Reliable Air Travel; Stratified Analysis of Flight Delay

DAVID OJOMO

M.Sc. Data Science Student
University of Hertfordshire

Github link: https://github.com/davidojomo/Flight-Delay-Analysis-MSc-Project/tree/main

Introduction

Motivation

- This project, “Reliable Air Travel; Stratified Analysis of Flight Delay,” aims to explore how reliability interact within the air travel industry.
- The study will look at the main factors that influence delays in domestic flights within the United States and how these vary between different routes.
- It will also test how well machine learning models can predict flight delays using operational data, and how these predictions may differ across routes and airlines.

Research question

1. What are the key factors influencing departure and arrival delays in domestic U.S. flights, and how do these vary by airline and airport?
2. To what extent do machine learning models predict flight arrival delays based on operational and environmental features, and how do these predictions differ across airlines and routes?

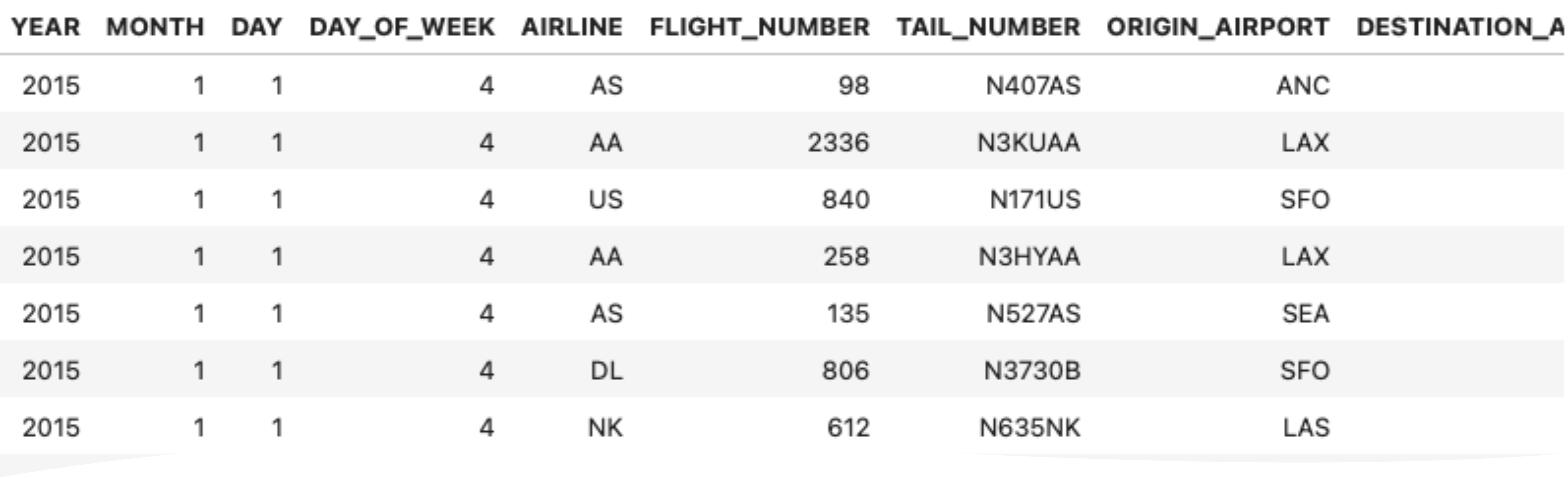
Overview of previous work

Title	Authors	Methods
Characterization and prediction of air traffic delays	(Rebollo & Balakrishnan, 2014)	k-Means clustering, Random Forests, Regression
Predictive Modeling of Flight Delays at an Airport Using Machine Learning Methods	(Hatıpoğlu & Tosun, 2024)	Logistic Regression, Naïve Bayes, Neural Networks, Random Forest, XGBoost, CatBoost, and LightGBM
Optimizing Flight Delay Predictions with Scorecard Systems	(Jacyna-Gołda et al., 2025)	operational forecasting Scorecard
Flight Delay Prediction Using Machine Learning	(Reddy et al., 2023)	SVM, Random Forest, Regression Decision Tree
Flight delay forecasting and analysis of direct and indirect factors	(Wang et al., 2022)	attention mechanism (LSTM-AM)

- Bibliography of references included at the end of the presentation

How does the project compare to the previous work?

- **Broader scope**: Compare delay patterns across airlines and airports using detailed time, flight, and cause data offering broader insights than prior studies focused on single airlines or the entire system.
- **Sharper insights**: Assess prediction accuracy to reveal key operational delay factors.
- **Real world impact**: Handle class imbalance, highlight feature importance, and link findings to practical airline decisions.



YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATION_A
2015	1	1	4	AS	98	N407AS	ANC	
2015	1	1	4	AA	2336	N3KUAA	LAX	
2015	1	1	4	US	840	N171US	SFO	
2015	1	1	4	AA	258	N3HYAA	LAX	
2015	1	1	4	AS	135	N527AS	SEA	
2015	1	1	4	DL	806	N3730B	SFO	
2015	1	1	4	NK	612	N635NK	LAS	

Dataset

Type of files - .csv files of flights data, airports and airlines

Size of the files - USA flights delay data (5.8 million rows, 592 MB), airports (24 KB), airlines (359B)

