Predicting Flight Delay Using Classification

CDS-303: Scientific Data Mining

12/15/2021

Team Members:

Beatriz Buquerin

Jamila Clayton

Gunnar Cukor

Mason Goss

Emma Resmini

Katie Stewart

Contents

[Executive Summary 3](#_Toc90499568)

[Business Problem 3](#_Toc90499569)

[Analytic Problem 5](#_Toc90499570)

[Project Plan 6](#_Toc90499571)

[Data 7](#_Toc90499572)

[Data Cleaning 7](#_Toc90499573)

[Exploratory Data Analysis 8](#_Toc90499574)

[Methodology 11](#_Toc90499575)

[Modeling and Testing 12](#_Toc90499576)

[Model type and justification 12](#_Toc90499577)

[Additional Data Preparation 13](#_Toc90499578)

[Hyperparameter tuning 15](#_Toc90499579)

[Testing and results 16](#_Toc90499580)

[Binary Classification Model 16](#_Toc90499581)

[Multi-class Classification Model 17](#_Toc90499582)

[Model output 18](#_Toc90499583)

[Model assumptions and limitations 19](#_Toc90499584)

[Evaluation 19](#_Toc90499585)

[Deployment and Maintenance 19](#_Toc90499586)

[Plans for Deployment 19](#_Toc90499587)

[Possible Deployment Pitfalls 20](#_Toc90499588)

[Monitoring 20](#_Toc90499589)

[Conclusion 21](#_Toc90499590)

[Future Work 21](#_Toc90499591)

[Appendix A: Timeline 23](#_Toc90499592)

[Appendix B 25](#_Toc90499593)

[Appendix C 26](#_Toc90499594)

[Binary classification model results 26](#_Toc90499595)

[Multi-class Classification model results 27](#_Toc90499596)

# Executive Summary

The client’s employees utilize air travel for offsite meetings with customers, and have recently experienced issues with flight delays. This ultimately has led to missed meetings and damaged customer relations. Our team identified that the primary issue lies with the client’s travel department, where staff members unintentionally book flights with airlines that tend to be delayed.

This report outlines the process and methods used to determine which airlines are better suited for the client’s needs. Our method employs supervised classification via a random forest algorithm to determine the on-time rate for United States (U.S.) domestic airlines, as well as determining the likely length of delay.

Two models were built for complementary purposes. The first model utilized binary classification and provides valuable output detailing the probability of a delay occurring for a given airline. The second model was a multi-class classification model which utilized five classes of delay lengths, ranging from no delay to 90+ minutes, and produces a prediction indicating how long the delay may last. Historical data from 2014-2018 was used. The data was prepared the same way for both models, and testing was uniform for both as well, which included statistical metrics and k-fold cross validation. Testing showed that the binary classification model is successful and reliable for use. Accuracy scores were above 90%, and other metrics tested returned promising numbers.

The outputs of the two models supply greater confidence to the client’s travel department when booking flights for staff. This in turn gives the client the opportunity to repair their damaged business relationships and strengthen customer relations. Additionally, this will reduce unnecessary monetary expenses incurred by booking flights for meetings that are ultimately cancelled. For the client’s employees who are travelling to these meetings, the models’ prediction will allow them to use their time more effectively. Our proposed solution will allow our client to focus less on putting out fires and more on doing what they do best, the work their customers hired them to do.

# Business Problem

The client’s employees utilize air travel on a weekly basis for offsite meetings with customers. The client has recently experienced issues with flight delays, resulting in missed meetings, loss of revenue from billable hours, increased travel expenses, opportunity costs attributable to delays, and damaged customer relations.

Flight delays and cancellations are experienced by all commercial airlines. Additionally, delays and cancellations have negative impacts on the U.S. economy. Inefficiencies within air transportation increases the cost of business for other sectors, which makes the associated businesses less productive.[[1]](#footnote-1) Our objective is to minimize missed meetings due to delayed flights. By optimizing flight schedules, our goal is to positively impact corporate synergy and ideation, while simultaneously reducing expenditures and increasing productivity.

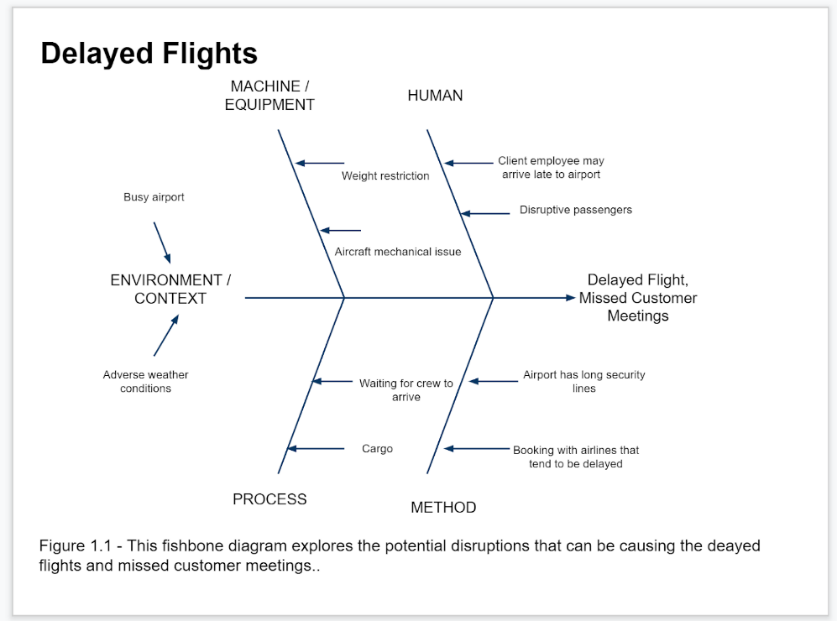


Figure 1: This fishbone diagram explores the potential disruptions that cause delayed flights and missed meetings.

The crux of the problem is that the client doesn’t know if there will be a delay until their employees are already at the airport. Our project will determine which airline carriers have the least amount of delays and how long a delay usually lasts for each airline.

A flight delay is defined as a flight departing and/or landing later than its scheduled time. The flight schedule includes the time to taxi out, airborne time, and the time to taxi in, which is shown in Figure 2. Following the definition set by the United States Department of Transportation (DOT), a flight is delayed if it is 15 minutes later than its scheduled time.

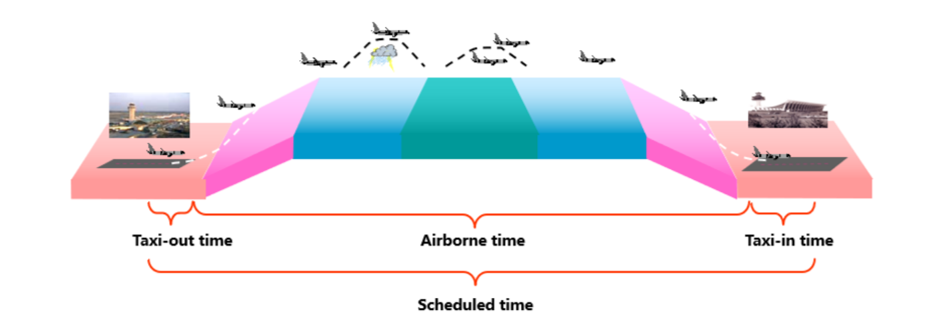


Figure : Flight route model[[2]](#footnote-2). A flight schedule includes taxi time with airborne time.

Delays happen for a multitude of reasons, including airline operations, tarmac traffic, air traffic, weather, security, etc. The DOT provides broad categories for airlines to classify flight delays: Air Carrier, Extreme Weather, National Aviation System (NAS), Late-arriving Aircraft, and Security.[[3]](#footnote-3) Table 1 shows the percentage of each delay type’s occurrence per year, from 2014-2020. Air carrier delays, late arriving aircraft, and NAS delays are the most common types.

Table : Delay Cause by Year, 2014-2020. Data provided by the Bureau of Transportation Statistics

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Delay Cause by Year, Percent of Total Delay Minutes (%)** | | | | | |
|  | **Air Carrier Delay** | **Aircraft Arriving Late** | **National Aviation System Delay** | **Security Delay** | **Extreme Weather** |
| **2014** | 30.2 | 41.9 | 23.5 | 0.1 | 4.3 |
| **2015** | 32.2 | 39.8 | 22.9 | 0.1 | 5.0 |
| **2016** | 32.6 | 39.2 | 23.7 | 0.1 | 4.4 |
| **2017** | 31.2 | 39.4 | 25.1 | 0.1 | 4.3 |
| **2018** | 30.1 | 39.6 | 24.5 | 0.1 | 5.6 |
| **2019** | 30.6 | 39.7 | 24.0 | 0.1 | 5.5 |
| **2020** | 41.0 | 30.2 | 21.6 | 0.2 | 6.9 |
| SOURCE: Bureau of Transportation Statistics | | | | | |

The stakeholders in this problem are the consulting firm who funds travel expenses, the travel office of the consulting firm who will implement the solution, and the consulting firm staff and clients who will be affected by the outcome of the project.

