# SkyFlow: AI-Powered Flight Delay Prediction for Optimized Airline Operations

Dipti Aswath | [LinkedIn](https://www.linkedin.com/in/dipti-aswath-60b9131) | [Email](mailto:dipti.aswath@gmail.com)

## Executive Summary

### Problem Statement:

Airlines and airports face significant operational challenges due to flight delays, which can be caused by a variety of factors including flight status, weather conditions, air traffic congestion, aircraft specifics, and inefficiencies in ground and passenger handling. The objective is to predict flight delays by developing a multi-class classification model that considers both departure and arrival delays, helping improve operational planning and customer satisfaction.

### Rationale:

Flight delays can have widespread consequences for airlines, from passenger dissatisfaction to operational disruptions. Developing a predictive model for flight delays not only addresses the core issue of minimizing delays but also enhances decision-making processes across various facets of airline operations.

#### Business Case 1: Enhancing Operational Efficiency

Predicting flight delays enables airlines to optimize their operations, routing, and resource management.

* **Route Optimization and Scheduling Adjustments**: Airlines can reroute flights to avoid congested airspace or adverse weather, minimizing delays. Predictions also allow real-time adjustments to schedules, gates, and crew to manage disruptions efficiently.
* **Resource Allocation**: By anticipating delays, airlines can proactively allocate ground crew, gates, and equipment, reducing the cascading effects on other flights.
* **Operational Resilience**: Dynamic rerouting and resource realignment minimize the operational impacts of weather or high-traffic delays, enhancing resilience in crisis situations.
* **Cost Management**: Avoiding delays lowers costs linked to operational disruptions, improving resource utilization and overall profitability.

#### Business Case 2: Improving Customer Experience

Accurate delay predictions lead to better customer service and proactive communication, enhancing the passenger experience.

* **Proactive Passenger Communication**: Accurate predictions allow airlines to update passengers promptly, manage expectations, and offer rebooking or compensation options.
* **Improved Customer Service**: Delay forecasts support better service recovery, leading to a smoother passenger experience and increased loyalty.
* **Competitive Advantage**: Effective rerouting and communication give airlines an edge in maintaining on-time performance and customer satisfaction.

By addressing these areas, airlines can significantly improve operational efficiency, enhance passenger experience with better customer satisfaction scores, and better manage resources and disruptions. Predictive modeling for flight delays is not just about minimizing delays but also about fostering a more responsive and resilient airline operation.

#### Example Usage: An AI system that predicts flight delays could:

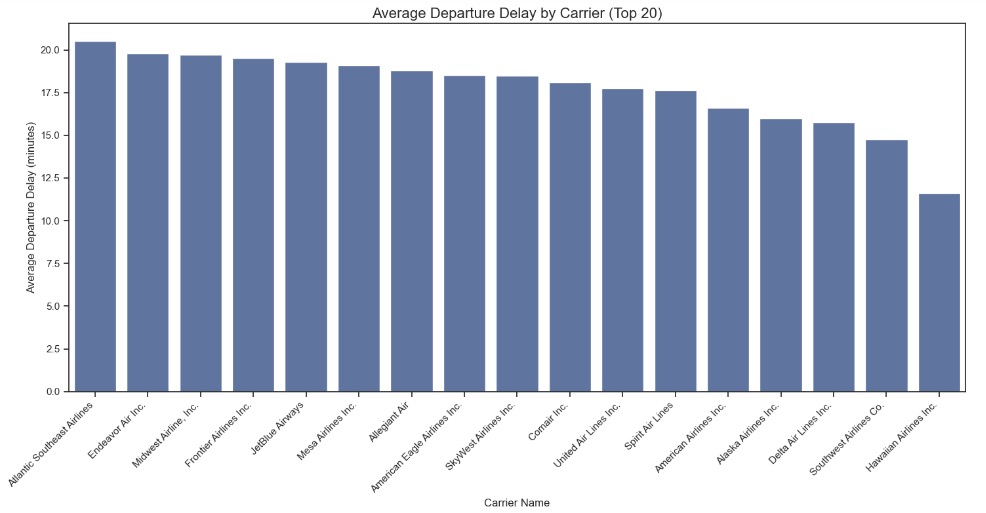
1. Suggest alternate flight paths that are less likely to experience delays.
2. Provide passengers with timely updates and rebooking options.
3. Dynamically adjust flight schedules to manage disruptions effectively.
4. Allocate resources efficiently to minimize the impact on subsequent flights.

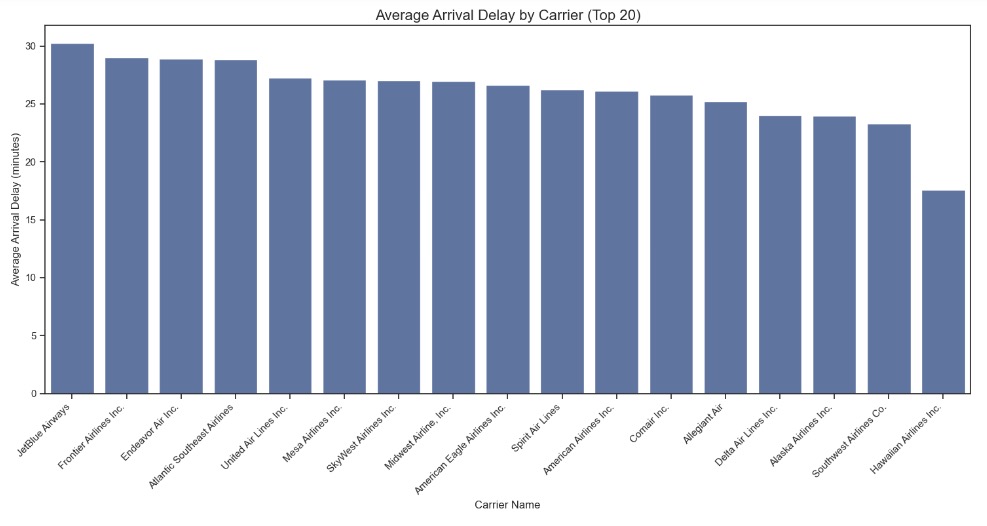
### Research Question:

How can a multi-class classification model be developed to accurately predict flight delays by assessing multiple factors, including departure and arrival delays, using data related to flight status, weather conditions, air traffic, aircraft specifics, and ground operations?

