Predicting Flight Cancellation

University of California, Los Angeles -- Statistics 101C

December 20th, 2020

Group Name: The Confusion Matrix
Avery Robinson -- averyrose98@g.ucla.edu
Albert Cao -- caoalbert@g.ucla.edu
Lue Li -- lilue@g.ucla.edu
Mabel Sekarputri -- mabelsekar@gmail.com

Abstract

Flight cancellation is a costly inconvenience that results in an airline's loss of time, money, and passenger confidence. Understanding the components that explain these events could help airlines optimize their resources and keep their clients' trust. Our report provides a detailed account of how we took a dataset containing various information on flights across the United States and used it to build a model that predicts the cancellation status of a flight with 99.85% accuracy. We discuss the technical process behind building our model as well as insights into which factors explain flight cancellation.

1. Introduction

Our initial data set consisted of 69,225 observations and 46 variables. It is a compilation of information describing flight details, locations, airports, passenger demographics, and more, on a series of flight routes in the United States. The goal of our project is to use these variables, along with variables we constructed from outside data sources, to predict whether or not a flight is cancelled. The variables we used were either categorical or continuous, and they fell into four categories: airport related, time related, airline related, and miscellaneous. With this data, we then constructed several different models in the process of obtaining our highest accuracy classifier: a random forest model with 99.85% accuracy.

2. Methodology

A. Data Cleaning

To prepare our data for analysis, we started by ensuring each variable was of the correct class -- “numeric” or “integer” for continuous variables, and “factor” for categorical variables.

Next, we investigated the completeness of our data. We noticed that we had several variables with missing value percentages upwards of 80%, as displayed in figure 1. Our approach was as follows: impute the variables with the lowest percentages of missing values and ignore the rest. We decided that it was not worth pursuing the imputation of the largely incomplete variables for several reasons. For example, if we tried to impute “share_white” by finding the missing data on the internet, the existing data under “share_white” might be from a different source, rendering the complete variable inconsistent. Additionally, it does not make sense to impute “share_white” based on the mean or median

Figure 1

Variable	Percent Missing
Pass.Traffic	0.9%
Aircraft.Movement	2.1%
TAIL_NUMBER	9.4%
AIR_SYSTEM_DELAY	85%
SECURITY_DELAY	85%
AIRLINE_DELAY	85%
LATE_AIRCRAFT_DELAY	85%
WEATHER_DELAY	85%
share_white	97.5%
share_black	97.5%
share_native_american	97.5%
share_asian	97.5%
share_hispanic	97.5%
Median.Income	97.5%
poverty_rate	97.5%
percent_completed_hs	97.5%

because this value will truly be unique to the area it describes, thus using the mean or median would be inaccurate. If we were to find ways to consistently and accurately complete the highly incomplete variables such as “share_white”, there is no assurance they will prove significant to our model. Therefore, we turned our attention to “Pass.Traffic”,“Aircraft.Movement”, and “TAIL_NUMBER”. The tail number is a code unique to each aircraft, so we could not impute this variable without information on each specific flight. To impute “Pass.Traffic”, we investigated the variable further to see if the missing values were randomly distributed; we found that all of the missing values came from observations with the destination airport of Honolulu International Airport, so we only had to impute the value of passenger traffic for this specific airport. Similarly, all of the missing values for “Aircraft.Movement” came from observations with the destination airport of Honolulu International or Nashville International Airport, so to complete this variable, we only had to search for data on aircraft movement from these two airports.

The next stage of our data cleaning process was combining levels of categorical variables to try to transform them into simpler, more powerful variables. For “DAY_OF_WEEK”, we noticed that there were higher percentages of cancelled flights on Friday, Saturday, and Sunday than there were for the rest of the week. We tried combining categories to reflect this pattern, but all in all, this resulted in a decrease in accuracy of our model. We attempted this same kind of transformation with several variables, but without success.

B. Feature Selection

To begin our process of finding the best set of variables to predict flight cancellation, we split the variables into two categories: categorical and numerical. In order to determine whether a categorical variable was important, we created a contingency table, as seen in figure 2, of each categorical variable against “YES” and “NO” in

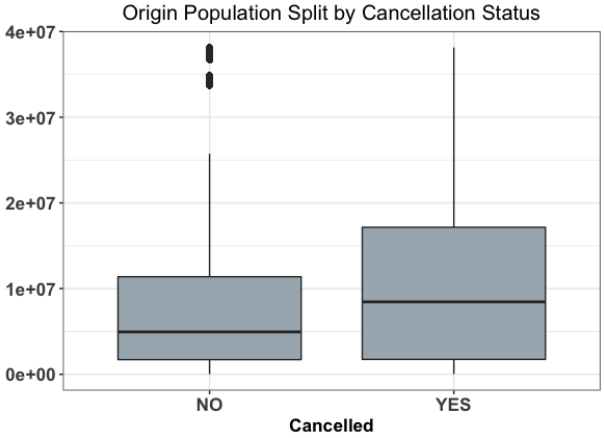
Figure 2

Month	NO	YES
January	21,234	7,203
February	18,496	11,587
March	6,355	4,350

our dependent variable, “Cancelled”. Then, using a chi-square test of associativity, we obtained a p-value that indicated whether the variable was significant or not.