https://www.kaggle.com/datasets/usdot/flight-delays

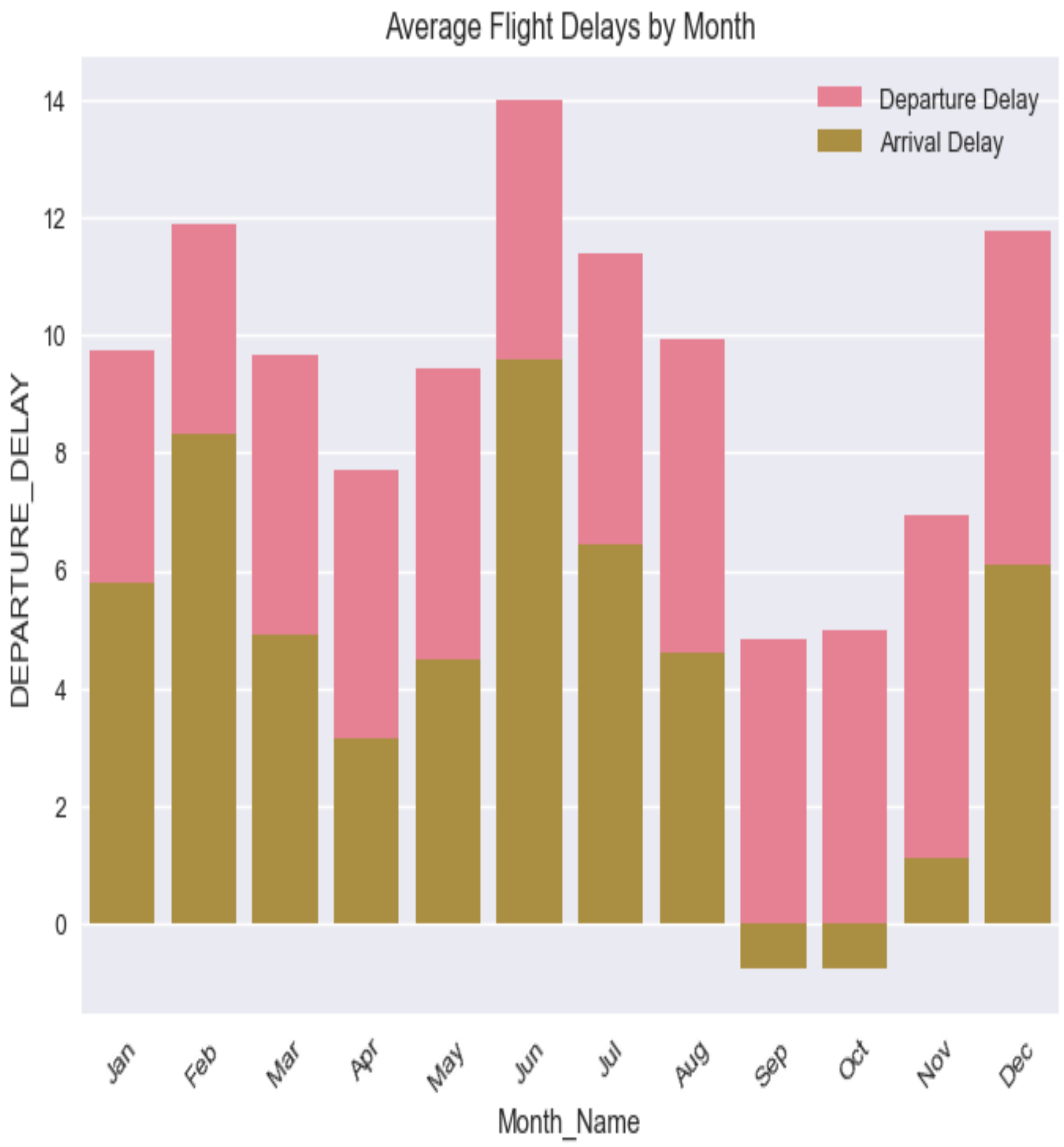
Ethical implications

- No personal data
- Consent was obtained to use the data for this project – The data is from a public website and has a public GitHub License.
- Data is anonymized
- Data storage - secure

EDA

- Flight Statistics

Total flights: 5,819,079
Cancelled Flights: 89,884 (1.54)%
Diverted Flights: 15,187 (0.26)%
Ontime Flights: 3,606,117 (61.97)%



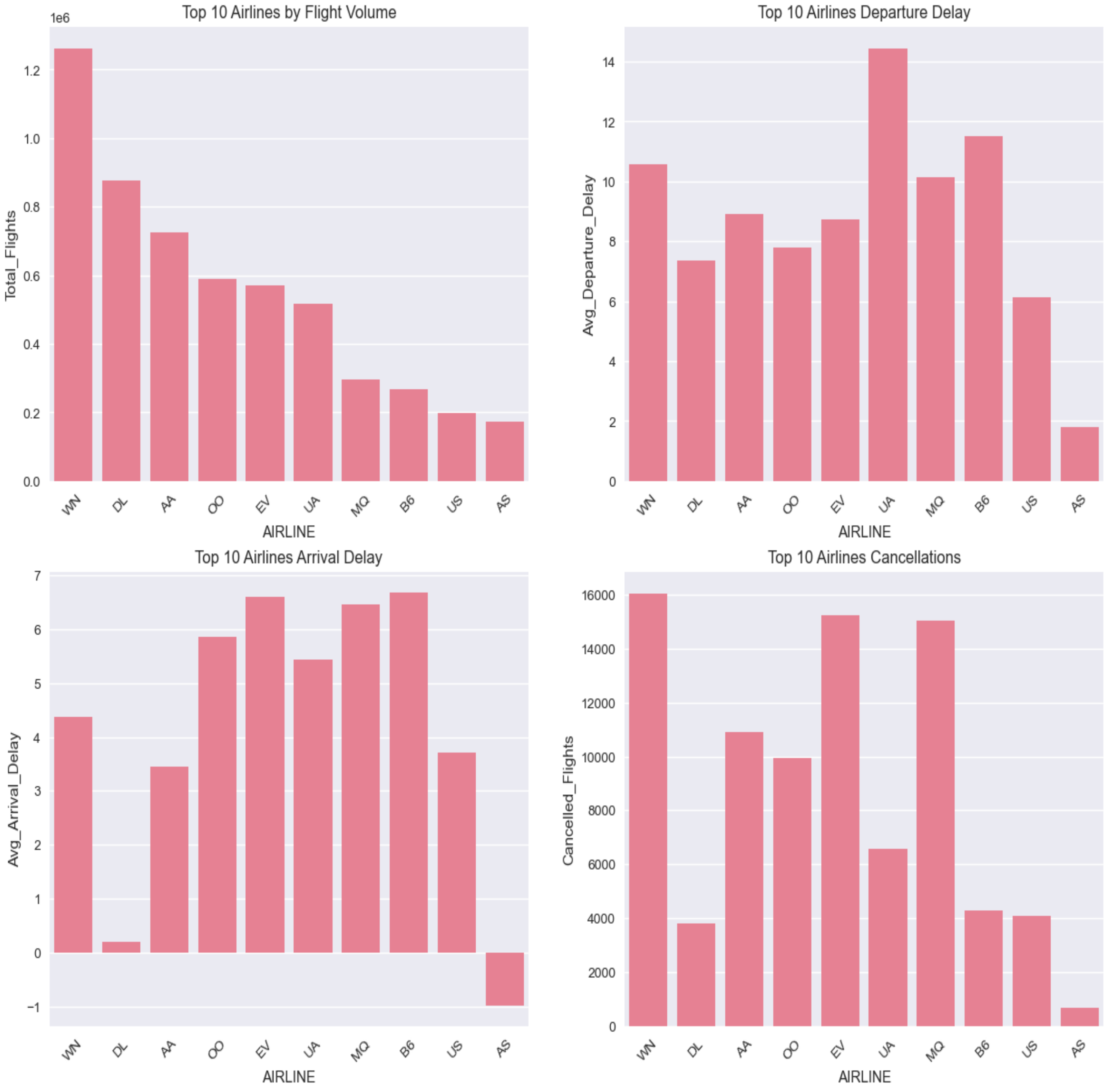
EDA

- Airline Statistics

Top 3 airlines by flight volume - Southwest Airlines, Delta Airlines and American Airlines

Top 3 airlines by departure delay - United Airlines, Southwest Airlines and JetBlue Airways

