Predicting Flight Delays Based On Flight Characteristics

Lena Kim
https://github.com/wkim31/DATA1030_FlightProject.git

Brown University

1. Introduction

The Bureau of Transportation Statistics (BTS)– a segment of the United States Department of Transportation– collects, analyzes, and publishes data on domestic flights each month for every year since June, 2003. The BTS classifies flight delays, as reported by the Air Carrier On-Time Reporting Advising Committee, in four categories: Air Carrier Delay, Aircraft arriving late, National Aviation System (NAS) delay, and Extreme Weather Delay [1]. Figure 1 shows the proportions of each, along with cancellations and on-time flights, for the month of January, 2020, as reported in the March 2020 Air Traveller Consumer Report [3].

Aircraft takeoff delays cost airlines millions of dollars annually, and hold adverse effects on other flights as well. Fuel consumption while delayed consumes expensive fuel supply. Disgruntled passengers are monetarily compensated for the stress of an unexpected flight delay. Other flights sharing the runway strip are delayed. The airline carrier's consumer reputation falls.

A 2022 study by Alla et al. used a BTS dataset on 2017 flight statistics to predict flight delays and mitigate some of these adverse effects. The team's objective was to identify and leverage new microscopic features which contribute to flight delays to develop a predictive model. Using a training and testing split ratio of 70:30, they apply K Nearest Neighbors, Decision Trees, and Random Forest, to achieve their highest accuracy score of 0.9356 with Random Forest. Their pipeline is shown in fig 2. Their methodology and applied results are proposed to be used as a decision support tool for aircraft traffic controllers [2].

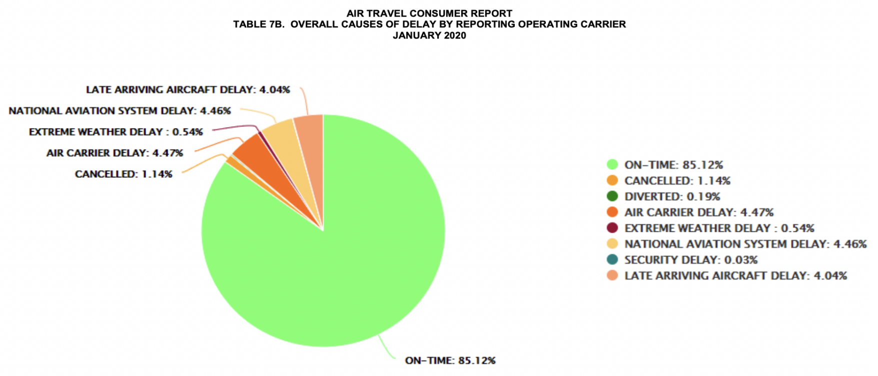

Fig 1: January 2020 Flight Status Stats

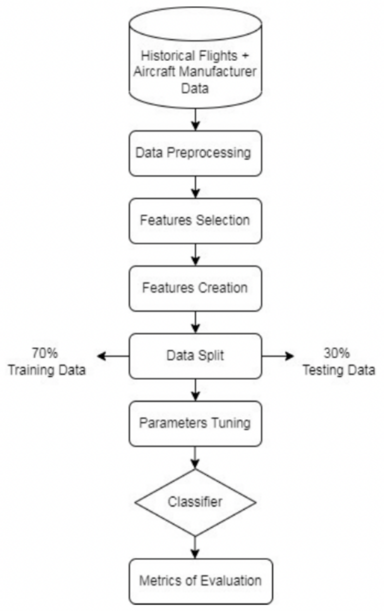

Fig. 2. Flowchart of the proposed method

The data used for *Predicting Flight Delays Based on Flight Characteristics* is a collection of domestic flight data from the BTS database for two months: January 2019 and January 2020. Throughout the United States, there were approximately 500,000 flights per year recorded, for a total of 1,168,277 flights [4].

