Flight Delay Prediction Analysis

Contents

Chapter 1: Introduction

Flight delays are a significant challenge in the aviation industry, impacting airline operations, passenger satisfaction, and economic efficiency. With millions of flights operating globally, unanticipated delays result in financial losses and logistical disruptions. Understanding the factors that contribute to flight delays is essential for improving airline scheduling, enhancing customer experience, and optimizing operational efficiency. This study presents a comprehensive analysis of flight delays using data science techniques, focusing on statistical analysis, data visualization, and machine learning models to identify key delay patterns. The dataset used in this research consists of historical flight records with attributes such as departure time, arrival time, weather conditions, airline carriers, and airport congestion. We employ various machine learning algorithms, to predict flight delays. Additionally, data visualization methods are used to uncover trends and correlations between different variables. The results indicate that factors such as weather conditions, flight distance, and departure time play a significant role in determining flight delays. The predictive models demonstrate significantly less error, suggesting that machine learning can be a valuable tool in mitigating delays and enhancing airline scheduling efficiency. This research contributes to the field of aviation analytics by providing data-driven insights that can help improve decision-making processes. Future work can explore deep learning models and real-time delay prediction systems for further advancements in flight delay mitigation strategies.

Some notable works have been conducted by many researchers on flights delay data analysis. Pineda-Jaramillo et al. used multiple classification models to detect flight delays, achieving around 77.2% accuracy with Gradient Boosting [1]. Mokhtarimousavi et al. proposed a hybrid model of Artificial Bee Colony (ABC) – a metaheuristic algorithm and Support Vector Machine (SVM). The ABC-SVM model attained a test accuracy of 94.7% on flights delay classification [2]. Henriques and Feiteira performed predictive modelling using a Multi-Layered Perceptron (MLP) model which achieved an accuracy of 85.63% on test dataset [3]. These works showcase the potential of statistical learning and neural networks in predicting flights delay.

Chapter 2: Problem Statement

Several factors contribute to flight delays, including adverse weather conditions, air traffic congestion, mechanical failures, and airline-specific inefficiencies. Understanding these causes through data-driven approaches can help airlines optimize scheduling, improve efficiency, and minimize financial losses.

This study focuses on analyzing historical flight delay data to identify key patterns and develop predictive models. By leveraging data analytics and machine learning, the study aims to provide insights that can help airlines, air traffic controllers, and policymakers make informed decisions to mitigate delays and improve the overall efficiency of air travel.

The primary objectives of this study are as follows:

1. **Data Collection and Preprocessing**: The dataset is obtained from Kaggle [4] which consists of 28,821 rows and 23 features, containing flights information.
2. **Exploratory Data Analysis (EDA) and Visualization**: Identify trends and patterns in flight delays through statistical analysis and graphical representations. Examine correlations between different factors influencing flight delays.
3. **Feature Extraction**: Identify the most significant factors contributing to flight delays. Apply feature extraction techniques to improve model performance and reduce computational complexity.
4. **Machine Learning**: Implement statistical predictive models to forecast flight delays. Compare model performance based on MAE, MSE, RMSE and R2-Score.

Chapter 3: Python Packages Used

Sl.	Library	Modules		Functions/Class imported	Explanation
1	Numpy [5]			-	For numerical operations.
2	Pandas [6]			read_csv	To load a DataFrame from a CSV File.
3	Matplotlib [7]	pyplot		figure, show	Basic plotting functions.
4	Seaborn [8]			histplot, boxplot, heatmap, pairplot	Advanced statistical visualizations.
5	Sklearn [9]	preprocessing		LabelEncoder	Encoding categorical data.
				StandardScalar	Scale numerical data.
		decomposition		PCA	Dimensionality reduction.
		model_selection		train_test_split	Splitting data into train & test.
		linear_model		LinearRegression	Implement Linear Regression model.
		ensemble		RandomForestRegressor	Implement Random Forest Regressor model.
		metrics		mean_absolute_error	Calculate MAE.
				mean_squared_error	Calculate MSE.
				root_mean_squared_error	Calculate RMSE.
				r2_score	Calculate R2-Score.
6	XGBoost [10]			XGBRegressor	To implement Extreme Gradient Boosting model.
7	Tensorflow [11]	keras	models	Sequential	To create an ANN model with multiple layers.
			layers	Dense	Layer of a number of units (neurons) with a predefined activation function.
				BatchNormalization	Stabilize and accelerate training by normalizing the inputs.
				Dropout	Regularization technique - during training, each neuron (activation) is **randomly "dropped"** (set to zero) with a probability.
				Input	Takes in input and passes it to the next layer.
			callbacks	EarlyStopping	Prevents overfitting in ANN by stopping the training process earlier than expected.

Chapter 5: Implementation Results

1. Data Visualization

To analyze the distributions of the numerical features present in the dataset, the histogram plot of every numerical feature has been made inside one figure for comparison and quick reference.