Airlines with least departure and arrival delay - Alaska Airlines

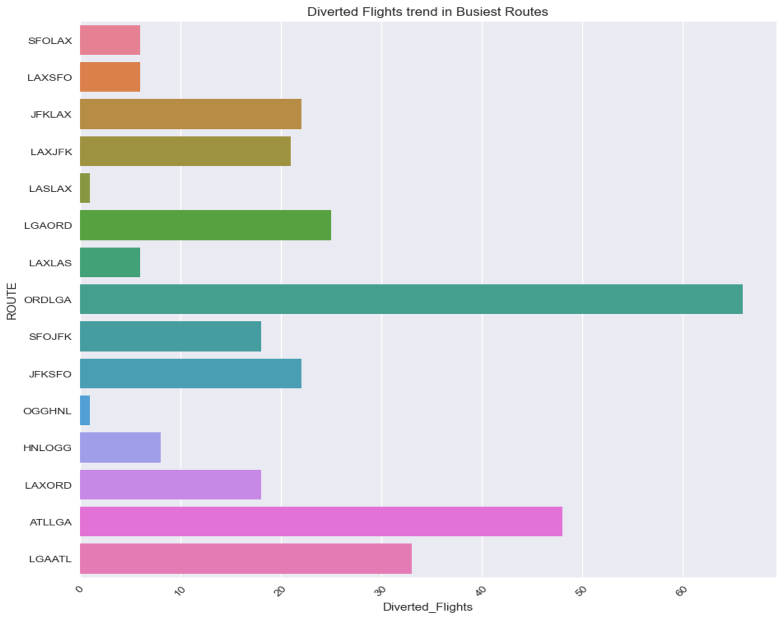


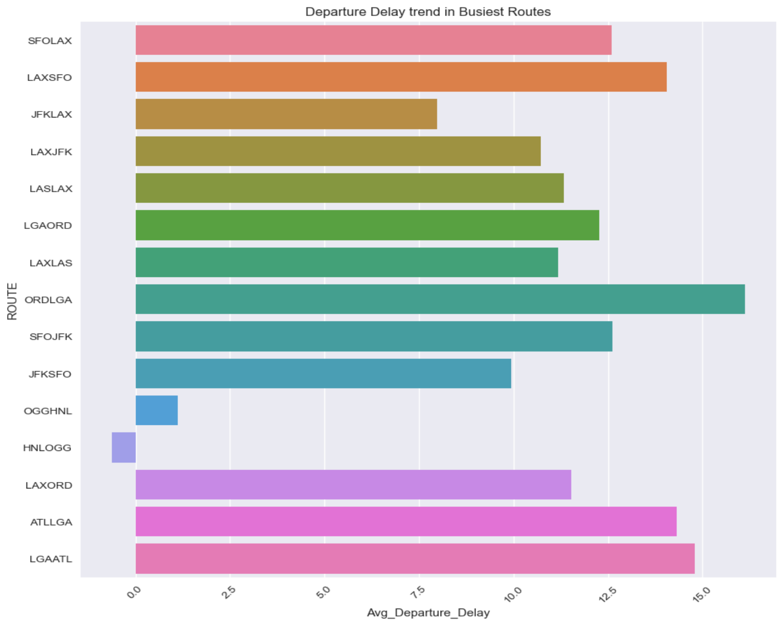
EDA

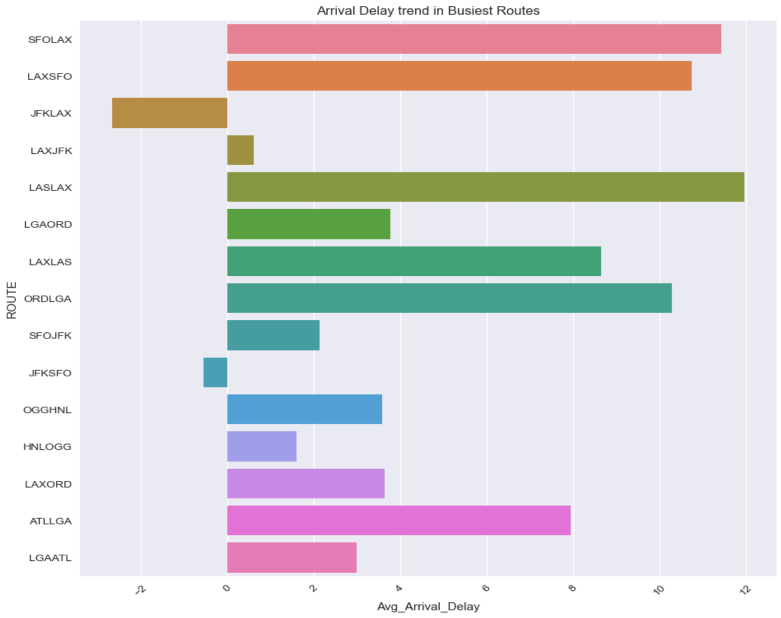
- Route Statistics

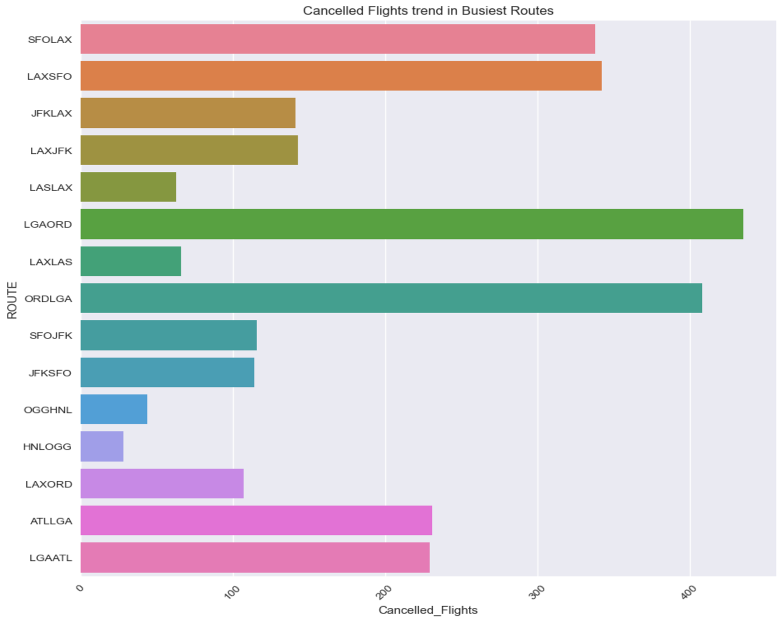
Routes with maximum departure delay - Chicago-La Guardia NYC and La Guardia NYC-Atlanta