To mitigate the adverse economic, reputative, and consumer effects of unexpected flight delays outlined above, *Predicting Flight Delays Based on Flight Characteristics* builds a Machine Learning classification model that predicts whether a flight will get delayed based on specific flight characteristics such as origin airport, time of departure, day of departure, airline carrier, and more. Including these features, flight_df –- the representative pandas dataframe for the aggregate flights across both years – contains 21 features. The target variable for prediction is DEP_DEL_15, a binary indicator variable that takes on a value of 1 if the flight was recorded as delayed by 15 minutes or more, and 0 if either canceled or on-time. DEP_DEL_15 assumes the definition of flight delayal based on the Air Carrier On-Time Reporting Advising Committee's four categories.

2. Exploratory Data Analysis

The first step in exploring the dataset was to explore the features of the dataset. A full description of each variable is summarized in Dataset_Description.txt in the linked GitHub. A summary of the key feature and types are described in the below table.

OP_UNIQUE_CARRIER	Unique Carrier Code of flight	object

ORIGIN_AIRPORT_ID	Airport ID of origin airport according to IATA standards	int64
DEST_AIRPORT_ID	Airport ID of destination airport according to IATA standards	int64
DEP_TIME_BLK	Departure Time Block in hourly intervals	object
DEP_DEL_15	Departure Delay Indicator, 15 Minutes or More (1=Yes, 0=No)	float64

Next, a generated Pearson Correlation Map on feature correlations. Except for ARR_DEL_15 and DEP_DEL_15, not a lot of features are correlated, discounting self-correlation. ARR_DEL_15 and DEP_DEL_15 are correlated because whether the flight arrived 15 minutes later is very much correlated with whether it departed 15 minutes later, which suggests multicollinearity. ARR_DEL_15 is later dropped in our flight_df.

Next, popular carriers: SouthWest Airlines, with an IATA (International Air Transportation Association) code of WN, is the most popular airline for domestic flights.