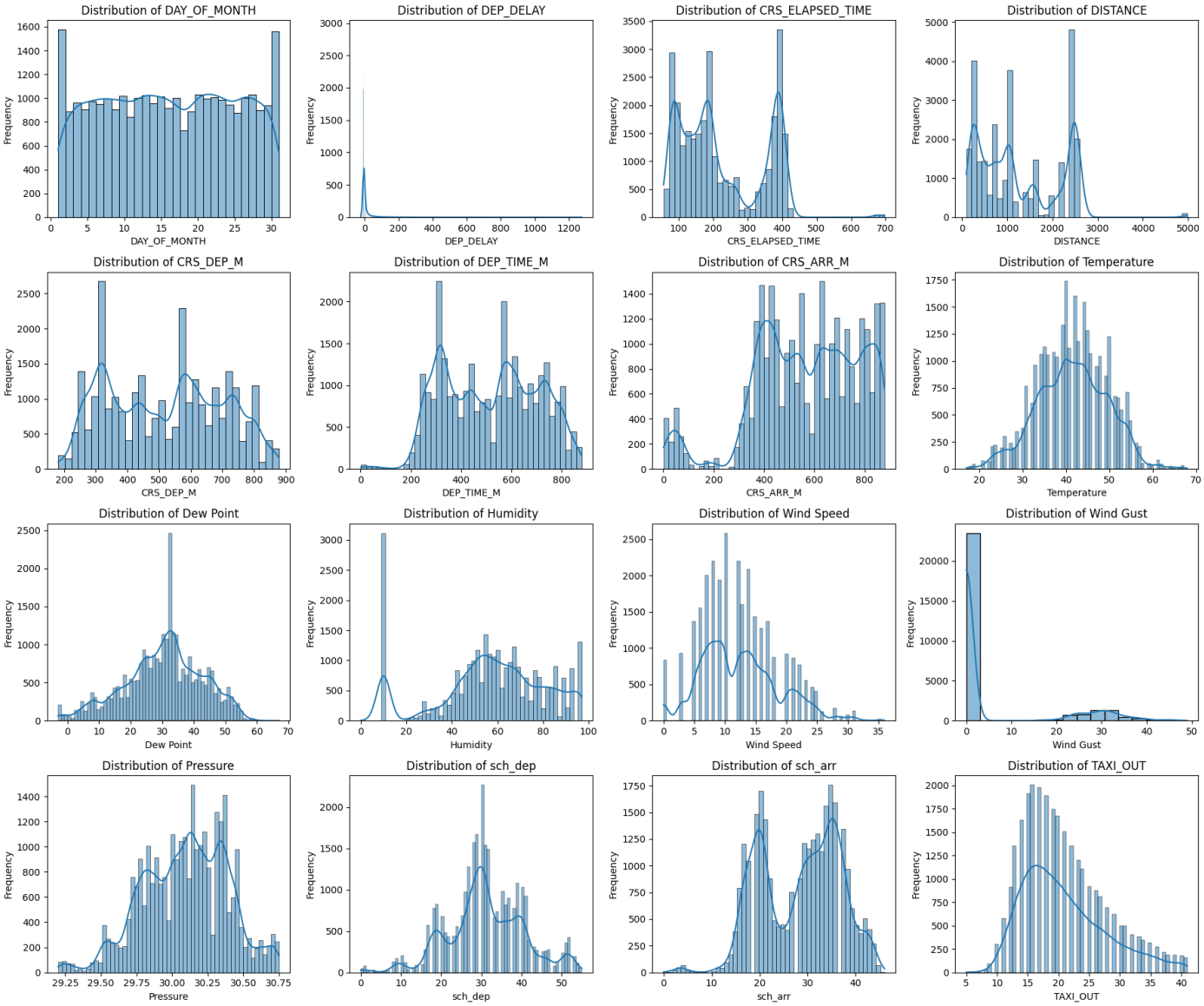

Figure 1: Distribution plot of Numerical Features

Categorical features play a role in determining the target variable. So, for that reason it is vital to know their distribution as well. Count plot of all the categorical features has been plotted in one figure.

