Prediction of U.S. Domestic Flights' Cancellation Status

21 December 2020

Miko Farin

Jack Krupinski

Wonjae Lee

Matt Turk

Abstract

Since the dawn of computing, the amount of data in the world has continued to increase exponentially, especially with the birth of the internet, and, in more recent years, social media. Statistical modeling has become increasingly relevant, and as a result statisticians are highly valued due to their ability to use statistical learning to identify trends and make classifications, which has proven to be extremely beneficial to nearly all fields and industries. Our team has applied several of these methods to predict flight cancellations using data collected by the U.S. Department of Transportation on over 69,000 domestic flights. After applying several methods of data analysis, transformation, and classification models, we created a random forest model which predicted flight cancellations with an accuracy of 99.7%.

Introduction

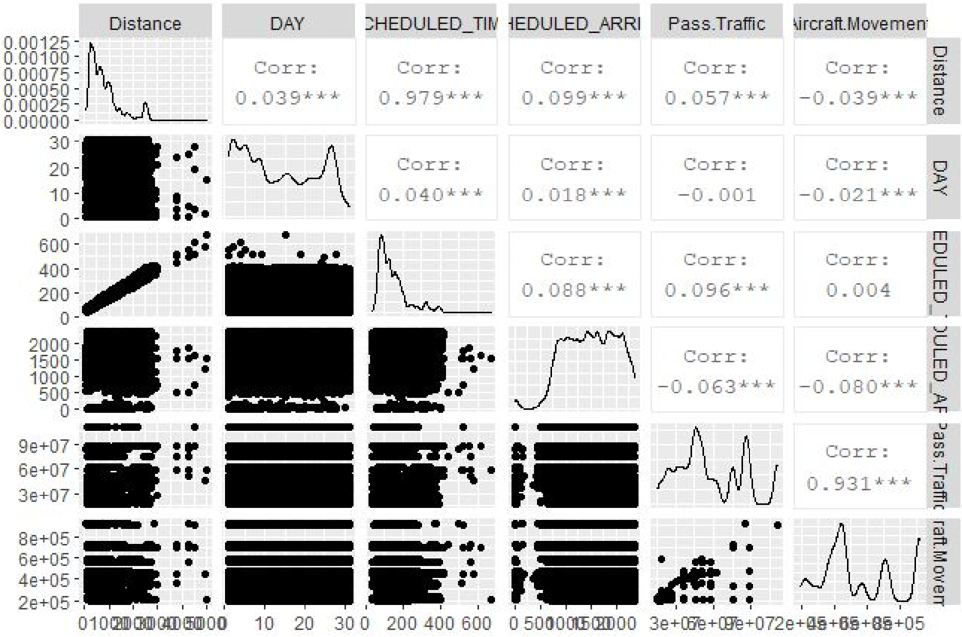
In 2019, the United States aviation industry supported over 10,000 jobs and made up 7.3% of the nation’s total GDP, yet an estimated 350,000 flights scheduled by U.S. airlines ended up being cancelled. These cancellations forced customers to change travel plans, impaired their trust in the airline industry, and consequently reduced airline profitability. Therefore, the ability to accurately predict flight cancellations would be beneficial both for travelers and airline operators, as the amount of resources wasted on cancelled flights could be minimized.

In order to develop a model to predict flight cancellations, several statistical learning methods may be applied to a large data set. The training data consisted of 69,225 flight observations and included information such as origin and destination location, date, scheduled arrival and departure times, passenger demographics, and a number of other features. The outcome variable—cancellation status—is a binary categorical variable with levels “YES” and “NO,” while the predictors include both numerical and categorical variables. Upon developing a model, its accuracy was tested on a separate data set which did not include the response variable, namely, whether or not the flight had been cancelled. The model's predictions were then compared to the actual flight cancellation data, and we calculated the model's classification accuracy rate. The accuracy rate of the model served as the metric that we strove to optimize throughout or procedure. Each method, or type of model, has its own advantages and disadvantages, but the goal of this project was to maximize the accuracy of the chosen model on the testing data set.

