A1909709 Shan Shan Bianca Tan

Big Data Analysis Project

Assignment 1: Part D (Report)

1 Problem Restatement and Summary

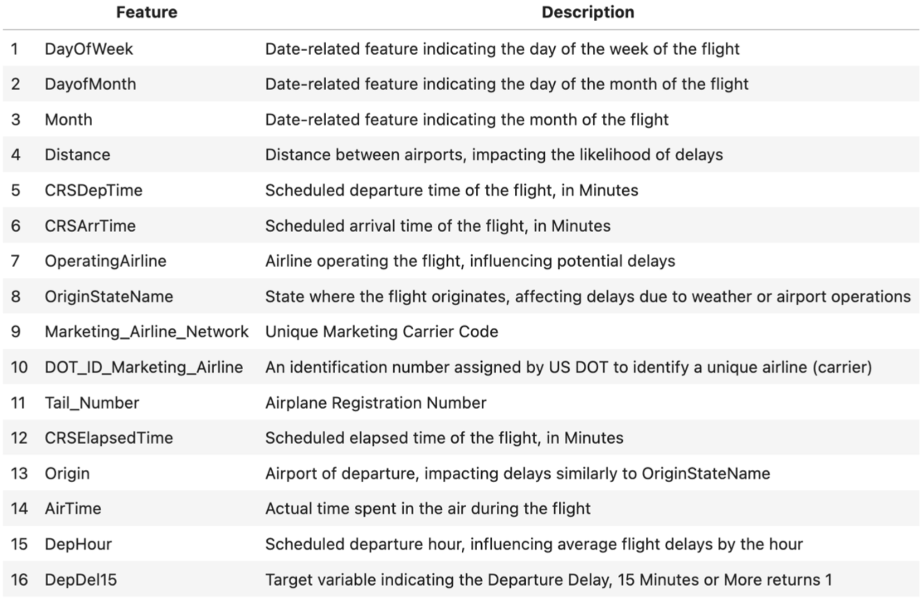
Problem Statement: Can we accurately predict whether a flight will experience a delay of more than 15 minutes with our dataset features?

Being able to predict flight status can benefit passengers, airlines and airports. It will help passengers make more informed decisions in their travel and reduce inconvenience. Airlines can allocate resources efficiently. Lastly, airports can manage space and staff effectively.

The original dataset covers US flights from 2018-2022, with 29,193,782 rows and 61 columns (*Flight status prediction*, 2022). To manage computational constraints, I focused on California, LAX airport, and flights from 2020-2021, reducing the data to 306,833 rows and 16 features shown in Figure 1.

The target variable is `*DepDel15*`, where 1 indicates a delay and 0 indicates no significant delay.

The input variables include `*DayOfWeek*`, `*DayofMonth*`, `*Month*`, `*Distance*`, `*CRSDepTime*`, `*CRSArrTime*`, `*OperatingAirline*`, `*CRSElapsedTime*`, `*AirTime*`, `*DepHour*`, `*Marketing_Airline_Network*`, `*DOT_ID_Marketing_Airline*`, `*OriginStateName*`, and `*Origin*`.



	Feature	Description
1	DayOfWeek	Date-related feature indicating the day of the week of the flight
2	DayofMonth	Date-related feature indicating the day of the month of the flight
3	Month	Date-related feature indicating the month of the flight
4	Distance	Distance between airports, impacting the likelihood of delays
5	CRSDepTime	Scheduled departure time of the flight, in Minutes
6	CRSArrTime	Scheduled arrival time of the flight, in Minutes
7	OperatingAirline	Airline operating the flight, influencing potential delays
8	OriginStateName	State where the flight originates, affecting delays due to weather or airport operations
9	Marketing_Airline_Network	Unique Marketing Carrier Code
10	DOT_ID_Marketing_Airline	An identification number assigned by US DOT to identify a unique airline (carrier)
11	Tail_Number	Airplane Registration Number
12	CRSElapsedTime	Scheduled elapsed time of the flight, in Minutes
13	Origin	Airport of departure, impacting delays similarly to OriginStateName
14	AirTime	Actual time spent in the air during the flight
15	DepHour	Scheduled departure hour, influencing average flight delays by the hour
16	DepDel15	Target variable indicating the Departure Delay, 15 Minutes or More returns 1