The timeline of this model is condensed in order to reduce loss of revenue and reputation for our client. The model will be implemented as soon as possible with the client’s approval. There were few physical constraints during production. The project team was able to utilize remote workspaces to complete the project. Some aspects of the modeling stage had to be modified due to financial and computing power capabilities.

# Analytic Problem

Supervised classification using historical data was determined to be the best approach to modeling this problem. Classification algorithms attempt to estimate the mapping function from the independent variables to categorical dependent variables. In this model, the algorithm takes the given features (airport, year, etc.) and matches it to the “target” variable (on-time or delayed, for example).

Two classification models were created, as both offered equally valuable input to provide to the client. The first model is a binary classification model, which means there are two possible outcomes, “on time” and “delayed”. The second model is a multi-class classification model, which means the model can have more than two possible outcomes. We created 5 classes (outcomes) which denote different delay lengths: No delay, 15-30 minutes, 30-60 minutes, 60-90 minutes, and 90+ minutes.

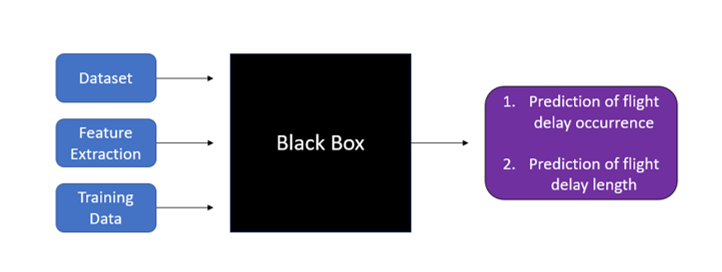


Figure : This black box model gives a simplistic overview

of how the user/client should expect the models to operate

The models will provide the client the probability of overall delay and the probability of delay length for each carrier. This allows the client to both avoid airlines that experience delays more regularly and airlines that specifically experience long delays more regularly

Our measure of success will come in two phases. In phase 1, the data team will test and validate the model to ensure acceptable accuracy. Phase 2 will be feedback from our stakeholders, specifically the clients travelling to meet with the customers and the customers themselves. This phase of feedback is the most important since if the root problem does not change, then we have not succeeded. We hope to increase the client’s level of satisfaction by experiencing less delays and missed meetings.

# Project Plan

Our team is split into a data team and an admin team. The former includes Emma Resmini, Beatriz Buquerin, and Gunnar Cukor. The latter includes Mason Goss, Katie Stewart, and Jamila Clayton. The data team is responsible for data cleaning, developing the model, testing, evaluation, deployment, and monitoring. The admin team is responsible for formulating the business and analytic problems, and developing the presentation and final report. Everyone contributed to the methodology and project plan. Our main source of communication was through Discord, which allowed us to send messages, documents, and links. We coordinated meeting times through Discord as well.

Code was shared via GitHub. Given the financial limits of the project and large storage needs, the data team worked on individual code scripts on their respective work environments. There was an agreed upon method for how to share the newest code onto GitHub without losing a documented contributor history. The code was sectioned into multiple files on GitHub: Data Cleaning, EDA/Visualizations[[4]](#footnote-4), and Modeling. This allowed the data team to work on individual parts of the project without potentially altering another part unintentionally and avoiding system crashes due to storage space and RAM usage.

All other materials – documents, datasets, images, etc. – were shared via Google Drive and OneDrive, which allowed easy simultaneous editing among the entire team, as well as a documented contributor history.

Major milestones include the Oct. 12 deadline for the initial proposal submission, the Oct. 21 deadline to have the model completed and ready for testing and validation, the Dec. 8 deadline to give the presentation, and the Dec. 15 deadline to submit the final report. The full timeline is available in Appendix A.

# Data

We utilized a dataset from Kaggle labelled “*Airline Delay and Cancellation Data, 2009 - 2018*”[[5]](#footnote-5). The Kaggle site lists the data source as the Bureau of Transportation Statistics (BTS), although that could not be confirmed. However, certain attributes of the dataset are similar/identical to those found in the publicly available dataset provided by BTS. Therefore, it may be assumed that either BTS has since changed their data collection methods since 2018 and no longer publicly provides that specific data, or the Kaggle dataset is a combination of BTS data and third-party sources. This dataset includes information including date, carrier, arrival delay, departure delay, scheduled arrival/departure, actual arrival/departure, types of delay (carrier, weather, NAS, security, cancelled, diverted), and origin/destination. A thorough explanation of the columns is available in Appendix B. In this model, we focused on the years 2014-2018.

## Data Cleaning

The first step in the data cleaning process was treating missing data. Upon close inspection, we noticed that flights whose delay lengths were zero ended up with a NaN (null) value. In other words, the null values were actually zeros. We used simple imputation to fill in those values to avoid computation errors later on.

After imputing, we removed the entries (rows) with flights delays caused by security and the National Air System. Our model focuses on delays attributable to the carrier, as well as flights delayed due to weather. The choice to include weather delays was influenced by the fact that when a major weather event occurs that causes widespread delays/cancellations to an airport or whole region, the choice is actually left to the airline if they will delay their flights, cancel their flights, run a reduced schedule, etc.[[6]](#footnote-6) So, while the weather itself is uncontrollable, the response from the airlines is not. Additionally, we included late aircraft delays. These are the “ripple effects” that occur when a past flight causes a future flight to be delayed for any reason, including carrier and weather.

The next major item in data cleaning was removing airlines that are no longer in business. The data spans from 2014-2018, but we want to use that data to make predictions about flights today, therefore we only included airlines that are currently operating. Shutdowns and mergers were accounted for when removing airlines from the data.[[7]](#footnote-7)

A minor enhancement to the dataset was splitting the FL\_DATE column into separate columns. FL\_DATE is in the yyyy-mm-dd format, and was separated into Year, Month, and Day columns. Additionally, we added a Weekday column, derived from FL\_DATE, which determined the day of the week for each date.

The final notable step of data preparation was the removal of columns. We determined a selection of attributes that would likely not be needed for the model. The selection included the FL\_DATE (as we created more usable date columns), flight number, departure and arrival time, and the group of cancelled/delay types.

Table : Columns removed during Data Cleaning process

|  |  |
| --- | --- |
| Column | Reason to remove: |
| FL\_DATE | We have created more useable date attributes (Year, Month, Weekday, etc.) |
| OP\_CARRIER\_FL\_NUM | We are looking at carriers as a whole, not just a single flight, therefore we don’t need the flight numbers. |
| DEP\_TIME  ARR\_TIME | We are more interested in the delay lengths and flight lengths than the scheduled departure/arrival times. |
| CANCELLED  CANCELLATION\_CODE  DIVERTED  CARRIER\_DELAY  WEATHER\_DELAY  NAS\_DELAY  SECURITY\_DELAY  LATE\_AIRCRAFT\_DELAY | We have filtered the flights that we want from these columns – however, there isn’t a need for the actual columns in the model, as they only convey the length of delay and we have that information from other columns. |

Other minor changes made were for the convenience of the data team, such as changing column names and reducing the storage size of the dataset.

## Exploratory Data Analysis

We started the exploratory data analysis (EDA) by looking at the distribution of flights across the airlines, as seen in Figure 4. Note that there isn’t an even distribution, with Southwest Airlines accounting for over a quarter of the flights in the dataset and Endeavor Air, PSA Airlines, Mesa Airlines, and Allegiant Air each accounting for less than 1% of the flights.