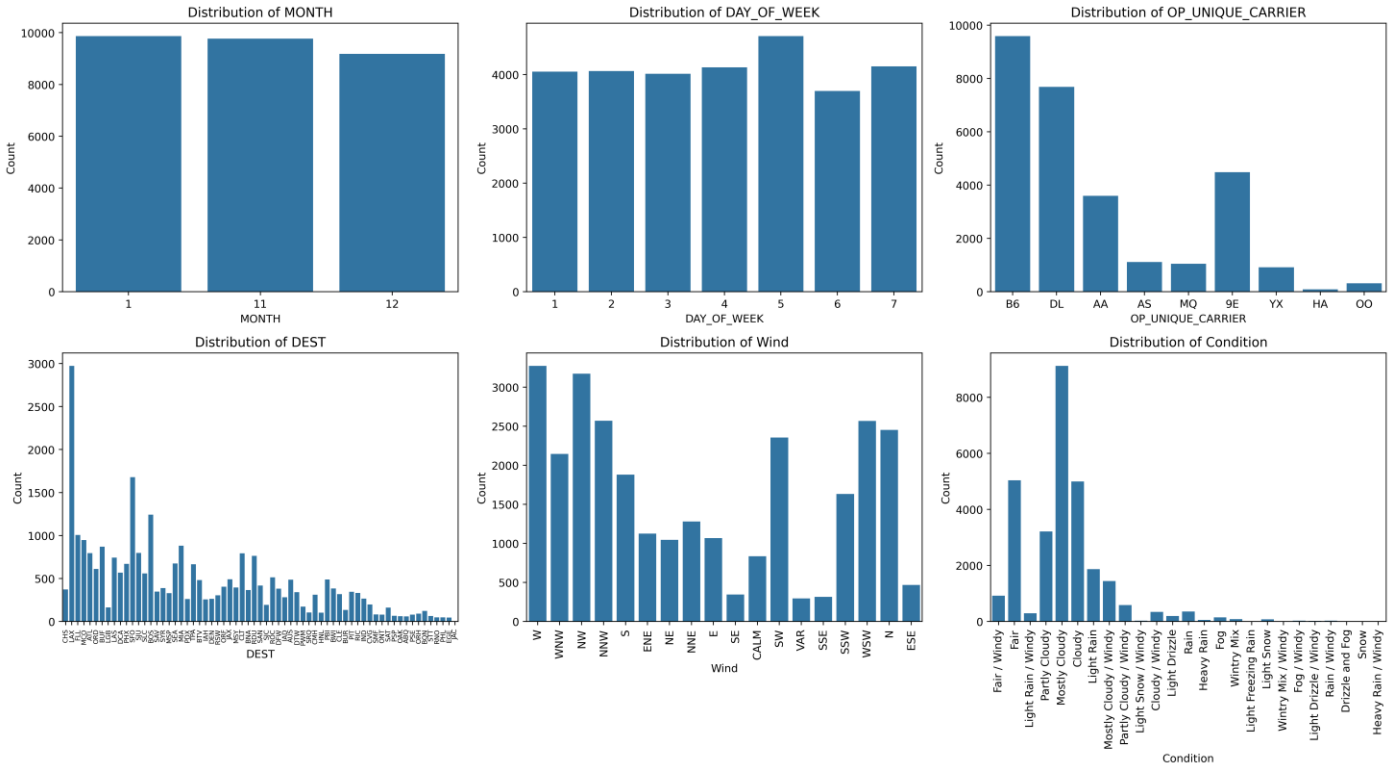

Figure 2: Count-plot of Categorical Features

The relation between each categorical feature with the target feature must be studied to understand key patterns, trends and categorical dependencies beforehand. This will make the data analysis process easier.

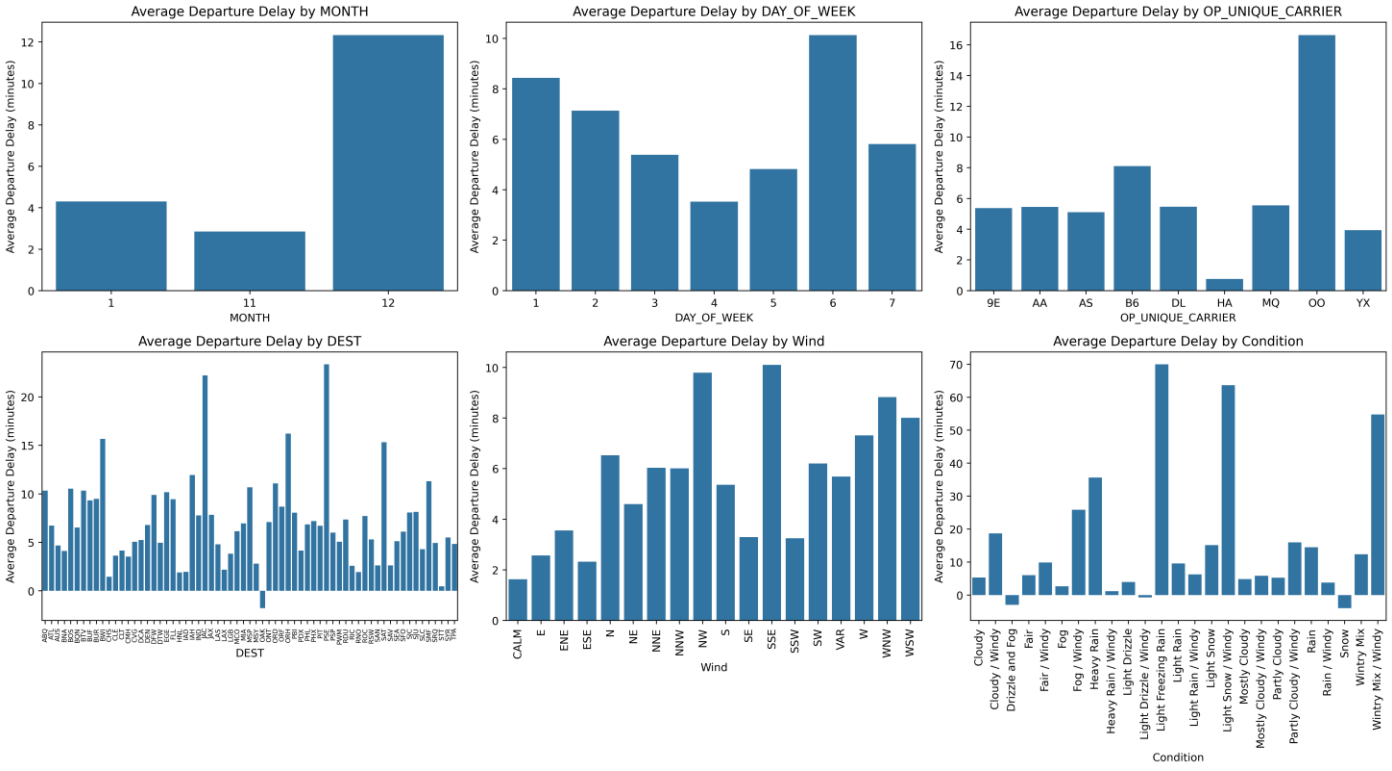

Figure 3: Average Departure Delay by Category

The correlation between the numerical variables give us an insight about the similarities between the features.