Figure 1 Summary of Variables

In the pre-processing steps, missing `*DepDel15*` values were imputed with 1 to reflect flight cancellations as delays, and missing `*AirTime*` values were filled with the median. StandardScaler was applied to numerical features for balanced scales (Upadhyay, 2021). To address class imbalance as seen in Figure 2, a combination of SMOTE and RandomUnderSampler (RUS) was used (Tamanna, 2023).

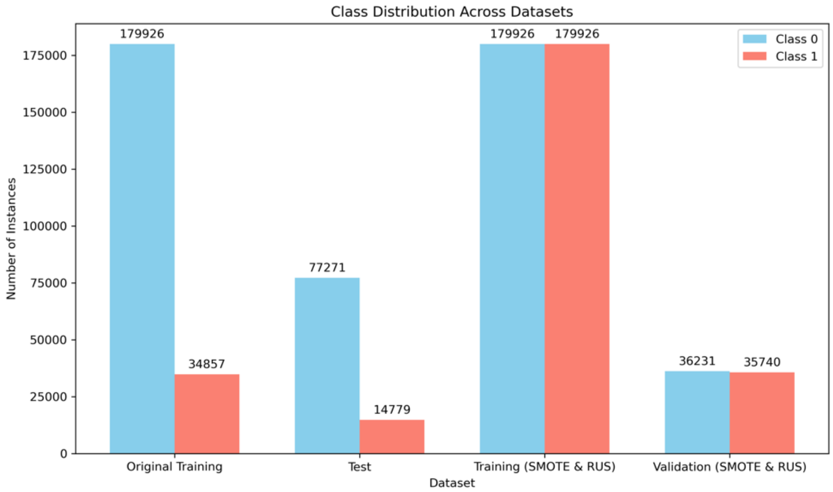


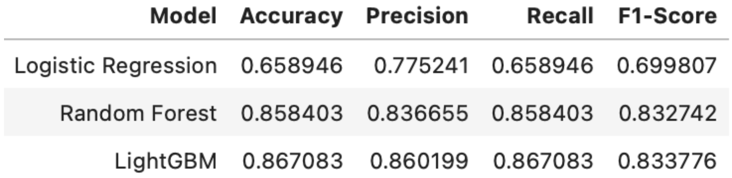
Figure 2 Class Distribution

2 Summary of Results

2.1 Model Selection and Performance

The models evaluated were Logistic Regression (LR), Random Forest (RF), and Light Gradient Boosting Model (LightGBM). LR was chosen for its simplicity and efficiency with large datasets. RF was selected for its ability to handle many features and complex interactions and for being less prone to overfitting (Sharma 2024). LightGBM was chosen for its efficiency, scalability, and suitability for imbalanced datasets (Anishnama 2023).

Training and validation results indicated that RF and LightGBM outperformed LR as illustrated in Figure 3 and 4.



Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.658946	0.775241	0.658946	0.699807
Random Forest	0.858403	0.836655	0.858403	0.832742
LightGBM	0.867083	0.860199	0.867083	0.833776

