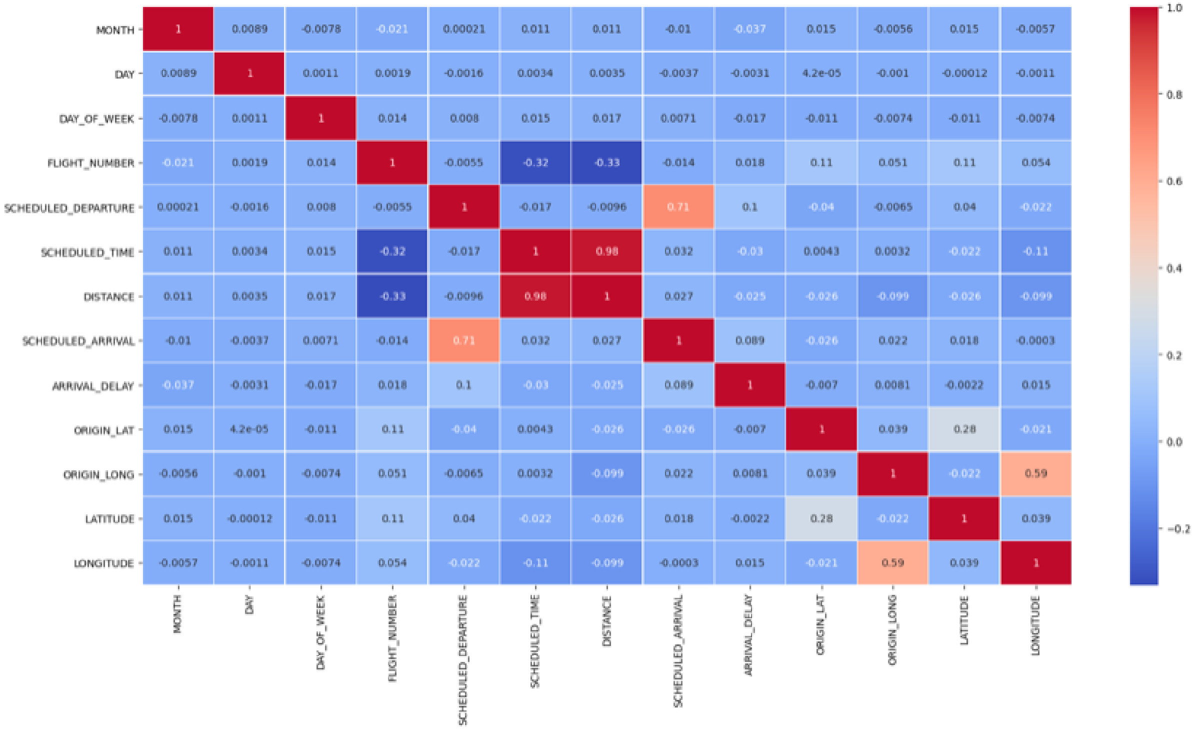
Problem Statement and Motivation

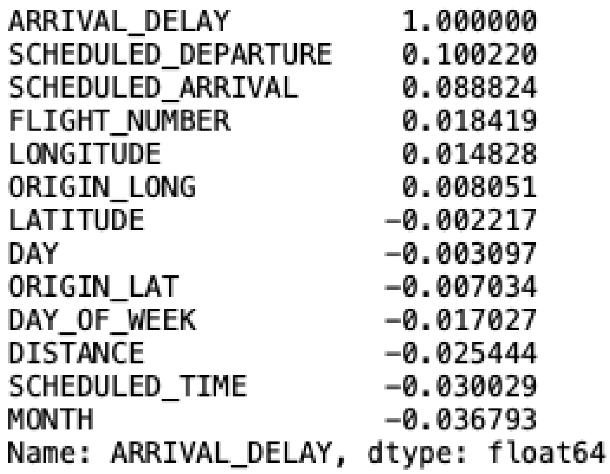
Flying is an amazing thing; 200 years ago, nobody would expect to be able to travel from New York to Los Angeles in less than one day, but that is our reality. Unfortunately, flying can be unreliable due to how often flights end up delayed, and recently this issue has been amplified due to a pilot shortage. Our goal with this project is to build a model that accurately predicts whether a flight will be on time or delayed and to figure out what you should pay attention to when scheduling a flight to give yourself the best chance of getting to your destination on time.