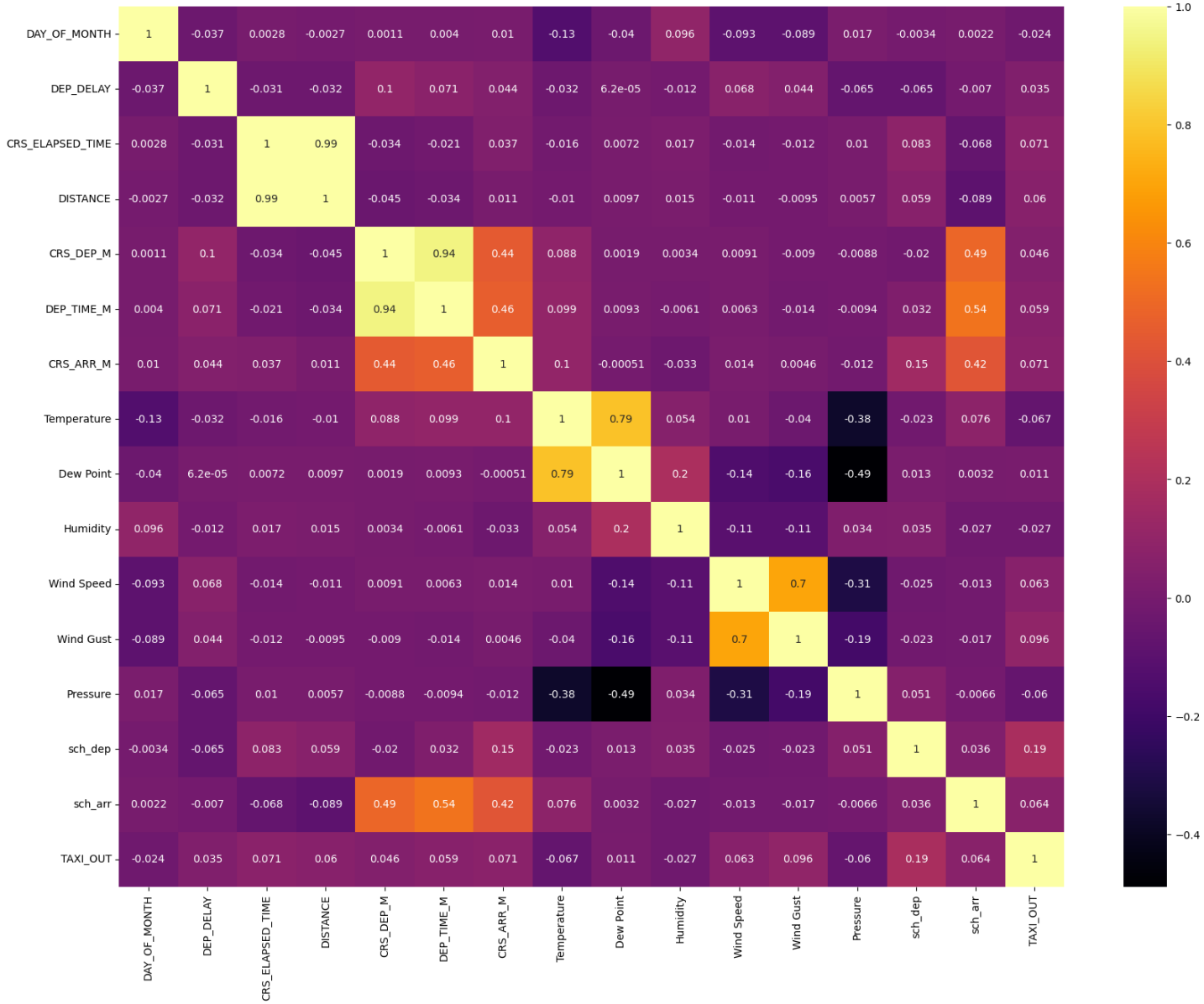

Figure 4: Correlation of Numerical Features

The Pair-plot is used to plot 2-dimensional relationships between features present in the dataset. It produces a Scatter plot if the two features to be compared are different, while a distribution plot (Histogram or sometimes KDE) is made when the features to be compared are the same.