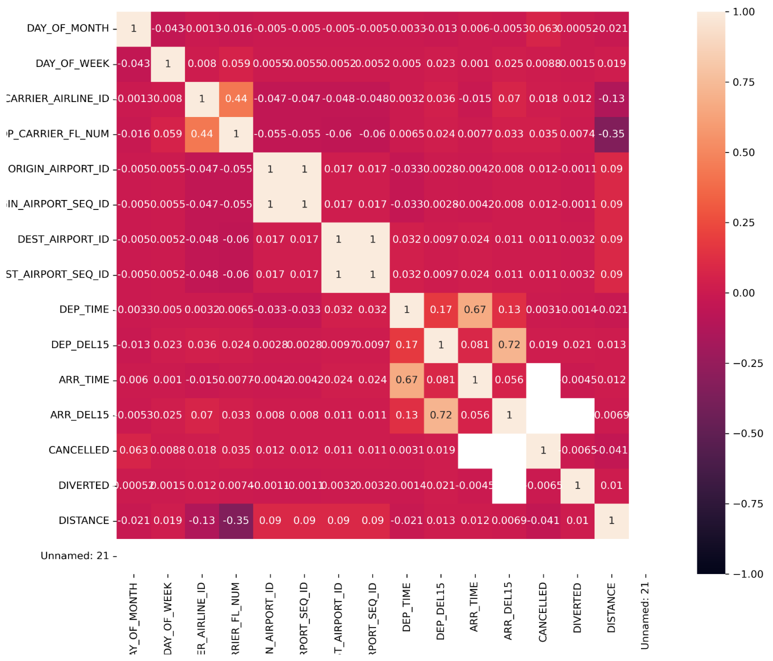

Fig 3: Pearson Correlation Map of Features

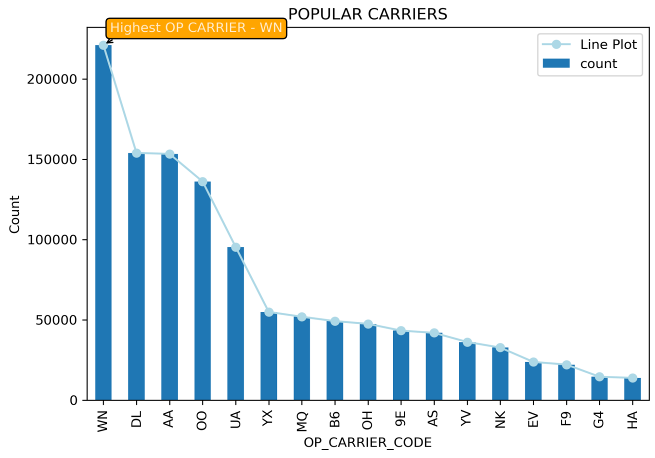

Fig 4: Airport Carriers by IATA code

It is also important here to consider the spreads of the time bulks that these flights departed in, as well as what day of the week most flights took off in. According to the figures below, morning and early to late afternoon flights are the norm. There is a lull in flights on the weekends of January, and Mondays are the busiest airport days.

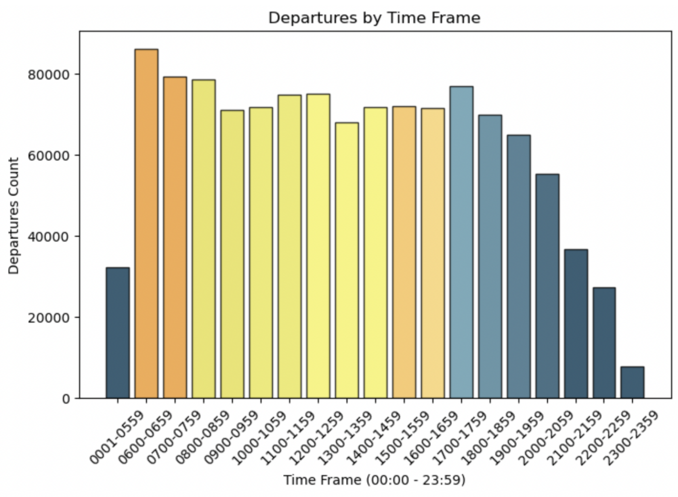

Fig 5 : Departures by Time Frame

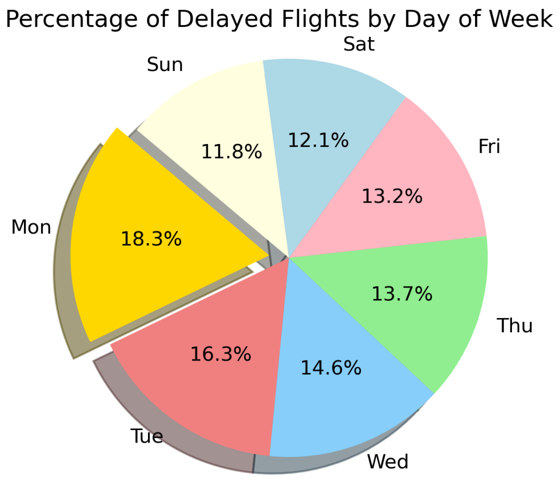

Fig 6: What Day did Flights Occur?

Finally, the fraction of the missing values of some features are described below.
Most features do not have missing values, but either human error or technical difficulties result in 3 % of TAIL_NUM, 1.9% of DEP_TIME and DEP_DEL_15 as missing in the records.

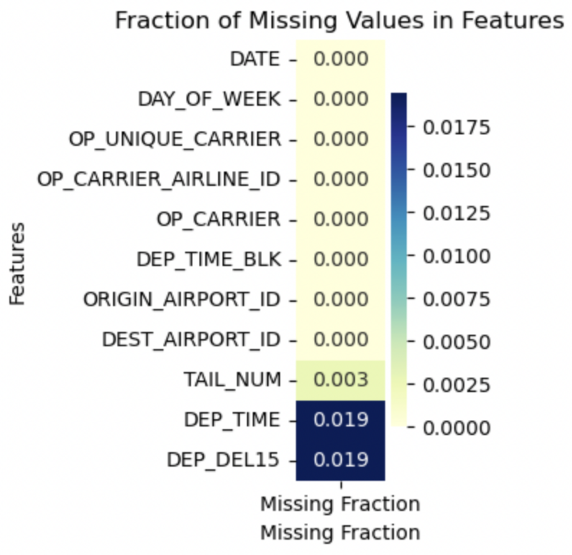

Fig 7: Missing Values

3. Methods

The splitting strategy is based on implementing a group-based K fold strategy, where groups are sectioned off by airport. Specific airports are strongly relevant to what influences flight delays due to different runway sizes, communication systems, and control systems. 2019 sets are used as training and 2020 for validation and testing to respect the temporal nature of df_flights, and to assess if the model can generalize between times. A 50-25-25 split is implemented using scikit-learn's train_test_split to apply all of the goals outlined above.