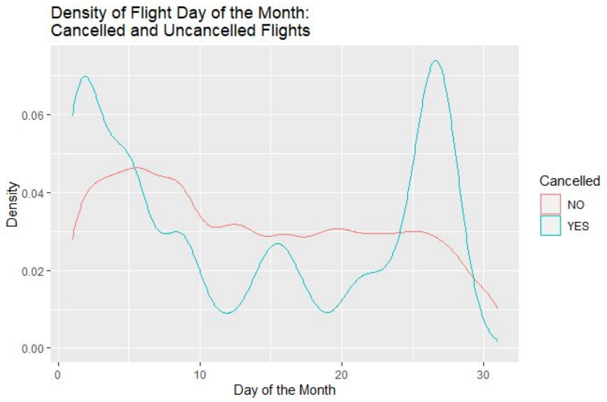
Methods

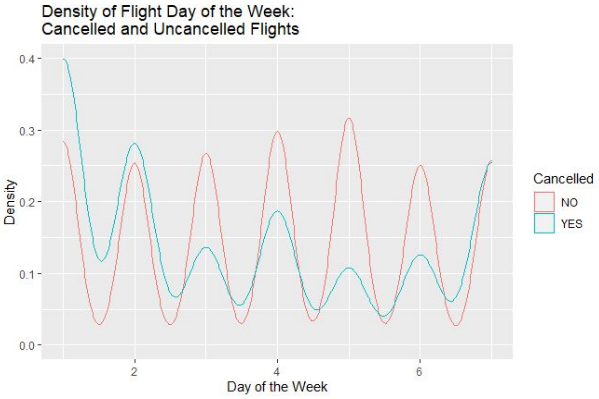
Exploratory Data Analysis

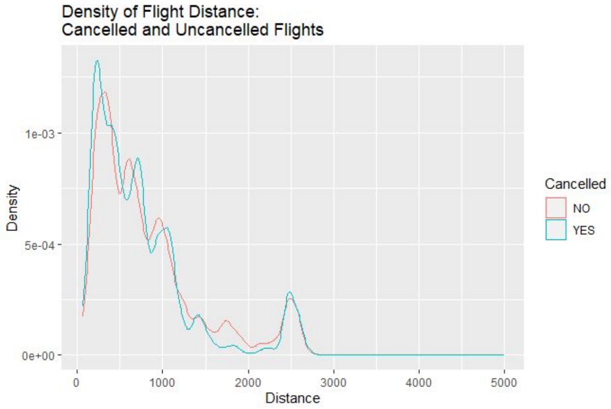
To begin our exploratory data analysis, we used the ggpairs function to visualize relationships and correlations between predictors. The ggpairs function has the benefit of displaying both the individual distributions of predictors and the interactions between predictors on a single plot. For example, the plot shows that the distribution of the "Distance" variable is right skewed and also very highly correlated with the "Scheduled Flight Time" variable. We also note that most of the features do not follow a normal distribution. For viewing accessibility, we include only the ggpairs plot for six predictors, although we applied the function to all predictors.

