Name: Suraj Nair

UID: 2023800066

TOPIC: AIRLINE DELAY ANALYSIS

Sardar Patel Institute of Technology, Andheri, India; Department: CSE

**Python For Data Science Project**

# Abstract:

This project tackles the increasing problem of flight delays, which significantly impact airline operations, passenger satisfaction, and overall travel efficiency. The main objective was to build a machine learning model capable of predicting whether a flight would be delayed based on a variety of historical and delay-related features. The dataset used was a publicly available flight delay dataset containing information such as scheduled departure times, arrival times, carrier codes, weather-related details, and previous delay history.

The project began with extensive data preprocessing, which included handling missing values, encoding categorical variables, removing irrelevant or redundant features, and standardizing numerical attributes. Following this, feature selection techniques were applied to reduce dimensionality and retain only the most informative variables. These techniques included Random Forest feature importance and Chi-Square statistical tests, which helped identify the key factors contributing to delays, such as scheduled departure time, airline carrier, and day of the week.

Multiple machine learning models were experimented with, including Random Forest, and Logistic Regression. Each model was evaluated using appropriate classification metrics such as accuracy, precision, recall, and F1-score. Among these, Logistic Regression was chosen as the final model due to its relatively simple structure, interpretability, and consistent performance on validation data.

To further enhance performance and reduce computational complexity, Principal Component Analysis (PCA) was explored as a dimensionality reduction technique. Although PCA did not significantly boost accuracy, it helped visualize the variance explained by different components and better understand the data structure.

Model evaluation was supported by visual tools such as learning curves, confusion matrices, and ROC curves, which helped diagnose issues like overfitting and underfitting. The learning curves revealed that while Random Forest achieved near-perfect training accuracy, it showed signs of overfitting, whereas Logistic Regression offered more generalizable performance with a smaller training-validation gap.

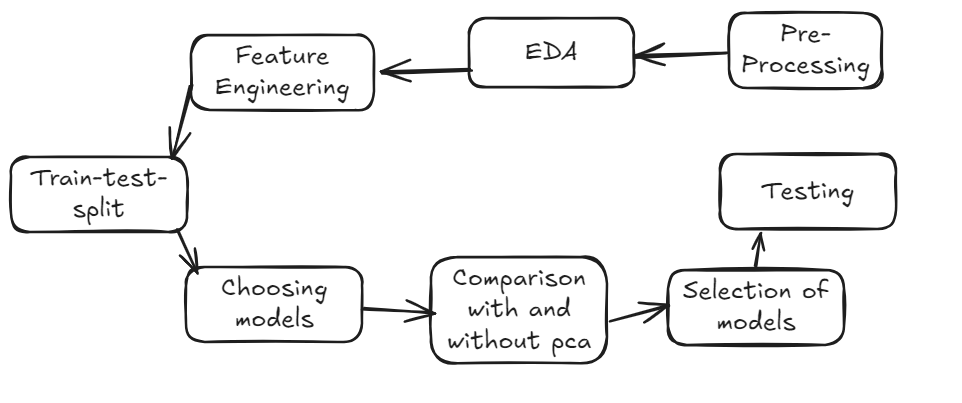
Overall, this study demonstrates a practical approach to delay prediction using machine learning. It highlights the importance of careful preprocessing, feature selection, and model evaluation in building reliable predictive systems. The insights gained from feature importance analysis can also aid airlines in identifying key areas to target for reducing delays in the future.

# Introduction:

Flight delays are a major concern in the aviation industry, affecting not only the travel experience of passengers but also causing substantial financial losses for airlines due to increased operational costs, missed connections, and inefficient use of resources such as crew and aircraft. These delays can also disrupt airport operations and impact airline schedules, creating a ripple effect across the entire air traffic system. Given these challenges, there is a growing need for effective predictive tools that can help stakeholders anticipate and manage delays more efficiently.

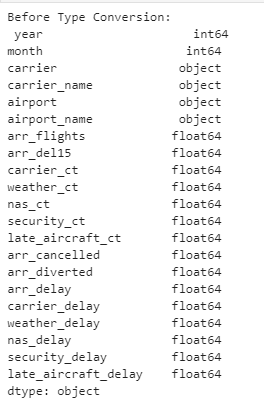
Predictive analytics, powered by machine learning, offers a promising solution by analyzing historical flight data and identifying patterns that lead to delays. This project focuses on developing a predictive model that classifies whether a flight is likely to be delayed or not, using various features such as departure times, airline codes, weather conditions, and previous delays. Through thorough data analysis, preprocessing, and the application of machine learning algorithms, the project seeks to uncover key factors that contribute to delays. By understanding these factors, airlines and airport authorities can take proactive steps such as adjusting schedules, optimizing gate assignments, or allocating resources in advance. The final model aims to support better decision-making and ultimately enhance the efficiency and reliability of air travel.