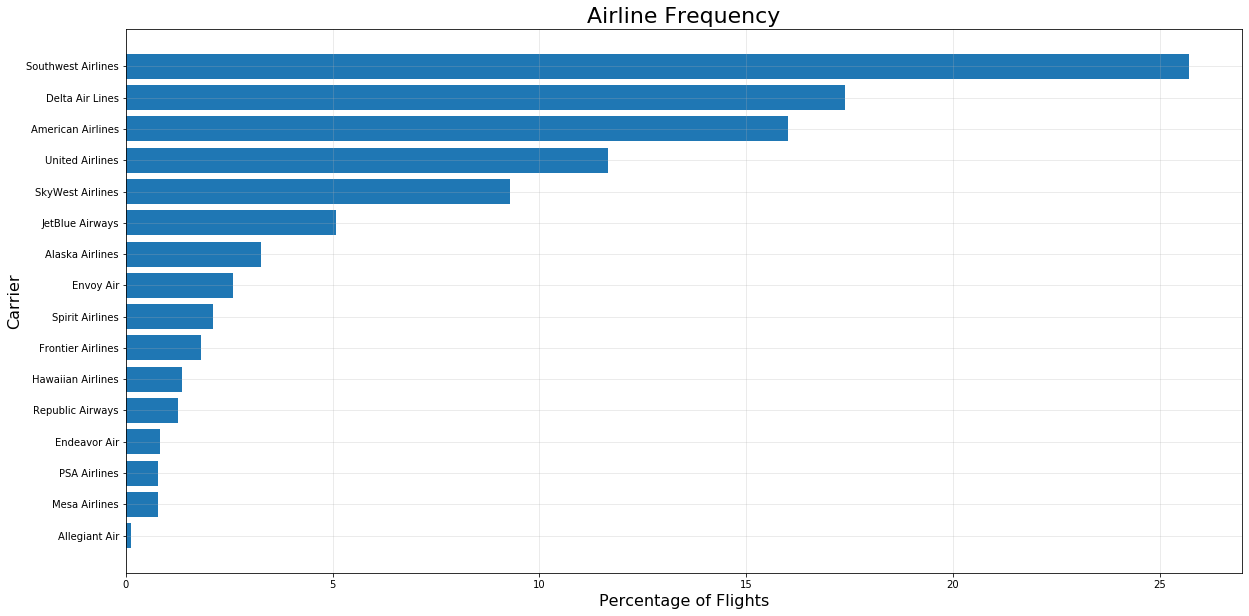


Figure 4: Distribution of Flights Across U.S. Airlines

Additionally, we explored the distribution of flights throughout the year, specifically focusing on weekdays and months. November and December are the most popular months to travel, likely due to Thanksgiving and the winter holidays. January and February are the least-travelled months. When looking at weekdays, there is a roughly even distribution. Monday is the most frequent travel day, however most other days of the week have a similar proportion. Saturday is the least popular day to travel, however the difference between the other weekdays is small.

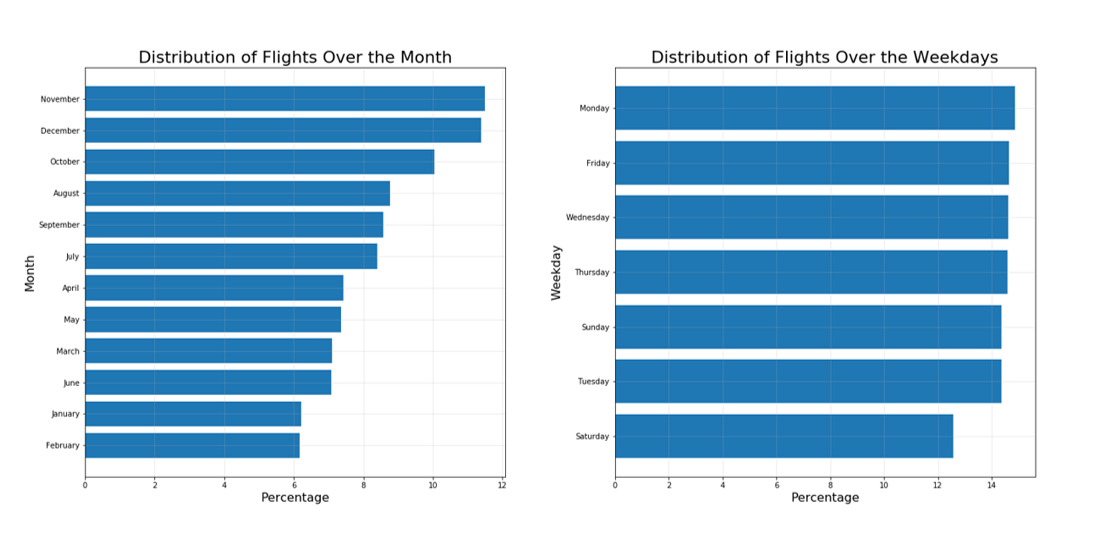


Figure 5: Distribution of flights over the months of the years (left); distribution of flights over the days of the week (right)

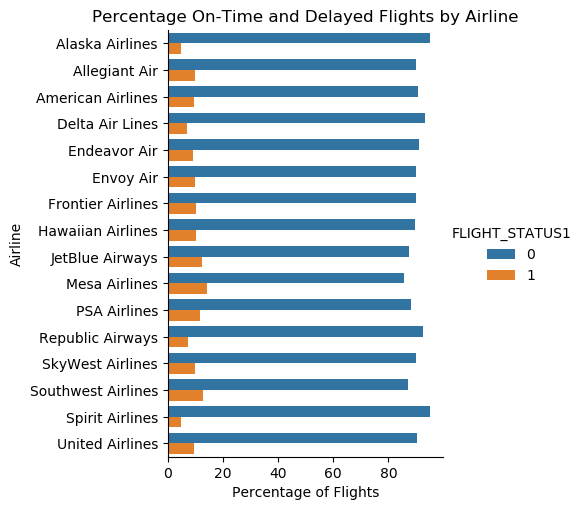
Looking at flight delays, we started by comparing the percentage of on-time flights against delayed flights per carrier. Figure 6 shows how our first model, using binary classification, will be split up. By far, there are more on-time flights than delayed. Every airline has an over 80% on-time rate, meaning a less than 20% delay rate. Alaska Airlines and Spirit Airlines have the lowest rates of delay, while Mesa Airlines and Southwest Airlines have the highest rates. Interestingly, Mesa Airlines is one of the least-frequent appearing airlines in the dataset while Southwest Airlines is the most-frequently appearing airline.

Figure 7 is a visual representation of the second model, the multi-class classification mode. The percentage of on-time (no delay) flights is still above 80% for all airlines. The change we see now is that the delayed flights are broken into four delay length classes: 15-30 mins, 30-60 mins, 60-90 mins, 90+ mins.

Figure 6: Comparing On-Time and Delayed Flights

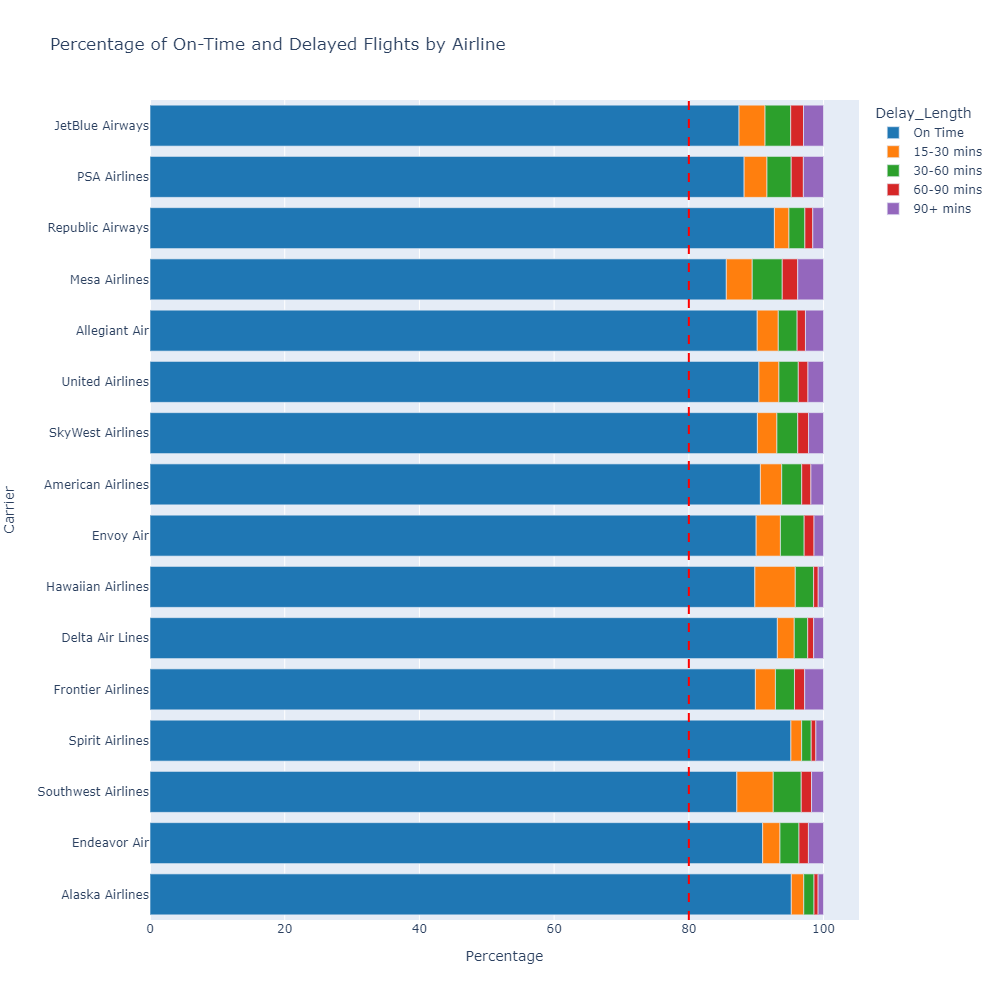


Figure : Flight Delay Lengths. All flights have more than 80% on-time flights.