Routes with least departure delay - Maui-Honolulu, Hawaii





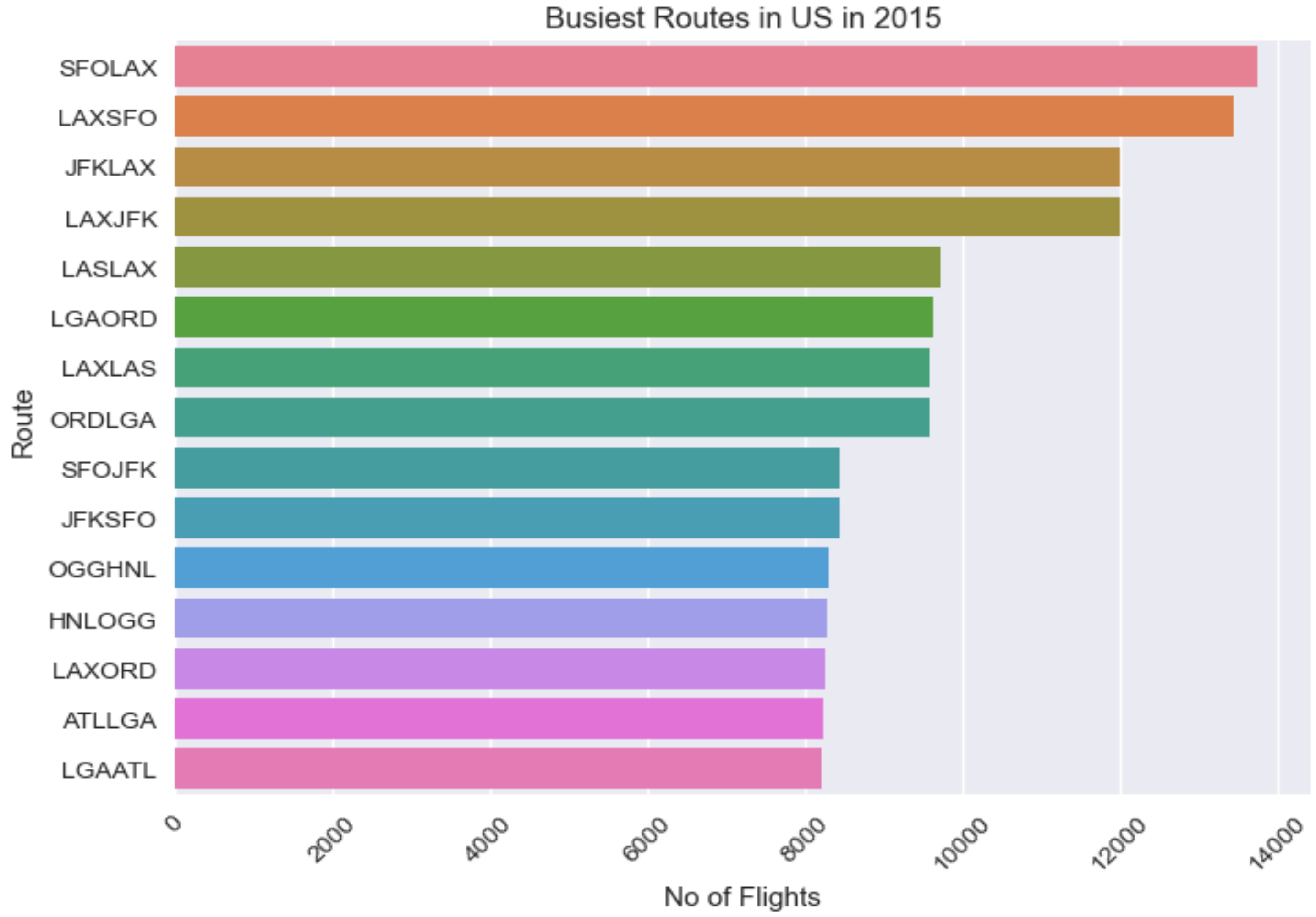




EDA

- Route Statistics

Top 3 busiest routes – San Francisco-Los Angeles, Los Angeles-San Francisco, NYC-Los Angeles



Data Balancing

- Dataset is imbalanced: 81.83% on-time flights vs 18.17% delayed flights
- Without balancing, models would bias toward predicting "on-time" for everything
- Used class_weight='balanced' parameter in sklearn models
- Automatically adjusts weights inversely proportional to class frequencies
- Recommended by research literature (Esmaeilzadeh & Mokhtarimousavi, 2020)

Code:

```
rf_model = RandomForestClassifier(
  n_estimators=100,max_depth=15,
  min_samples_split = 100,
  min_samples_leaf = 50,
  class_weight = 'balanced',
  random_state=42, n_jobs=-1)
```

Pre-processing

Raw Data (5.8M flights, 30 columns)

↓

Feature Engineering (7 new features created)

↓

Label Encoding (7 categorical → numerical)

↓

RFE Feature Selection (16 → 12 features)

↓

Train-Test Split (80/20 stratified)

↓

Ready for Modeling

Data Augmentation: No synthetic augmentation used; Class weighting applied instead to handle imbalance efficiently while preserving 5.8M real-world flight records

Model used

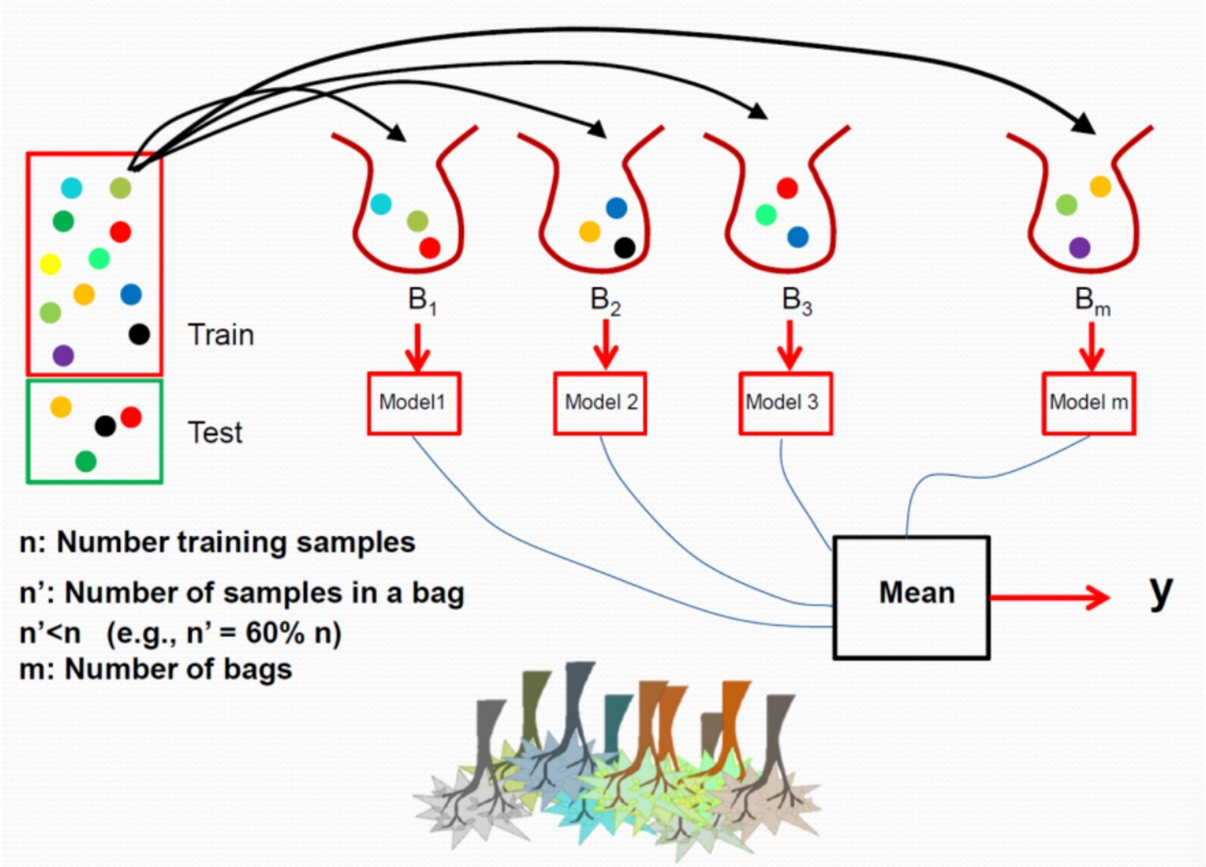
Random Forest:

Ensemble method that builds multiple independent decision trees on random data subsets.

With each tree voting on the final prediction; reduces overfitting through averaging and provides built-in feature importance rankings

Why Used:

Robust to imbalanced data when combined with class weighting; Provides interpretable feature importance for RQ1 analysis.



Source: Lecture 2 Page 74 of Machine Learning and Neural Networks

Model used

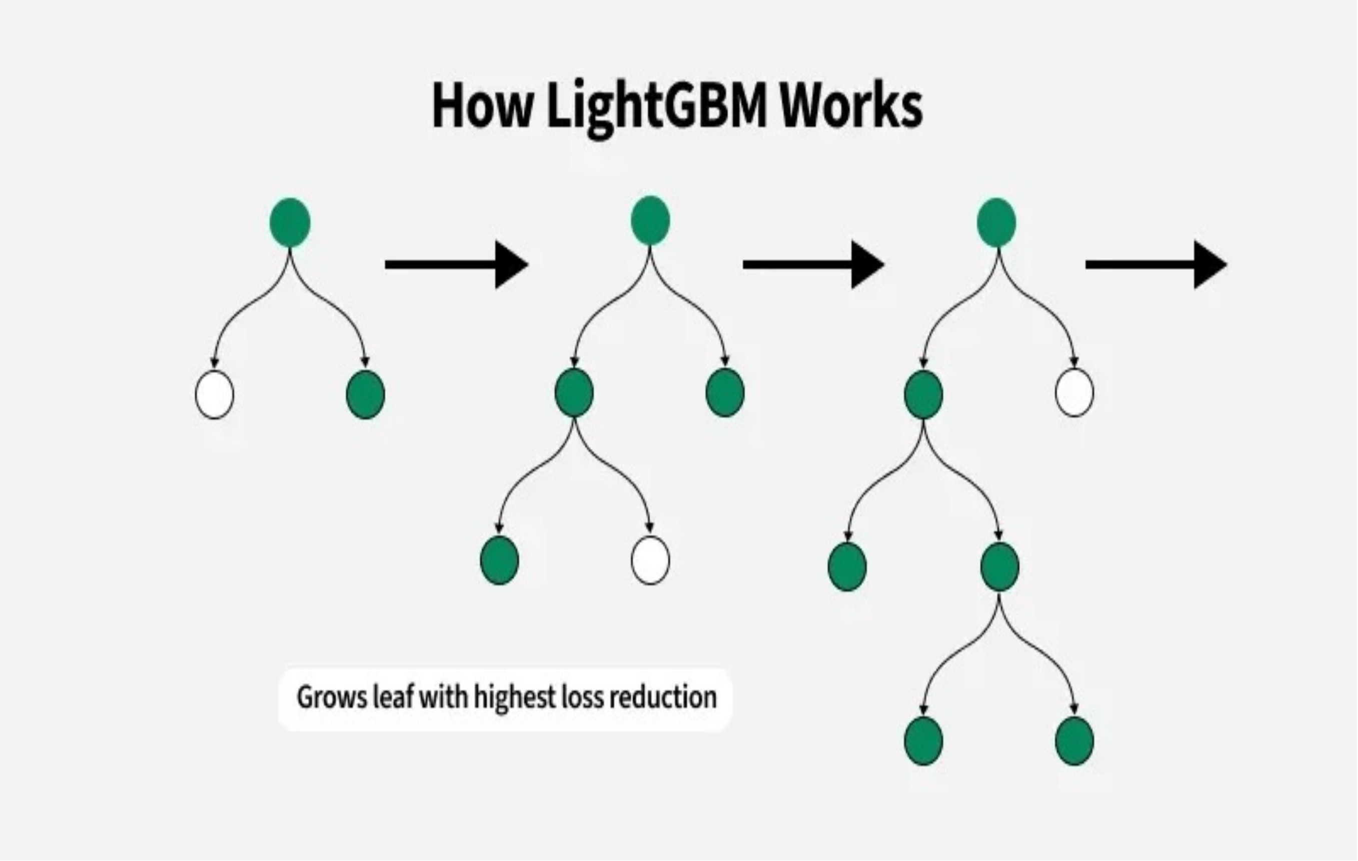
Light GBM:

Sequential ensemble method where each tree corrects errors from previous trees; uses efficient leaf-wise growth strategy optimized for large datasets and faster training

Why Used:

Handles large datasets efficiently; Often achieves higher accuracy than Random Forest.