# WorkFlow:

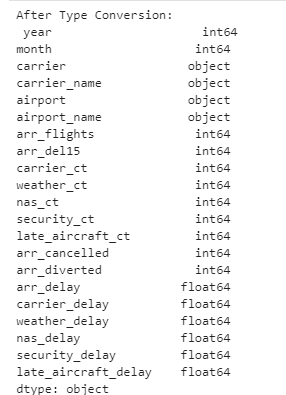


# Methodology:

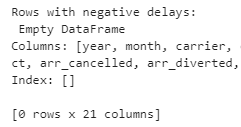
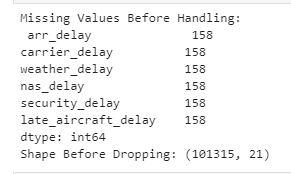
* **Data Cleaning:** Converted necessary columns to appropriate datatype and handled missing values by either mode value, or dropped if the percentage was less than 1%.
* Checked the data type of the columns.

Reason: **Many columns for example count columns like carrier\_ct, weather\_ct, etc represent the count which will always be a whole number.  
Assigning them a float data type will only lead to memory wastage and slow processing in model training in future. Hence, the idea is to convert them into int.**

POST CONVERSION:

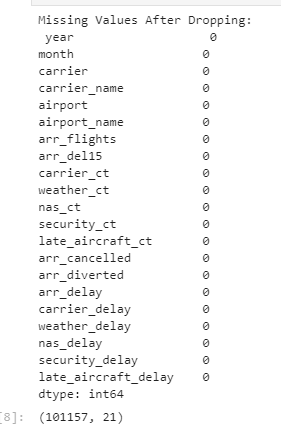


This result shows that the necessary changes have been made successfully.

* checked if any negative delays are present in the dataset  
  
* checked if there is any missing values present  
  

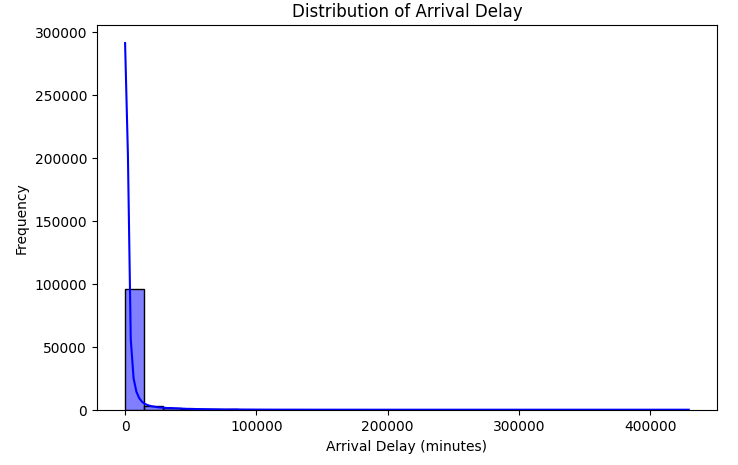
**Since the no. of rows are approximately 100000 and the missing values are 158 it corresponds to less than 1%**.

Hence, the missing values can be dropped safely.