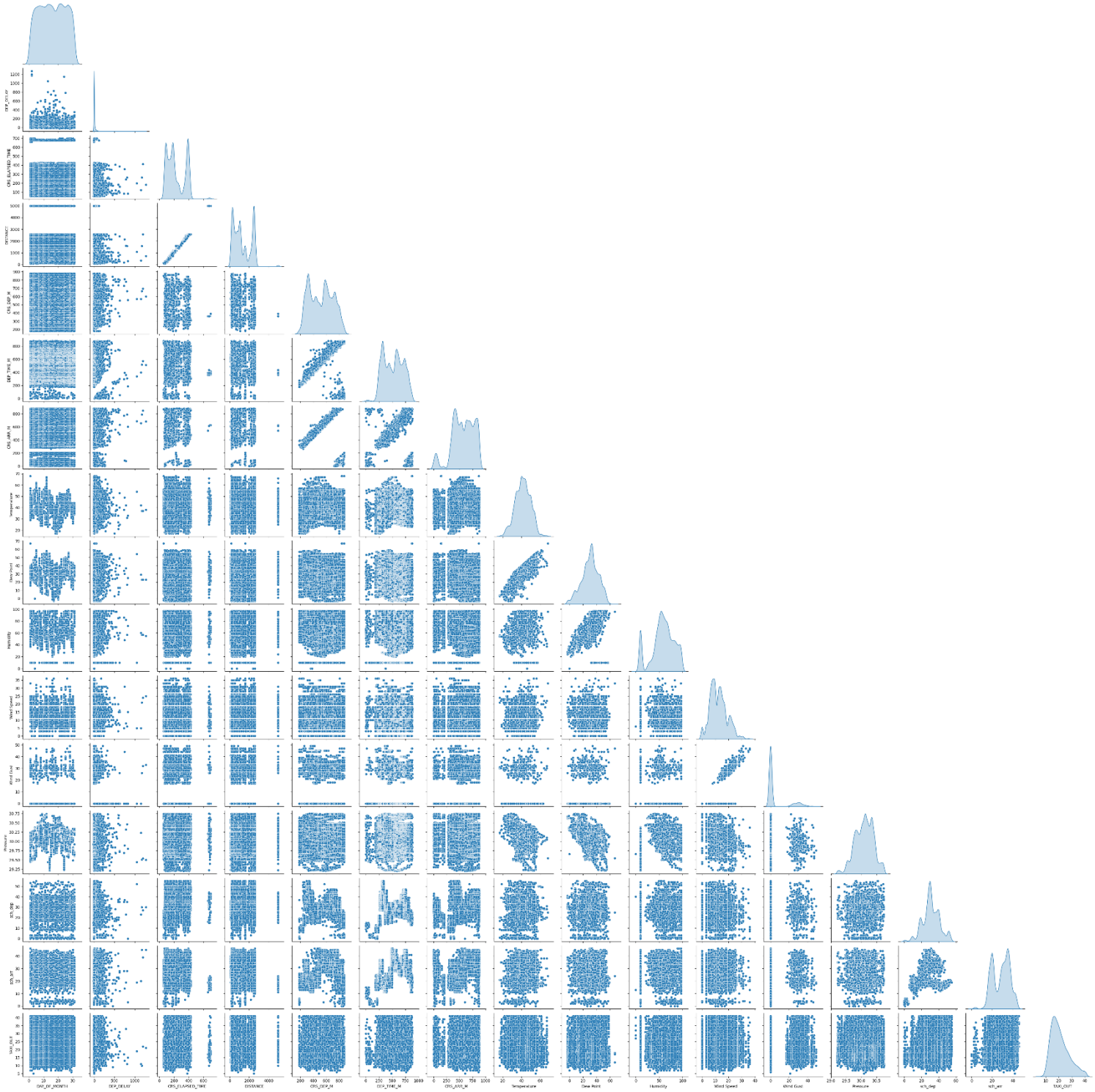

Figure 5: Pair-plot of Numerical Features

Box plot showcases the Inter-Quartile Ranges of the dataset and also the Outliers present in the dataset as well. Tuckey Fences method was employed to removed the outliers. This is the box plot of the numerical features before removal of the outliers.