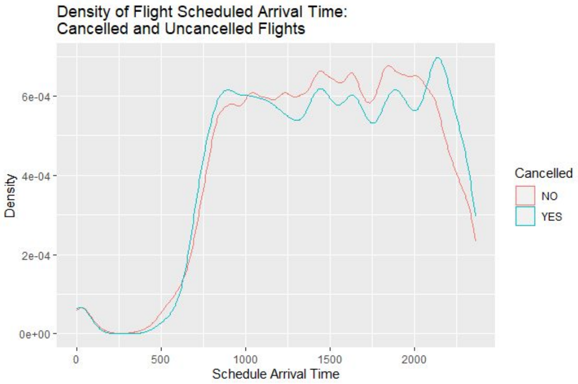


The next goal of our exploratory data analysis was to identify variables that successfully separate cancelled flights from uncancelled flights when graphed on a density plot. If a predictor's density plot has clear demarcations between cancelled and uncancelled flights, then the feature has potential to be a useful variable in the classification models. The "Day of the Month" and "Day of the Week" variables produced the greatest separation between cancelled and uncancelled flights on their respective density plots. A greater proportion of cancelled flights occurred early in the week compared to uncancelled flights, and cancelled flights tended to occur more often at the beginnings and ends of months. While these two density plots adequately separated the levels of our outcome variable, the majority of the predictors were less successful in differentiating between flights of different cancellation status. The density plots of these less effective variables show considerable overlap for both cancelled and uncancelled flights. We display the density plots of the variables "Distance" and "Schedule Arrival Time" as examples of predictors that fail to separate the outcome variable. Note that most of the density plots for our features exhibited the characteristics of less effective predictors.





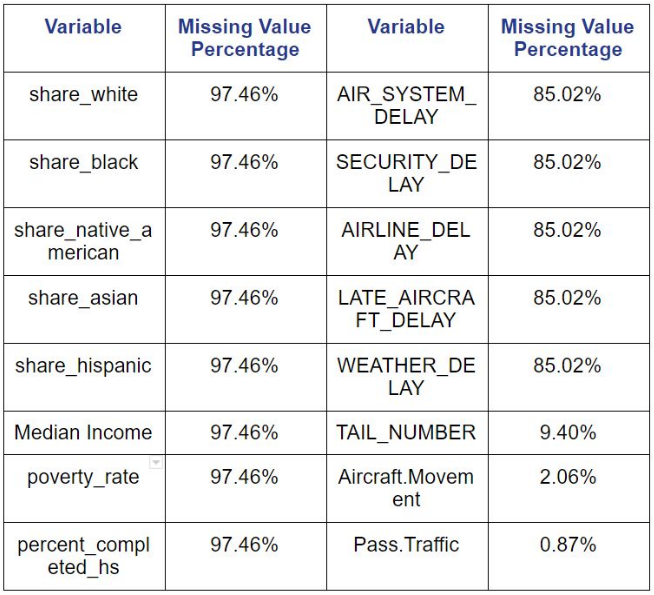




Missing Values and Data Trimming

The data has 44 predictors with a combination of both numeric and factor variables, and we note that the predictors, “Destination_airport”, “O.city”, “O.state”, “Origin_airport”, “Origin_city”, “Destination_city”, “AIRLINE”, and “TAIL_NUMBER” are factor variables with multiple levels. These predictors have a high possibility to make the statistical models more complex and even result in meaningless prediction models. Thus, we decided to transform the predictors into numeric variables where applicable. For example, the longitudes and latitudes of the destination and origin airports were used as numerical variables that encode nearly the same information as the factor variables that list each airport as its own level. Additionally, we dropped some factor predictors altogether due to their duplicacy. Specifically, “O.city”, “O.state” and “Origin_city” contained the same information. “O.city” contains the information of origin cities, “O.state” contains the information of origin states, and “Origin_city” contains the combined information of both city and state. To simplify and maximize efficiency of the data set, we chose to include “Origin_city” only and drop “O.city” and “O.state.” For “TAIL_NUMBER”, the variable includes missing values and will be discussed in a later part of this section. As discussed previously, we used the numerical longitude and latitude variables to incorporate geographic features, so we could omit the “Destination_airport”, “Origin_airport”, “Origin_city”, “Destination_city" and “AIRLINE” factor variables. In the next section, we discuss our creation of cancellation rate variables for airports and cities that further capture the susceptibility of particular locations to cancellation.

In the process of data cleaning, we discovered that 16 out of 44 predictors contain missing values, and we computed the percentage of missing values to total observations in the training set. From the table below, the predictors Aircraft.Movement and Pass.Traffic have a relatively low percentage of missing values, but the other predictors have significant missing value percentages that render the variables unhelpful for classification. They do not contain enough information to predict the flight cancellation and may even affect the model negatively if they are considered.

