Capstone2: Predicting Flight Takeoff Time Delays

Flight delays are a significant issue for airline companies, affecting operational efficiency, customer satisfaction, and financial performance. Accurately predicting whether a flight will be delayed, and by how long, can enable airlines to proactively manage operations and improve the passenger experience. Knowing that a flight is likely to be late, airlines can establish policies to immediately assuage customer complaints. The goal of this project is to create a model that predicts the difference in scheduled and actual takeoff times of flights.

Data Collection and Cleaning:

We used three datasets. The main dataset was of commercial domestic flights in the United States of America for the year 2018, taken from the United States's Department of Transportation's Bureau of Transportation Statistics. The 2nd and third datasets were of Airline Carrier names and initials, and airport names and codes.

There were various issues with the main dataset that needed to be addressed prior to EDA.

We started with a data frame of 7,213,446 rows and 28 columns. We removed columns that were not needed for our specific problem, such as taxiing time, time in the air, and a few others. We then renamed the remaining 20 columns to make it easier to identify what each variable was. Next, we converted the date column and the multiple time columns into datatimes to make them easier to work with. The next step was to remove any flights that were marks as diverted, as these entries had incomplete data.

Then we imported the other two datasets to add the airport names and airline names to the main dataset. We did this by creating a dictionary from each of the two secondary data frames and then mapping them to the main data frame. Checking the updated data frame, there were some entries that were now missing location names. These locations were not in the dataset used to create the dictionary for mapping, so we looked up these locations and their corresponding code names and manually created a dictionary to map to our main data frame. We then created a new column that displayed whether a flight was delayed/cancelled or not. At this point we had a total of 7,195,587 entries.

We began looking at columns that contained missing information. We started with the Planned Arrival Time Column, which had 91 missing values. Upon investigating, all flights that had a planned arrival time of 00:00 (midnight) were marked as 'NaT' when the column was converted to a datetime. We replaced these 'NaT' values with the value 00:00. The same was true for the actual arrival times, planned departure times, and actual departure times, so we imputed these with 00:00 for all non-cancelled flights. Next, we noticed that there were some flights that had an actual departure time when the flight was cancelled.

As this is an impossibility, we removed these entries as the information stored in them were likely errors.

We then noticed that the number of entries for departure delay and actual departure time were not equal. Upon investigation, all flights that had the same planned departure time and actual departure time had a departure delay of “NaN” instead of 0, so we imputed 0 inplace of these “NaN” values. This was the same for flights with the same planned and actual arrival times.

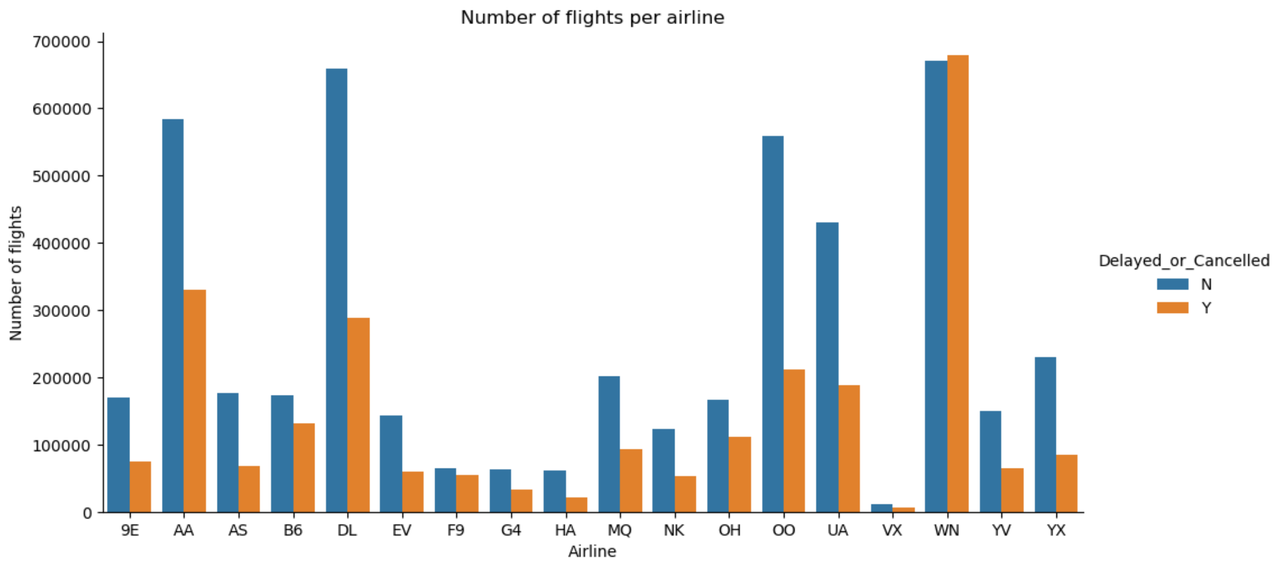
Finally, we removed any entries that had no departure time but did have a departure delay and entries that had a departure time, but still had no departure delay, as these would have been data entry errors. This left us a data frame with the shape 7,191,323 by 25.

