threshold was 7.4%, which is even better than Lasso Logistic Regression.

K-Nearest-Neighbour (KNN)

K-Nearest-Neighbors (KNN) is a model-free classification method. It predicts an input according to its nearest-neighbors and then classifies according to a majority vote. Because KNN method heavily depends on distances between data points, we transferred all the categorical variables into numeric variables, except the flight carrier (CARRIER) and the flight destination (DEST). Also, to avoid variables with large variation determine the distances between points, we standardized all numeric variables before fitting the data in the KNN model. In order to choose the number of neighbors, parameter k, we ran KNN models with 1 to 40 nearest-neighbors, and k=13 gives the lowest misclassification rates when comparing predictions to 16 test data.

Boosting

Extreme Gradiant Boosting(xgboost) is an efficient algorithm that can be applied in supervised learning because it has high predictive power and fast computation. We decided to apply xgboost on training set (70% of flight activities in 2015) and tried to find the best model by hyper-tuning booster parameters with 5 folds cross-validation. For the booster parameters, we tuned on learning rate(`eta`), `gamma`, maximum depth of the tree (`max_depth`), and subsample of training instance (`subsample`) and built 1000 trees in the model (`nrounds`). The learning rate specifies step size shrinkage used in update to prevents overfitting, the smaller `eta` the more conservative the algorithm will be. `gamma` represents the minimum loss reduction required to make a further partition on a leaf node of the tree, so the larger the gamma more conservative the algorithm is.

After 5 folds of cross validation, we have the model with largest AUC (0.8806) value with `max_dempth = 6`, `ets = 0.01`, `gamma = 1`, `subsample = 0.5`. Figure 9 shows the resulting AUCs given the values of learning rate and tree depth, the ligher color and bigger size of the dot suggests higher AUC value.

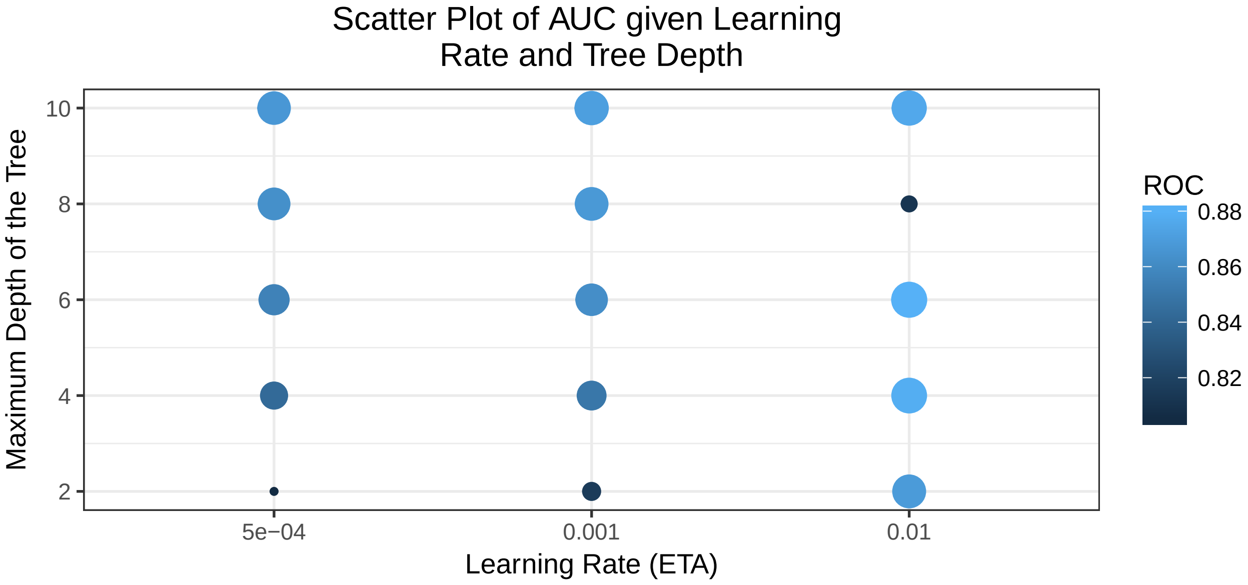


Figure 9

Then we used the selected model to predict the flight delays on testing set (30% of 2015 dataset). Figure 10 shows the histogram of the probability of departure delay given the actual flight delay information with the two classes colorcoded. Similar or even better than random forest, xgboost did a great job in separating the two classes.