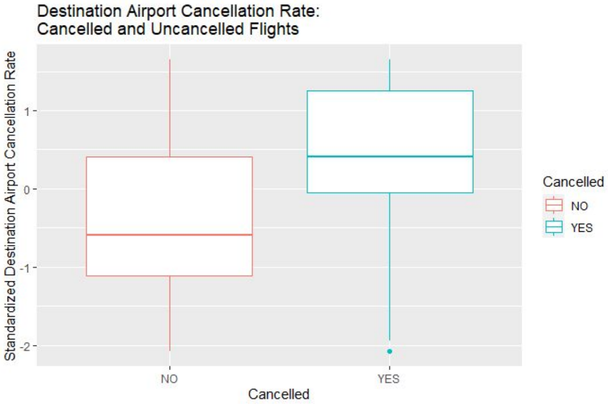


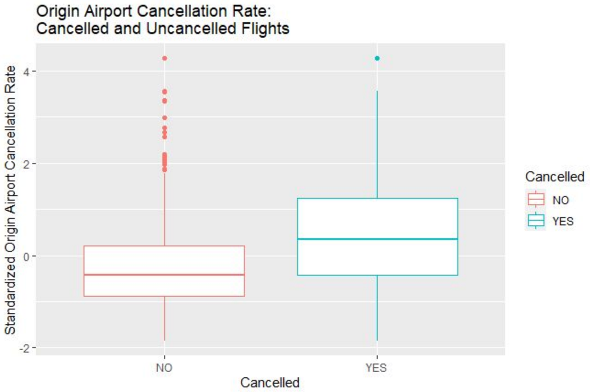
Variable	Missing Value Percentage	Variable	Missing Value Percentage
share_white	97.46%	AIR_SYSTEM_DELAY	85.02%
share_black	97.46%	SECURITY_DELAY	85.02%
share_native_american	97.46%	AIRLINE_DELAY	85.02%
share_asian	97.46%	LATE_AIRCRAFT_DELAY	85.02%
share_hispanic	97.46%	WEATHER_DELAY	85.02%
Median Income	97.46%	TAIL_NUMBER	9.40%
poverty_rate	97.46%	Aircraft.Movement	2.06%
percent_completed_hs	97.46%	Pass.Traffic	0.87%

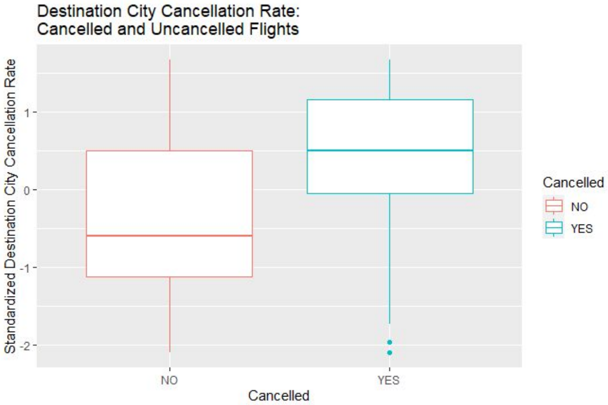
Next, we imputed missing values for the Aircraft.Movement and Pass.Traffic predictors using the `mice` package (Multivariate Imputation via Chained Equations). Specifically, we used random forest imputation—a univariate imputation method in the package—that implements Breiman's random forest algorithm (based on Breiman and Cutler's original Fortran code) for classification and regression. After imputation and trimming, the cleaned data consisted of only numeric variables.