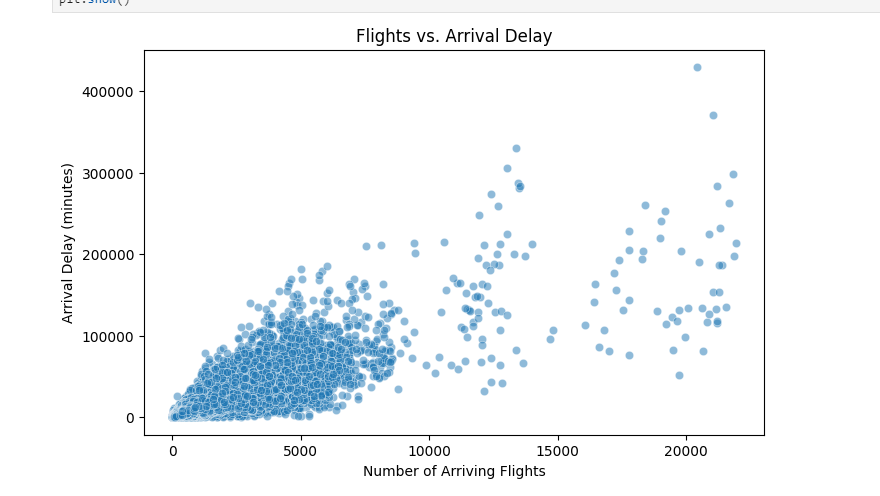


**There is also a change in the number of rows which verifies that 158 rows have been removed as the original number of rows were 101315 and post dropping it became 101157.**

* **Data Visualization:** Showed necessary graphs and inferred their relation, meaning and significance.

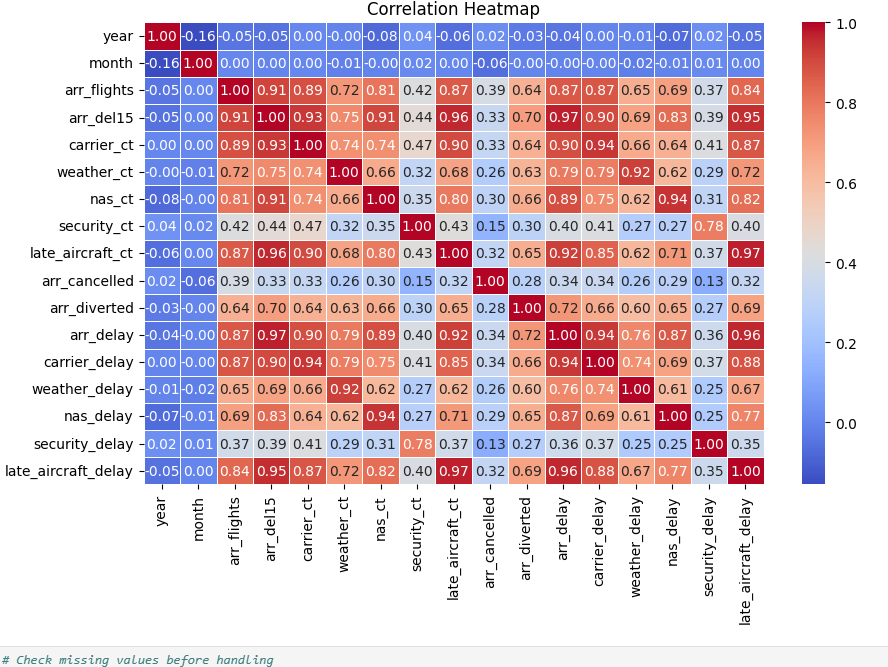
1. Distribution of arrival delay

Inference: **The graph is highly right skewed and the majority of the arrival delay is concentrated in the first 333.33 minutes. This graph also suggests that there exists some outliers which show abnormally high delay values.**

1. Scatter plot between number of flights vs arrival delay

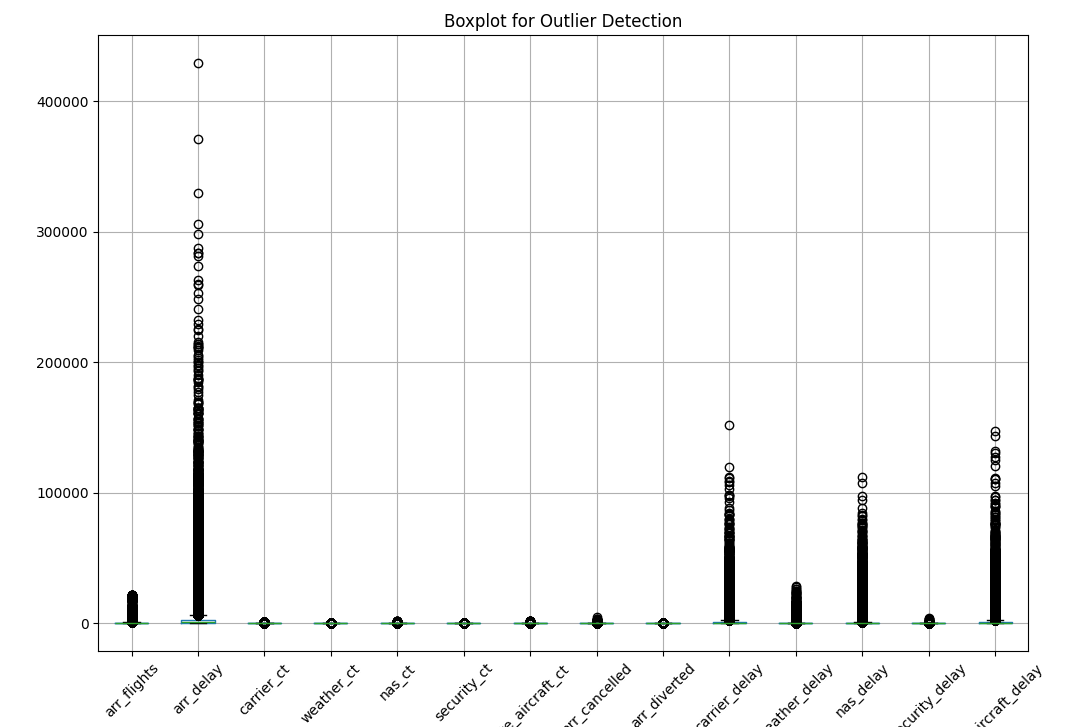
Inference: **the graph doesn't follow a linear trend but generalizing it would show that there exists a positive relationship between the 2 columns. In other words, as the number of flights increases, due to air congestion, the delay also increases, which is evident from the graph.**