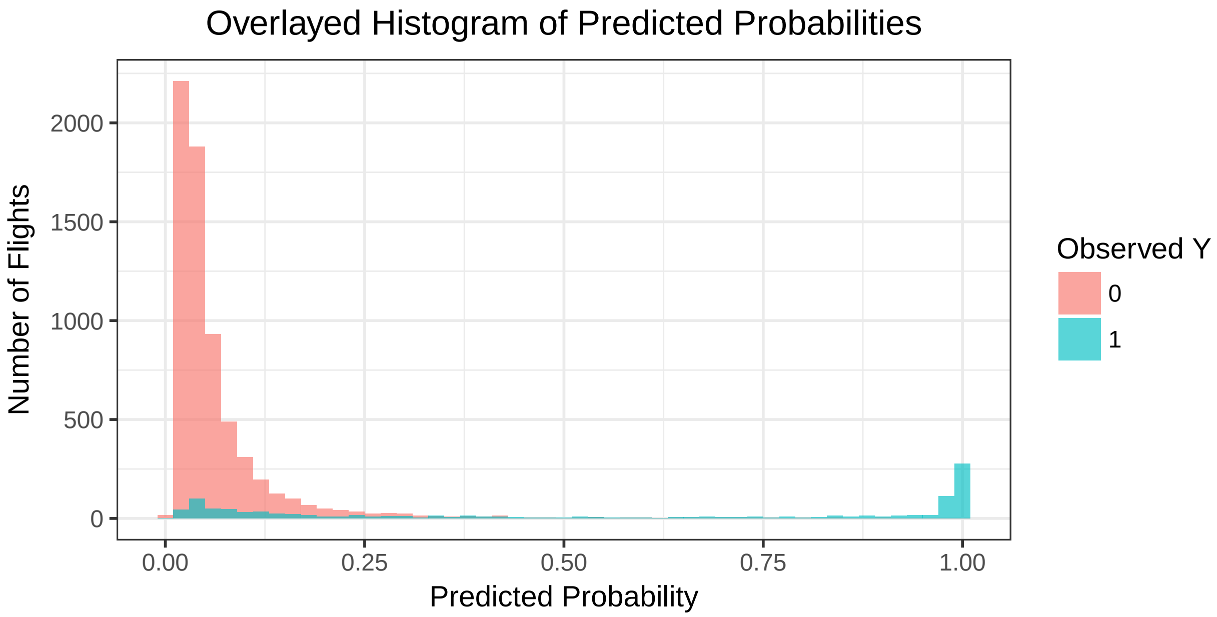
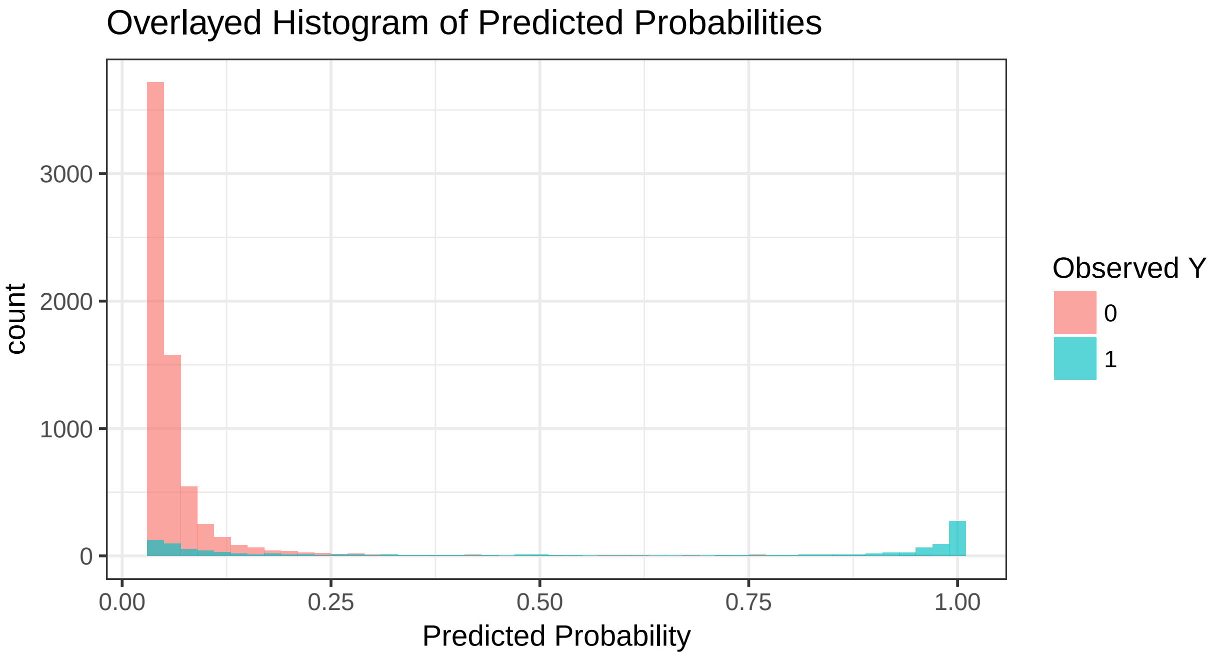