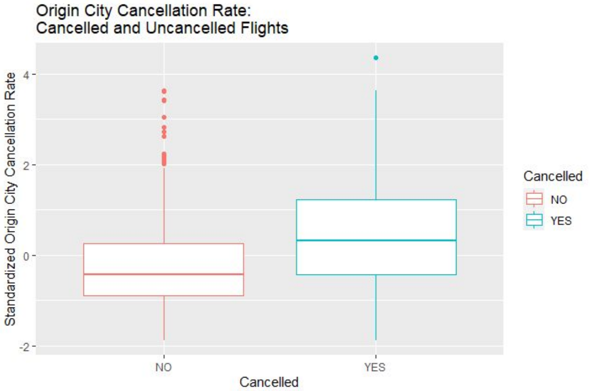
Variable Creation

In our exploratory data analysis, we found that most of the given predictors in the data set were not particularly effective at separating cancelled from uncancelled flights. This motivated us to create our own features that clearly differentiate between outcome variable classes. We produced new variables that represented the cancellation rate for origin city, destination city, origin airport, and destination airport by dividing the number of cancelled flights by the total number of flights for each locale. By creating a unique cancellation rate for each city and airport, we also justify the omission of the airport and city factor variables, which yields a data set that consists entirely of numerical variables. Unsurprisingly, there is a high correlation between destination city cancellation rate and destination airport cancellation rate, as well as between origin city cancellation rate and origin airport cancellation rate. By construction, our created variables do a good job of separating cancelled and uncancelled flights on the training data. We hoped that our created variables would also be useful predictors in the classification models when applied to the testing data.





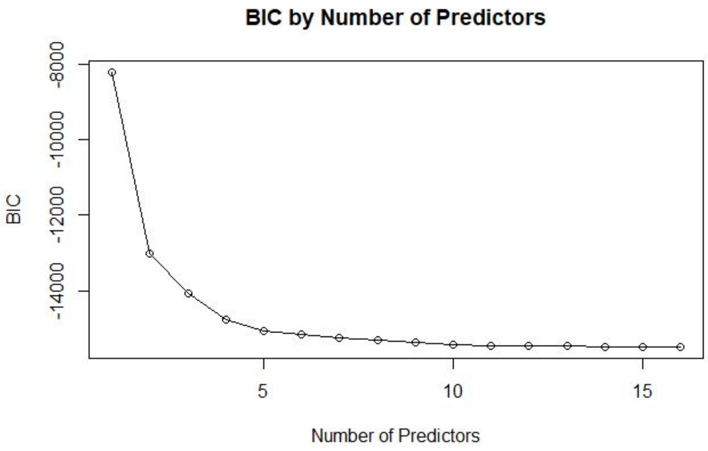




Feature Selection and Variable Importance

First, we used the “stepclass” function from the `klaR` package for feature selection because it is built specifically for classification models. The function uses ten-fold cross validation and adds predictors to an LDA model until the prediction correctness rate fails to increase by 1%. The stepclass function ultimately selected a model with two predictors: destination airport cancellation rate and origin airport cancellation rate. Note that these two predictors were created during the data cleaning process and were not included in the original data set.

Next, we used traditional stepwise feature selection with the regsubsets function and BIC as our criterion. From the graph below, we see that BIC stops decreasing significantly after around five predictors and plateaus entirely beyond ten predictors. Although we identified certain predictors as most important, we included 16 predictors in our final model. Since our objective was to maximize prediction accuracy on Kaggle, we were willing to trade off model simplicity for increased accuracy. Nevertheless, identifying the most important features gives us insight and intuition into which predictors will likely play prominent roles in the classification models.



Choosing a model

After preparing the data and creating new variables, we begin the model creation process. In order to create the appropriate model, we selected a few criteria that we wanted to achieve: to be effective with the large data set we have, to keep the model simple and efficient, while still being able to achieve a high accuracy rate on the training and testing data, and lastly to minimize the bias-variance trade-off. Before fitting the model, we scaled our numerical variables to be more “versatile” and identified the response variable as ‘Cancelled’, which is a factor consisting of two levels: “YES” and “NO”. For the purposes of testing our potential models, we used the random sampling methods to split the data into 80% new training data and 20% new testing data. Then, we used our models to test the accuracy on the new testing data to effectively evaluate the models performance, minimize effects of data discrepancies and better understand the characteristics of the model.