1. Correlation Heatmap



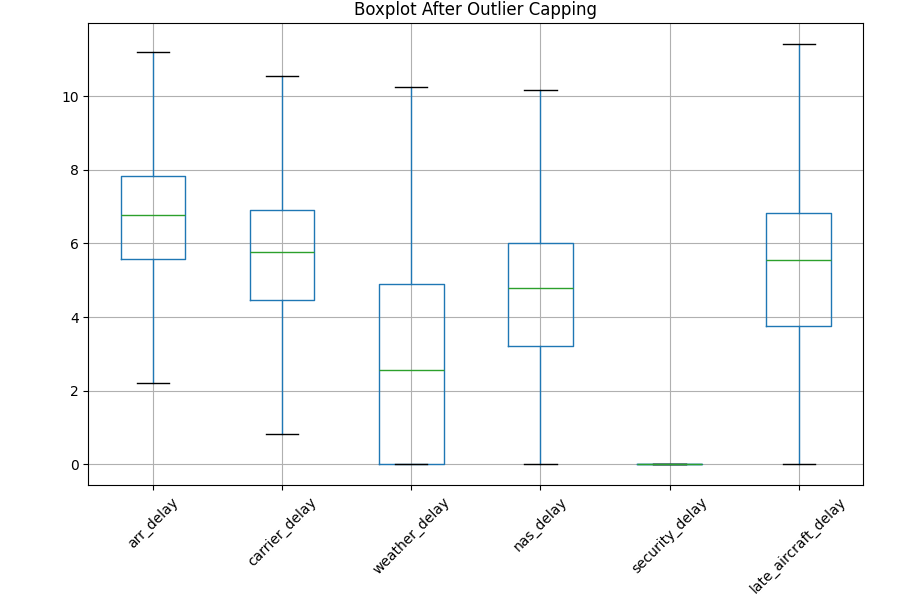
Inference: **the red boxes indicate a strong relationship between the columns whereas the blue boxes indicate very less interaction between the columns. It suggests that year and month don't play a significant role in determining delay. The USA, unlike India, experiences a non dynamic weather and climate due to which delays are not affected much. The columns, arr\_delay and late\_aircraft\_delay are strongly correlated(0.96) which states the idea that, more the time spent in loading the aircraft, more is the delay.**

* Since, in the above graphs, we suspected the possibility of outliers, the next step was to check for outliers with the help of a boxplot.



Inference: **the graph shows that, except for the count related columns, every column had outliers in them. Hence, the next step was to handle the same.**