Figure 8 compares departure delays and arrival delays per airline. The solid blue bars are the departure delays, and the hatched lines are the arrival delays. Overall, all airlines have a greater average departure delay than average arrival delay, and most airlines even have a “negative” avg. arrival delay (another way of denoting an early arrival). The graph implies many flights “make up” for the departure delay in the sky, resulting in a lesser arrival delay or, as seen below, an early arrival. In the context of this project, we see that modeling using arrival delay may be more important than modeling using departure delay. There is an understandable inclination to fixate on the departure delay, however the arrival delay is the real length of delay once the trip has ended.

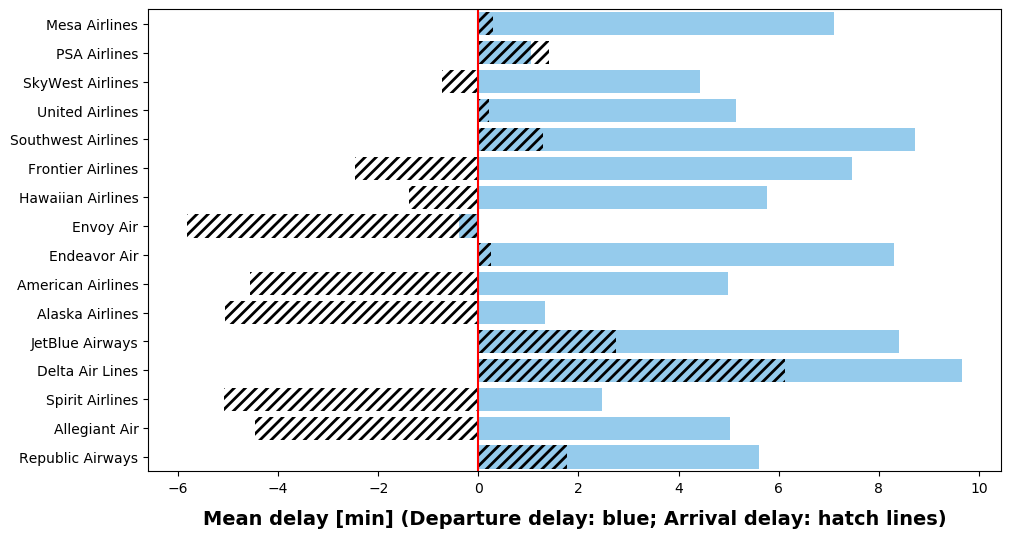


Figure : Comparing departure and arrival delays. Departure delays tend to be longer than arrival delays.

# Methodology

Random forest classification was used for both models. The random forest classifier is an ensemble learning method for classification that creates decision tress on data samples, gets a prediction from each tree, and chooses the best solution via “vote.” The most popular class is ultimately chosen as the final output.

There are many advantages in using random forest classification. It has excellent accuracy and runs large databases in an efficient manner. Thousands of input variables can be handled without deletion, and it has the capability to estimate the importance of the input features, which is a useful tool to have when testing goes awry. Additionally, it better suited for unbalanced data classes, an issue that will be discussed later in the report. [[8]](#footnote-8)

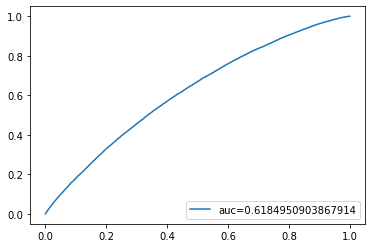
Figure : Random Forest diagram.

Source: <https://medium.datadriveninvestor.com/classification-algorithms-in-machine-learning-85c0ab65ff4>

Since the random forest algorithm is being used for classification, model outputs will be in the form of probabilities of an airline falling into each class. The provides the data in an easy accessible format for the client that can be utilized by any member of the client’s travel department.

# Modeling and Testing

## Model type and justification

We initially made the assumption that logistic regression would be the best-suited model for this problem. Logistic regression is a classification algorithm used to find the probability of event success and event failure. Part of the initial appeal was its simplistic construction, as well as adaptability from binary classification to multi-class classification and ability to handle large datasets.

However, our expectations were not met, with the algorithm returning subpar metrics. Possible complications included overfitting due to the high-dimensional dataset, too many features, and outliers. Figure 10 shows a receiving operating characteristic (ROC) curve for the binary classification logistic model, focusing on JetBlue Airways flights. The ROC curve plots the True Positive Rate versus the False Positive Rate, and the area under the curve (AUC) indicates the measure of performance. Figure 10 has an AUC of 0.618, which means the model is only correct 61.8% of the time. That is not a reliable model.

Figure : ROC curve for Logistic Regression

on JetBlue Airways. This is a poor result.

Due to time constraints, it was more optimal to test other algorithms than try to improve on logistic regression. Therefore, we turned to random forest classification, which proved much more successful. The use of random forests comes with many advantages. The algorithm is considered very accurate and robust, and does not fall into the trap of overfitting as easily as other algorithms. It also runs efficiently with large datasets.[[9]](#footnote-9) Lastly, the coding package that was used to run random forest classification had few hyperparameters to tune, and random forest returns a very accurate prediction even on the default settings.

We used scikit-learn, a free machine learning library for the Python programming language, to build the models. The advantage of scikit-learn is that it includes tools for essentially all aspects of building a model – preprocessing, model selection, algorithms, evaluation, etc. – which helps to keep our code consistent. Some scikit-learn tools may also require the use of NumPy, a free Python library for handling arrays and matrices and high-level mathematics.

Important tools used from scikit-learn:

* RandomForestClassifier()[[10]](#footnote-10), to build the random forest models
* StratifiedKFold()[[11]](#footnote-11), to split the data into test/train sets
* metrics[[12]](#footnote-12), to evaluate the quality of model predictions
* cross\_val\_score()[[13]](#footnote-13), to validate the models

## Additional Data Preparation

We included some steps in the modeling process which involved additional data cleaning. It was decided to leave a few steps in the data preparation until right before the modeling phase in order to avoid inhibiting the exploratory data analysis and visualizations.

We started by removing multicollinearity, which occurs when independent variables are correlated. Such a phenomenon hinders the statistical significance of an independent variable, which reduces the model’s ability to identify independent variables that are statistically significant. In Figure 11 there is obvious multicollinearity. A moderate/strong correlation value is usually 0.5 and above. There are correlation values of 0.6 and above – the yellow and light green squares. The main diagonal is fine, however, as that is a variable being correlated with itself.

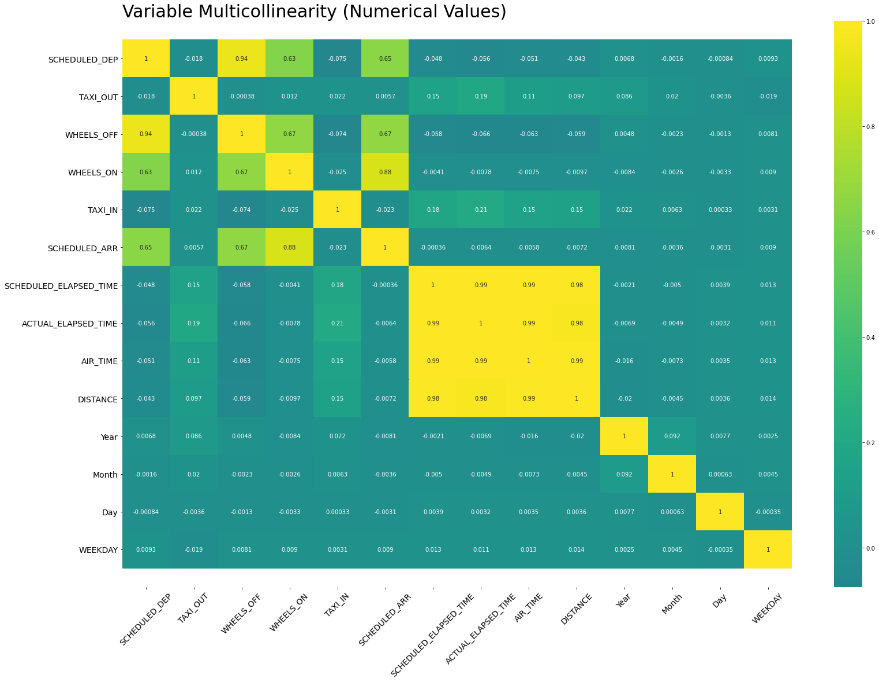
 Upon closer inspection, the fix is quite easy. Many of the columns exhibiting multicollinearity are not crucial to the model, so removing them altogether likely has no effect on the model output. DISTANCE, AIR\_TIME, SCHEDULED\_ELAPSED\_TIME, and SCHEDULED\_ARR are all correlated to each other. Those columns were removed, along with WHEELS\_ON, WHEELS\_OFF, and SCHEDULED\_DEP, which had moderately high correlation. Except for DISTANCE, these columns were all time values, information that didn’t seem particularly useful to utilize for the model, and were not in a easily-usable format. With DISTANCE, as we did not look too closely at geographic information (except for simply the origin/destination airport) and how that affects delays, it did not seem necessary to keep the attribute. In Figure 12, we see the improved heatmap after removing the columns, now showing no significant multicollinearity.