Figure 3

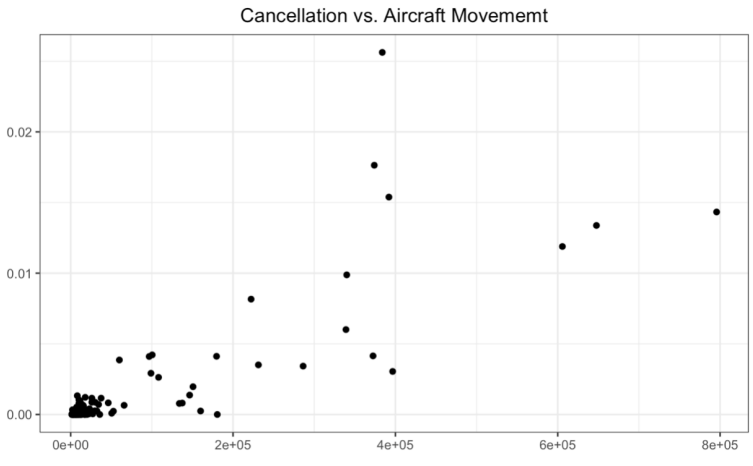
Origin Population Split by Cancellation Status



For our numeric variables, we used box plots and t-tests to determine significance. A t-test tells us whether the means of the variable’s distributions for “NO” and “YES” are statistically significantly different. If they are, the variable may prove to be a useful predictor. In figure 3, we can observe that the two distributions split by “Cancelled” for the variable “Origin_population” seem to be different, which was confirmed by our t-test results. This result tells us that origin_population may be a useful predictor.

Figure 2(b)

Cancellation vs. Aircraft Movememt



Additionally, we plotted the percentage of cancellation for each level of a categorical variable against other numeric variables. In figure 2(b), for instance, the percentage of flights cancelled at each airport was plotted against the airport’s total aircraft movement. The figure indicates that there is a positive correlation between the probability of flight cancellation and the capacity of the airport. By using illustrations generated with probability tables, we were able to transform flight cancellations from a categorical variable to a numeric variable, which further helped us in our feature selection process.

Overall, t-tests and chi squared tests gave us a good idea of which variables to start with. Depending on the classifier we were using, we used subsequent methods to edit our parameter set further. For example, with logistic regression, the model output of which variables were significant allowed us to tune our logistic regression model further. Logistic regression coefficient estimates, however, are not always stable and accurate. In extension, its estimates on which parameters are significant may be off, so we did not use this information as a concrete guide for our other methods. Additionally, when using random forests, we used variance importance plots to perfect our parameter set for that classifier, which proved to be extremely useful in increasing our prediction accuracy.

Considering all of the above, a general set of our most important features and the categories in which they fall are depicted in figure 4.

Figure 4

Airport.Related	Time.Related	Airline.Related	Miscellaneous
Destination_airport	SCHEDULED_DEPARTURE	FLIGHT_NUMBER	DISTANCE
O.City	SCHEDULED_TIME		
Org_airport_long	SCHEDULED_ARRIVAL		
	DAY_OF_WEEK		
	MONTH		

C. Feature Engineering

We found supporting data from the Federal Aviation Administration official website to enrich our dataset. With this new data, we constructed additional variables that we thought may be useful in predicting flight cancellation. Considering our findings in the feature selection section, our goal was to add variables that were either airport related or time related. In total, we added 26 new variables to our dataset.

After adding our 26 new variables, we went through a similar feature selection process as described above. In the end, we used 6 of our added variables in our models. These six variables are described in figure 5.