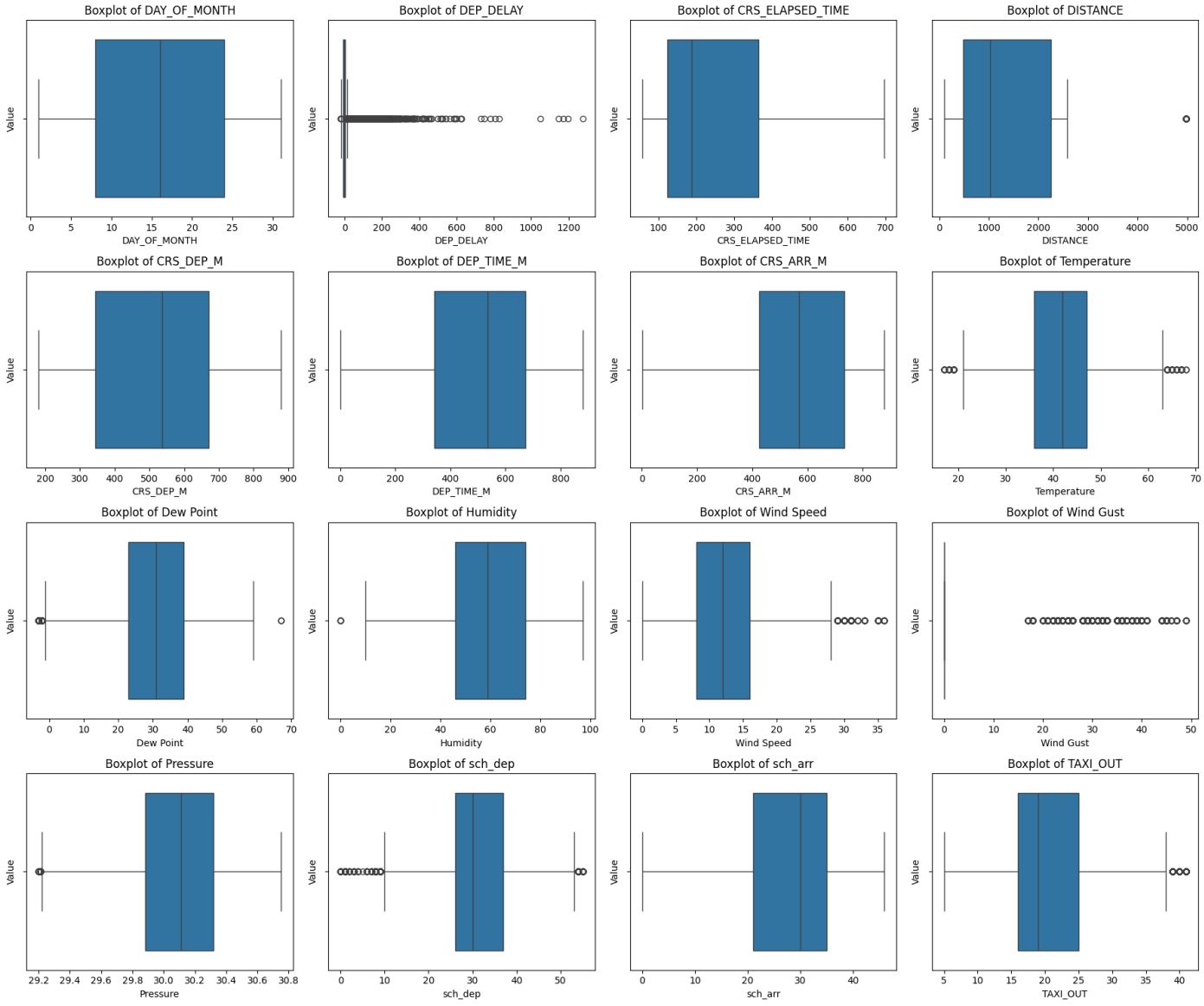

Figure 6: Box-plot of Numerical Features with outliers.

After removal of outliers, again a box plot visualization of the dataset is performed to validate the results. This is the box plot of the numerical features after the removal of the outliers.

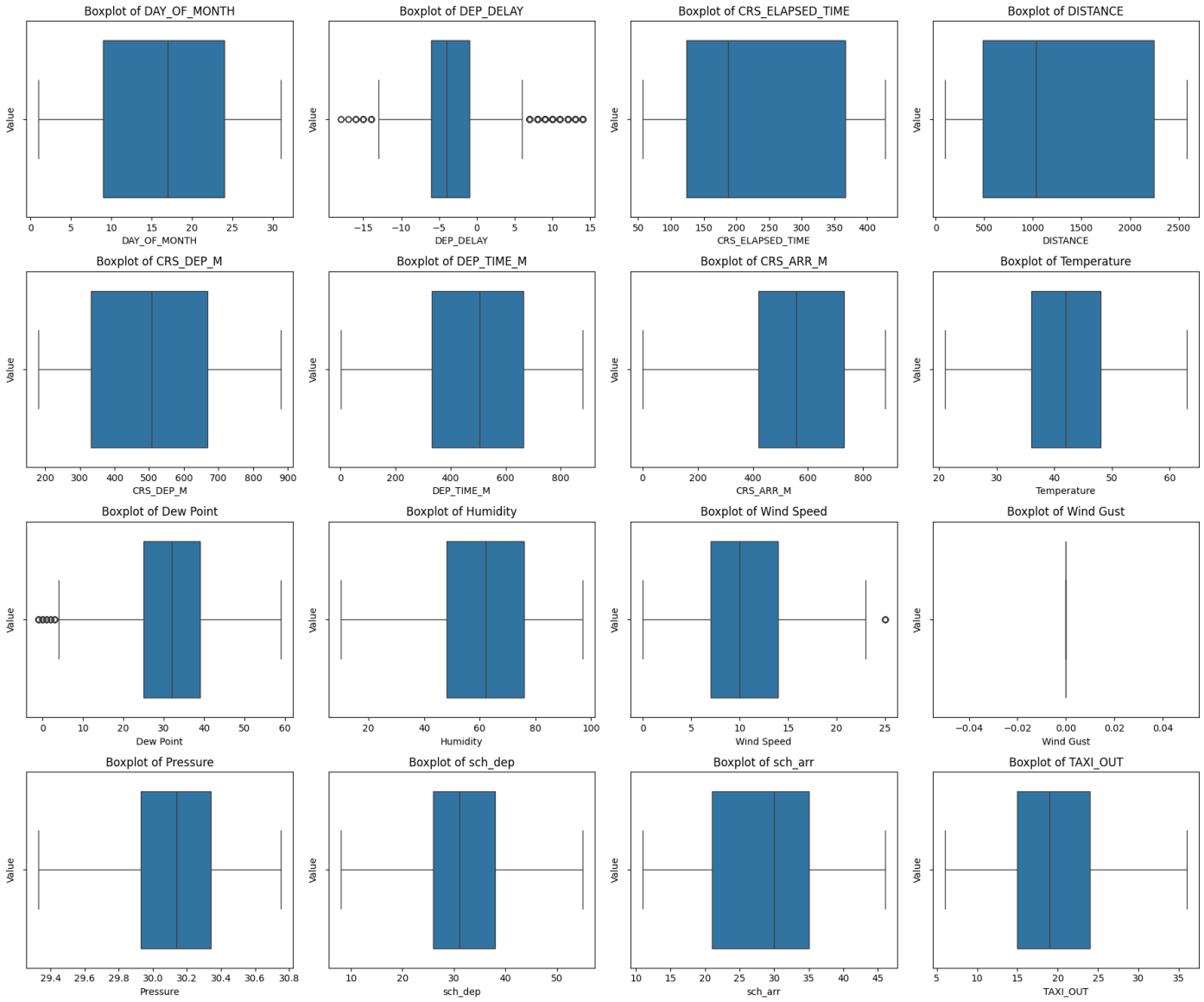

Figure 7: Box-plot of Numerical Features after outlier removal.