Supports class imbalance through scale_pos_weight parameter; Fast training time despite dataset size

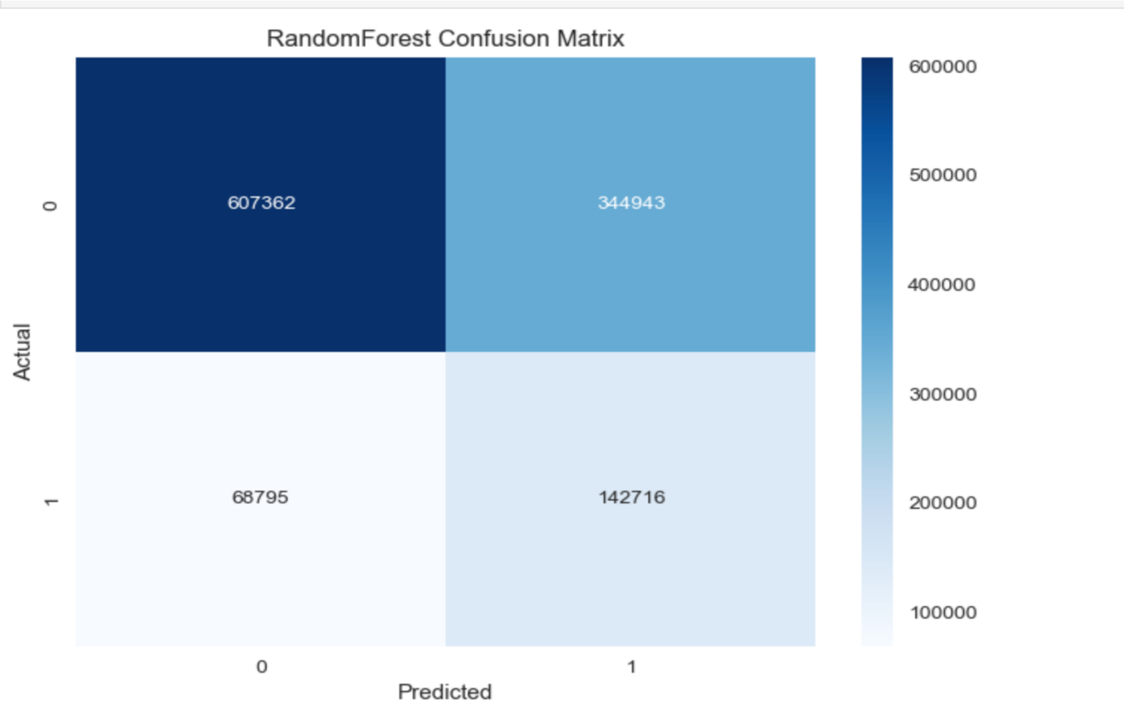


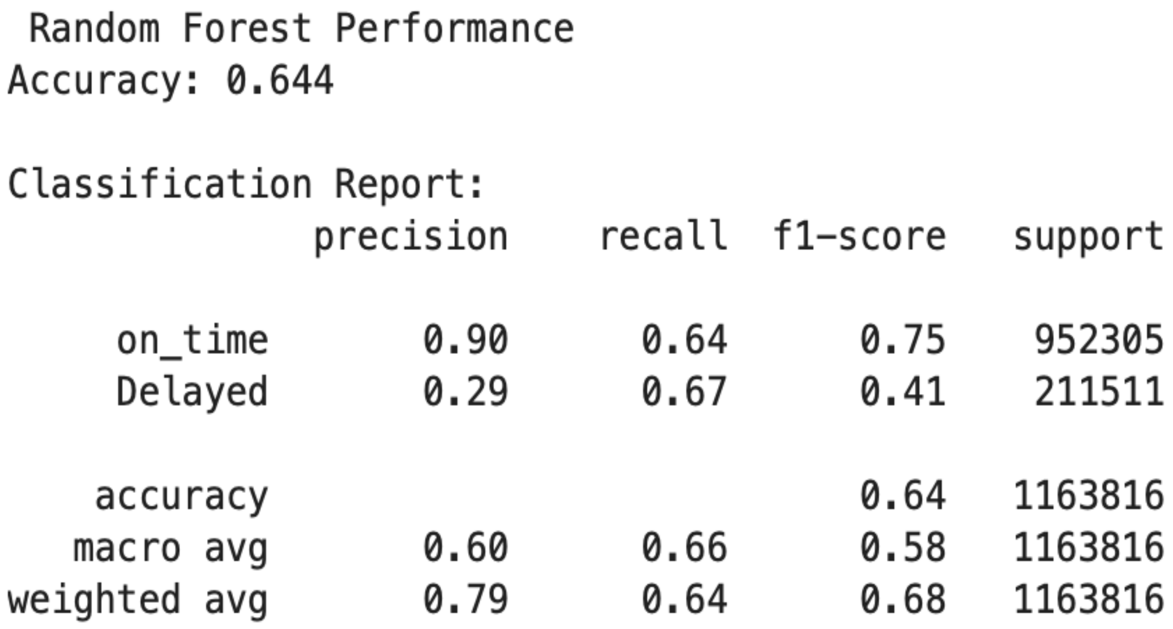
Source: LightGBM (Light Gradient Boosting Machine), https://www.geeksforgeeks.org/machine-learning/lightgbm-light-gradient-boosting-machine/

Model result

Random Forest Performance

- Accuracy: 64% with balanced precision-recall trade-off for both delayed and on-time classes
- Strong feature importance extraction capability, identifying DISTANCE, TAXI_OUT, and SCHEDULED_HOUR as top predictors
- Robust generalization with consistent performance across different data subsets; minimal overfitting due to ensemble averaging



