Using Data Science to predict the probability of a flight delay

There are fewer things that captivate the human spirit more than flying. Whether for leisure or work-related purposes, flights have proven to be an invaluable part of human society. However, as the demand for flights has increased, so has the potential for delays. Perhaps one of the most dreaded parts of aviation, pilots and passengers alike lament the possibility of a flight delay.

Step 1: Download the Dataset

- Using Kaggle, I found a phenomenal dataset containing over 1,200,000 flights between 2004-2017. Including over 50 categories such as weather, flight distance, and day of the week, it's a sufficiently large dataset that was relatively easy to explore and interpret.
  Airline Delay Data - Mendeley Data
  The aforementioned link contains both the dataset itself, available for download as an Excel CSV, and an accompanying PDF that explains the variable names.

- Per the FAA, a flight delay is when an airline takes off and/or lands more than 15 minutes later than its scheduled time. For the sake of this project, I will define a delayed flight such that the combined delay of both departure and arrival is more than 15 minutes. In this capstone, I will use Data Science to predict flight delays given a dataset. The codes for every step of this report are available in an accompanying Jupyter notebook.

Step 2: Data cleaning

Perhaps the most time-consuming part of this project, data cleaning requires removing unnecessary and/or unhelpful values. I accessed the dataset via Jupyter notebook, on which the rest of this project will take place. Once I opened the dataset, I explored it to familiarize myself with the different columns.

1) I dropped the ones I deemed unnecessary. There were a couple of seemingly unnecessary columns, e.g. metro population of flight city origin & destination, carrier code for planes, etc.
2) The next step was to get rid of NaN/null values in the dataset. I managed to pinpoint five columns containing such values, namely minutes of flight arrival delay, temperature at departure, windspeed, wind speed squared, and the wind gust speed. I replaced these missing values with the means of their respective columns.
3) Voila! The dataset now contains only numeric values; in particular, binary classifications (0 for no, 1 for yes), and continuous values.

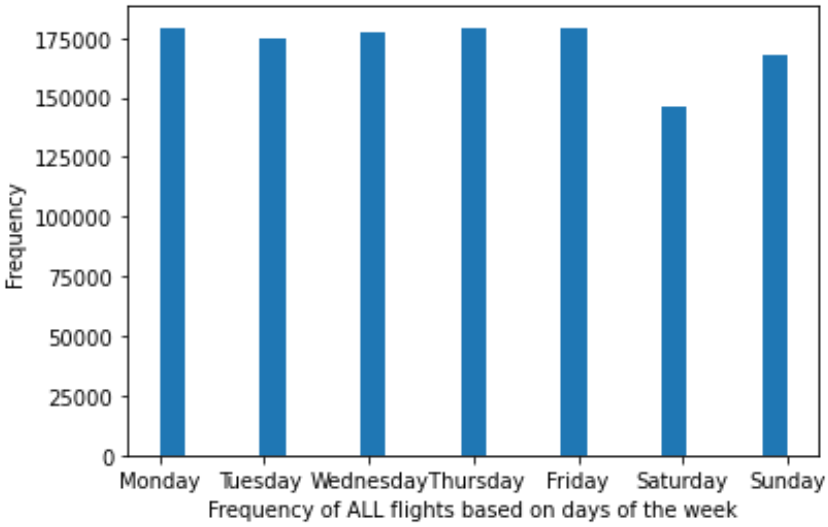
Step 3: Exploratory Data Analysis

Now that the data is clean, I then explored the columns (i.e. different factors) of this dataset:

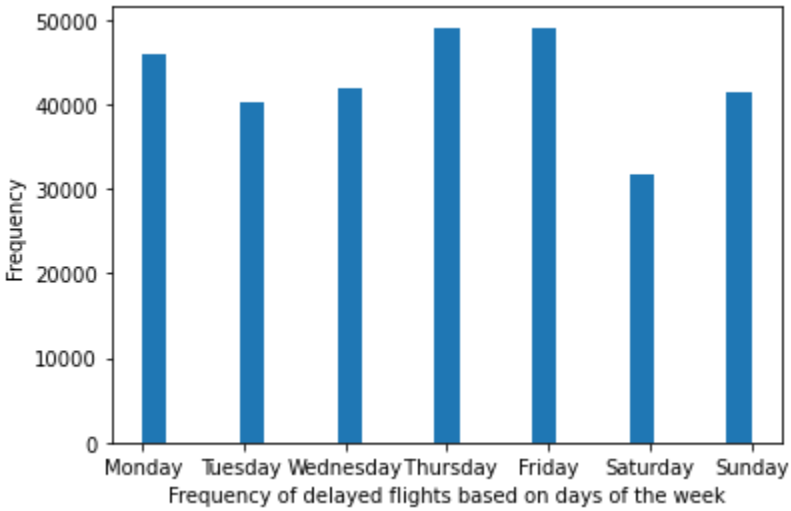
depdelay: Departure delay in minutes; negative denotes early delay
arrdelay: Arrival delay in minutes; negative denotes early arrival
scheduleddepartdatetime: Scheduled departure time
origin: Code of origin airport
dest: Code of destination airport
uniquecarrier: Carrier code
marketshareorigin: Market share of airline at origin airport
marketsharedest: Market share of airline at destination airport
hhiorigin: Market concentration at origin airport
hhidest: Market concentration at destination airport
nonhubairportorigin: The origin airport is a non-hub
smallhubairportorigin: The origin airport is a small hub
mediumhubairportorigin: The origin airport is a medium hub
largehubairportorigin: The origin airport is a large hub
nonhubairportdest: The destination airport is a non-hub
smallhubairportdest:The destination airport is a small hub
mediumhubairportdest: The destination airport is a medium hub
largehubairportdest: The destination airport is a large hub
nonhubairlineorigin: The airline is a non-hub at the origin airport
smallhubairlineorigin: The airline is a small hub at the origin airport
mediumhubairlineorigin: The airline is a medium hub at the origin airport
largehubairlineorigin: The airline is a large hub at the origin airport
nonhubairlinedest: The airline is a non-hub at the destination airport
smallhubairlinedest: The airline is a small hub at the destination airport
mediumhubairlinedest: The airline is a medium hub at the destination airport

largehubairlinedest: The airline is a largehub at the destination airport
year: Year of flight
month: Month of flight
dayofmonth: Day of the month
dayofweek: Day of the week
scheduledhour: The hour of the day that the flight is scheduled for (24 hour clock)
originairportid: Numeric code for origin airport
destairportid: Numeric code for destination airport
tailnum: A unique plane identifier
capacity: Number of passengers per flight
loadfactor: Percentage of seats that are occupied (monthly)
numflights: Systemwide Number of flights on the given day
origincityname: Name of origin city
originstate: Name of origin state
distance: Flight distance
monopolyroute: Whether airline is the only airline to offer flights between two cities in a particular month (dummy)
temperature: Temperature in degrees Celsius
temp_ninfty_n10: Temperature is less than -10°C
temp_n10_0: Temperature is between -10°C and 0°C
temp_0_10: Temperature is between 0°C and 10°C
temp_10_20: Temperature is between 10°C and 20°C
temp_20_30: Temperature is between 20°C and 30°C
temp_30_40: Temperature is between 30°C and 40°C
temp_40_infty: Temperature is greater than 40°C
windspeed: The wind speed
windspeedsquare: The wind speed square
windgustdummy: Tells if there are wind Gusts
windgustspeed: Speed of wind gust
raindummy: Tells if there is rain
raintracedummy: Tells if there are trace amounts of rain
snowdummy: Tells if there is snow
snowtracedummy: Tells if there trace amounts of snow