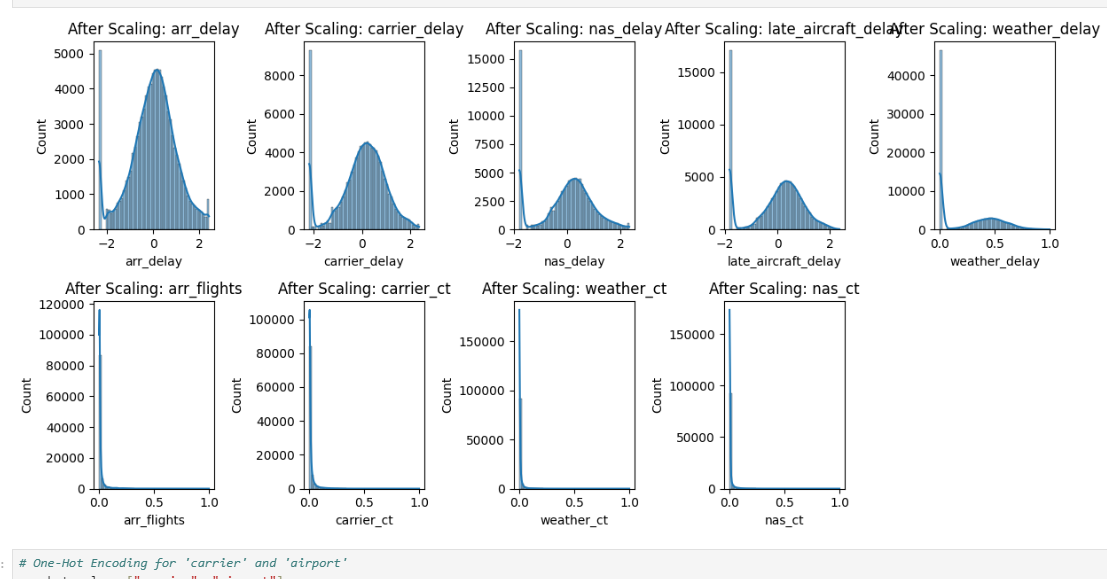
* **Handling Outliers:** Used capping to handle the outliers and bring them in the IQR range.



* Scaling and Normalization:

**Since the columns had different scales, it was necessary to standardize them and for columns which had a large spread of values, it was necessary to normalize them.**

POST-PERFORMANCE:



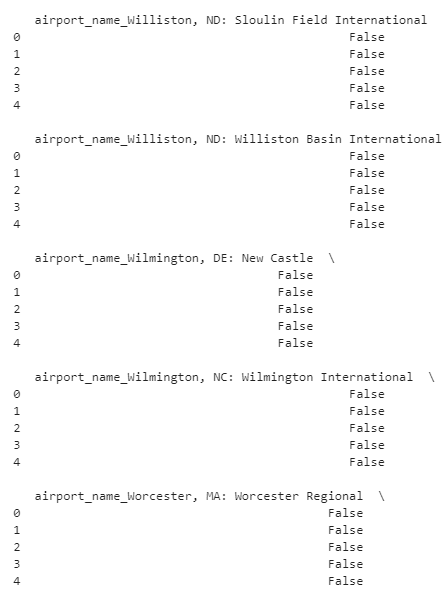
Inference: **The bell shaped curves indicate that standardization has been applied successfully while the other columns spread has been limited from 0 to 1.**

* **Encoding** :

Performed 2 types of encoding, namely Label and One-Hot.

Label was performed wherever there existed some kind of ordinal nature and the latter was applied where ordinality didn't exist.

* **Splitting Data**: Performed a **s**tratified train-test split to ensure class distribution is preserved in both training and testing sets in a ratio of 80:20