Exploratory Data Analysis

We started our EDA by removing the move columns that were not needed. This included arrival times columns, as they do not help us with departure delays. We removed the cancellation code column as well as the type of delay columns. While these would have been useful, there was simply too few entries within these columns to do anything significant with (of the close to 7.2 million flights, only 1.35 million had this information entered.

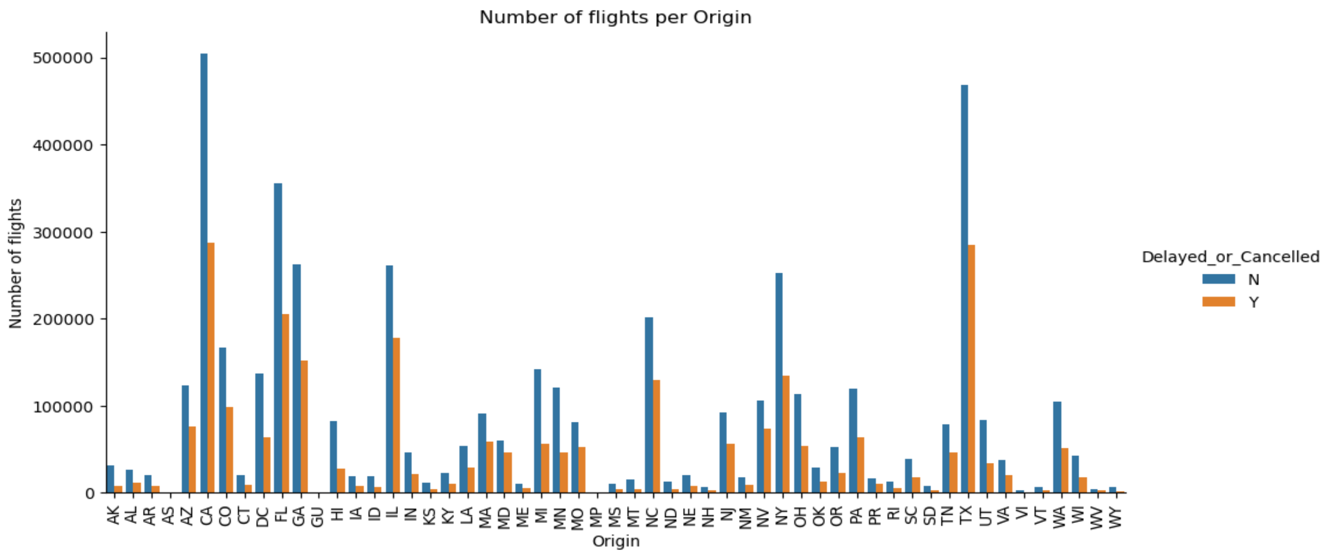
We then graphed the number of flights delayed/cancelled vs left on time across multiple different variables to see if any patterns emerged.

By Airline:



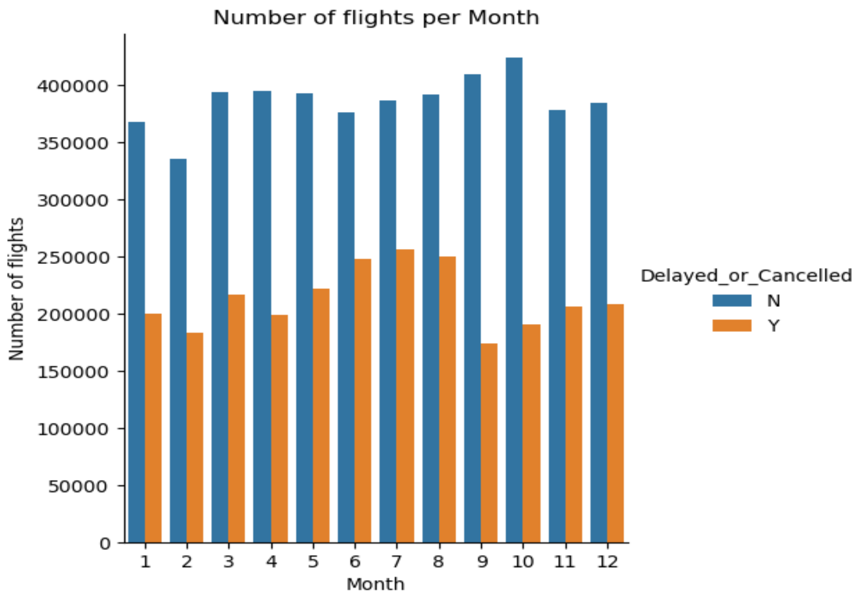
Most airlines have approximetly 33% or less of their flights cancelled or delayed. The notable exceptions are WN (Southwest Airlines), with more delayed/cancelled flights than on time flights, and B6 (Jetblue), F9 ('Frontier Airlines), OH (Jetstream Intl), which have between 40%-46% of their flights are delayed/cancelled.