Figure 3 Table Summary of Metrics

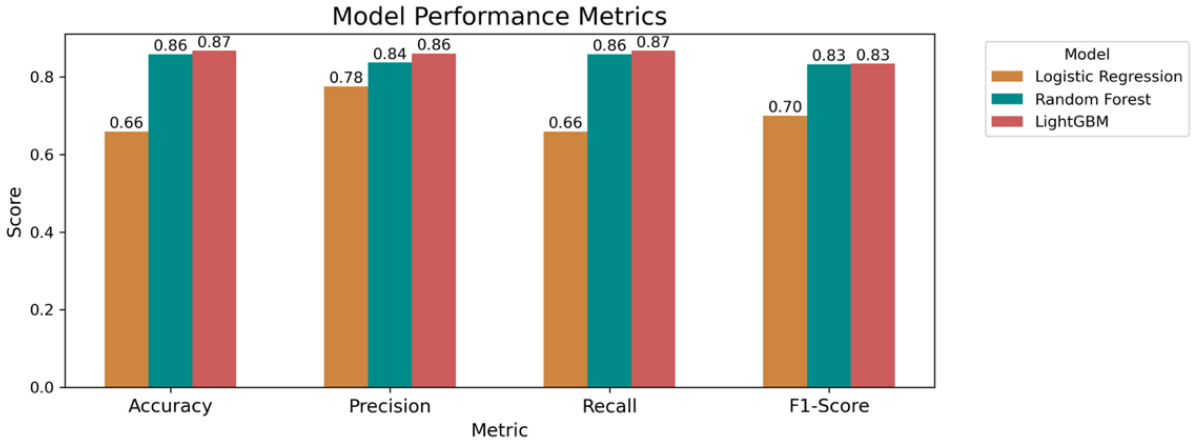


Figure 4 Bar Chart Summary of Metrics

2.2 Results Analysis

Hyperparameter tuning used RandomizedSearchCV, optimizing for the F1-score to balance class performance due to its efficiency over GridSearchCV (Satheesh 2021).

As shown in Table 1 and Figure 5, LightGBM outperformed RF in most metrics, with higher accuracy (0.91 validation, 0.87 test) compared to RF (0.74 validation, 0.72 test). LightGBM excelled in predicting non-delayed flights (class 0), showing superior precision and recall. For delayed flights (class 1), LightGBM achieved higher precision and recall on the validation set, though RF's metrics were lower. The F1-scores further indicate LightGBM's stronger overall performance.

However, LightGBM's performance on the test set for delayed flights (class 1) dropped significantly, with recall falling from 0.83 on the validation set to 0.23 on the test set, signaling overfitting. The F1-score for class 1 dropped from 0.90 to 0.36, highlighting the model's struggle to generalize. This indicates a need for further refinement to address overfitting.

Table 1 Summary of Optimized Models

	LightGBM		Random Forest	
Metric	**Validation**	**Test**	**Validation**	**Test**
Accuracy	0.91	0.87	0.74	0.72
Precision (0)	0.86	0.87	0.74	0.88
Recall (0)	0.99	0.99	0.75	0.77
F1-Score (0)	0.92	0.93	0.75	0.82
Precision (1)	0.99	0.82	0.75	0.28
Recall (1)	0.83	0.23	0.74	0.48
F1-Score (1)	0.90	0.36	0.74	0.35