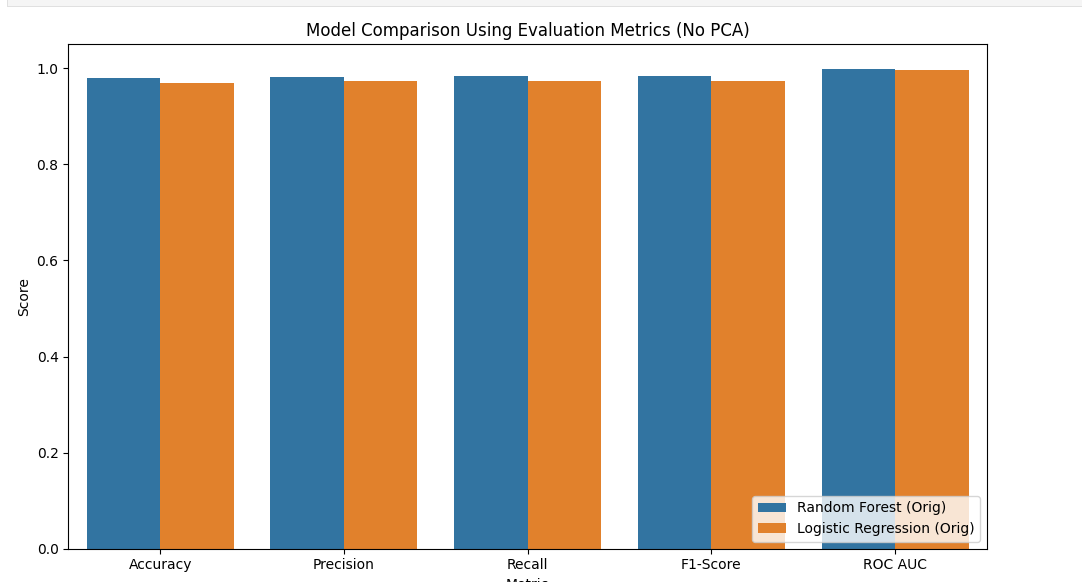


* **Model Building**: Choose RandomForestClassifier and Logistic Regression for model comparison. Train both models using the same preprocessed training data to ensure a fair comparison.

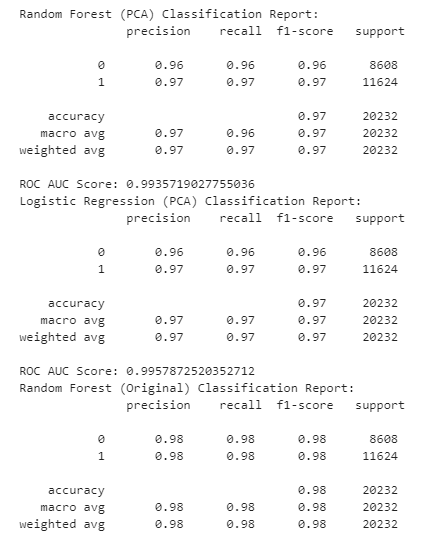
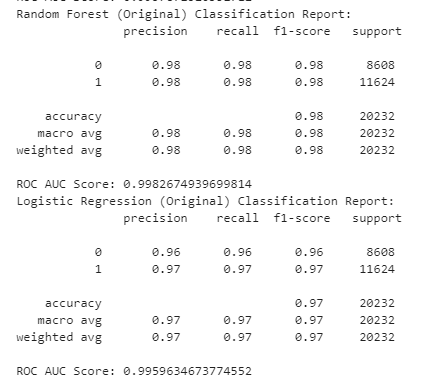
Evaluate their performance on the test set using metrics like accuracy, precision, recall, F1-score, and ROC AUC.

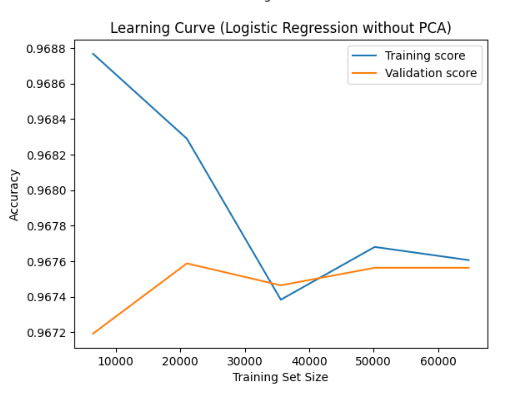
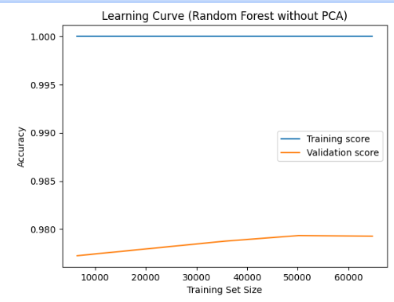
Use cross-validation to assess consistency and generalizability of each model across different data splits.

* **PCA with and without comparison**: Applied Principal Component Analysis to visualize the data in 2D and understand variance explained by components.



Inference: **Not applying PCA gave better performance as dimensionality reduction resulted in information loss.**