By Origin Location



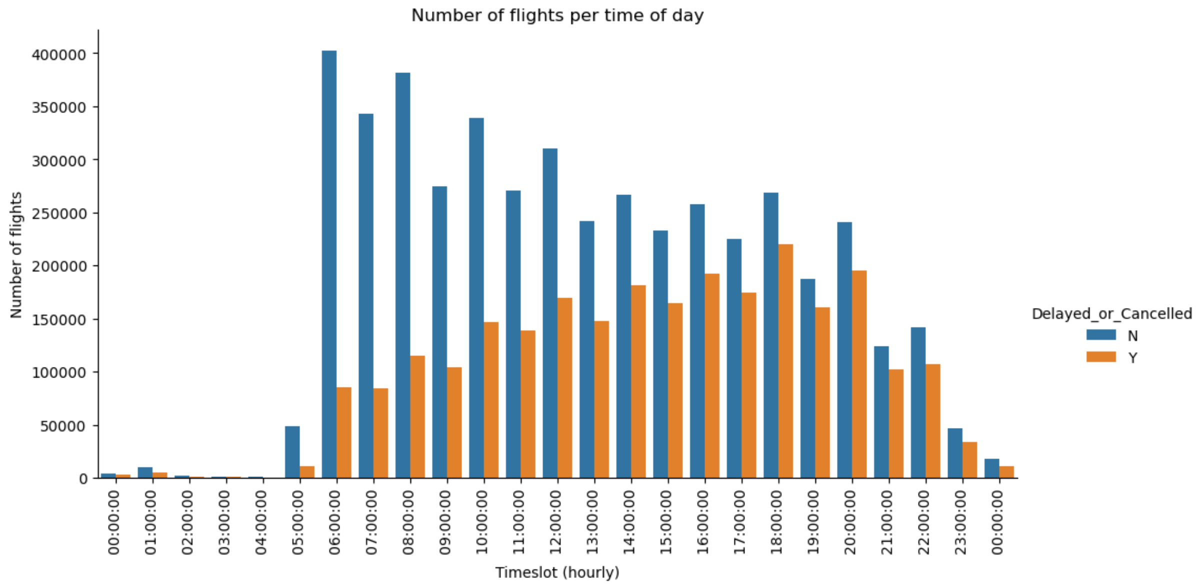
Due to the number of airports in the dataset, we grouped the airports by state/territory. Delayed/Cancelled flight percentages ranged from 19.6% to 47.5%, with both the mean and median being 32%.