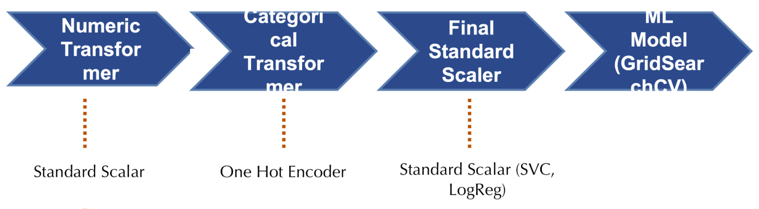

Fig 8: Preprocessing

For data preprocessing above, an implementation of a Standard Scalar on numerical features such as departure times and distances were implemented. An implementation of a OneHotEncoder on categorical features such as day of week, carrier IATA ids, and destination major airports were applied as well. For linear models, standard scalars were applied before OneHotEncoding for later analysis of feature importance. The training set was fit_transformed, and testing and validation were transformed as well.

Four models (Section on Implementing Algos on Split Data) were implemented using GridSearchCV to find the best hyperparameters and best models with cross validation. These models are two linear models (Support Vector Classifier (SVC), Logistic Regression (LogReg), and two nonlinear models (Random Forest (RF), and XGBOOST).

Table 1: Models and Tuned Hyperparameters

ML Model	**Hyperparameters**	**Values**
SVC	C	'C': [0.01, **0.1,** 1, 10]
LogReg	C	'C': [0.01, 0.1, 1, 10, **100]**

RF	max depth max features	'max_depth': [3, 6**, 9**, None], 'max_features': ['sqrt', 'log2', **None**]
XGBOOST	max depth reg alpha	'max_depth': [3, **6**], 'reg_alpha': [0, 0.1, 0.5, **1.0**]

Attempts to tune the gamma parameter in SVC were not appropriate in a linear kernel, as gamma controls the influence of individual training samples in kernel based SVMs. Attempts to tune the penalty term in LogReg encountered future warnings in Jupyter notebook. Best hyperparameters found by GridSearch are bolded.

The evaluation metrics chosen were accuracy score. See next section for more.

4. Results

The main evaluation metric considered was accuracy score for its interpretability. The calculated baseline was 0.864, meaning a random guess predicting that it is not delayed (class 0 for DEP_DEL_15) is right 86.4% of the time. Based on the mean accuracies of the models on the test set, XGBOOST performed the best with a mean accuracy score of 0.94. This score is by far the best. Random Forest stood second best at around 0.89. See the below table for more:

Table 2: Accuracy Scores of Models. Baseline: 0.864

ML Model	**Accuracies**	**Mean Accuracies**	**Std Dev Above Baseline**
SVC	Test Accuracy for model {'C': 0.01}: 0.8670768902918181 Test Accuracy for model {'C': 0.1}: 0.8669037439565269 Test Accuracy for model {'C': 1}: 0.8660346825428538 Test Accuracy for model {'C': 10}: 0.8658715254192139 Test Accuracy for model {'C': 100}: 0.8658781848936482	**0.87**	**0.6**
LogReg	Test Accuracy (C=0.01): 0.8738662244775642 Test Accuracy (C=0.1): 0.891706956486994 Test Accuracy (C=1): 0.8968181031153022 Test Accuracy (C=10):	**0.89**	**2.6**