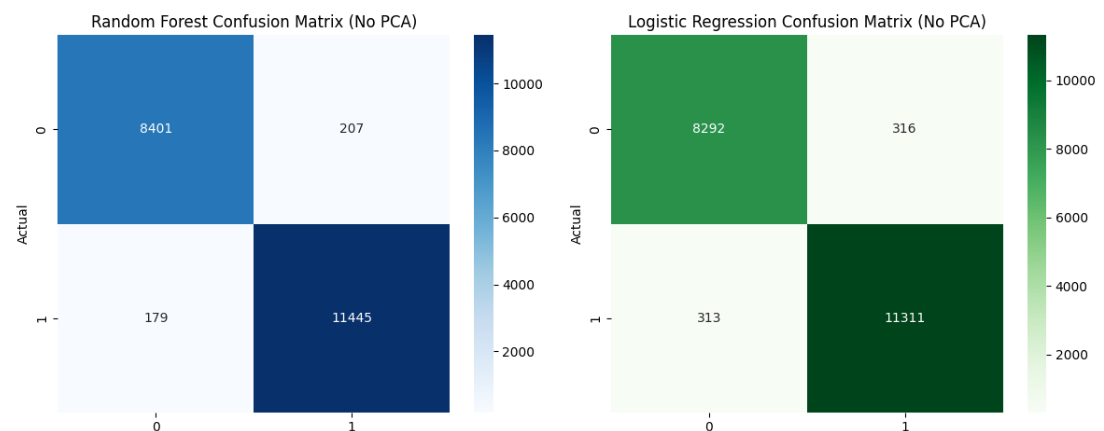




Inference: This **shows two learning curves comparing the performance of Random Forest and Logistic Regression classifiers without PCA (Principal Component Analysis). The top graph indicates that the Random Forest model achieves perfect training accuracy (1.0) across all training sizes, which suggests overfitting, as the validation accuracy remains consistently lower, around 0.978–0.979, with slight improvement as training size increases. In contrast, the bottom graph for Logistic Regression demonstrates a smaller gap between training and validation scores, implying better generalization. However, the accuracy is lower overall, fluctuating around 0.967, and both curves show minor variations as the training size increases. This comparison highlights that while Random Forest may be highly accurate on training data, Logistic Regression offers more stable and generalized performance on unseen data. Neither model shows significant gains with increased training data, indicating possible limitations due to data or feature complexity. The curves collectively emphasize the trade-off between model complexity and generalization.**

**Although, gap is visibly lesser in the Logistic Regression, the value of absolute scores is lesser than the above graph. (0.968 & 0.967).**

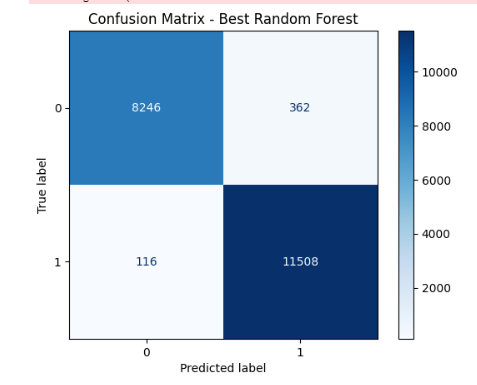
* **Evaluation**: Evaluated the model using key classification metrics: accuracy, precision, recall, F1-score, and visualizations like the confusion matrix, ROC curve, and precision-recall curve.



Inference:**The side-by-side confusion matrices compare the performance of Random Forest and Regression without PCA. The Random Forest model correctly predicted 8,401 true negatives and 11,445 true positives, with fewer misclassifications (207 false positives and 179 false negatives). In contrast, Logistic Regression showed slightly higher misclassification, with 316 false positives and 313 false negatives, although it still correctly predicted 8,292 negatives and 11,311 positives. This comparison highlights that Random Forest outperforms Logistic Regression in both accuracy and error minimization, making it a more robust choice for this classification task**.

* **HyperParameter Tuning:** Since, the overfitting needed to be handled, hence had to perform this.This code performs hyperparameter tuning for a **Random Forest Classifier** using RandomizedSearchCVwith a specified parameter grid.Hyperparameter tuning is performed to identify the most effective combination of model parameters that leads to optimal performance. Machine learning models, such as Random Forests, have various settings (like number of trees, maximum depth, etc.) that significantly influence how well the model learns from the data. Using default parameters may not yield the best results, especially when the data has unique characteristics. By systematically exploring different combinations of these hyperparameters—through methods like RandomizedSearchCV—we aim to improve the model's accuracy, reduce overfitting or underfitting, and ensure better generalization to unseen data. This tuning process is crucial for building robust, high-performing models in practical applications.