Introduction and Description of Data

Our dataset is made up of nearly 6 million flights that occurred in the U.S. in the year 2015, split into three tables (flights, airlines, and airports) that need to be joined together to include all of the information. With all the tables joined we have 38 columns in our dataset, but not all of them are useful for analysis, as some of the columns include information about the result of the flight, or information that is redundant and covered by other features (e.g. “Year”, “Airline”, “Tail Number”, “Cancellation Reason”, “Taxi In Time”, “Wheels on” and “Wheels off”). Those unnecessary features are not considered when building our models.

Apart from the irrelevant features, we also used the correlation matrix and heat map to gauge how useful each of our predictors may be.





```
ARRIVAL_DELAY          1.000000
SCHEDULED_DEPARTURE    0.100220
SCHEDULED_ARRIVAL      0.088824
FLIGHT_NUMBER          0.018419
LONGITUDE              0.014828
ORIGIN_LONG            0.008051
LATITUDE              -0.002217
DAY                   -0.003097
ORIGIN_LAT            -0.007034
DAY_OF_WEEK           -0.017027
DISTANCE              -0.025444
SCHEDULED_TIME        -0.030029
MONTH                 -0.036793
Name: ARRIVAL_DELAY, dtype: float64
```

Though some predictors have lower correlations than others, we decided to use all non-redundant features giving us a total of 12 features we can use when building our predictive models.

- SCHEDULED_DEPARTURE
- SCHEDULED_ARRIVAL
- FLIGHT_NUMBER
- LONGITUDE (that is, the longitude of the destination)
- ORIGIN_LONGITUDE
- LATITUDE (that is, the latitude of the destination)
- DAY
- ORIGIN_LAT
- DAY_OF_WEEK
- DISTANCE
- SCHEDULED_TIME
- MONTH

Of these, the biggest factor in whether or not a delay is likely to happen seems to be the scheduled arrival and departure times. This is one thing to watch carefully when selecting flights, and it will likely be very useful to our models.

One thing this dataset is missing is the actual result of the flight, that is, whether the flight was delayed or arrived on time. This is an easy fix with a little data preprocessing because we do have information on the expected and actual arrival time. Thus, we added a new feature called “FLIGHT_STATUS” which serves as the result of whether a flight is delayed or not. There are two values in this column. Any time the actual arrival time is greater (i.e. later) than the expected arrival time, we encode the flight as being delayed (or a 1 in the binary column denoting delays).

Modeling Approach

Model 1-(Baseline): Decision Tree Classifier

For our baseline model, a decision tree classifier was used. The 'FLIGHT_STATUS' was defined as the target. As the data is imbalanced, prior to developing a train/test split, random oversampling was used to enhance our accuracy as an additional pre-processing step. Data was split with a 30/70 test/train split. The use of the Decision Tree Classifier yielded an initial baseline of a 0.621 accuracy score with a 0.589 AUC score.

Model 2: Logistic Regression

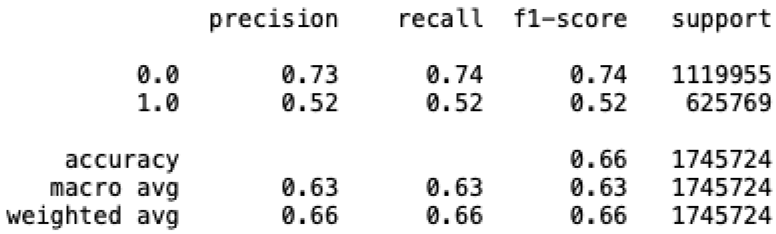
The second model employed was a Logistic Regression model with a max iteration of 150 and a liblinear solver. This model performed poorer than our baseline model, yielding an accuracy score of 0.568 and an AUC of 0.569.

Model 3: Random Forest Classifier

Our third tested model was a Random Forest Classifier with 100 estimators. This performed better than the Logistic Regression Model and our baseline model, with an accuracy score of 0.675 and an AUC of 0.635.