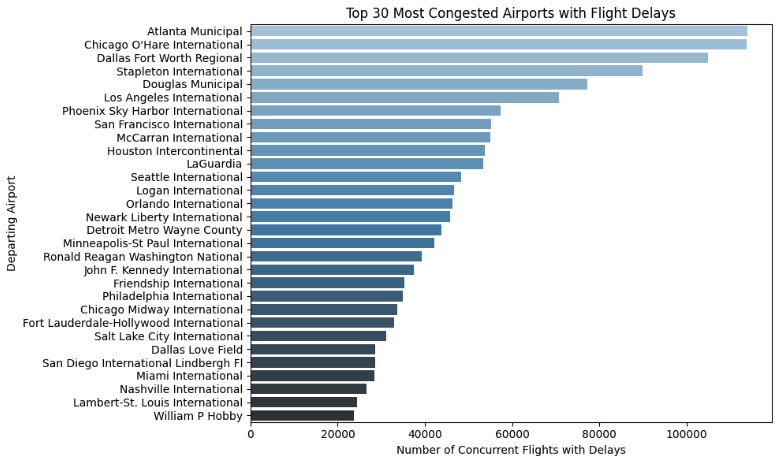
### Key Findings from Exploratory Data Analysis:

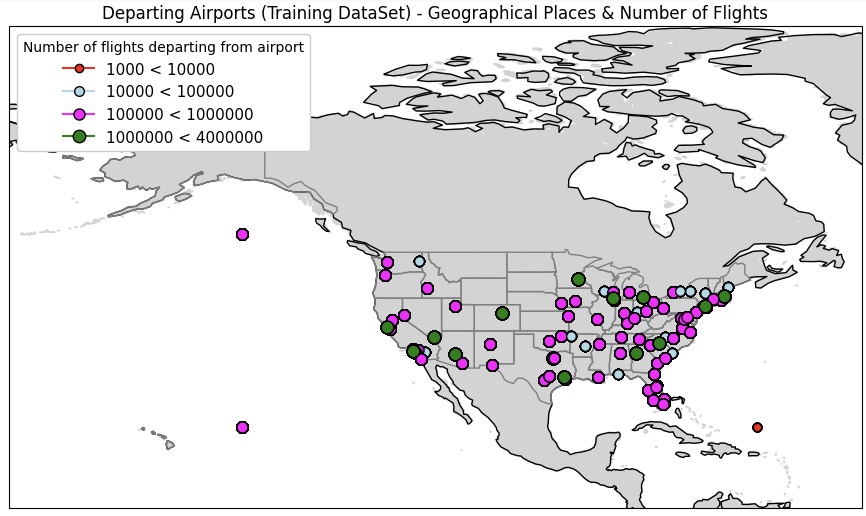
**Highest Departure and Arrival delays by Carriers (2019):** Identifying the carriers with the highest delays directly relates to **improved customer experience and financial impact**. By pinpointing these carriers, airlines can better manage customer expectations, offer targeted support, and address issues that could lead to costly disruptions and compensation claims.



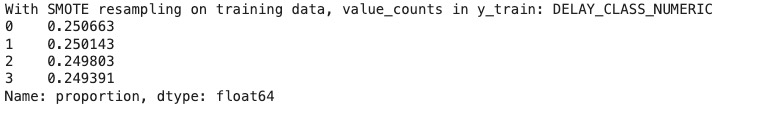


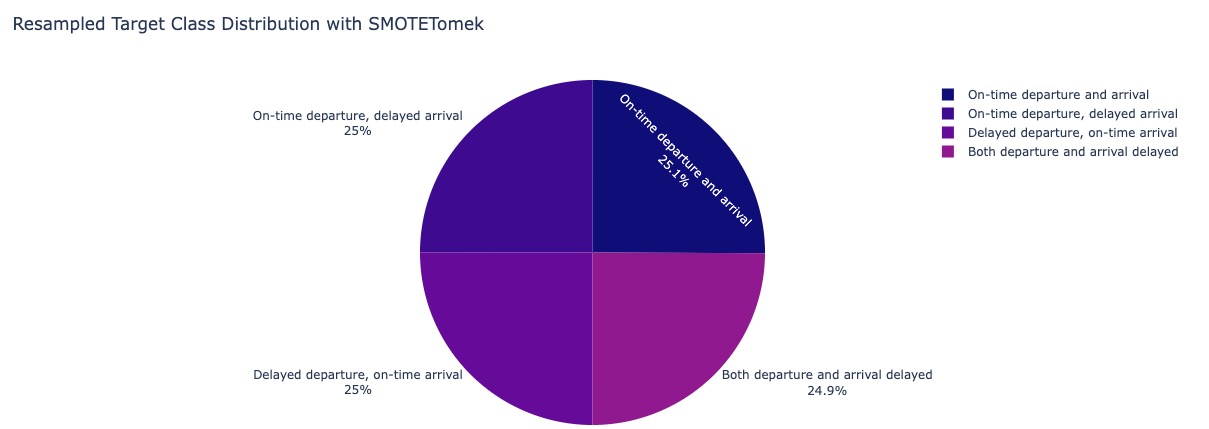
**Top 30 Congested Airports with Flight Delays (2019):** This finding supports **enhanced operational efficiency and operational resilience**. By focusing on the most congested airports, airlines can optimize resource allocation and improve scheduling to alleviate delays at these critical points, leading to smoother operations and better crisis management.





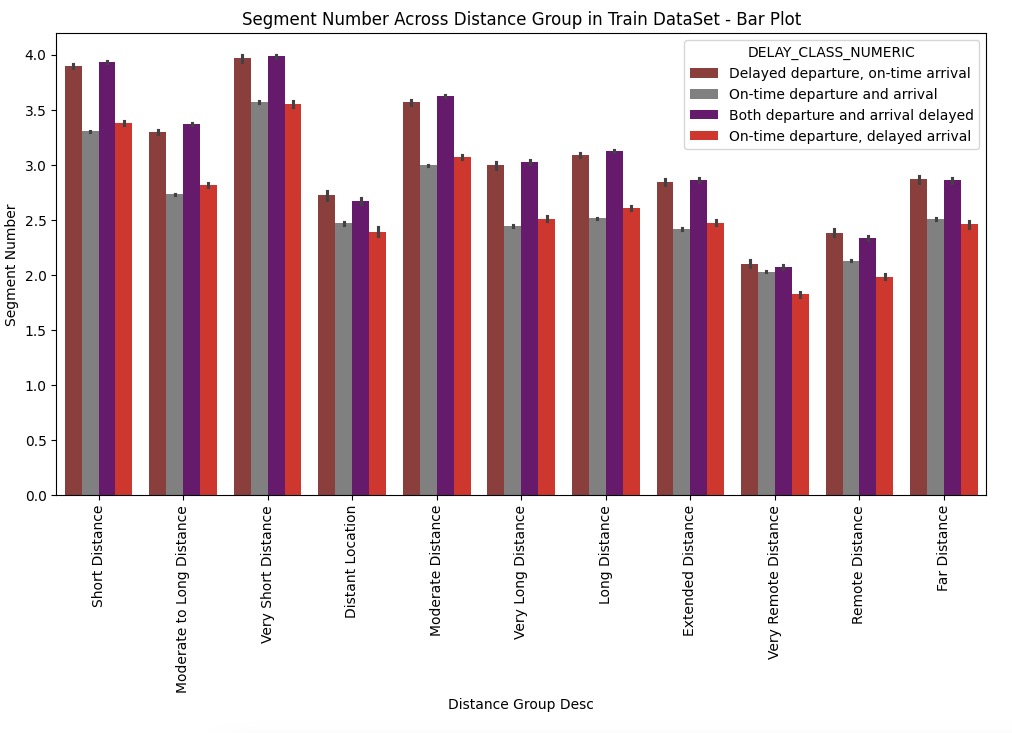
**SMOTE Resampling on Training Data:** Demonstrates the importance of **data-driven decision making**. By improving model performance through resampling, airlines can make more accurate predictions about delays, leading to better strategic planning and performance monitoring.