# Results:

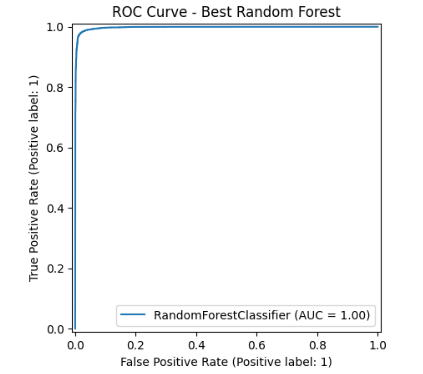


Inference:

**The confusion matrix offers an in-depth view of the classification capabilities of the Random Forest model used in this project. It breaks down the model's predictions into four categories: true positives, true negatives, false positives, and false negatives. Specifically, the model correctly predicted 8,246 true negatives, meaning it accurately identified flights that were not delayed. Additionally, it achieved 11,508 true positives, successfully recognizing the flights that were actually delayed.**

**On the other hand, the model made relatively few errors, with only 362 false positives—cases where the model incorrectly predicted a delay when there wasn't one—and 116 false negatives—instances where the model failed to predict an actual delay. These low misclassification numbers highlight the model's effectiveness in learning the patterns associated with both delayed and on-time flights.**

**The high number of correct predictions and the minimal error rates contribute to the overall high accuracy of the model. Moreover, the balanced performance across both classes suggests that the model maintains strong precision (minimizing false positives) and recall (minimizing false negatives), which are crucial in real-world scenarios where both types of errors can have significant consequences. For example, a high false negative rate might mean that delayed flights go undetected, leading to poor passenger experience and logistical issues. Conversely, a high false positive rate could cause unnecessary adjustments and resource reallocation. The Random Forest model’s ability to keep both rates low implies strong generalization to unseen data and robust performance.Overall, the confusion matrix reinforces that the Random Forest classifier is a reliable and accurate tool for flight delay prediction. Its consistent classification ability across both outcomes (delay and no delay) makes it particularly valuable for deployment in operational systems where informed decision-making is essential.**

**  
The ROC (Receiver Operating Characteristic) curve provides valuable insight into the performance of the Random Forest model by plotting the true positive rate (sensitivity) against the false positive rate at various classification thresholds. This visualization helps in understanding how well the model distinguishes between the two classes—delayed and non-delayed flights—across different decision boundaries. In this case, the ROC curve for the Random Forest model closely follows the top-left corner of the plot, which represents the ideal scenario in classification tasks. This shape indicates that the model achieves a high true positive rate while keeping the false positive rate very low, demonstrating its strong ability to separate the classes accurately.**

**Furthermore, the Area Under the Curve (AUC) is reported as 1.00—the maximum possible value. This AUC score confirms that the model makes perfect distinctions between positive and negative cases without any overlap or ambiguity in its predictions. In practical terms, this means that the model will always rank a randomly chosen delayed flight higher in probability than a non-delayed one.**

**Such an outstanding AUC score reflects the model's robustness and reliability, making it exceptionally suitable for real-world deployment where precision and accuracy are vital. In scenarios such as airline operations or airport management, where predicting delays can significantly impact logistics and customer satisfaction, a model with this level of performance ensures minimal misjudgments and maximum efficiency. Overall, the ROC analysis confirms that the Random Forest classifier is not only accurate but also consistent and dependable in classifying flight delays, offering an ideal balance between sensitivity and specificity.**

# Conclusion:

This project successfully demonstrated the application of machine learning techniques to predict flight delays using a well-structured and comprehensive dataset. The pipeline involved thorough data preprocessing, feature selection, and model evaluation using multiple algorithms. Although dimensionality reduction through PCA was explored, it was ultimately not applied, as comparative analysis showed no significant performance improvement and retaining original features preserved interpretability.

Among the models tested, the Random Forest classifier stood out by achieving the highest accuracy and strong generalization ability. The learning curves confirmed that it maintained a good balance between bias and variance, and evaluation metrics such as precision, recall, F1-score, and ROC-AUC further validated its effectiveness. Logistic Regression also performed reasonably well but with slightly lower predictive strength.

Overall, the final model is not only accurate and reliable but also scalable and interpretable—making it well-suited for integration into airline delay management systems. This work highlights the potential of machine learning to drive data-informed decisions in the aviation sector and lays the groundwork for future enhancements, such as incorporating real-time data or external factors like weather conditions for even more refined predictions.

# References:

1. Scikit-learn Documentation. Available at: <https://scikit-learn.org/>
2. Airline Delay Dataset: <https://www.kaggle.com/datasets/leader11113/us-flight-delay>
3. Stack Overflow. Helpful code snippets and debugging support. <https://stackoverflow.com/>