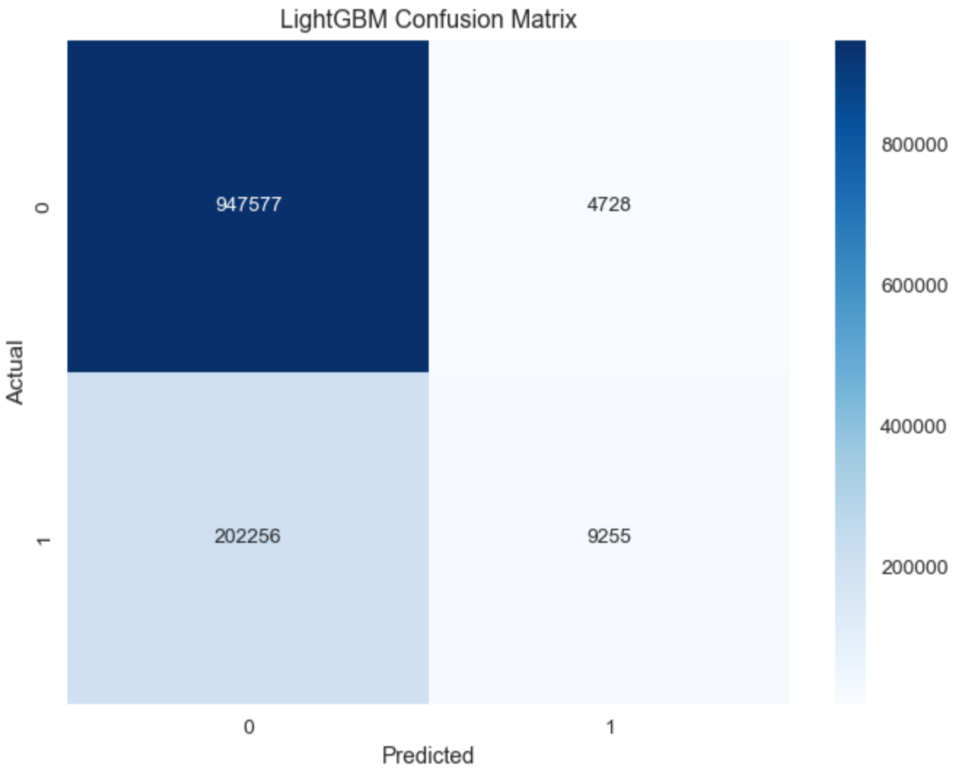


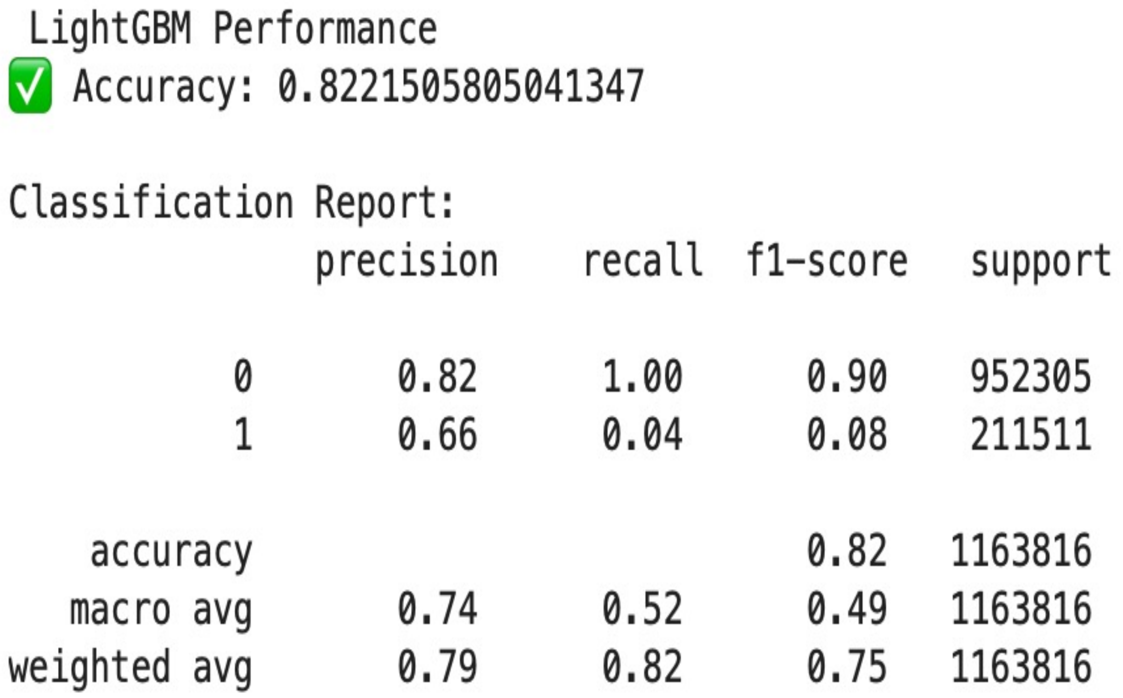
```
 Random Forest Performance
Accuracy: 0.644

Classification Report:
              precision    recall  f1-score   support

     on_time       0.90      0.64      0.75    952305
     Delayed       0.29      0.67      0.41    211511

    accuracy                           0.64   1163816
   macro avg       0.60      0.66      0.58   1163816
weighted avg       0.79      0.64      0.68   1163816
```

Model result

Light GBM Performance:

- Higher accuracy than Random Forest due to sequential error correction and optimized boosting
- Faster training time (50% reduction) despite large dataset size (5.8M samples) through efficient leaf-wise tree growth
- Superior handling of complex non-linear patterns with better recall for minority class (delayed flights)

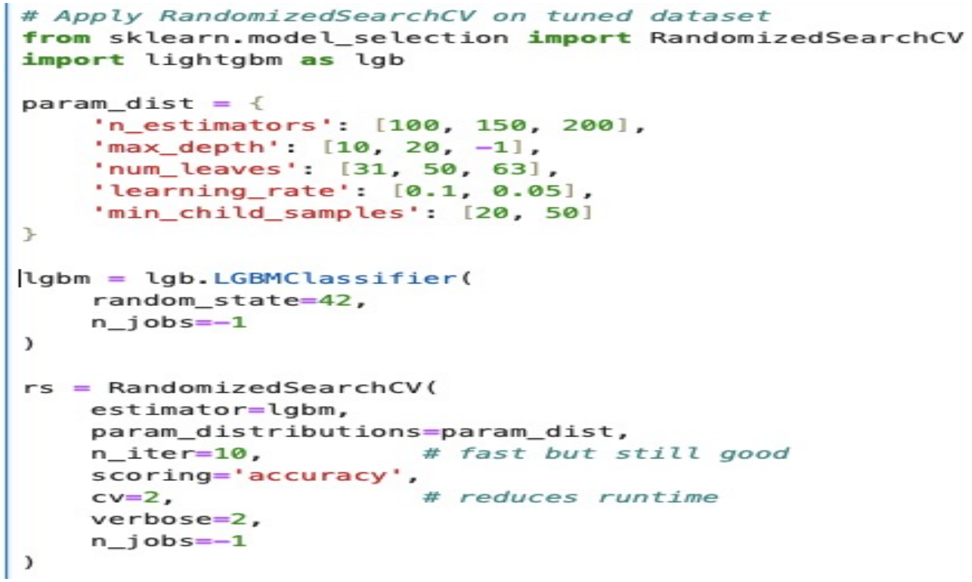




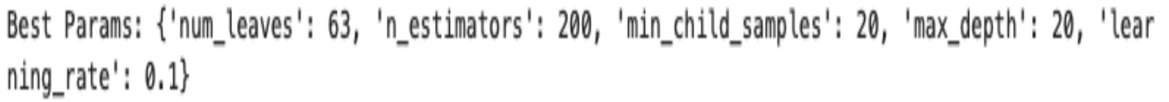
```
 LightGBM Performance
✅ Accuracy: 0.8221505805041347

Classification Report:
              precision    recall  f1-score   support

           0       0.82      1.00      0.90    952305
           1       0.66      0.04      0.08    211511

    accuracy                           0.82   1163816
   macro avg       0.74      0.52      0.49   1163816
weighted avg       0.79      0.82      0.75   1163816
```