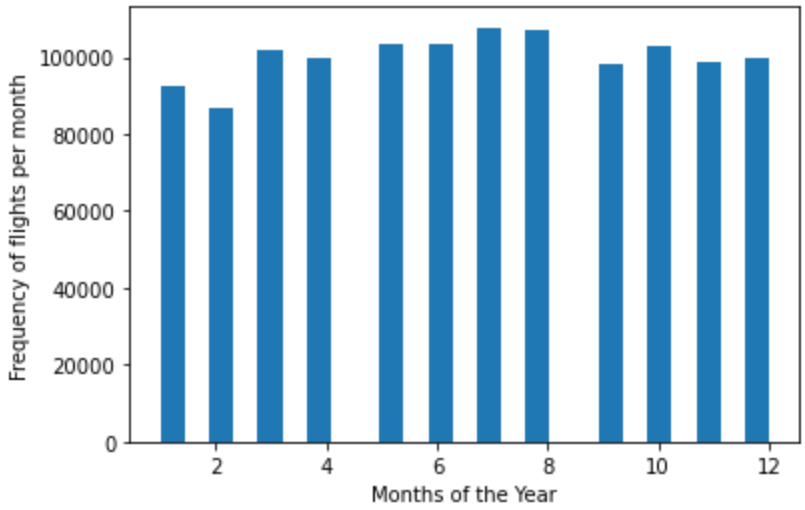
Given not only the quantity of data but also the variety, this looks like a fantastic dataset that will allow me to explore and successfully implement several different metrics. In addition to the aforementioned categories, I also created a new column, delay_sum, that gives me the combined times of both departure and arrival delays. Now that we've defined the variables, let's see what some of them look like:



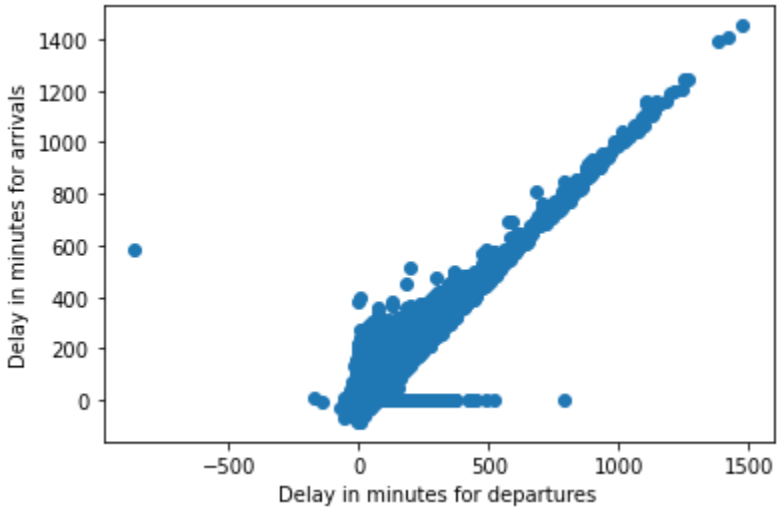
We see that Monday is the most common day of the week for all flights to take place;



And that Thursday and Friday are the most common day of the week for delayed flights to take place.



Unsurprisingly, July and August are the most common months for flights.



There is a positive correlation between flights with both delayed departures and delayed arrivals.

Step 4: Dummy Features, Scaling, Fitting and Test/Train Sets

Once I explored the data and familiarized myself with it, I then added an additional column, called 'delayed,' where 1 represents a delayed flight (i.e. more than 15 minutes of a combined departure & arrival delay), and 0 represents an on-time flight (i.e. 15 minutes or less of a combined departure & arrival delay). I did this because I ultimately want to use a classification metric, which requires that the dependent variable be of a classification type (i.e. 1 for yes, 0 for no). That dataset already contained several other dummy features (columns with 0 or 1 values), and I added another one which will become my target variable.

After the inclusion of the 'delayed' column, I scaled the dataframe in order to normalize the data as well as speed up the modeling & testing processes. I then split my data frame into

two distinct groups: y, the target/dependent variable, and x, the independent variables. I assigned y to the 'delayed' column, as this is what I ultimately want to correctly predict (whether a flight is delayed or on-time), and x is the entire dataset other than this last column. I then subdivided my dataset into their training and test sets, where 80% of the data will be used for training, and the remaining 20% will be used for testing.