Figure : Heatmap of variable multicollinearity. A correlation coefficient of 0.5 or higher between independent variables is considered multicollinear.

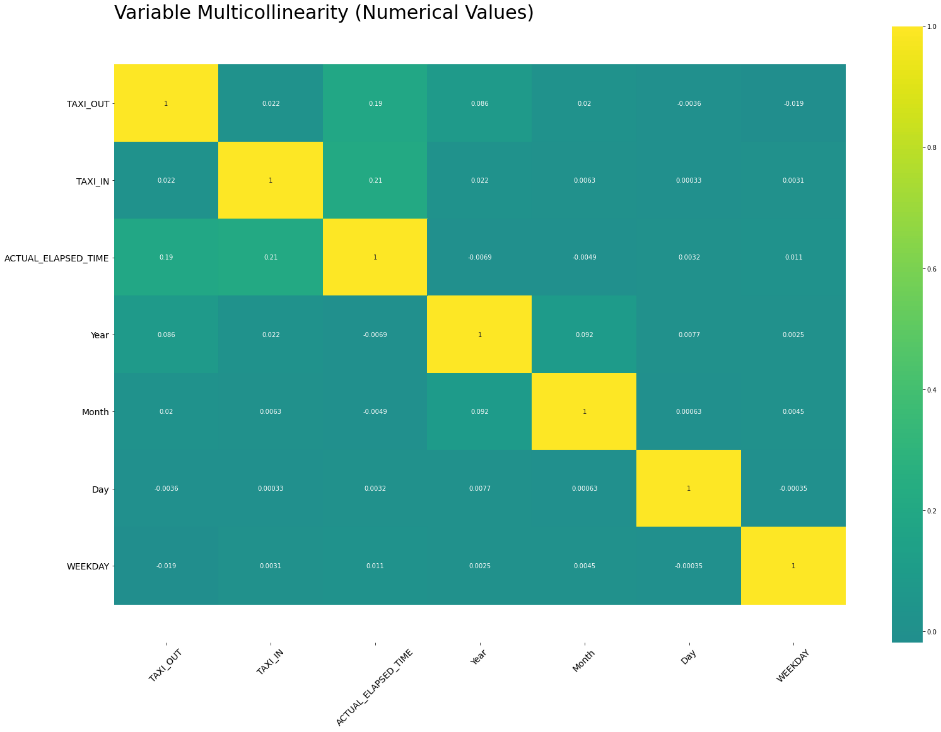
Lastly, we transformed the categorical variables. Most machine learning algorithms either handle them incorrectly handle categorical variables or will not handle them at all. The dataset has numeric categorical data (Month, Weekday, etc.) and string categorical data (origin, carrier, etc.). We used dummy variables, also referred to as one-hot encoding, to tackle this challenge. A dummy variable takes the values 0 and 1, where 1 is the presence of the variable in the data entry. When a categorical variable has more than two categories (which is the case for all of our variables), a set of dummy variables can represent it. For example, when we dummy code the Weekday column, there will be one column for Monday, one column for Tuesday, and so on.

Figure : Improved heatmap.

There no multicollinearity between the independent variables.

## Hyperparameter tuning

Hyperparameters were tested “by hand” for both models. Limited computing power made automated parameter tuning not possible in the given timeframe. Important parameters to consider in the random forest classifier were the number of trees, maximum depth of the tree, number of features, class weighting, and function to measure quality of data split. For parameters which required a numerical input, an appropriate lower value and upper value were compared. Other inputs where a selection of options was available, all options were tried. The hyperparameter combination which returned the best metrics and validation scores was chosen. Table 3 shows the results of hyperparameter testing for both models.

Table : Results of hyperparameter testing.

Values that returned the best model metrics and validation scores were chosen as the "best" hyperparameters.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Binary Classification Model |  | Multi-class Classification Model |  |
| **Hyperparameter** | **Description** | **Tested values** | **Chosen value** | **Tested values** | **Chosen value** |
| n\_estimators | No. of trees in forest | 100, 400 | 400 | 200,500 | 200 |
| max\_depth | Max. depth of tree | 5, 10 | 10 | 5,10,20 | 10 |
| class\_weight | Weights associated with classes | None, ‘balanced’ | ‘balanced’ | None, ‘balanced’ | ‘balanced’ |
| max\_features | No. of features to consider when looking for the best split | ‘auto’, ‘sqrt’, ‘log2’, ‘None’ | None | ‘auto’, ‘sqrt’, ‘log2’, None | None |
| criterion | The function to measure the quality of a split | ‘gini’, ‘entropy’ | ‘gini’ | ‘gini’, ‘entropy’ | ‘gini’ |
| n\_jobs | No. of jobs to run in parallel | None, -1 | -1 | None, -1 | -1 |

The classes of the models are highly unbalanced. There are far more on-time flights than delayed flights. This is a positive case for the client, however can bring trouble when trying to model the data. Most machine learning algorithms operate best when the provided classes are approximately equal. When severe imbalance is present, models can often get a high accuracy simply by always predicting the “majority” class. In context, all of the airlines have at least an 80% on-time rate, so if the model guessed on time no matter what, it would be 80% correct. But often the point of building models is to predict the minor classes (as this project is doing), so we want to avoid that scenario.[[14]](#footnote-14)

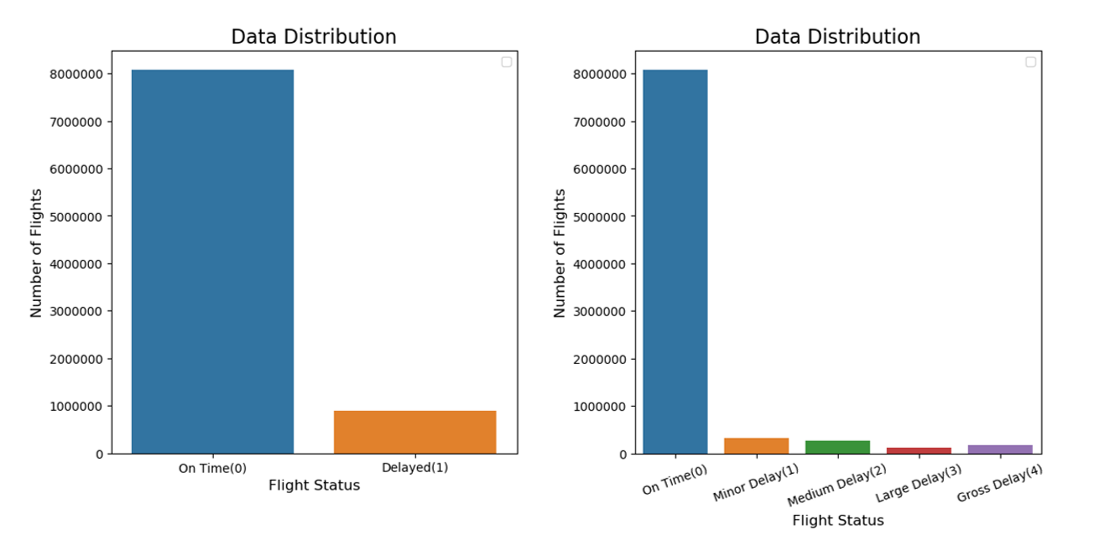


Figure : Class imbalance in the binary classifier (right) and multi-class classifier (left)

This class imbalance affected how the data was split. We utilized stratified k-fold cross validation, which splits the data into k (chosen number) folds while ensuring each fold is approximately representative of the data, i.e. maintaining the class weightings. With stratified k-fold, we are able to obtain a representative sample for the model, but also ensure that the minor class(es) is not completely ignored. For both models, we used two folds as it returned the best model results.

## Testing and results

Multiple forms of metrics were utilized to evaluate the models, including visuals, which will be discussed later, and statistical metrics. Accuracy is the percentage of predictions that were correct out of the total number of predictions. However, accuracy alone is not generally reliable. Additionally, since the classes are imbalanced, the reliability of accuracy alone further breaks down. Therefore, precision, and recall scores were included in the model analysis. Precision is the rate of true positive predictions out of all positive predictions. Recall, also known as the “true positive rate”, is the rate of positive predictions out all actual positive predictions.[[15]](#footnote-15)

Lastly, the K-fold cross-validation feature from scikit-learn was utilized for further testing. This feature randomly splits the training data set into a user-specified number of subset called “folds”, trains and evaluates the random forest algorithm model the specific number of times. We chose 5 folds, therefore the data was split into 5 subsets and tested five times per model run. The results are given as an array, but for simplicity we took the average of the scores. Additionally, the standard deviation is provided to show how precise the estimate is. [[16]](#footnote-16)

### Binary Classification Model

The binary classification model returned very promising results. Table 4 shows the results for the three most frequently occurring airports in the dataset: Southwest Airlines, Delta Air Lines, and American Airlines. (Full results for all airlines in Appendix C.) The model has a high accuracy score, as well as high recall. The precision score is lacking, around 10% lower than its respective scores for its entry. This means the false-positive rate is higher than desired. However, as the precision scores are near 90%, the false-positive rate is only around 10%, meaning the model is still quite reliable.