Linear Discriminant Analysis

Linear discriminant analysis is an effective supervised learning method for dimensionality reduction in a dataset, as it is able to maximize the separation between classes. LDA is able to find the features in a dataset that separates classes by utilizing Bayes’ Theorem to estimate the probabilities of an observation belonging to a certain class. The advantages of the model are that it avoids overfitting and reduces computation costs compared to more complex methods.

For the project, the LDA model’s main purpose was to be a pre-processing step for our later classification models, acting as a baseline model to extract information about the variables and features that were included in our cleaned data set. As a result, we were not concerned with the prediction accuracy of this model. To be more specific in our method, we used all of the 16 predictors from our cleaned Flight data set to build a LDA approach model to find the axes or predictors that would maximize the separation between the two classes, “Cancelled” or “not Cancelled” flights. From there, we looked at the predictors that caused the greatest reduction in our misclassification error rate and moved forward in using them for our stronger algorithms and classification methods.

Neural Network

Neural networks utilize an algorithm known as gradient descent to find local minima of a loss function, which is similar to the concept of minimizing the residual sum of squares in a simple linear regression problem. Minimizing the loss function maximizes the accuracy of the network on the training data set, and therefore the prediction power of the neural network depends on the size of the training data set (a larger training set is expected to reduce bias), and how accurately the available data can be fed into the network (all input data must be represented as a decimal between 0 and 1 for a neural network). So while neural networks are extremely powerful when it comes to numerical data, in a data set like ours with many categorical features, the neural network has no way of incorporating these values into its calculations. The lower accuracy of the network on the testing data compared to the training data is evidence that in this case, the nature of the neural network model caused it to be heavily biased towards the training data.

Another drawback of neural networks is their computation time. The network we could run with our limited computation power had only one hidden layer, which means there was greater room for error when compared to a network with two or more hidden layers. However, a more complex network like this would have taken much longer to run and it is unlikely that RStudio would have supported these computations.