Ensemble Model: Voting Classifier (DTC, LR, RFC)

For our final step, we created an ensemble model using a Voting Classifier that combined all three of the previous models. This yielded an accuracy of 0.695 and an AUC of 0.628.



```
              precision    recall  f1-score   support

         0.0       0.73      0.74      0.74   1119955
         1.0       0.52      0.52      0.52    625769

    accuracy                           0.66   1745724
   macro avg       0.63      0.63      0.63   1745724
weighted avg       0.66      0.66      0.66   1745724
```

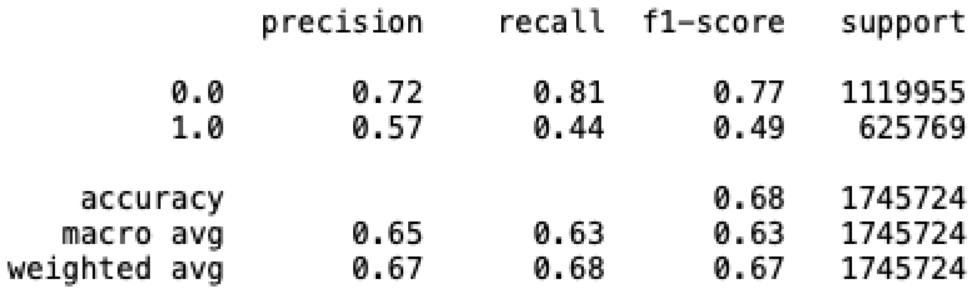
Ensemble Model: Blending

As an additional step, we ran the same models through a blending ensemble. This performed very similarly to our voting classifier ensemble, with a very slight improvement in accuracy (0.679 vs. 0.675) and slightly lower AUC (0.626 vs. 0.628

Project Trajectory, Results and Interpretation

The trajectory of our project remained relatively unchanged following the selection of the airline data set. Where our approach changed was in the feature selection for inclusion in the model. Initially, delays had been included as a factor towards cancellation. However, this gave us AUC and Accuracy scores that were far too large (0.995 for the final model). We then recognized that delays would have been recorded after a flight had landed, meaning that these features heavily skewed our data as the delay reasons meant a flight had been delayed (thus leading to high accuracy as using this as an X value means our Y status is delayed). We also realized early on that we needed to create a separate binary-coded column for flight status (delayed or not delayed) as this information was not an innate part of the data set (listing mainly reasons for delays and arrival time versus expected arrival time). We also had considered various target factors and ways to build our model, such as departure delay, or amount of delay in minutes based on reasons for the delay and other factors; but decided on the binary flight status (delayed or not delayed based on arrival delay) due to the manageable scope and easy factor selection.

The results of our final ensemble model yielded a fair accuracy score of 0.679. This means that it predicted the flight status correctly 67.9% of the time. While this could certainly be improved, it is far higher than our baseline model and the individual accuracy scores of our individual component models.The table below displays the accuracy of our model as well as the precision and recall scores which all range on average from 0.63-0.68.



```
              precision    recall  f1-score   support

         0.0       0.72      0.81      0.77   1119955
         1.0       0.57      0.44      0.49    625769

    accuracy                           0.68   1745724
   macro avg       0.65      0.63      0.63   1745724
weighted avg       0.67      0.68      0.67   1745724
```

Conclusion and Future Work

Based on the accuracy and AUC metrics for our final ensemble model, we can say that our model has met our goal of predicting flight delay fairly well based on the selected factors. One key strength of our model is the combination of a Decision Tree Classifier with a Random Forest Classifier, and a Logistic Regression model. While none of the three models performed very well on their own, the ensemble of them boosted the accuracy enough that we can say it performs fairly well in predicting flight status. However, the weaknesses of our model lie in a few key areas: Our dataset is imbalanced, though we did account for this with oversampling, it may still factor into the ability of the model to properly predict flight status when fed a new dataset. The algorithms that compose our ensemble are also not hyper-tuned, and a grid search to find the best hyperparameters for these algorithms may make our ensemble model stronger.

In the future, given more computational power we would like to hyper-tune all three of our algorithms prior to putting them together in an ensemble so as to increase our accuracy. It would also be interesting to look at other factors we had considered such as Departure Delay and Cancellation (vs. delay) as these factors would be useful to look at as well as arrival delays; however, when trying to implement these, the scope of our modeling efforts became too large and unfocused for the nature of this project. Other possible models could predict the amount of delay in minutes using the reasons for delay (weather, cabin pressure issue, etc.) and/or the airline and airport as factors. These were interesting correlational relationships we found in our data exploration; but were outside the scope of our model.