Table : Binary classification results for 3 most-frequently occurring airports

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Carrier | On-Time | Delayed | Accuracy | Precision | Recall | Validation |
| *Southwest Airlines* | 83.8694 % | 16.1306 % | 96.4025 % | 89.2751 % | 97.3717 % | 0.95 accuracy, 0.0286 std. dev. |
| *Delta Air Lines* | 90.6162 % | 9.3838 % | 97.3058 % | 86.0444 % | 97.9879 % | 0.97 accuracy, 0.0203 std. dev. |
| *American Airlines* | 88.1315 % | 11.8685 % | 97.3436 % | 89.1113 % | 98.1383 % | 0.97 accuracy, 0.0010 std. dev. |

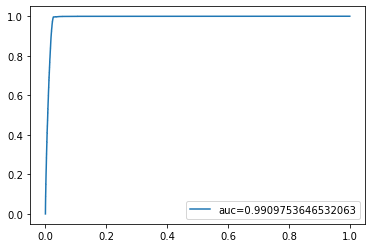
Figure 14 shows the receiver operating characteristic (ROC) curve for the binary classification results on Alaska Airlines, which has the best on-time rate of 93%. The ROC curve shows the ability of a binary classifier by plotting the true positive rate (recall) against the false positive rate. At the basic level of understanding, the goal is for the area under the curve (AUC) to be as close to 1 as possible. [[17]](#footnote-17) We have an AUC of 0.991, which is near-perfect. This result, along with the previously-discussed metrics, means that the binary classification model is highly reliable.

Figure 14: ROC curve for binary classification on Alaska Airlines.

The AUC should be as close to 1 as possible.

### Multi-class Classification Model

Table 6 shows the results of the multi-class classification model for the three most frequently occurring airlines in the dataset: Southwest Airlines, Delta Air Lines, and American Airlines. (Full results for all airlines in Appendix C.) All three metrics are lower across the board, but especially recall and precision, which means a lower rate of predictions are being labelled positive, and of those positive predictions, fewer are true positive. However, the validation tests returned a smaller standard deviation than the previous model, which means the results are more steady across runs.

Table 6: Multi-class classification results for 3 most-frequently occurring airports

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Carrier | No Delay | 15-30 mins | 30-60 mins | 60-90 mins | 90+ mins | Accuracy | Precision | Recall | Validation |
| *Southwest Airlines* | 82.4754 % | 9.7144 % | 4.2072 % | 1.8184 % | 1.7846 % | 93.6084 % | 80.0785 % | 89.7748 % | 0.94 accuracy, 0.0074 std. dev. |
| *Delta Air Lines* | 89.3481 % | 6.0749 % | 2.0775 % | 1.0705 % | 1.429 % | 95.1704 % | 75.7975 % | 88.0152 % | 0.95 accuracy, 0.0046 std. dev. |
| *American Airlines* | 87.1742 % | 6.3701 % | 2.9717 % | 1.6457 % | 1.8384 % | 95.0838 % | 77.5634 % | 87.8423 % | 0.95 accuracy, 0.0020 std. dev. |

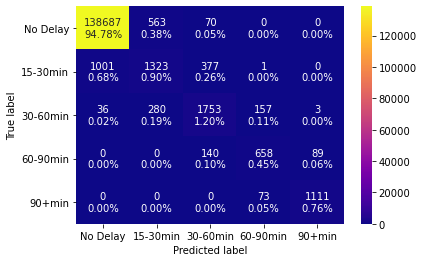
Future versions of this model could demonstrate additional enhancements. Feature importance could be inspected to verify if input variables need to be removed, and previously-removed columns could be added back. A high accuracy but lower precision and recall score is often expected with unbalanced classes, so other ways of balancing the classes could be tested, such as undersampling or oversampling. Also, additional data would help with the imbalance.

## Model output

The model takes in user input for a U.S. airline, then filters the dataset to only include that airline’s flight before running the model. The output from the binary classification model consists of the probability of each class, metrics (accuracy, precision, recall, F1), validation test score, and visuals which include the ROC curve and a confusion matrix. The multiclass classification model output consists of similar output, but without the ROC curve.

Below is a sample output of what the user would see for Alaska Airlines on the multi-class classification model:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Carrier | No Delay | 15-30 mins | 30-60 mins | 60-90 mins | 90+ mins | Accuracy | Precision | Recall | Validation |
| Alaska Airlines | 92.6935 % | 4.2468 % | 1.5056 % | 0.7497 % | 0.8044 % | 96.6143 % | 74.2479 % | 86.1226 % | 0.97 accuracy, 0.0064 std. dev. |



The confusion matrix, also known as an “error matrix”, shows the percentage of predictions that were accurate (the main diagonal), as well as the inaccurate predictions. This layout allows the user to see where inaccurate predictions happen most commonly.

## Model assumptions and limitations

There are few limitations with random forests. It has no distribution assumptions, allowing skewed and multi-modal data and categorical data to be used as inputs. However, a major limitation with random forests is that more accurate predictions require more trees, but a large number of trees can slow down the algorithm, making it too costly for real-time predictions. Random forest algorithms are fast to train, yet can be slow to make predictions. Additionally, as random forest is a tool for prediction. Its use as a descriptive tool is limited, and it would be more effective to use a different algorithm. Lastly, overfitting can occur if the dataset is “noisy”, which means it contains an abundance of meaningless information.

Use of this model assumes that the client is only interested in domestic U.S. flights as our data did not account for international travel. Further, the model assumes that the client is only booking a flight based on the carrier and not the departure and arrival airport. This draws a limitation to our model. Plans to improve the model and solve this limitation is discussed in the Future Work section of this report.

# Evaluation

Our model takes user input and delivers the probability that a domestic U.S. flight from a carrier will be delayed, and what the likely delay range will be. This allows the client to book flights with more confidence and have more realistic expectations as to what will happen with the chosen carrier. From testing and evaluation, it was determined that Alaska Airlines has the highest rate of on-time flights, with a 93% on-time rate. This may drive the client to choose them as the airline carrier if Alaska Airlines services the flight from their departing airport to the arrival one. Furthermore, our model found that Southwest Airlines experienced the lowest rate of on-time, 83%, of the domestic U.S. airlines tested. This insight may drive the client to avoid them when booking a flight if they determine the risk of a delay is too high.

# Deployment and Maintenance

## Plans for Deployment

Once our team determines that the models are ready for release and has passed our performance checkpoints, we will clear the model for release from our sandbox environment. From this point, we will begin building the release package for the client.

The release package will include the model as a .py (Python) file, the user interface that will run as the front end of the models, documentation, the data used by the model, and the system specifications that ensures the models run on the client’s system. Additionally, there will be instructions for how to safely run, shut down, and idle the model in the production environment.

The package will be released to the client’s login portal in our system. This allows the user to login to our cloud environment, use the models, download the results, and safely log out. This production environment will exist on a Microsoft Azure cloud server maintained by the data team. By keeping the model in a cloud-based production environment, security, updates, and system safety prioritizations will be met. A cloud-based server will have the elasticity to easily expand or retract its space if it is determined to be needed.

The client’s dedicated space on the server will be a small part of our server used by multiple clients. This fraction also works as its own sandbox, meaning that the client cannot interfere with any other environments, and is not susceptible to damaging the physical hardware in the event the model is compromised.

The travel office holds primary responsibility for arranging employee travel and will be the primary users of the model. Every member of the travel office will have access to the model and will continually consult it in tandem with booking flights for company employees. Other employees of the office will not have access to the cloud environment, as only the travel office and managers (stakeholders) need access to the model, and this reduces the possibility of model corruption. Employees who travel will be affected by the new implementation, but ideally will not experience any significant change in their work environment.

Finally, we will include the contact information for our emergency technical helpline if the client has an immediate problem related to the model that must be rectified. Since the model is cloud-based and under the project team’s control, the technical helpline will have direct access to the model to see what the client sees and deliver a solution in real time. This contact information will be hyperlinked into the model documentation. This hyperlink allows us to update the contact information at any point as the travel office changes and grows.

Once the package has been released to the client’s login portal, we will deliver a final report to the client and present the features of the model. At the conclusion of the presentation, we will conduct a short walkthrough of the model in the client’s system with them present. This will allow them to have any questions answered by our team. Additionally, a monthly report will be provided to the managers, which will outline the success rate of the model.

## Possible Deployment Pitfalls

Future updates to workstation operating systems (Windows, Mac, Linux, etc.), Python, and our cloud system may not be compatible with each other.

The consulting firm assumes the security risks, which include viruses, property theft, and intellectual theft.