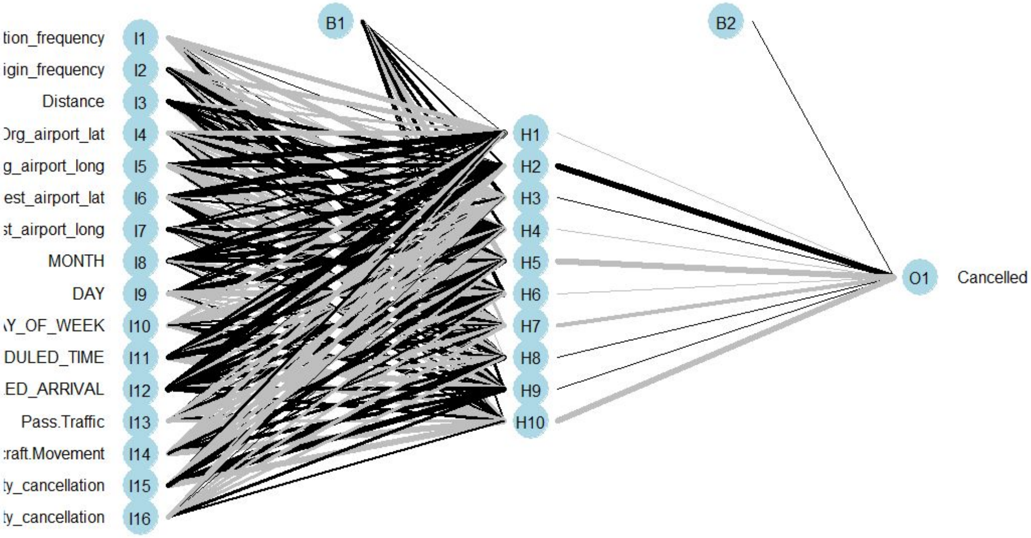


Figure x: Neural network diagram, one hidden layer

Boosting

Boosting is another advanced machine learning algorithm we chose to experiment with. Like the neural network, boosting involves an algorithm that minimizes a loss function. However, rather than using gradient descent to compute a weight matrix, boosting uses basis functions (y = 1, y = sin(x), or y = cos(x)) to model the response. The main idea of boosting is that it begins with training a decision tree and grows multiple trees sequentially, using information about the previously grown tree to increase the weights of those observations that are difficult to classify and lower the weights for those that are easy to classify. The information that is used in the process is the change in prediction error: if the change in the prediction for an observation causes a large enough drop in classification error, the weight of that particular observation will be large. On the other hand, if there is little to no change, then the weight will be small.

For our project, we used the library `gbm` to create our boosting model. When fitting the model we used the following parameters to tune our boosting model:

- **distribution** = “bernoulli”, since it is a classification problem
- **n.trees** : number of trees in the model. There can be over-fitting so we were able to use cross-validation to see the best number of trees
- **shrinkage =** 0.01, set to default value; not messed with since we had enough time to weight for a small learning rate
- **splits** = default value

After tuning our model, we then began testing using similar methods for our other modes like using the split data, changing the parameters occasionally, and performing predictions and submitting to Kaggle. Finally, using the steps outlined above we calculated our average classification error rate and settled for the best average error rate.

Random forest

When a random forest model is created, the computer runs a set number of decision trees on the data (in our case, 100), and creates the final model based on a bootstrap aggregate of the tree models. A decision tree is similar in nature to a flowchart, where each “branch” splitting represents a test on an attribute of the data. After a certain number of branches (which number is determined by the model itself), the decision tree ends in “leaves” which are the classification predictions based on how each entry in the data flows through the tree. With the random forest, 100 tree models were created based on random samples of the data, which determined the decision boundary of the model.

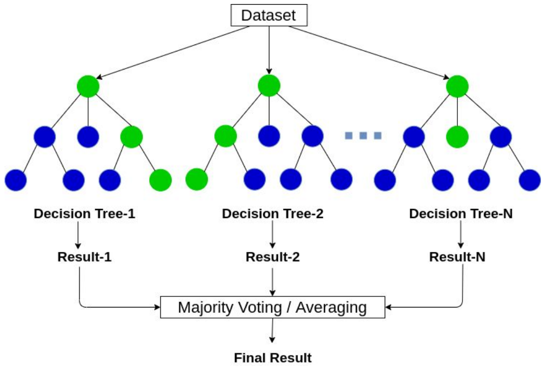


Figure x: Random forest model visualization

Results

LDA Confusion Matrix	NO	YES
NO	7923	2384
YES	1290	2248

Neural Network Confusion Matrix	NO	YES
NO	8367	966
YES	783	3729

Boosting Confusion Matrix	NO	YES
NO	8597	1047
YES	616	3585

Random Forest Confusion Matrix (16 predictors)	NO	YES
NO	9182	11
YES	31	4621

LDA	NO	YES
NO	7923	2384
YES	1290	2248

Boosting	NO	YES
NO	8597	1047
YES	616	3585

Neural Network	NO	YES
NO	8367	966
YES	783	3729

Random Forest	NO	YES
NO	9182	11
YES	31	4621

Model:	LDA	Neural Network	Boosting	Random Forest
Accuracy:	73.46%	87.37%	87.99%	99.70%

With the random forest model, we were able to achieve nearest to maximum accuracy with only 8 predictors, while the other models required at least twice as many to achieve a similar testing accuracy.