By Month



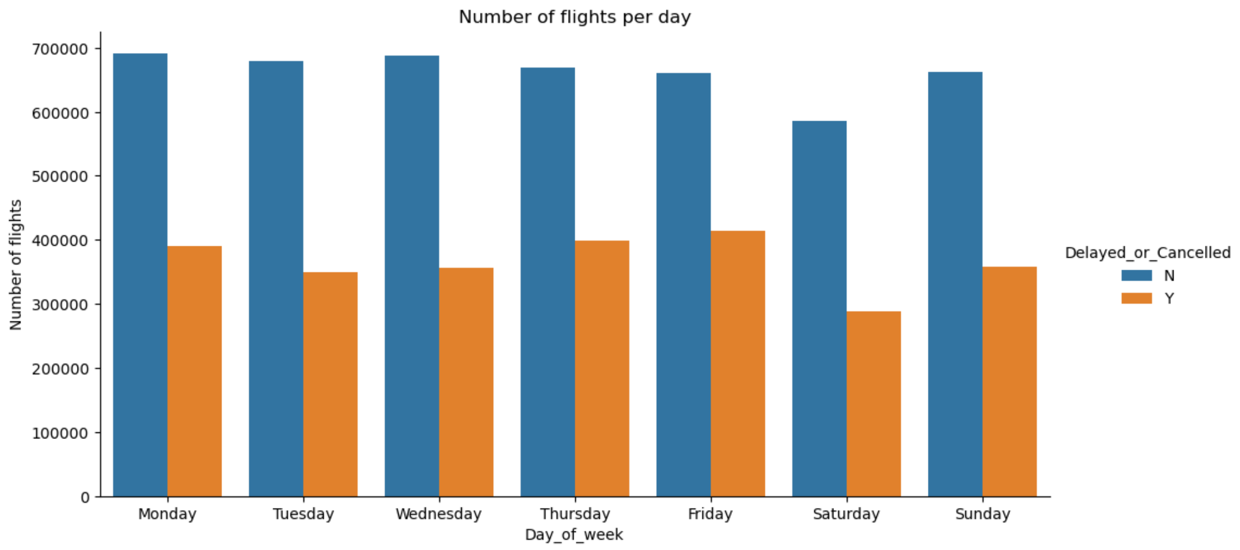
Delayed/cancelled flights vary between 30-40% based on month. The months with the most delays were the summer months (July at 39.85%, June at 39.77%, and August at 38.96%). The months with the fewest delays were September (29.83%) and October (31.04%).

By Time of day



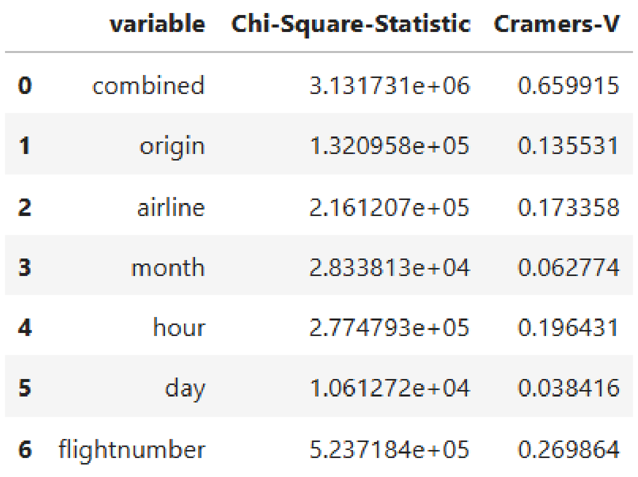
Departure times were rounded to the nearest hour. Flights in the evening (6pm-9pm hours) are delayed the most at approximately 45-46%. Flight times in the early morning (5am-8am hours) are delayed the least at approximately 17-23%. Delays during the rest of the hours of the day range between 27-44%.

Per Day



Delays based on day are consistent throughout the week, with Friday having the highest at 38.5% and Saturday having the lowest at 32.9%.

Next, we calculated the chi square, and Cramer's-V for the origin location, airline, month, hour, day of the week, flight number, and all 6 features combined.



	variable	Chi-Square-Statistic	Cramers-V
0	combined	3.131731e+06	0.659915
1	origin	1.320958e+05	0.135531
2	airline	2.161207e+05	0.173358
3	month	2.833813e+04	0.062774
4	hour	2.774793e+05	0.196431
5	day	1.061272e+04	0.038416
6	flightnumber	5.237184e+05	0.269864

Based on the above, flight delays and cancellations are affected by many factors, including the airline, the location of take off, and time of day, week, and month. While day of the week and the month are on the lower end, the flight number has the biggest impact on whether a flight will be delayed or cancelled because it combines airline, time of day, and location into 1 feature.

We also created the same graphs and statistics for counting flights as delayed on if they were delayed by more than 15 minutes and by more than 50 minutes. While the graphs looked different, the chi-square and Cramer's V stayed relatively the same, so we did not continue with having multiple delay subgroups.

Pre-processing

As we were only interested in flights that left on time or were delayed, we started by removing all rows of cancelled flights. We then removed a number of columns from the dataframe. 'dcn15', 'dcn60', 'dcn','Delay15m', 'Delayed60m', 'fnstring', and 'airflightnumber', were all temporary columns made for EDA calculations. 'Airline_Name', 'Origin_Name', 'Dest_Name' are the same information as 'Airline', 'Origin', and 'Dest'. 'Date', 'Planned_depart_time', and 'Actual_depart_time' were previously combined into different groupings to reduce the number of features. Instead of 365 dates, and over 1,350 departure times, the 'month', 'hour_depart', and 'day" columns group this data into more manageable features. 'Origin', and 'Dest' were previously combined into different groupings to reduce the number of features. Instead of 358 airports, the 'Origin_State' and 'Dest_State' columns group the airports by state/territory, reducing the features to 55. This left us with a data frame of 7,079,005 rows and 8 columns.