	0.8966649352033138 Test Accuracy (C=100): 0.8965317457146282		
RF	Test Accuracy for model {'max_features': 'log2', 'max_depth': None}: 0.8763835058137212 Test Accuracy for model {'max_features': 'sqrt', 'max_depth': None}: 0.9057384691200171 Test Accuracy for model {'max_features': 'sqrt', 'max_depth': 3}: 0.8641567107523874 Test Accuracy for model {'max_features': None, 'max_depth': 9}: 0.9128641067646941 Test Accuracy for model {'max_features': None, 'max_depth': 6}: 0.8824869141327366	**0.89**	**2.6**
XGBOOST	Test Accuracy for model {'reg_alpha': 0.1, 'max_depth': 3}: 0.9397350861070044 Test Accuracy for model {'reg_alpha': 0.1, 'max_depth': 6}: 0.9410070457239514 Test Accuracy for model {'reg_alpha': 0, 'max_depth': 3}: 0.9393754744875534 Test Accuracy for model {'reg_alpha': 1.0, 'max_depth': 6}: 0.9412068299569798 Test Accuracy for model {'reg_alpha': 0.5, 'max_depth': 3}: 0.9399681677122042	**0.94**	**7.6**

*Standard Devs from Baseline is calculated from baseline mean of 0.864 and sd 0.01

To study which features are the most important for the predictions, 5 global importances and local feature importances were explored for our best model, XGBOOST. First, XGBOOST's 5 importance metrics were calculated, including: weight, gain, cover total gain, and total cover. It looks as though the most important features were the time bulks that the flight took off in, especially the afternoon times.