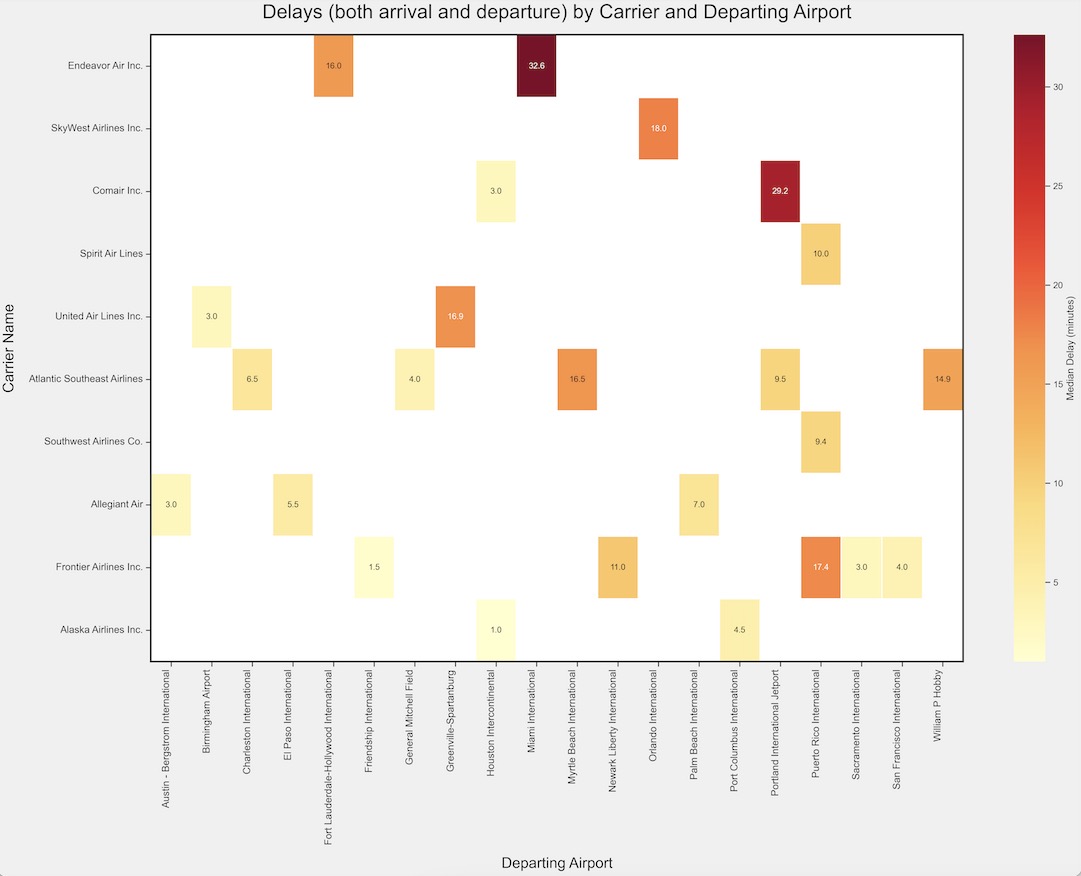
**Delay Trends Across Distance Groups and Flight Segments (2019):** This finding helps provide valuable insights into how aircraft operational schedules and the number of daily flights contributed to 2019 delays, effectively addressing **operational efficiency and contingency planning**. Understanding how delay patterns vary with flight distance and segment numbers helps airlines plan better turnaround times and manage operational schedules more effectively to prevent delays.

* **Segment Number Decreases with Distance**: As flight distance increases, the number of segments (flights) decreases. Aircraft flying longer routes complete fewer flights in a day due to time constraints.
* **Delays Correlate with Higher Segment Numbers**: Flights scheduled for more segments in a day are more prone to delays, regardless of distance. These delays are likely due to operational factors, such as shorter turnaround times, leading to delayed departures and arrivals.



**Median Departure and Arrival Delays per Carrier (2019):** Identified the top 20 carriers with the highest median delays. For each carrier, the top 20 airports with the most significant contribution to delays were also identified. By examining median delays, airlines can gain insights into typical delay experiences and ensure compliance with regulations. Focusing on specific carriers and airports with high delays can enhance **overall safety and customer satisfaction**.

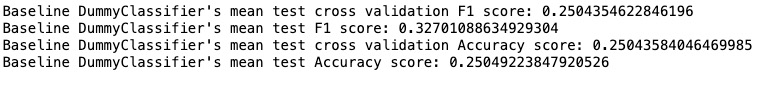
* **Comprehensive Delay Analysis:** By considering both departure and arrival delays, we provide a more holistic view of 2019 airline performance and airport efficiency. Endeavor Air Inc shows a highest delay at Miami International Airport. Comair Inc follows with the next highest delay at Portland International Airport.
* **Focus on median delays**: The use of median delays helped identify typical delay experiences, filtering out the effect of extreme delays that skewed averages.
* **Unique Operational Factors:** The variation in delay trends suggests that delays may be influenced by distinct factors specific to each carrier and airport, rather than being caused by common issues across multiple locations. For instance, both Endeavor Air Inc and Comair Inc experienced higher-than-usual precipitation at the airports on their flight day, which could have contributed to their delays.