Unpredictable external events may cause a personnel change of the project team, which could delay deployment and maintenance.

Unpredictable external events may cause stakeholders to withdraw from the project.

Unpredictable external events such as market crashes and new government regulations could quickly affect the airline industry, which could consequently result in an inadequate model.

## Monitoring

While the initial model utilized a static dataset, over time the model will acquire new data. Data will need to include, at minimum, information on air carrier, date, origin, destination, origin/arrival delay, and the type of delay which impacted the flight (as defined by the Department of Transportation). Ideally, new data will be added to the model quarterly or bi-annually. Additionally, as new data comes in, old data may have to be adjusted to account for new airline mergers/acquisitions or shutdowns.

With every addition of new data, validation tests will be performed to verify model accuracy. This will help prevent negative impacts to the consulting firm. Tests should also evaluate variables in the model and the effect of removing such features. If a variable becomes irrelevant over time, multicollinearity occurs between variables, etc., the model will be adjusted and variables possibly removed.

The consulting firm may decide that the chosen classifications of delay lengths are either too restrictive – that potentially acceptable flight options are eliminated – or too generous – that employees are still missing meetings at too high of a rate. At the request of the consulting firm, further testing will be done to optimize the classifications.

If it is determined that maintenance and recalibration needs to be performed to this model, we will deliver an update package for the client to download and install when they decide to do so. If the client decides to terminate the project, we will freeze the client’s login credentials and download the package contents to our storage system for closed accounts. If the client wishes to reactivate their account, we will bring the model back online for use.

# Conclusion

Throughout this report, we addressed our clients concern in regard to mitigating the booking of delayed flights. Next, we walked through how we would analytically address this client issue by dissecting the various causes of delayed flights. The data from Kaggle allowed our team to build a comprehensive model to attack this issue. We then described how binary and multi-class classification models would solve our client’s problem of experiencing many flight delays when flying to meet with customers.

By using random forest classification, the best method accounting for time, computing power constraints, we were able to build a model that returned specific useful outputs with great accuracy. First, the probability that a flight from a carrier inputted by the user would be delayed or on time as well as the predicted range of the delay if the flight was in fact predicted to be delayed. These ranges were as follows: no delay, 15-30 mins, 30-60 mins, 60-90 mins, and 90+ minutes. Our model also provides the user with values for the accuracy, precision, and recall values that will show the travel agents the performance of our model.

These outputs give the user confidence in our output and allows them to draw insight into selecting the flight carrier of their choice. In doing so, the client’s travel agency staff has stronger confidence in the flight selection for their team. This complete solution allows the client to book flights more cost effectively and ensure their staff is arriving to meetings on time to maintain a strong relationship with their customers.

## Future Work

With more time and resources, the multi-class classification model could be further improved. While the client did not express a specific need to know the likely delay length of a flight, such a model provides more information, and thus could potentially benefit the consulting firm’s business operations in an even more positive way. The multi-class model could be further tuned to be more accurate, or other algorithms could be tested. For example, linear regression can provide a specific estimate for each carrier, instead of sorting it into a class.

An additional next step would be including airports in the analysis. There was not enough data on the airports in the dataset to produce reliable results. The model would require more data, and the data team would have to ensure that smaller airports (e.g., Roanoke-Blacksburg Regional Airport) were thoroughly represented, along with the major airports (e.g., Denver International Airport). With this additional information, the client could input their desired carrier into the model, but now include an origin and destination airport. Combining the airport data with the results of our carrier delay project could prove to be a more tailored solution.

# Appendix A: Timeline

**Definitions:**

All: Refers to the entire group.

Admin Team: Jamila Clayton, Mason Goss, and Katie Stewart.

Data Team: Beatriz Buquerin, Gunnar Cukor, and Emma Resmini.

|  |  |  |
| --- | --- | --- |
| **Date** | **Task** | **Responsible Party** |
| 1-Sep | First Group Meeting to discuss roles and coding language | All |
| 8-Sep | Second Group Meeting to determine problem and general approach | All |
| 13-Sep | Found the data set | Data Team |
| 17-Sep | Business problem brainstorm and draft begin | Admin Team |
| 22-Sep | Business problem complete | Admin Team |
| 23-Sep | Analytic Problem Begin Draft | Admin Team |
| 27-Sep | Analytic Problem Final Draft Due | Admin Team |
| 28-Sep | Data Wrangling and Cleanup Begin | Data Team |
| 5-Oct | Data Cleanup Complete | Data Team |
| 7-Oct | Begin Model Coding | Data Team |
| 12-Oct | Final Team Review of Proposal | All |
| **12-Oct** | **Turn in Proposal** | - |
| 21-Oct | Complete Model Build, Begin Validation and testing | Data Team |
| 4-Nov | Complete Testing | Data Team |
| 5-Nov | Begin prepping for production and conclusions | All |
| 10-Nov | Meet to begin building final report | All |
| 15-Nov | Final Report Drafted and Reviewed | All |
| 21-Nov | Thanksgiving Recess Begin | - |
| 28-Nov | Thanksgiving Recess End | - |
| 1-Dec | Final Report Re-Reviewed | All |
| 4-Dec | Begin Prepping for presentation | All |
| **8-Dec** | **Present in Class** | **All** |
| **15-Dec** | **Turn in Final Report** | **All** |

# Appendix B

|  |  |
| --- | --- |
| Column | Explanation |
| FL\_DATE | Date of flight, yyyy-mm-dd |
| OP\_CARRIER | Airline ID (IATA) |
| OP\_CARRIER\_FL\_NUM | Flight number |
| ORIGIN | Origin airport code |
| DEST | Destination airport code |
| CRS\_DEP\_TIME | Scheduled departure time |
| DEP\_TIME | Actual departure time |
| DEP\_DELAY | Total departure delay (minutes) |
| TAXI\_OUT | The time elapsed between departure from origin gate and wheels off |
| WHEELS\_OFF | The time when the plane’s wheels leave the ground |
| WHEELS\_ON | The time when the plane’s wheels touch the ground |
| TAXI\_IN | The time elapsed between wheels on and arrival at destination gate |
| CRS\_ARR\_TIME | Planned arrival time |
| ARR\_TIME | Actual arrival time |
| ARR\_DELAY | Total arrival delay (minutes) |
| CANCELLED | Flight cancelled (1=cancelled) |
| CANCELLATION\_CODE | Reason for cancellation:  A - Carrier  B – Weather  C - National Air System (NAS)  D - Security |
| DIVERTED | Plane had to land at another airport |
| CRS\_ELAPSED\_TIME | Planned time needed for flight |
| ACTUAL\_ELAPSED\_TIME | Actual elapsed time of flight |
| AIR\_TIME | Time spent in the air |
| DISTANCE | Distance between origin and destination airports |
| CARRIER\_DELAY | Delay caused by airline (minutes) |
| WEATHER\_DELAY | Delay caused by weather (mintutes) |
| NAS\_DELAY | Delay caused by National Air System (minutes) |
| SECURITY\_DELAY | Delay caused by security concern/disruption (minutes) |
| LATE\_AIRCRAFT\_DELAY | Previous flight with the same aircraft caused current flight to be delayed, i.e. ripple effect (minutes) |

# Appendix C

## Binary classification model results

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Carrier | On-Time | Delayed | Accuracy | Precision | Recall | f1 Score | Validation |
| *Alaska Airlines* | 93.3182 % | 6.68184 % | 97.9777 % | 85.3043 % | 98.3141 % | 91.3483 % | 0.98 accuracy,  0.0049 std. dev. |
| *Allegiant Air* | 89.2795 % | 10.7205 % | 98.1469 % | 93.3472 % | 96.7765 % | 95.0309 % | 0.98 accuracy, 0.0064 std. dev. |
| *American Airlines* | 88.1315 % | 11.8685 % | 97.3436 % | 89.1113 % | 98.1383 % | 93.4072 % | 0.97 accuracy, 0.0010 std. dev. |
| *Delta Air Lines* | 90.6162 % | 9.3838 % | 97.3058 % | 86.0444 % | 97.9879 % | 91.6286 % | 0.97 accuracy, 0.0203 std. dev. |
| *Endeavor Air* | 89.3055 % | 10.6945 % | 98.0061 % | 91.3794 % | 98.0551 % | 94.5996 % | 0.97 accuracy, 0.0108 std. dev. |
| *Envoy Air* | 88.242 % | 11.758 % | 97.9776 % | 91.9016 % | 98.2718 % | 94.9800 % | 0.98 accuracy, 0.0063 std. dev. |
| *Frontier Airlines* | 87.5065 % | 12.4935 % | 97.2403 % | 89.6174 % | 97.619 % | 93.4472 % | 0.93 accuracy, 0.0363 std. dev. |
| *Hawaiian Airlines* | 85.7354 % | 14.2646 % | 94.5941 % | 83.0065 % | 94.0815 % | 88.1977 % | 0.94 accuracy, 0.0156 std. dev. |
| *JetBlue Airways* | 84.83 % | 15.17 % | 96.9629 % | 90.5522 % | 97.5545 % | 93.9228 % | 0.96 accuracy, 0.0110 std. dev. |
| *Mesa Airlines* | 84.0028 % | 15.9972 % | 97.5422 % | 93.4516 % | 97.2741 % | 95.3245 % | 0.94 accuracy, 0.0265 std. dev. |
| *PSA Airlines* | 86.7956 % | 13.2044 % | 98.2417 % | 93.8973 % | 98.368 % | 96.0807 % | 0.96 accuracy, 0.0196 std. dev. |
| *Republic Airways* | 91.1973 % | 8.80267 % | 98.3084 % | 90.8683 % | 98.5013 % | 94.5310 % | 0.98 accuracy, 0.0076 std. dev. |
| *SkyWest Airlines* | 88.5638 % | 11.4362 % | 98.1874 % | 92.4537 % | 98.5441 % | 95.4018 % | 0.97 accuracy, 0.0250 std. dev. |
| *Southwest Airlines* | 83.8694 % | 16.1306 % | 96.4025 % | 89.2751 % | 97.3717 % | 93.1478 % | 0.95 accuracy, 0.0286 std. dev. |
| *Spirit Airlines* | 93.0377 % | 6.96225 % | 97.7123 % | 84.1394 % | 97.9601 % | 90.5253 % | 0.98 accuracy, 0.0052 std. dev. |
| *United Airlines* | 87.8132 % | 12.1868 % | 97.1029 % | 88.6531 % | 97.6868 % | 92.9510 % | 0.96 accuracy, 0.0190 std. dev. |