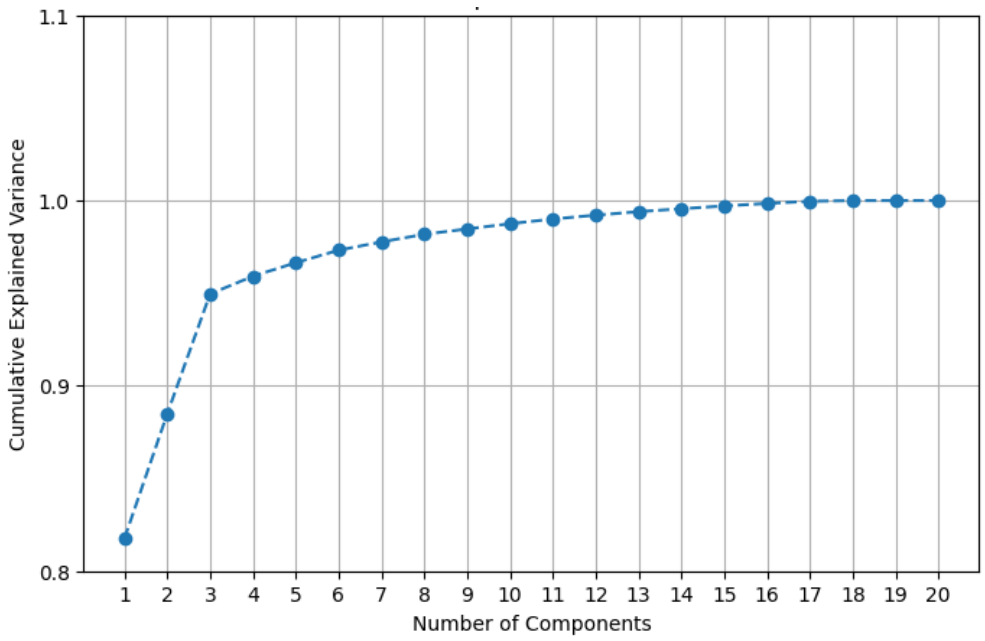

Figure 8: PCA Explained Variance

2. Model Evaluation

- **Linear Regression**

  It is unable to fit a regression hyperplane properly to the given dataset and hence performs poorly during testing phase.

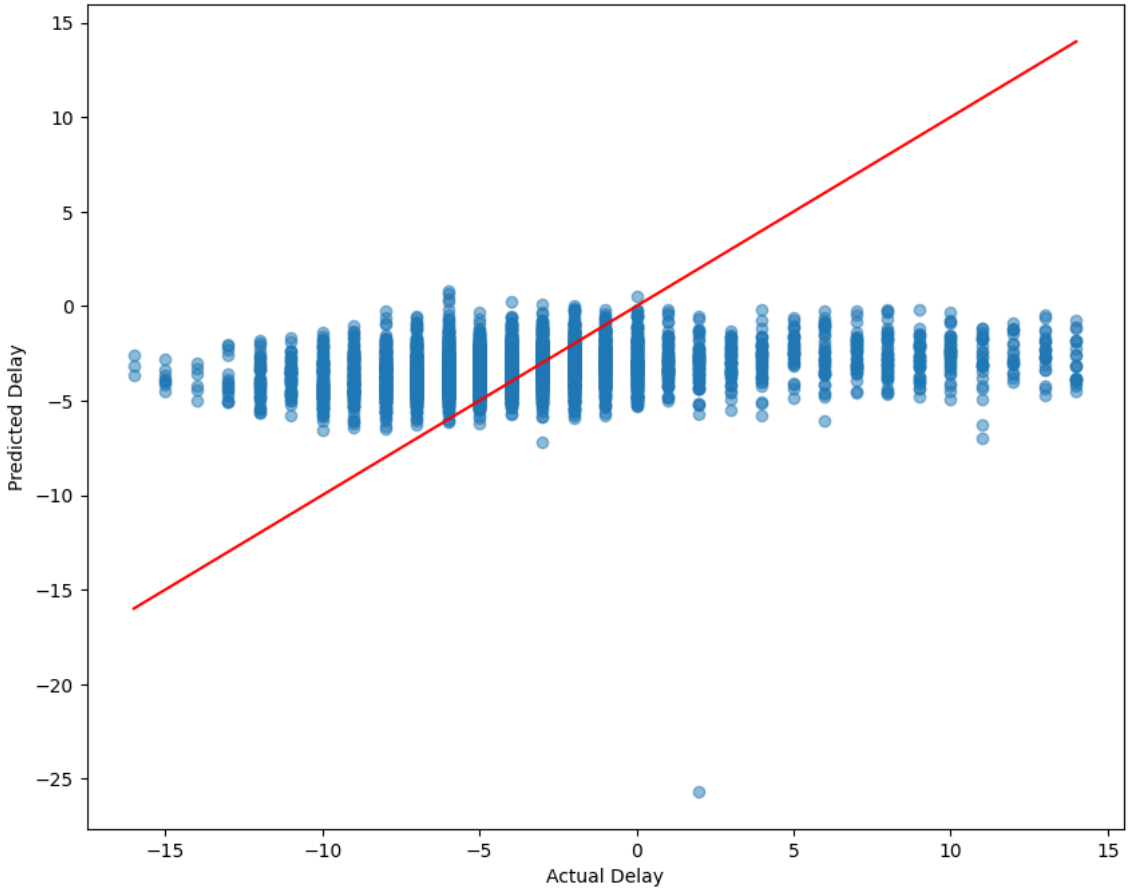

Figure 9: Actual vs. Predicted (Linear Reg.)