### A graph of a number of aircraft carrier names Description automatically generated

### Model Evaluation Summary and Performance Metrics:

#### Baseline Model with Dummy Classifier:



#### Multinomial Logistic Regression Classifier:

Performance Summary:

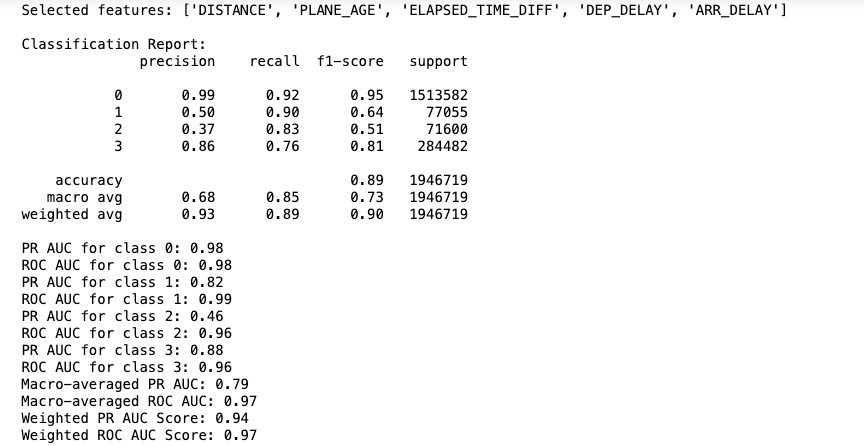
* Class 0: Excellent precision (0.99), high recall (0.92), and top F1-score (0.95). The model performs very well on this majority class.
* Classes 1 & 2: Low precision and F1-scores (0.50 and 0.64 for Class 1; 0.37 and 0.59 for Class 2) indicate many false positives. These classes have high recall, suggesting the model detects them but struggles with classification accuracy.
* Class 3: Balanced performance with a good F1-score (0.81), showing moderate effectiveness.

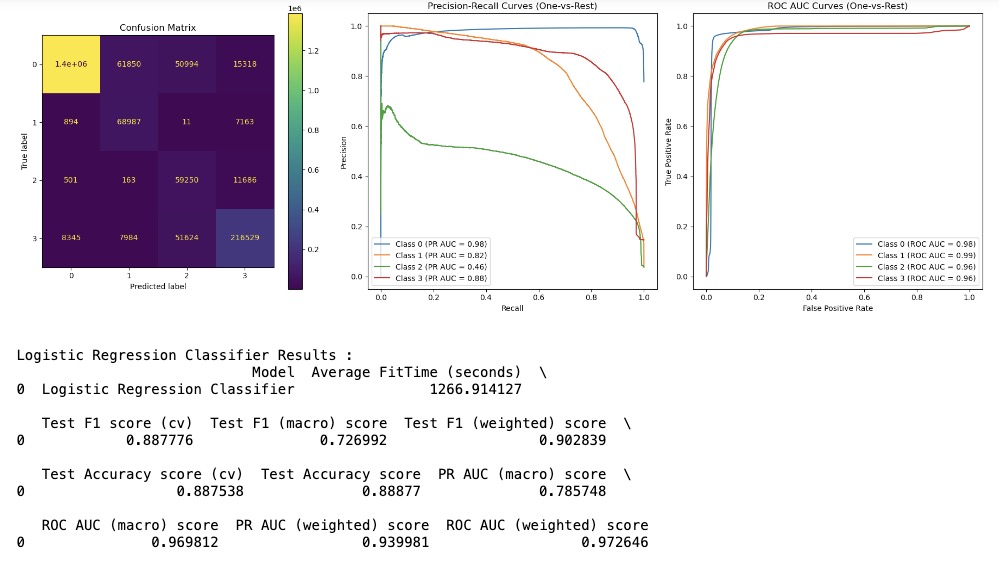
Bias vs. Variance Analysis Summary:

* High Variance: The model exhibits high variance by overfitting to the majority class (Class 0) and performing poorly on minority classes (Classes 1 and 2). This discrepancy indicates overfitting, as the model captures the majority class well but generalizes poorly to others.
* High Bias: The model exhibits high bias as well, as the model’s overall performance metrics (macro-averaged) are lower, reflecting that the model is too simplistic to handle the complexities of minority classes.

Summary:

* This model exhibits high variance, as evidenced by its strong performance on the majority class but weaker performance on the minority classes. This indicates overfitting to the majority class. It also shows signs of high bias, with poor performance on minority classes. To address this, we will next evaluate ensemble bagging methods, such as Random Forest, using a Decision Tree as a base model to potentially improve generalization.



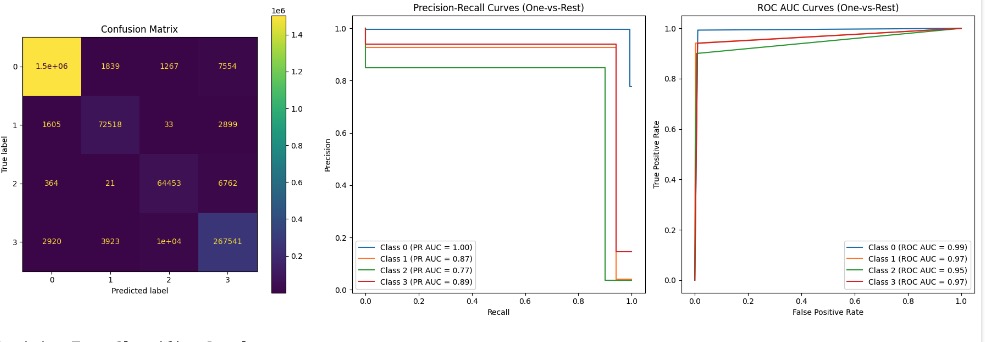
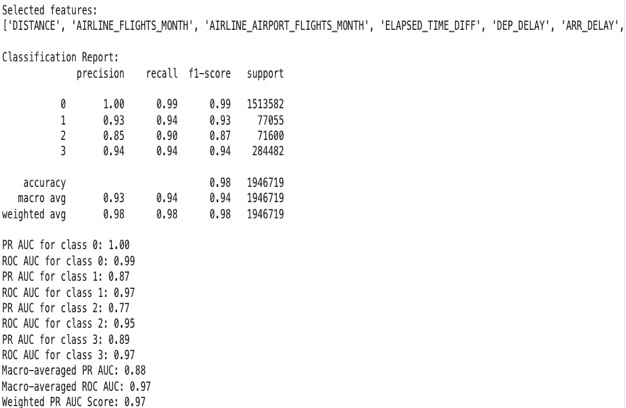