Hyperparameter tuning

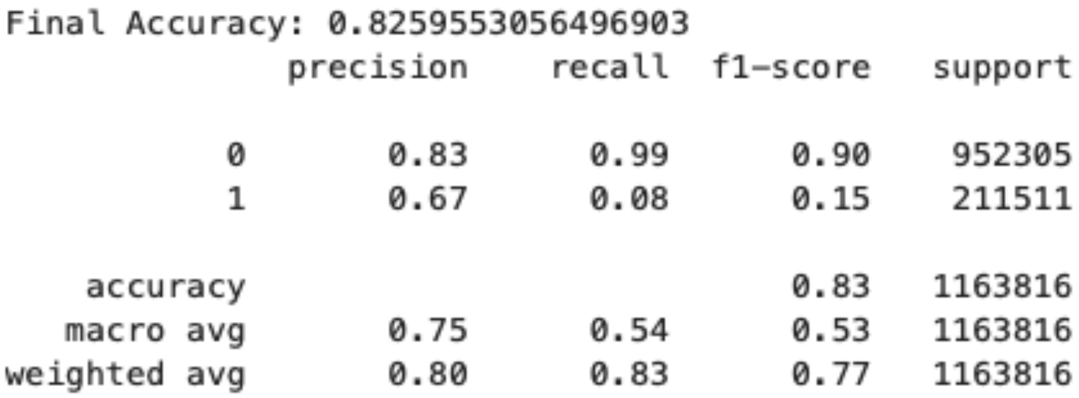
- Applied RandomizedSearchCV on Light GBM for faster and efficient parameter optimization; accuracy improved marginally from baseline, confirming model robustness
- Explored Random Forest hyperparameter tuning but discontinued due to excessive computational time
- Tuning parameters included n_estimators, max_depth, learning_rate, and min_samples_split; optimized for F1-score



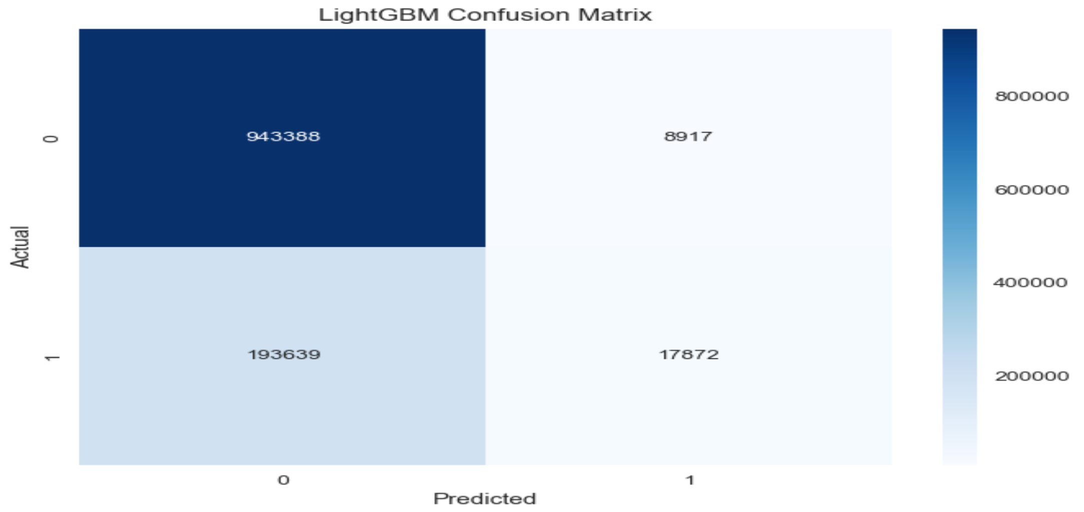
```
# Apply RandomizedSearchCV on tuned dataset
from sklearn.model_selection import RandomizedSearchCV
import lightgbm as lgb

param_dist = {
    'n_estimators': [100, 150, 200],
    'max_depth': [10, 20, -1],
    'num_leaves': [31, 50, 63],
    'learning_rate': [0.1, 0.05],
    'min_child_samples': [20, 50]
}

lgbm = lgb.LGBMClassifier(
    random_state=42,
    n_jobs=-1
)

rs = RandomizedSearchCV(
    estimator=lgbm,
    param_distributions=param_dist,
    n_iter=10,          # fast but still good
    scoring='accuracy',
    cv=2,               # reduces runtime
    verbose=2,
    n_jobs=-1
)
```



```
Best Params: {'num_leaves': 63, 'n_estimators': 200, 'min_child_samples': 20, 'max_depth': 20, 'lear
ning_rate': 0.1}
```



```
Final Accuracy: 0.8259553056496903
              precision    recall  f1-score   support

           0       0.83      0.99      0.90    952305
           1       0.67      0.08      0.15    211511

    accuracy                           0.83   1163816
   macro avg       0.75      0.54      0.53   1163816
weighted avg       0.80      0.83      0.77   1163816
```



LightGBM Confusion Matrix

Analysis

- Satisfied with 82-83% accuracy; Random Forest (64%) served its purpose for feature importance analysis (RQ1); Light GBM performance is operationally useful and aligns with aviation research standards
- 82-83% accuracy is good and operationally valuable; correctly predicts 8 out of 10 cases with balanced performance across both delayed and on-time classes
- Industry standard for delay prediction: 75-85% accuracy due to inherent complexity and unpredictable external factors

Next Steps

- Experiment with other faster and efficient models such as XGB, Logistic Regression
- Develop context-specific models per airline/airport (RQ2 analysis) which might typically achieve higher accuracy for targeted scenarios

Summary and conclusions

- Summary - Applied Random Forest and Light GBM models to predict fight delay from features such as distance, day of week, season, Taxi_out etc.
- Day of Week, Taxi_out, Distance and Month were top features influencing departure delay (RFE method)
  Results - Got 64% accuracy for Random Forest and 82% accuracy for Light GBM models
- With hyperparameter tuning performance moderately increased as Light GBM already handles complex datasets well
- What next?- Implement more models and conduct a stratified analysis of model performance by airline and airports

References

- Esmaeilzadeh, E. and Mokhtarimousavi, S. (2020) “Machine Learning Approach for Flight Departure Delay Prediction and Analysis,” Transportation Research Record: Journal of the Transportation Research Board, 2674(8), pp. 145–159. Available at: https://doi.org/10.1177/0361198120930014.
- Hatıpoğlu, I. and Tosun, Ö. (2024) “Predictive Modeling of Flight Delays at an Airport Using Machine Learning Methods,” Applied Sciences, 14(13), p. 5472. Available at: https://doi.org/10.3390/app14135472.
- Jacyna-Gołda, I. et al. (2025) “Optimizing Flight Delay Predictions with Scorecard Systems,” Applied Sciences, 15(11), p. 5918. Available at: https://doi.org/10.3390/app15115918.
-

References

- Rebollo, J.J. and Balakrishnan, H. (2014) “Characterization and prediction of air traffic delays,” Transportation Research Part C: Emerging Technologies, 44, pp. 231–241. Available at: https://doi.org/10.1016/J.TRC.2014.04.007.
- Reddy, R.T. et al. (2023) Flight Delay Prediction Using Machine Learning, 2023 IEEE 8th International Conference for Convergence in Technology (I2CT). IEEE. Available at: https://doi.org/10.1109/I2CT57861.2023.10126220.
- Wang, F. et al. (2022) “Flight delay forecasting and analysis of direct and indirect factors,” IET Intelligent Transport Systems, 16(7), pp. 890–907. Available at: https://doi.org/10.1049/itr2.12183.