Discussion and Model Limitations

As seen in the results, the accuracy rates of different models differ considerably. In the

Linear Discriminant Analysis model, or LDA, the accuracy rate is about 73%, which is a relatively low score compared to other models. LDA computes a linear combination of features that characterizes two or more classes through dimensionality reduction. However, the limitation of LDA is that it assumes a one-dimensional normal distribution of each predictor in the data. Since the density plots in the data exploration section indicate that the predictors do not closely follow the Gaussian distribution, the LDA model's efficacy will be limited. Thus, we have chosen this model as the baseline model to extract information about the variables and features. As a result, the accuracy rate of the model is the lowest among all the models used.

Neural networks and Boosting have almost identical accuracy rates of about 87%. The limitation of a neural network is its nature of heavy bias towards the training set. During the model evaluation, its training accuracy score and testing accuracy score have a significant difference, with the training score exceeding the testing score. Thus, either an even larger data set or the addition of another hidden layer may be needed to improve the accuracy, which would be computationally expensive. Similarly, boosting also has limitations in its algorithm. Boosting corrects the errors of each sequential tree, but it is vulnerable to outliers in the data. Thus, this may reduce the accuracy rate. Lastly, although our random forest model recorded the highest accuracy score, we recognize that random forests can exhibit overfitting behavior by creating excessive terminal nodes.

Finally, we acknowledge that there are limitations in our project. Although four powerful models are implemented in this project to compare their accuracy rates, there exist other different models that could produce comparable accuracy scores. Additionally, the focus of our project was purely on predicting flight cancellations using the classification model. Given that our optimal random forest model lacks interpretability, we cannot make causal conclusions about the underlying reasons for flight cancellations. Similarly, our project did not specifically analyze the reasons that certain cases were misclassified. In order to improve upon our model accuracy, we may need a richer data set that includes fewer missing values about flight delays and passenger demographics. Overall, the applicability of our model is somewhat limited by its lack of clear-cut interpretability.

Conclusion

From the process of data mining, we realized the importance of the data cleaning process on predicting the flight cancellations. The analysis on data preparation and data cleaning enabled us to transform predictors with multiple levels to numerical values, which are simple and effective in implementation of our models, and determine which predictors to use in the model. After implementation of the four models (LDA, Neural Networks, Boosting and Random Forest), we have got the accuracy rates of about 73%, 87%, 88% and 99% respectively. By getting 99% accuracy rates in random forest, we believe that we have succeeded in the meaningful process of data preparation and analysis.

However, we acknowledge the limitations of our paper. Our paper is mainly focused on finding the model with the highest accuracy rate. There must be further research and analysis on interpreting the misclassification cases of each model. By doing so, we will not only gain the understanding of each model in depth but also improve the accuracy rates by adjusting misclassified cases in our model. Therefore, in the future, the investigation on the misclassified cases are needed to reduce the number of misclassified cases and acquire the model that could precisely predict the flight cancellation.

We recommend that future studies attempt to collect more complete data on passenger demographics and flight delays, as our data set contained mostly missing values for these predictors. In particular, flight delays may be a particularly useful predictor of flight cancellations. Future studies may strive to produce more interpretable models that improve understanding about the underlying causes of flight cancellation, as our desire to optimize classification accuracy detracted from model simplicity and accessibility. Nevertheless, our project has successfully identified a set of predictors that can be used to produce a highly accurate classification model, so this can serve as a starting point for further investigations into flight cancellation classification.

References & Acknowledgements

https://www.sciencedirect.com/science/article/pii/S1877050919320241

https://arxiv.org/ftp/arxiv/papers/1903/1903.06740.pdf

https://stat-or.unc.edu/wp-content/uploads/sites/182/2018/09/Paper3_MSOM_2012_AirlineFlightDelays.pdf

https://www.airlines.org/data/

https://www.transtats.bts.gov/HomeDrillChart.asp

Peterson, Everett B., et al. “The Economic Cost of Airline Flight Delay.” *Journal of Transport Economics and Policy*, vol. 47, no. 1, 2013, pp. 107–121. *JSTOR*, www.jstor.org/stable/24396355. Accessed 19 Dec. 2020.

https://cran.r-project.org/web/packages/mice/mice.pdf