#### Decision Tree Classifier

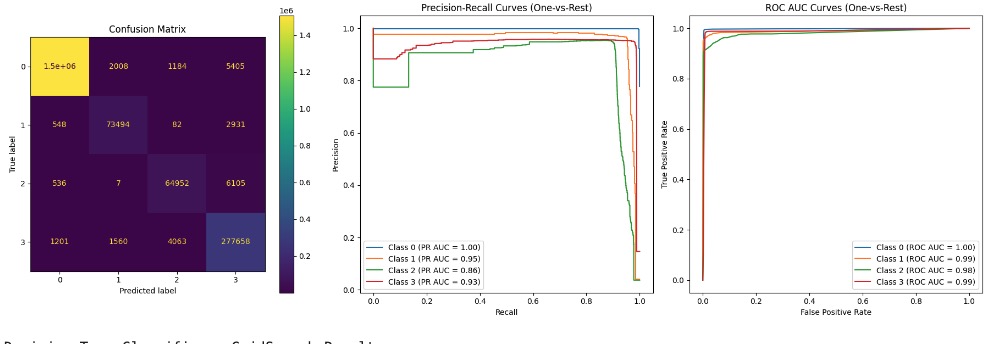
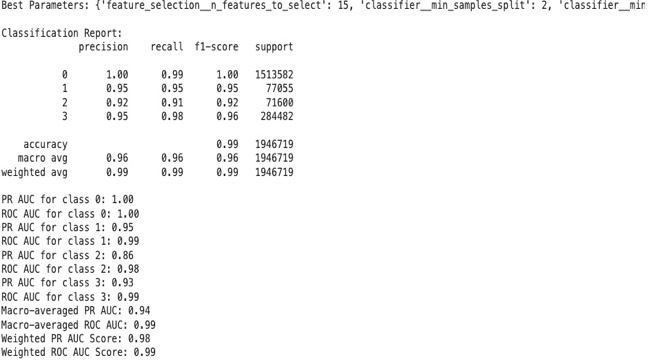
Two distinct decision tree models were developed for comparison: the original decision tree and the hyperparameter-tuned decision tree.

|  |  |  |
| --- | --- | --- |
|  | **Original Decision Tree** | **Hyperparameter-Tuned Decision Tree** |
| Performance Analysis Comparison | - Accuracy: 98%  - Class 0: Exceptional precision (1.00) and recall (0.99)  - Class 1: High precision (0.93), good recall (0.94)  - Class 2: Good precision (0.85) and recall (0.90)  - Class 3: Strong precision (0.94) and recall (0.94)  - PR AUC: 0.88 (macro), 0.97(weighted) - ROC AUC: 0.97 (macro), 0.98 (weighted) | - Accuracy: 99%  - Class 0: Near-perfect precision (1.00) and recall (0.99)  - Class 1: Improved precision (0.95), recall (0.95)  - Class 2: Significant improvement in precision (0.92) and recall (0.91)  - Class 3: Higher precision (0.95) and recall (0.98)  - PR AUC: 0.94 (macro), 0.98 (weighted)  - ROC AUC: 0.99 (macro), 0.99 (weighted) |
| Bias vs. Variance Comparison | - Low Bias: Captures patterns well with high performance across classes  - Moderate Variance: Shows signs of overfitting, especially in minority classes (lower precision for class 2) | - Low Bias: Captures complex patterns well across classes  - Reduced Variance: Tuning helped balance performance between classes, especially for class 2 and class 3 |
| Summary | - Features such as 'DISTANCE' and 'DEP\_DELAY' play a significant role in predictions - Good performance across majority and minority classes, but some overfitting on training data is noted.  - Ideal for quick insights with strong accuracy, but could benefit from further refinement | - Tuned parameters improved performance, especially for minority classes like 2 and 3 - More balanced across all classes with enhanced generalization  - Features like 'ELAPSED\_TIME\_DIFF' and 'DEP\_PART\_OF\_DAY' were instrumental in improving model performance - Tuned model shows high performance with low bias and high variance. To improve robustness and generalization, exploring ensemble methods like Random Forest is recommended. Random Forest can reduce overfitting and enhance generalization by averaging multiple decision trees. This approach will be tested next to further validate and fine-tune this model's performance. |