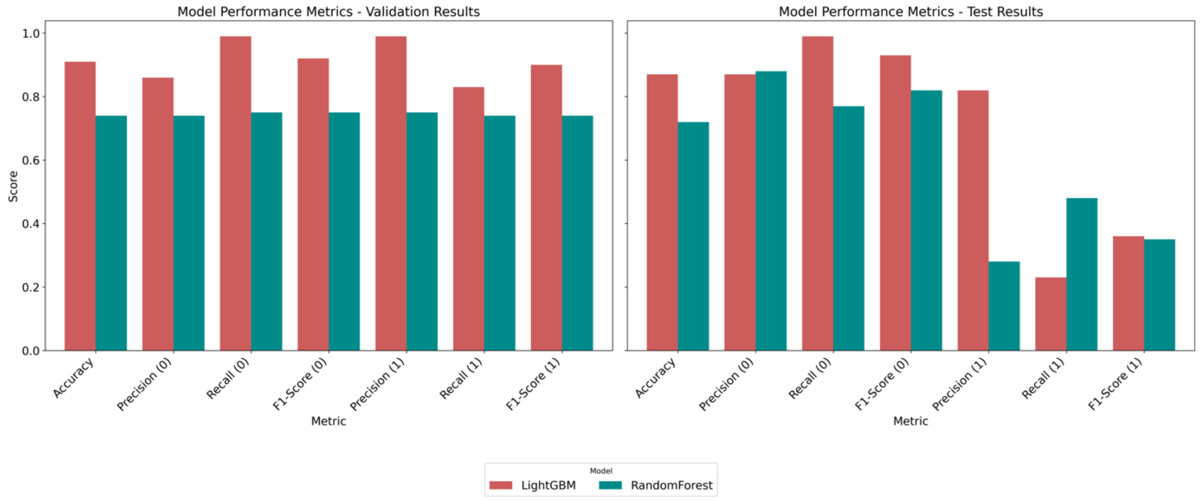


Figure 5 Summary of Optimized Models

3 Recommendations (Improvement of Situation)

While LightGBM shows strong accuracy for non-delayed flights, its performance on the test set indicates a need for further tuning to address overfitting and improve performance in the minority class. Specific actions to improve results include:

- **Enhance Model Tuning:** Refine LightGBM by focusing on recall for delayed flights (class 1) through hyperparameter adjustments, including learning rate, number of leaves, and boosting iterations. Employ strategies like Hyperband, a bandit-based algorithm, to optimize these hyperparameters by discarding less promising configurations early (Hong, 2024). This helps refine the hyperparameter space and reduce overfitting.

- **Explore Alternative Models:** Investigate other machine learning models or ensemble methods like XGBoost, CatBoost, or combining multiple algorithms to enhance prediction accuracy for the minority class and improve overall robustness.

- **Feature Engineering:** Incorporate features like weather conditions, historical delay patterns, or real-time air traffic data to better capture factors contributing to flight delays.

- **Data Augmentation:** Apply advanced techniques such as synthetic data generation or GANs (Thakran, 2023), to create more representative samples of delayed flights, mitigating class imbalance and improving model training.

These actions can enhance accuracy and reliability of flight delay predictions, benefiting passengers, airlines, and airports through better resource allocation and planning.

4 Conclusion & Future Work

The analysis shows that LightGBM generally outperforms Random Forest (RF) in accuracy and F1-scores. While LightGBM excels in predicting non-delayed flights, its performance for delayed flights (class 1) on the test set indicates potential overfitting. This suggests that LightGBM, while effective on training data, struggles with generalization to new data, especially in detecting delayed flights.

To address these issues, future research should focus on mitigating overfitting and enhancing model generalization. Improving model regularization and exploring advanced hyperparameter tuning methods, such as Hyperband, could help. Combining multiple models through ensemble techniques may also improve robustness. Additionally, incorporating new features like real-time data or environmental factors and applying data augmentation strategies could enhance model performance and balance the dataset. Future work should explore real-time data integration and operational deployment to continuously refine and validate predictions. Implementing these strategies can significantly improve the model’s accuracy and reliability in predicting flight delays, offering more actionable insights for stakeholders.

5 References

Flight status prediction (2022). https://www.kaggle.com/datasets/robikscube/flight-delay-dataset-20182022/data.

Sharma, A 2024, *Random Forest vs Decision Tree | Which Is Right for You?*, https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/.

Tamanna 2023, *Handling Imbalanced Datasets in Python: Methods and Procedures*, *Medium*, https://medium.com/@tam.tamanna18/handling-imbalanced-datasets-in-python-methods-and-procedures-7376f99794de.

Satheesh, V 2021, *Hyper Parameter Tuning (GridSearchCV vs RandomizedSearchCV)*, *Medium*, https://medium.com/analytics-vidhya/hyper-parameter-tuning-gridsearchcv-vs-randomizedsearchcv-499862e3ca5.

Upadhyay, A 2021, *StandardScaler and Normalization with code and graph*, *Medium*, https://medium.com/analytics-vidhya/standardscaler-and-normalization-with-code-and-graph-ba220025c054.

Hong, Z. (2024) 'Hyperparameter Optimization at Scale: Strategies for Large-Scale Experiments,' *Medium*, 13 January. https://medium.com/@zhonghong9998/hyperparameter-optimization-at-scale-strategies-for-large-scale-experiments-8e59e1525c9a.

Thakran, D. (2023) *Enhancing Datasets with GANs for Data Augmentation*. https://astconsulting.in/blog/2023/08/28/enhancing-datasets-gans/.

(Word count: 834)