## Multi-class Classification model results

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Carrier | No Delay | 15-30 mins | 30-60 mins | 60-90 mins | 90+ mins | Accuracy | Precision | Recall | f1 score | Validation |
| *Alaska Airlines* | 92.6935 % | 4.2468 % | 1.5056 % | 0.7497 % | 0.8044 % | 96.6143 % | 74.2479 % | 86.1226 % | 79.7456 % | 0.97 accuracy, 0.0064 std. dev. |
| *Allegiant Air* | 88.6214 % | 4.5203 % | 2.9442 % | 1.1604 % | 2.7537 % | 96.6228 % | 82.445 % | 87.1837 % | 84.7482 % | 0.96 accuracy, 0.0027 std. dev. |
| *American Airlines* | 87.1742 % | 6.3701 % | 2.9717 % | 1.6457 % | 1.8384 % | 95.0838 % | 77.5634 % | 87.8423 % | 82.3835 % | 0.95 accuracy, 0.0020 std. dev. |
| *Delta Air Lines* | 89.3481 % | 6.0749 % | 2.0775 % | 1.0705 % | 1.429 % | 95.1704 % | 75.7975 % | 88.0152 % | 81.4507 % | 0.95 accuracy, 0.0046 std. dev. |
| *Endeavor Air* | 88.199 % | 5.0781 % | 2.8975 % | 1.6638 % | 2.1616 % | 95.7227 % | 77.1312 % | 86.773 % | 81.6685 % | 0.96 accuracy, 0.0048 std. dev. |
| *Envoy Air* | 87.339 % | 6.2069 % | 3.3555 % | 1.6390 % | 1.4597 % | 95.9931 % | 81.556 % | 90.0097 % | 85.5746 % | 0.96 accuracy 0.0065 std. dev. |
| *Frontier Airlines* | 86.1155 % | 6.5687 % | 2.8310 % | 1.7657 % | 2.7191 % | 94.6354 % | 76.7879 % | 86.901 % | 81.5320 % | 0.94 accuracy, 0.0141 std. dev. |
| *Hawaiian Airlines* | 79.4219 % | 15.3333 % | 3.7339 % | 0.7399 % | 0.7711 % | 87.4075 % | 70.1515 % | 82.6604 % | 75.8940 % | 0.87 accuracy, 0.0251 std. dev. |
| *JetBlue Airways* | 83.2232 % | 7.8703 % | 3.7662 % | 2.2438 % | 2.8965 % | 93.7893 % | 77.7701 % | 87.2703 % | 82.2468 % | 0.94 accuracy, 0.0071 std. dev. |
| *Mesa Airlines* | 81.1611 % | 8.5454 % | 3.9848 % | 2.6323 % | 3.6765 % | 92.9205 % | 77.1049 % | 84.9946 % | 80.8577 % | 0.93 accuracy, 0.0060 std. dev. |
| *PSA Airlines* | 85.9395 % | 5.6323 % | 3.4733 % | 2.0033 % | 2.9516 % | 96.2211 % | 82.4294 % | 89.7085 % | 85.9150 % | 0.96 accuracy, 0.0032 std. dev. |
| *Republic Airways* | 90.5025 % | 4.2977 % | 2.2971 % | 1.2915 % | 1.6112 % | 96.5079 % | 77.2061 % | 86.15 % | 81.4332 % | 0.96 accuracy, 0.0041 std. dev. |
| *SkyWest Airlines* | 87.6578 % | 5.3730 % | 2.9399 % | 1.8234 % | 2.2059 % | 96.213 % | 81.2719 % | 89.8781 % | 85.3586 % | 0.96 accuracy, 0.0024 std. dev. |
| *Southwest Airlines* | 82.4754 % | 9.7144 % | 4.2072 % | 1.8184 % | 1.7846 % | 93.6084 % | 80.0785 % | 89.7748 % | 84.6499 % | 0.94 accuracy, 0.0074 std. dev. |
| *Spirit Airlines* | 91.9108 % | 4.6091 % | 1.5800 % | 0.7932 % | 1.1069 % | 95.9876 % | 74.4467 % | 87.5961 % | 80.4879 % | 0.96 accuracy, 0.0066 std. dev. |
| *United Airlines* | 86.0996 % | 7.0871 % | 2.8082 % | 1.7616 % | 2.2436 % | 94.0582 % | 75.3458 % | 86.276 % | 80.4413 % | 0.94 accuracy, 0.0160 std. dev. |

1. Ann Brody Guy, “Flight delays cost $32.9 billion, passengers foot half the bill”, *Berkeley News*, <https://news.berkeley.edu/2010/10/18/flight_delays/> [↑](#footnote-ref-1)
2. Ye, B., Liu, B., Tian, Y., Wan, L., A Methodology for Predicting Aggregate Flight Departure Delays in Airports Based on Supervised Learning. *Sustainability*, 3-3 (Figure 1). <https://www.mdpi.com/2071-1050/12/7/2749>. [↑](#footnote-ref-2)
3. “Understanding the Reporting of Causes of Flight Delays and Cancellations”, *Bureau of Transportation Statistics*, <https://www.bts.gov/topics/airlines-and-airports/understanding-reporting-causes-flight-delays-and-cancellations#q6> [↑](#footnote-ref-3)
4. EDA: Exploratory Data Analysis [↑](#footnote-ref-4)
5. “Airline Delay and Cancellation Data, 2009-2018”, *Kaggle*, <https://www.kaggle.com/yuanyuwendymu/airline-delay-and-cancellation-data-2009-2018> [↑](#footnote-ref-5)
6. Barbara Peterson, “Why Only Some Flight Are Cancelled in Bad Weather”, *Afar*, <https://www.afar.com/magazine/why-bad-weather-cancels-or-delays-some-flights-but-not-others> [↑](#footnote-ref-6)
7. “U.S. Airline Mergers and Acquisitions”, *Airlines for America*, <https://www.airlines.org/dataset/u-s-airline-mergers-and-acquisitions/> [↑](#footnote-ref-7)
8. L. Breiman & A. Cutler. “Random Forests”. *Berkeley Statistics*. UC Berkeley. <https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#intro> [↑](#footnote-ref-8)
9. Afroz Chakure, “Random Forest Classification”, *Medium*, <https://medium.com/swlh/random-forest-classification-and-its-implementation-d5d840dbead0> [↑](#footnote-ref-9)
10. Documentation for RandomForestClassifier(): <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html> [↑](#footnote-ref-10)
11. Documentation for StratifiedKFold(): <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html> [↑](#footnote-ref-11)
12. Metrics and scoring: quantifying the quality of predictions: <https://scikit-learn.org/stable/modules/model_evaluation.html> [↑](#footnote-ref-12)
13. Documentation for cross\_val\_score(): <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html> [↑](#footnote-ref-13)
14. “10 Techniques to deal with Imbalanced Classes in Machine Learning”, *Analytics Vidhya*, <https://www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/> [↑](#footnote-ref-14)
15. Provost, F., & Fawcett, T. (2013). *Data Science for Business*. O'Reilly Media, Inc. [↑](#footnote-ref-15)
16. Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc. [↑](#footnote-ref-16)
17. T. Tape, “ROC Curves”, University of Nebraska Medical Center, <http://gim.unmc.edu/dxtests/Default.htm> [↑](#footnote-ref-17)