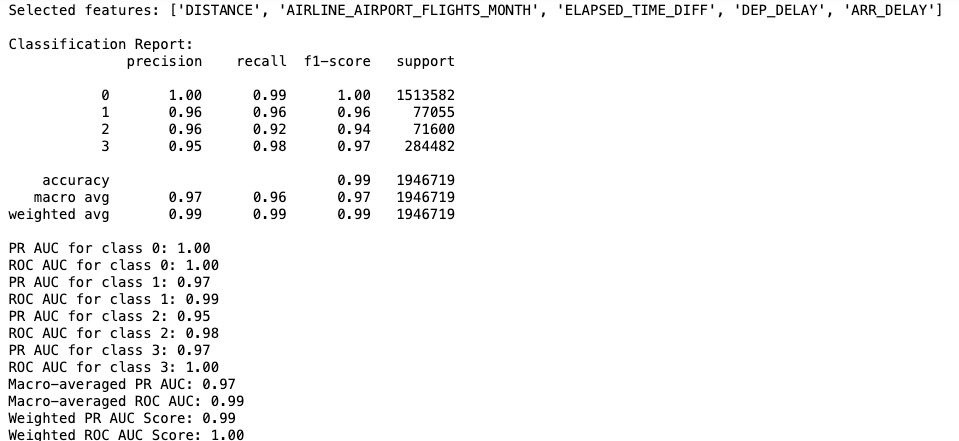
**Original Decision Tree Metrics:**

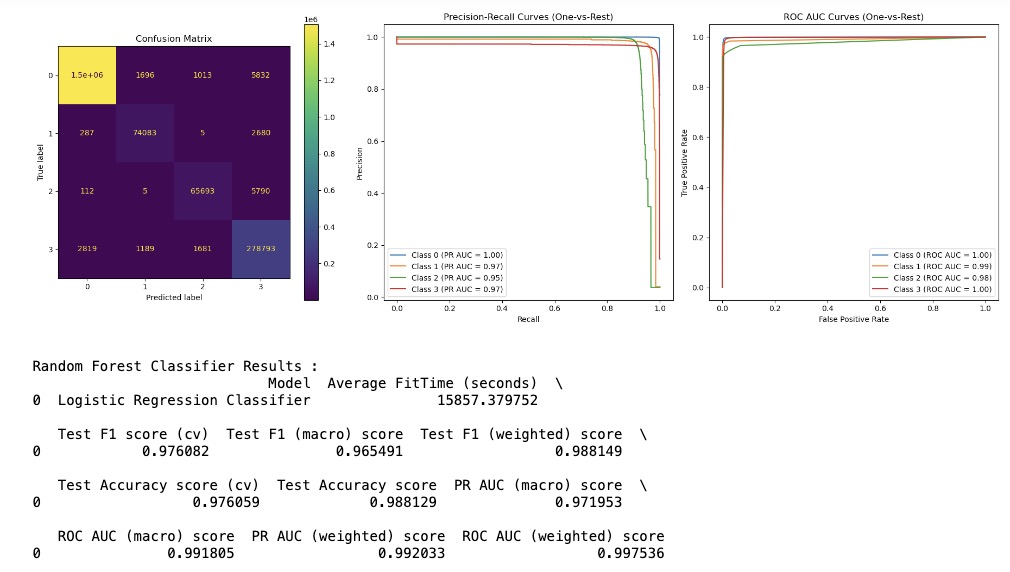


**Hyperparameter-Tuned Decision Tree Metrics:**

****

**Random Forest Classifier:**





### Summary of Model Evaluations:

**Model Ranking**: Based on performance metrics, the models can be ranked as a) Random Forest, b) Decision Tree, c) Logistic Regression, with Random Forest being a superior model for predicting flight delays, particularly in handling class imbalance and maintaing high performance across all classes. The Decision Tree classifier also handles class imbalance much better than Logistic Regression and its performance is more consistent across classes compared to Logistic Regression.

**Simplicity vs. Performance**: Decision Tree performs nearly as well as Random Forest, despite being a simpler model. Its significantly better than Logistic Regression. This suggests that the decision boundaries in the data are relatively clear and can be captured by a single tree.

**Interpretability**: Decision Trees offer better interpretability compared to Random Forest, while maintaining high performance. This could be valuable for understanding the key decision points in flight delay prediction.

**Robustness**: The high performance of the Decision Tree suggests that the important patterns in the data are strong and consistent

**Feature Importance**: Analyzing the structure of the Decision Tree could provide clear insights into the most critical factors for predicting flight delays.

**Deployment**: Given its high performance and interpretability, the Decision Tree could be an excellent choice for deployment, especially if model explanation is important.