Figure 10

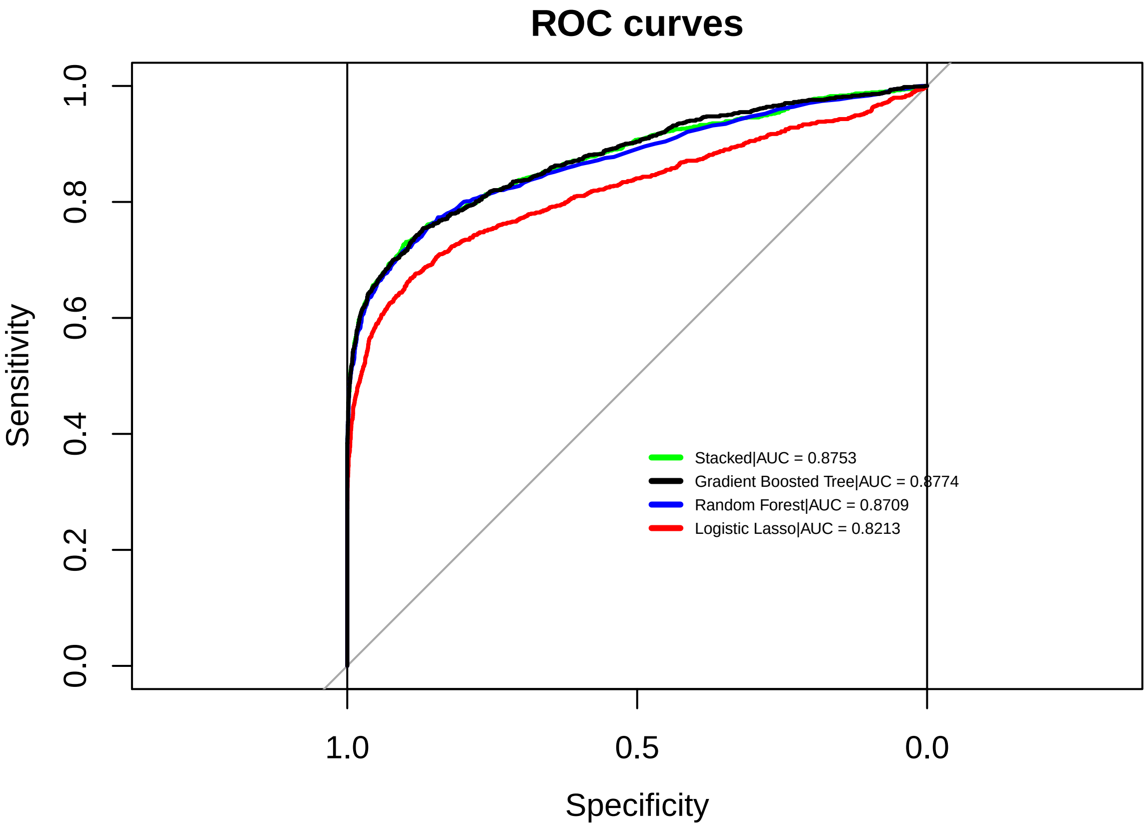
The resulting misclassificatin rate with a 50% threshold is 5.9%, which is extremely low compared to the other models and the base rate of 14%.

Stacking

Stacking is an ensemble learning technique that incorporates several classifiers into one single combined classifier. It is usually done by fitting a logisitc regression of y.test on the predicted probabilities from several classifiers. The advantage of this method is that it combines the strength of different classifiers, which produces surprisingly good predictions with low computational cost. This is project, we ran "stacked" the predicted probabilities of logisitc lasso, random forest, extreme gradient boosted trees and k-nearest-neighbors. The following histogram shows the resulting predicted probabilities.



We compared the potential predictive power of each classifier against the stacked classifier using the ROC. As one can see in the plot below, gradient boositng is surperior to any other single classifiers we have tried, and is very similar to the results of the stacked classifier.



Final predictions on guess 2016 flight dataset was made using the stacked classifier.

Discussion

In reality, flight delays are extremely difficult to predict because there are often a lot of unobservable factors that are in effect. Flight arrival time and taxi duration are two good examples of variables that would not be available in practice when predicting flight delays. Because having this information would essentially mean that we would also know whether a flight is delayed, so there is no point in making predictions in this case.

For the purpose of this project, those two "important" variables are available to make the problem easier to approach. From our results, we found that given appropriate tuning, gradient boosting can outperform random forest or any other classifers, and stacking of individual classifiers can outperform any single classifier being stacked. In the future, we should try to remove unrealistic variables and re-validate our model to see if it is useful for predicting flight delays in practice.