Figure 5

Name	Type	Description
Biggest2	Categorical	“1” if the origin airport is one of the top ten largest airports in the U.S.
tot	Numerical	The sum of # of arrival seats + # of departure seats on the given day by the origin airport
X..Delayed	Numerical	Percentage chance that a flight is delayed on a given day
Percent	Numerical	Percentage chance that a flight is cancelled on a given day
Cancellation	Numerical	Number of cancellation on a given day
arrival.Delay	Numerical	Percentage chance that the flight is delayed upon arrival at the destination airport

3. Modelling

We began our modelling process with four fundamental machine learning methods in order to gain insight into the type of decision boundary separating “YES” and “NO” in our dependent variable, “Cancelled”. The four methods were logistic regression, K-Nearest neighbors, linear discriminant analysis, and quadratic discriminant analysis. We trained each model on 70% of our data and obtained test prediction accuracies by using the model to predict flight cancellation on the remaining 30%. The prediction accuracies recorded from these methods indicated a highly complex decision boundary, which led us to try random forests and boosting.

A. Logistic Regression

Logistic regression is a classification technique centered around the logit function -- the log of the odds of an event happening; for our purposes, the event is flight cancellation. We used information from our variable exploration, as well as trial and error, to fit the most optimal logistic regression model for the data. In this scenario, trial and error meant taking out the least significant variables, according to the logistic model, and observing the effect on the prediction accuracy. Our best prediction accuracy with logistic regression was 75.66448%.

B. K-Nearest Neighbors

K-Nearest neighbors (KNN) is a classification technique that forms a decision boundary based on the classes of the K nearest neighbors surrounding each point. The optimal value of K for a particular data set tells us a lot about the decision boundary. For our data, we found the optimal value of K to be 1. We ultimately chose to use K=3 as a step against overfitting, which produced an accuracy value of 95.50751%. In KNN, the higher the value of K, the more confident we can be in a linear decision boundary. Given that extremely low values of K resulted in the best accuracy ratings, we can conclude that our decision boundary is highly non-linear.

C. Linear Discriminant Analysis and Quadratic Discriminant Analysis

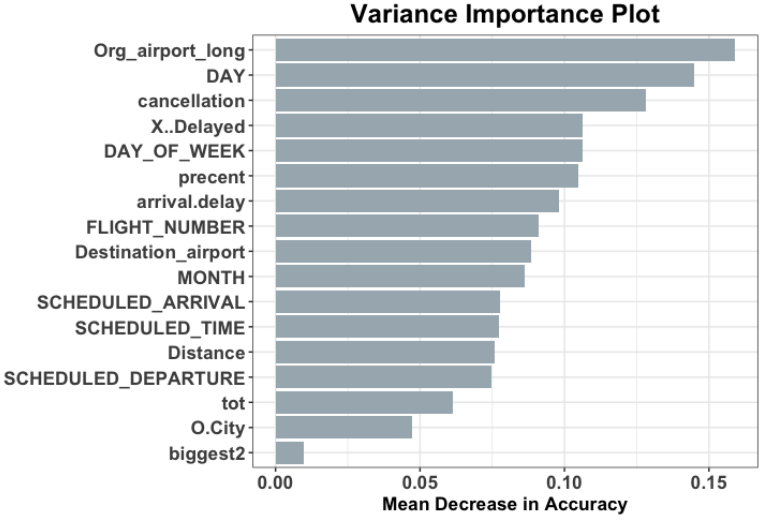
Linear discriminant analysis and quadratic discriminant analysis reduce the dimensions of the data in a way that maximizes separability between the distinct classes of the dependent variable. Similarly to logistic regression, linear discriminant analysis will perform well when the true decision boundary of the data is linear. Quadratic discriminant analysis, on the other hand, will perform well when the true decision boundary of the data is quadratic. Our best accuracy using LDA and QDA were 80.52292% and 79.0735%, respectively.

Out of our four original techniques, KNN (K=3) had the highest accuracy. From this, we conclude that the decision boundary is highly non-linear. We can also conclude that the decision boundary is non-quadratic -- if it were, QDA would have performed much better. We attempted a voting method between these four original techniques, which did not improve our accuracies. Overall, our results with these methods led us to try random forests and boosting, two techniques that do well with complex decision boundaries.

D. Random Forests

A random forest is a collection of uncorrelated decision trees, each made from a bootstrapped sample of the training data and a randomly selected set of variables. Each tree will classify a specific observation into a certain class, and the class with the highest number of votes wins. Using this powerful machine learning technique with the parameters at default values, we obtained a promising prediction accuracy upwards of 91%. As mentioned in the feature selection section, we were able to use variance importance plots, such as the one depicted in Figure 6, to efficiently optimize our variable set for our random forest classifier. Once we had our best set of variables, we focused on tuning the parameters of the model. Using loops, we found the optimal values of *ntree* and *mtry* to be 500 and 5 respectively. *Ntree* represents the number of trees in our "forest", and *mtry* is defined as the "number of variables randomly sampled as candidates at each split" (R documentation). With these optimizations, we were able to get our best accuracy of 99.841%. Considering the high performance of our random forest classifier, we can be confident that the true decision boundaries in our data set are hyper-rectangular and highly complex.