Step 5: Algorithms/Machine Learning

Now that I've separated the dataset into my training sets and test sets, I will decide which model is best. There are two categories of models: classification and regression. Classification is for discrete labels, but not continuous values (e.g. 0 or 1, yes/no); regression is for continuous values. In my case, I want to find a discrete label (i.e. 1 for a delayed flight, and 0 for an on-time flight). As such, I will explore different classification metrics.

The first one that I tried was the Precision/Recall/F1-Score combination. Precision measures the count of true positives out of all of the predictions made; recall measures the count of true positives out of all positive values in total; and the F1-score is the harmonic mean of precision & recall, i.e. (2 x Precision x Recall)/ (Precision + Recall). I got the following results:

Precision: 0.8181818181818182, so 81.82%
recall: 0.3879310344827586, so 38.79%
F1 score: 0.5263157894736843, so 52.63%

I think that the suboptimal scores are due to false positives: given that roughly 75% of the flights are on-time and 25% are delayed, I think the Precision/Recall/F1-Score tests may have misclassified delayed flights as on-time, as there are disproportionately more on-time flights than delayed ones (902334 compared to 299330).

The second metric that I used was Logistic Regression. It is a classification algorithm that uses several independent variables to predict the probability of a categorical dependent variable. The dependent variable is a binary value (0,1) which, in my case, means an on-time flight (0) or a delayed flight (1).

After running a Logistic Regression, I got a result of 0.9595, or 95.95%. Notably better than the aforementioned tests. I think that the Logistic Regression was successful, as it had a plethora of independent variables from which to work, as well a notably large dataset.

The third and final metric from these tests that I used was the Random Forest Classifier, a classification metric that fits a series of decision trees on multiple samples from the dataset, and then averages values from those samples to improve predictive accuracy. The training data improves via bagging, i.e. when random data samples from the training set are selected with replacement. It is worth noting that Random Forest Classifications generally take longer to execute, due to its fundamentally more complex structure.

After running a Random Forest Classifier, it achieved a result of 0.964, or 96.4%. It achieved such remarkable success due to the abundance of data, which enabled the Random Forest to conceive a surfeit of decision trees and produce such great results. While the Random Forest Classifier achieved a result only .45% better than the Logistic Regression, it took nearly a minute of runtime, whereas the Logistic Regression was almost instantaneous.

Step 6: Conclusion

Following the Data Science method, I became progressively more comfortable with the project and with my own competency. Using visuals, I better understood the components of this dataset, and how to ultimately achieve what I wanted to prove: whether or not a flight is delayed based on the

dataset. After executing the aforementioned tests, the Random Forest Classifier yields the highest score, but at a cost: it took significantly longer than the other two tests, given its complexity. In conclusion, I find the Logistic Regression to be the best of the three tests, as its score is an iota less than that of the Random Forest, but at a fraction of the cost (cost being runtime).

Step 7: Client Recommendations/Further Research

I have successfully created from scratch a Machine Learning model that predicts whether or not a flight is delayed with 95.95% success and takes mere seconds to execute using a dataset containing more than 1,200,000 flights and more than 50 categories. You can use this to avoid hefty cancellation fees as well as dissatisfactory reviews from clients, and avoid a terrible ripple effect as a result. You can use my findings to foresee the potential of such a dreaded event and to plan accordingly.

Now that I've successfully created and deployed a Machine Learning model, one thing I would like to further research is which category or categories do delayed flights have in common. Is it the weather? The day of the week? A particular temperature threshold? There is always the possibility of other categories, and I think such findings would enable airline companies to prepare even more thoroughly for cancellations.

Final Thoughts

It has been fascinating to watch this project grow little by little, and am astounded with how much I personally have learned about Data Science over the last few months. Applying and understanding every step of the Data Science Method has proven to myself that I have learned truly invaluable skills, and am thrilled to have successfully completed my first personal capstone. A special thanks to my advisor, David Yakobovitch, who's provided phenomenal insight and guidance throughout the duration of this project.