Next, We created dummy variables. Then we looked at and removed outliers in the departure delay and distance columns. We removed all rows with values greater than 3 standard deviations. Lastly, we created our training and testing sets and used a log transformer to scale and standardize our data.

Initially this was where we finished our pre-processing step. However, during the modelling step, we decided to test a model that can use categorical data, so we went back and created a second set of training/testing sets that did not utilize the dummy variables. On these sets, we also removed outliers and used a log transformer to scale and standardize out data.

Modeling

We created and tested a variety of models, including K Nearest Neighbors, Random Forest, XGBoost, and two LightGBM. With a dataset this large, we created a subsample train and test set and ran each model though Bayesian Optimization to get the best hyperparameters for each model. The initial subsample was 20% or .2 of the full data, which we used for the XGBoost model and the two LightGBM models, however this sample was too large for the K Nearest Neighbors and Random Forest models, so we had to use a smaller sample size of .02 for them. Although it likely would not work well on a data set this large, we still started with a KNN model to see if it would even work. As expected, the KNN model is too slow at predicting on a dataset this large. It also scored the worst on all evaluation metrics, so it is not a good model to use in our case. Next we tried a random forest model. It performed slightly better than the KNN model on its metrics, but is slow on training, so it also may not be the best model to use.

Next we moved on to different gradient boosting machines. Both LightGBM and XGBoost performed better than Random Forest in all evaluation metrics. They also have a much faster training speed compared to the Random Forest, so these models would be better than either of the two previous models.

Finally, we tried the LightGBM model again, but this time kept the categorical data, instead of using one-hot encoding. For this model, we split the also converted the month feature into day of the year and week of the year. We also used cyclical sin/cos encoding on the month to address the rollover of December to January, and on the time to address the 23:00 to 00:00 rollover.

Documention

Model Metric Comparison Table

MODEL	MAE	MSE	RMSE	R-Squared
KNN	13.9348	505.1558	22.4757	0.0321
Random Forest	13.6746	503.6235	22.4416	0.0676
LightGBM	13.4947	493.1229	22.2064	0.0870
XGBOOST	13.4557	491.4054	22.1677	0.0902
LightGBM - Categorical	12.8110	455.9757	21.3536	0.1601

The LightGBM model using categorical featrues (without one-hot encoding is the best model to use of the model's tested. That being said, there is much room for improvement. Currently, this model only explains approximately 16% of the variance in the data.

LightGBM-Categorical

Features	Value
Airline	category
Origin	category
Dest	category
Distance	float64
day_of_week	category
day	int32
week_of_year	UInt32
month_sin	float64
month_cos	float64
hour_sin	float64
hour_cos	float64
minute_sin	float64
minute_cos	float64

LightGBM-Categorical

Hyperparameters	Value
colsample_bytree	0.819758663105755
learning_rate	0.107194943614919
max_depth	49
n_estimators	409
num_leaves	95
subsample	0.831183308051619

Metrics	Value
Mean Absolute Error (MAE)	12.811
Mean Squared Error (MSE)	455.9757
Root Mean Squared Error (RMSE)	21.3536
R-squared (R2)	0.1601

Recomendations:

1. Airlines can use this model to determine in advance if there will likely be a delay in a flight, and estimate how long the delay will be. This in turn can be used to improve customer service by being able to notify passengers more in advance of flight delays. However, airlines should note that the MAE is 12.8 minutes, so they should take this into account when deciding how to post delay notifications. We

recommend rounding this number to 15 minutes, so airlines can inform passengers that the estimated delayed length posted may be off by approximately 15 minutes.

2. Airlines can better manage staffing personal. Knowing the length of a delay in advance can allow airlines to better schedule staff and other operation needs.
3. The work from this project not only gave a model that estimates the difference in time between the scheduled flight takeoff and the actual flight takeoff, but also produced a clean data set with flight times. As many airlines schedule the same flights every day, airlines should use this data as a starting off point to adjust flights that are consistently delayed.

Improvements:

The main area for improvement would be collecting more data and creating more unique features to get a more accurate predictive model. This could include the reason for each delay (weather, mechanical, staffing, etc), data across multiple years, if the previous flight was delayed, or whether the day was a holiday.

Next Steps:

1. Track the same flights throughout the year to see if there are specific flights that are always delayed.
2. Track the same plane throughout the day to see if there is a cascade effect that contributes to delays.