Will my Flight be late?

Team: Simon Guichandut, Tim Hallat, Ketan Sand

GitHub: https://github.com/simonguichandut/WhyIsMyFlightLate
Website - https://willmyflightbelate.streamlit.app/

Overview

Flight delays are not only bothersome but also widespread, causing over 200,000 hours of combined delay annually in just 20 of the busiest airports in the United States. This results in a staggering $32.9 billion annual economic loss for the US. The ability to understand the contributing factors and predict delays is crucial for better preparation and minimizing the impact.

To address this issue, we utilized 12 years' worth of data from the Bureau of Transportation Statistics in the US. The dataset was refined to focus on flights between the top 20 busiest airports, operated by the top 10 airline carriers. We employed a random forest model for training, predicting both the likelihood of delay and quantifying the delay duration.

A user-friendly website (https://willmyflightbelate.streamlit.app/) was developed to enhance the overall experience.

Stakeholders: Travelers, Airlines, Airport Authorities, Government (Tourism Sectors), Insurance companies.

Key Performance Indicators (KPIs) - Precision/Recall: False Positive vs. False Negative rates, Receiver Operating Characteristic (ROC), Detection Error Tradeoff (DET)

Approach - We took 12 years of data from 2012 to 2023. We then removed the canceled flights and only kept flights from the 20 of the busiest airports of the US. We made another cut to keep top 8 flight carriers.

We then preprocess our data to make it ready for modeling, with following steps -

- **Predicted variable**: Delay times >15 min are categorized as TRUE
- **Scaling**: Normalize flight duration to 0-1
- **Categorical variables**: One-hot encore carrier and airport variables
- **Cyclic variables**: Convert day & year fraction to sin & cos components

We split the data into six chunks to identify the best training set for our model.
2020-2023, 2020-2023 (No COVID), 2016-2023, 2016-2023 (No COVID), 2012-2023, 2012-2023 (No COVID). Where No COVID means removing years 2020 and 2021 as we found the data to be unreliable during our exploratory data analysis.

Finally we run 3 models on each of these datasets: Logistic Regression, Random Forest and XGBoost.

As a second step we also divided our delays into 3 categories - 15 to 30 mins, 30 to 60 mins and more than 60 mins, to train a model that can provide an estimate on delay. We used the same models as above. Though further optimization is needed here.

Results & Strategies- We looked at the Precision-Recall Curve, Gain Curve, ROC and DET for each date range and each model (shown in slides). We found that training on the last 2 years of data that is the best and the most optimized model is Random Forest.
Finally our training set was 2022 to June 2023 and we tested our model on July and August 2023 dataset. Here are the final values -

On delayed flight category	Precision	Recall	F1-score	Accuracy
Validation set	0.69	0.72	0.71	0.87
Testing set	0.5	0.7	0.58	0.72

We see our precision is low on the test set, but it is to be noted that the fraction of delayed flights is actually quite low as well. Our precision and recall are more than 90% in predicting non delayed flights since we have more data there.

In terms of features we find that Flight duration, time of the day and time of the year matter the most in predicting delays. Carrier and origin airports have a small but significant part to play. This behavior can be understood better.

For delay range prediction we found the random forest model works the best, but further optimization is needed there.

We finally uploaded this all to a web-app to make this experience more user-friendly.

Future Iterations -

- Calibration of model: predict probability, rather than Yes/No
- Delay range prediction: Optimize and Add to the Website
- Hyperparameter tuning: RandomForest takes too long
    - n_estimators, max_features, max_depth, min_samples_split, min_samples_leaf, bootstrap
    - Using RandomForestRegressor and RandomizedSearchCV
    - Notebook in repo
- Train Model that also provides the cause of the delay
- Get more parameters and apply neural networks