Figure 6



E. GBM Boosting

Next, to build upon our random forest success, we tried boosting. Boosting is a type of random forest classifier in which each tree is built upon the errors of the previous trees. The degree to which the errors of the previous tree are emphasized in the new tree is controlled by the learning rate of the algorithm -- a manipulatable parameter. The parameters we tuned were *n.tree*, *shrinkage*, and the cutoff point, which represent the total number of trees grown, the

learning rate of the algorithm, and the value between 0 and 1 that separates the decision between classifying the observation into one class or another. Again, we used loops to optimize. Our best boosting model had an accuracy rate of 99.29218% with 8780 trees, a learning rate of 0.525, and a cut-off value of .912. While this method did quite well, our random forest classifier still did better. A summary of our results can be seen in figure 7.

Figure 7

Method	Accuracy
Logistic Regression	75.66448%
KNN (K=3)	95.50751%
LDA	80.52292%
QDA	79.0735%
Random Forest	99.841%
Boosting	99.29218%

4. Results

Our modelling process not only allowed us to predict flight cancellations with 99.841% accuracy, but it also gave us further insight into the decision boundary of “Cancelled”, our dependent variable. As mentioned above, our accuracy ratings tell us that the boundary is highly nonlinear, nonquadratic, and hyper-rectangular. This is supported from the lower accuracies in methods that do well with linear boundaries, such as logistic regression, LDA, and KNN with a large K. It is also supported by low accuracy rating with QDA, which does well with quadratic boundaries. Finally, it is confirmed by our high accuracies with KNN (K=3), random forest, and boosting, which all do well with extremely complex boundaries.

5. Discussion and Limitations

Our discovery that the decision boundary between “YES” and “NO” in our dependent variable is highly non-linear helps point us in the direction of which machine learning techniques we could

explore to improve our prediction accuracy even further: Gradient Boost, XGBoost, SVMs, and Bagging. Given the nature of these algorithms, we would expect these classifiers to work well with our data. Each machine learning method, however, has its short-comings. For example, random forests tend to over fit, leading to higher variance; KNN with a low K also tends to overfit. Therefore, combining the highest accuracy models into a voting system will alleviate some of the errors introduced by the short-comings of these methods, as well as produce a more robust classifier. Our initial voting method had low accuracy because we used our four original classifiers -- logistic regression, KNN (K=3), LDA, and QDA -- to vote. We expect that using our highest performing classifiers instead, such as random forests, boosting, and potentially Gradient Boost, XGBoost, SVMs, and Bagging, would lead to a much more successful voting classifier.

In addition to trying new classifiers, a possible limitation of our method was ignoring variables that were highly incomplete. Spending more time imputing “AIR_SYSTEM_DELAY”, ”AIRLINE_DELAY”, and other variables we initially ignored may prove beneficial to our model. Finally, when we combined our new data with the existing data, we merged based on destination airport. We may find that some of our added variables would prove more significant if, instead, we merged on origin airport.

6. Conclusion

Our process of obtaining our final model came with machine learning lessons and bits of insight into flight cancellation patterns. To start, we learned the importance of accurate variable imputation and how sometimes the fastest approach will have a detrimental effect on the variable’s significance. For example, when imputing “Pass.Traffic", we needed to find the passenger traffic figure for Honolulu International Airport. We found many different figures on the web, and had to ensure, via cross-checking, that the figure we used was consistent with the

existing “Pass.Traffic” data. Additionally, we learned the importance of having a plethora of machine learning techniques in our toolbox. Comparing accuracy ratings across techniques informed us on the nature of the decision boundary and pointed us in the direction of more successful techniques to implement for this particular problem. Finally, through the use of many models, we saw patterns in which variables were consistently helpful in predicting flight cancellation -- month, day, time of arrival, time of departure, the origin airport’s longitude, the size of the destination airport, and more. This information could help airlines make informed, data-driven flight schedules that work to save resources and preserve their punctual reputation.

References

Gohel, David. “Flextable Overview.” • *Flextable*, davidgohel.github.io/flextable/articles/overview.html.

Liaw, Andy. “RandomForest.” *Function* | *R Documentation*, www.rdocumentation.org/packages/randomForest/versions/4.6-14/topics/randomForest.

Passenger & All-Cargo Statistics, 13 Oct. 2020, www.faa.gov/airports/planning_capacity/passenger_allcargo_stats/.