## Data Sources:

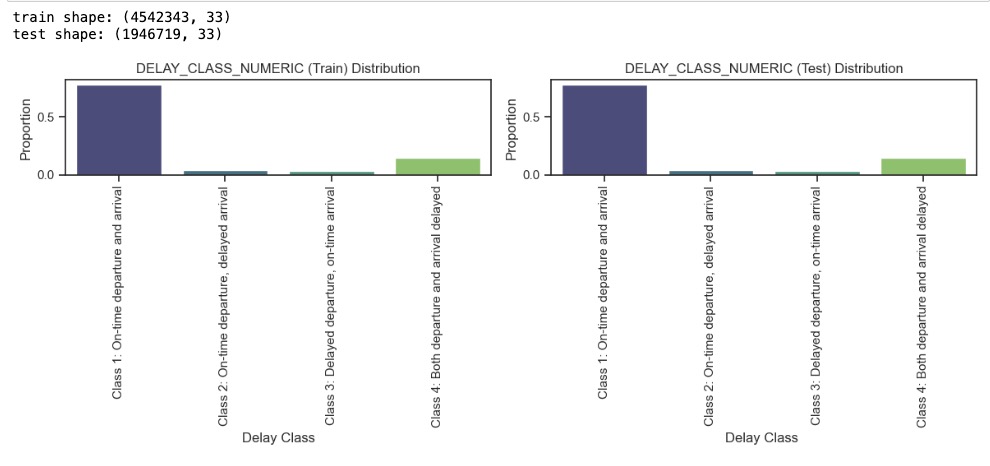
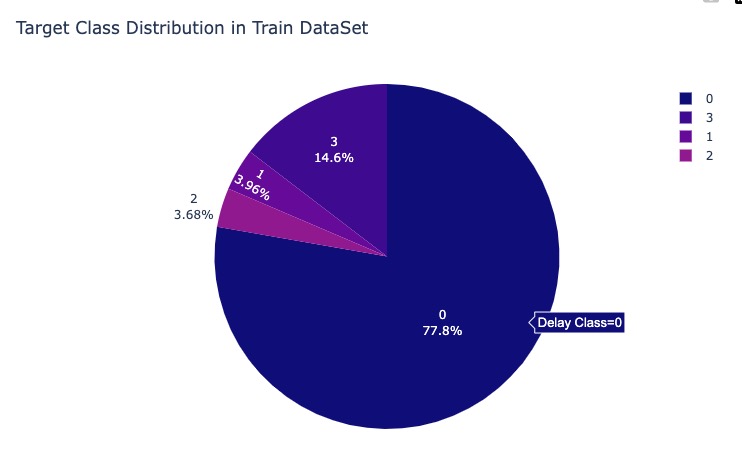
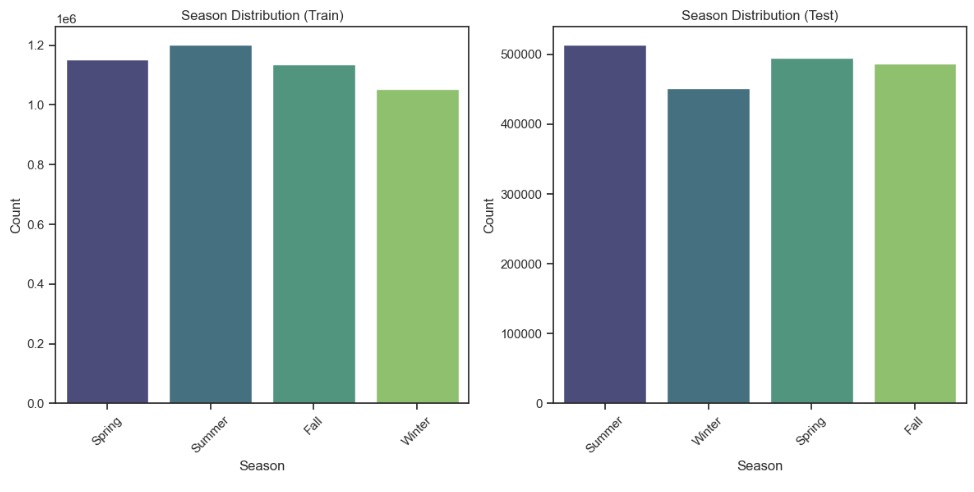
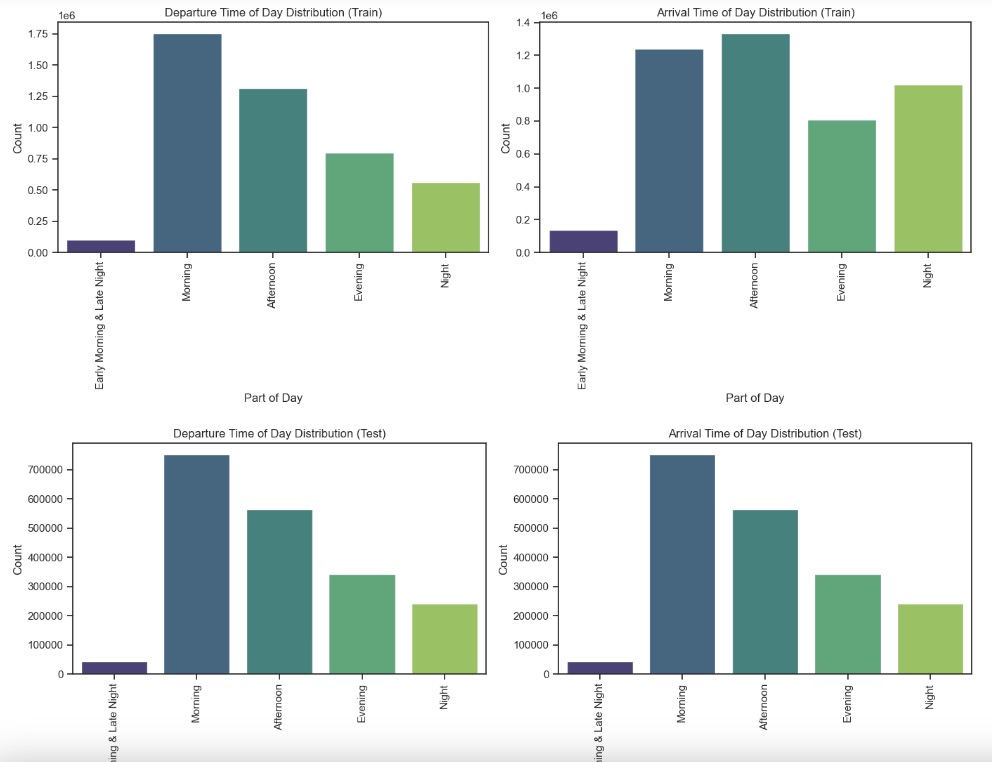
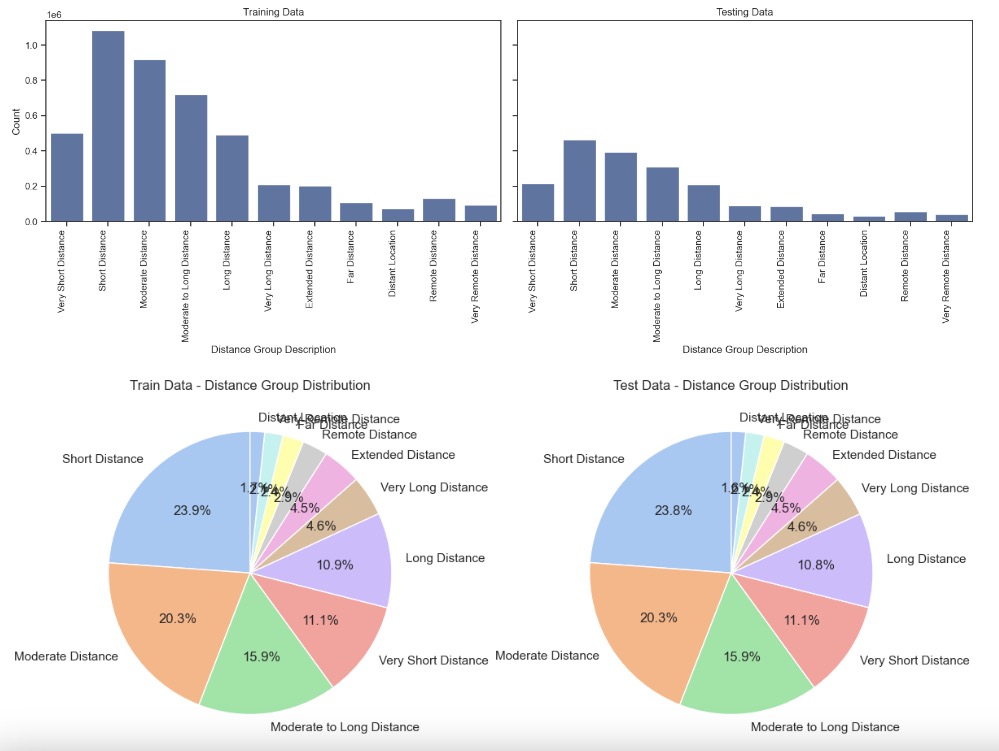
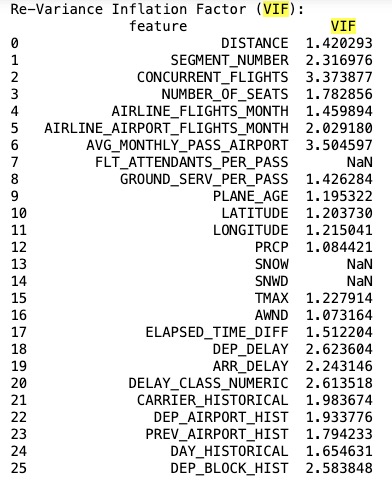
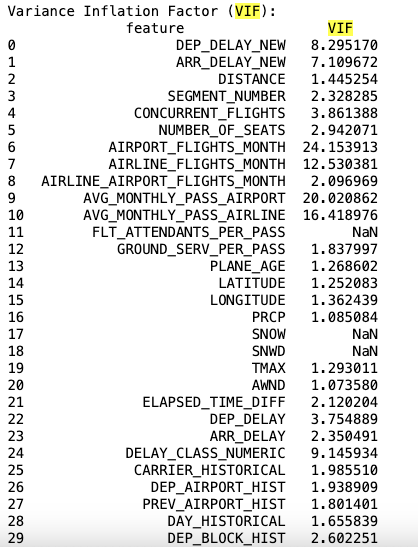
Kaggle Dataset from [here](https://www.kaggle.com/datasets/threnjen/2019-airline-delays-and-cancellations/data), that is comprised of multiple csv's listed below.

* Air Carrier Summary
* Aircraft Inventory
* Air Carrier employee support (Ground Crew, Flight Attendants)
* Flight On Time Reporting Status with Air Carrier info for 2019-2020
* Airport Weather
* Airport and Carrier look-up codes

## Methodology Used:

**Data Preparation:** Involved cleaning and merging multiple raw CSV files to create a unified data-set with ~4M entries (for training) and ~2M entries (for testing) with 34 predictor variables and 1 target variable. Raw data-set description is [here](https://github.com/diptiaswath/airlineFlightDelayPrediction/blob/main/raw_data/raw_data_documentation.txt).