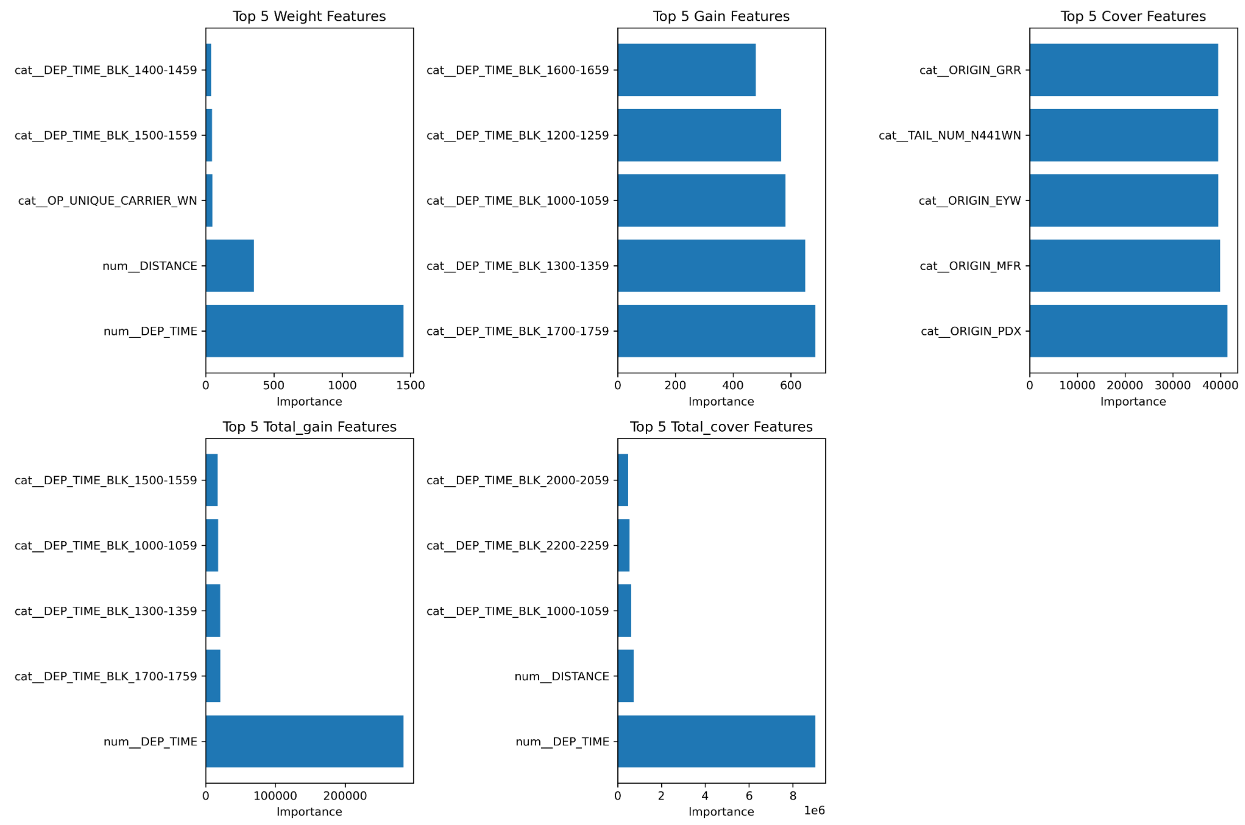

Fig 9: Importance Metrics

SHAP local feature importances were also calculated for the model. For local importances, the most important metrics were, again, departure time. Figure 10 shows a SHAP force plot of the impacts of each feature on the model's prediction.For this data point, the actual flight was delayed. The model predicted that it indeed, was delayed (class 1 for DEP_DEL_15).

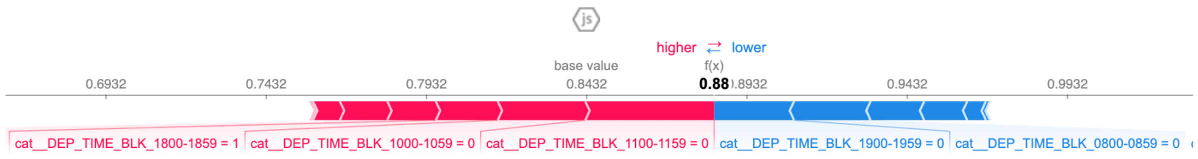

Fig 10: Shap for Importance

The features that drove the prediction up– that is, it made that flight more likely to be predicted delayed– were the fact that that flight did not take off from 10:00 am - 10:59 am, but rather took off from 6:00 - 6:59 p.m. The features that drove the prediction down – it made the flight less likely to be predicted delayed– was the fact that it did not take off from 7:00-7:59 p.m.

From this, it is not folly to conclude that evening flights (from 7:00pm on) are more likely to make a flight delayed, and morning flights (before noon) are more likely to make a flight delayed as well.

The least important feature that was mildly surprising that failed to be displayed anywhere in the importance metrics was day of week. Whether the flight is scheduled for departure on Monday or Saturday makes no difference in flight delays.

5. Outlook

To improve the project further, strategies to increase model predictability and interpretability can be taken. For example, more hyperparameters can be tuned, such as increasing the C parameter in LogReg to 1000. A lot of the hyperparameters tuned were at the edges of the list to be tuned, so given more time and a larger Jupyter RAM, an improvement to the predictive power (and thus the accuracy score evaluation metric) can be strained out. Additionally, more models such as Gradient Boosting can be implemented to see if there are models out there that are better predictively [5]. A brief implementation of Gradient Boosting resulted in a 90% accuracy score, but additional hyperparameters would have to be tuned.

For interpretability, LIME (Local Interpretable Agnostic Explanations) can further be used to explore the nuances of even more local features for individual predictions. Additionally, in terms of expanding the model further, data across years instead of data across months can be collected and used in this pipeline as well to implement a more representative data set.

References

[1]https://www.bts.gov/topics/airlines-and-airports/understanding-reporting-causes-flight-delays-and-cancellations

[2]https://www.researchgate.net/publication/363223003_Impact_of_Aircraft_Performance_and_Time_of_the_Day_on_Flight_Arrival_Delays_Prediction_in_the_United_States_a_Machine_Learning_Classification

[3]https://www.transportation.gov/sites/dot.gov/files/2021-02/March%202020%20ATCR%20rev%202-11-21.pdf

[4] https://www.kaggle.com/datasets/divyansh22/flight-delay-prediction

[5]https://towardsdatascience.com/all-you-need-to-know-about-gradient-boosting-algorithm-part-1-regression-2520a34a502