**Feature Engineering:**

* Delay Categories: Classified delays into four distinct categories for more granular analysis of flight performance:
* Class0: On-time Departure and Arrival - Flights that depart and arrive within their scheduled times.
* Class1: On-time Departure, Delayed Arrival - Flights that experience delays during arrival, but depart on time.
* Class2: Delayed Departure, On-time Arrival - Flights that experience delays during departure but still arrive on time.
* Class3: Delayed Departure and Arrival - Flights that experience delays both in departure and arrival times.
* 
* Aggregation Features: Developed historical delay averages, to identify patterns and trends in airline operations.
* CARRIER\_HISTORICAL = captures the historical average delay rate of each carrier per month
* DEP\_AIRPORT\_HIST = captures historical average delay rates for flights departing from specific airports per month
* PREV\_AIRPORT\_HIST = captures historical average delay rate for the airport from which the aircraft arrived before the current departure
* DAY\_HISTORICAL = captures historical average delays associated with each day of the week, adjusted monthly
* DEP\_BLOCK\_HIST = captures historical average delay rate for different departure time blocks, aggregated by month
* Time-Based Features: Extracted seasonal information from the month and categorized parts of the day using departure and arrival time blocks to enhance temporal analysis of flight data.
* 
* 
* Distance-Based Features: Mapped distance groups to descriptive labels, providing clearer insights into flight range categories for more intuitive analysis.
* 
* Delay-Based Features: Created new features by combining actual departure and arrival times with scheduled times, generating detailed delay metrics to enhance analysis of flight performance and punctuality.
* ELAPSED\_TIME\_DIFF, DEP\_DELAY, ARR\_DELAY
* Employee Statistics Features: Developed features to analyze staffing and resourcing in airline and carrier operations, providing insights into workforce allocation, scheduling efficiency, and resource optimization.
* FLT\_ATTENDANTS\_PER\_PASS, PASSENGER\_HANDLING
* Removed highly correlated features with VIF
* 

**Data Pre-Processing:** Missing values and outliers detected were removed. SMOTETomek was applied to just the training data-set. This combined SMOTE's oversampling of the minority classes (classes 1,2 and 3) and Tomek links' under-sampling. Categorical features were also target encoded and Numerical features were scaled.

**Modeling and Evaluation:** Classification algorithms used were Decision Trees, Random Forest, and multi-nomial Logistic Regression, with evaluation metrics: F1 Score, PR AUC, ROC AUC and Accuracy scores. Feature Selection, specifically Recursive Feature Elimination (RFE) was used to select features from among the 34 predictor variables for Decision Treee and Logistic Regression Classifier.

## Project Structure

**Data:**

* [Engineered Features Documentation](https://github.com/diptiaswath/airlineFlightDelayPrediction/blob/main/combined_data/dataset_documentation.txt)
* Merged Datasets: [Train](https://github.com/diptiaswath/airlineFlightDelayPrediction/blob/main/combined_data/train.pkl) | [Test](https://github.com/diptiaswath/airlineFlightDelayPrediction/blob/main/combined_data/test.pkl)
* [Raw Data](https://github.com/diptiaswath/airlineFlightDelayPrediction/tree/main/raw_data)
* [Raw Data Documentation](https://github.com/diptiaswath/airlineFlightDelayPrediction/blob/main/raw_data/raw_data_documentation.txt)

**Analysis and Visualization:**

* [AutoViz Plots](https://github.com/diptiaswath/airlineFlightDelayPrediction/tree/main/plots) (Credit: [AutoViML/AutoViz](https://github.com/AutoViML/AutoViz))
* [README Images](https://github.com/diptiaswath/airlineFlightDelayPrediction/tree/main/images)

**Notebooks:**

* [Data Preparation and Feature Engineering](https://github.com/diptiaswath/airlineFlightDelayPrediction/blob/main/notebooks/flight-delays-data-prep-and-eda_v1.ipynb)
* [Additional Data Exploration](https://github.com/diptiaswath/airlineFlightDelayPrediction/blob/main/notebooks/flight-delays-data-exploration_v1.ipynb)
* [Data Pre-processing, Modeling, and Evaluation](https://github.com/diptiaswath/airlineFlightDelayPrediction/blob/main/notebooks/flight-delays-data-preproc-and-modeling_v1.ipynb)
* [Utility Functions](https://github.com/diptiaswath/airlineFlightDelayPrediction/blob/main/notebooks/utils/common_functions.ipynb)

**Git Large File Storage(LFS):**

This project uses Git Large File Storage (LFS) to handle large files efficiently. Git LFS replaces large files with text pointers inside Git, while storing the file contents on a remote server.

#### To work with this repository:

* Ensure you have Git LFS installed. If not, install it from [git-lfs.com](https://git-lfs.com).
* After cloning the repository, run: git lfs install git lfs pull
* When adding new large files, track them with: git lfs track "path/to/large/file"
* Commit and push as usual. Git LFS will handle the large files automatically. For more information on Git LFS, refer to the [official documentation](https://git-lfs.com/).

## Project Infrastructure

This project utilized Google Colab Pro to handle computationally intensive notebook operations for data exploration and modeling. Key components include:

**Notebooks:**

* Data exploration and modeling results from Colab Pro are captured in notebooks available in this GitHub repository.
* Direct links to key external notebooks for results: [Exploration Notebook](https://drive.google.com/file/d/136lYzQDpJ9rODL6nGHwUe3S37fY8L1c_/view?usp=drive_link), [Modeling Notebook](https://drive.google.com/file/d/1wR0uXhx9T_DXFRhNy-dKAz7XZkCIQQdy/view?usp=drive_link)

**AutoViz Visualizations:**

* Comprehensive AutoViz plots generated during data exploration are externally stored [here](https://drive.google.com/drive/folders/1N_Drv8Gvx0ANEk3fiaMAguF1JY8ptAd3?usp=drive_link) due to size constraints on GitHub.

**Decision Tree Artifacts:**

* Decision tree structures and rule sets are available in two locations:
  + Externally: View [here](https://drive.google.com/drive/folders/1qXDYyuo2lqJBwFTBoI7KCV45SZC-w163?usp=drive_link)
  + Locally: In the [images](https://github.com/diptiaswath/airlineFlightDelayPrediction/tree/main/images) folder of this repository

## Next Steps:

* Use Dimensionality Reduction and Clustering to reduce dimensions, and cluster features together to reduce the count of 34 predictors. Relying on Feature Selection techniques alone, takes a while to train any of the classification models.
* Investigate features and decision rules contributing the most to predicting flight delays with both the Decision Tree and the Random Forest classifier.
* Explore ensemble methods that could combine Decision Tree with other models to potentially improve performance further.
* Use StreamLit and Fast API to serve flight prediction delays via an application interface