Metrics	Score
Mean Absolute Error	3.4622
Mean Squared Error	22.9459
Root Mean Squared Error	4.7902
R2-Score	0.0681

- **Random Forest Regressor**

  It a Bagging technique which creates multiple decision trees, trains them parallelly and aggregates all their predictions to form a single prediction. It performs quite well on the test dataset.

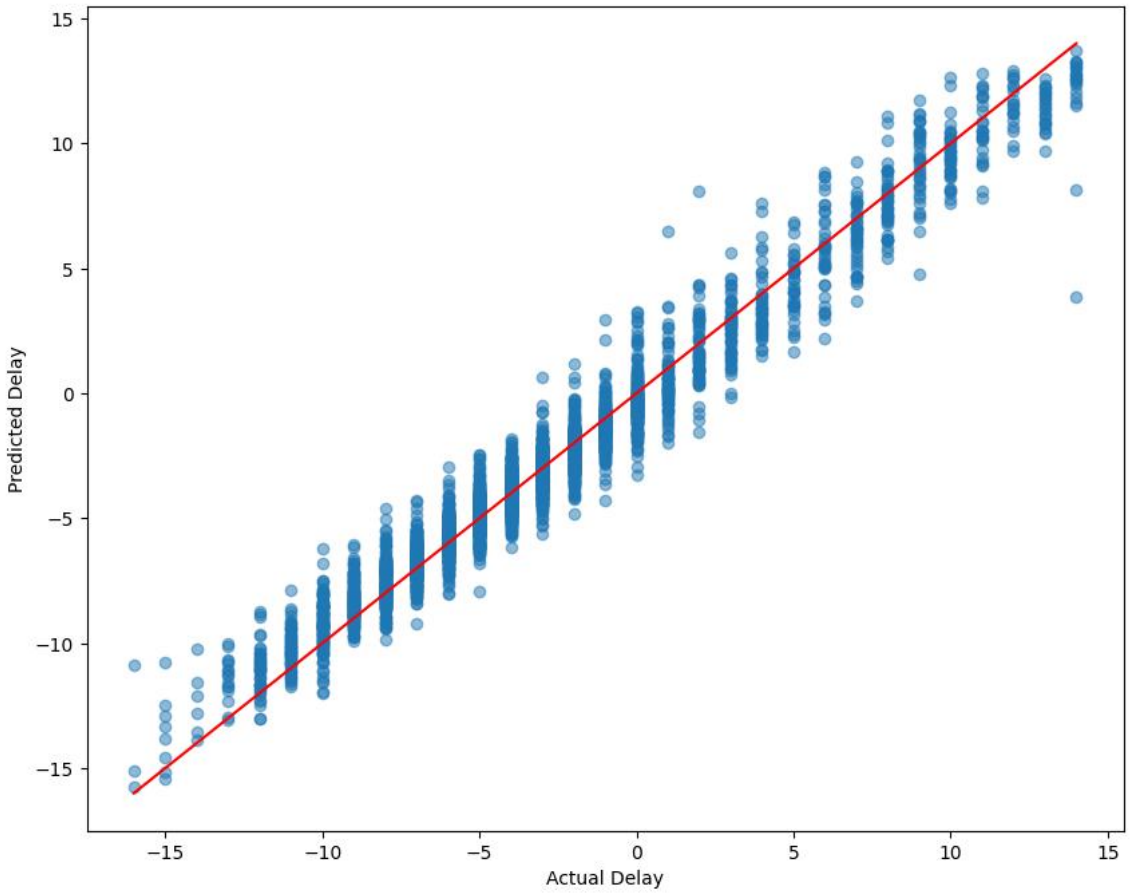

Figure 10: Actual vs. Predicted (Random Forest)

Metrics	Score
Mean Absolute Error	0.7578
Mean Squared Error	1.0195
Root Mean Squared Error	1.0097
R2-Score	0.9586

- **Artificial Neural Network**

An **Artificial Neural Network (ANN)** model consists of multiple interconnected units called **neurons**, each associated with **weights** that determine the strength of connections between them. These neurons are organized into **layers**—an **input layer**, one or more **hidden layers**, and an **output layer**.

An ANN model was created having **23,553** trainable and **672** non-trainable parameters.

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 256)	5,376
batch_normalization (BatchNormalization)	(None, 256)	1,024
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 64)	16,448
batch_normalization_1 (BatchNormalization)	(None, 64)	256
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 16)	1,040
batch_normalization_2 (BatchNormalization)	(None, 16)	64
dropout_2 (Dropout)	(None, 16)	0
dense_3 (Dense)	(None, 1)	17

Figure 11: ANN Model Structure

By analysis the training loss and validation loss we can get insights on how well the model is able to train itself and how quickly it is able to optimize its weights.

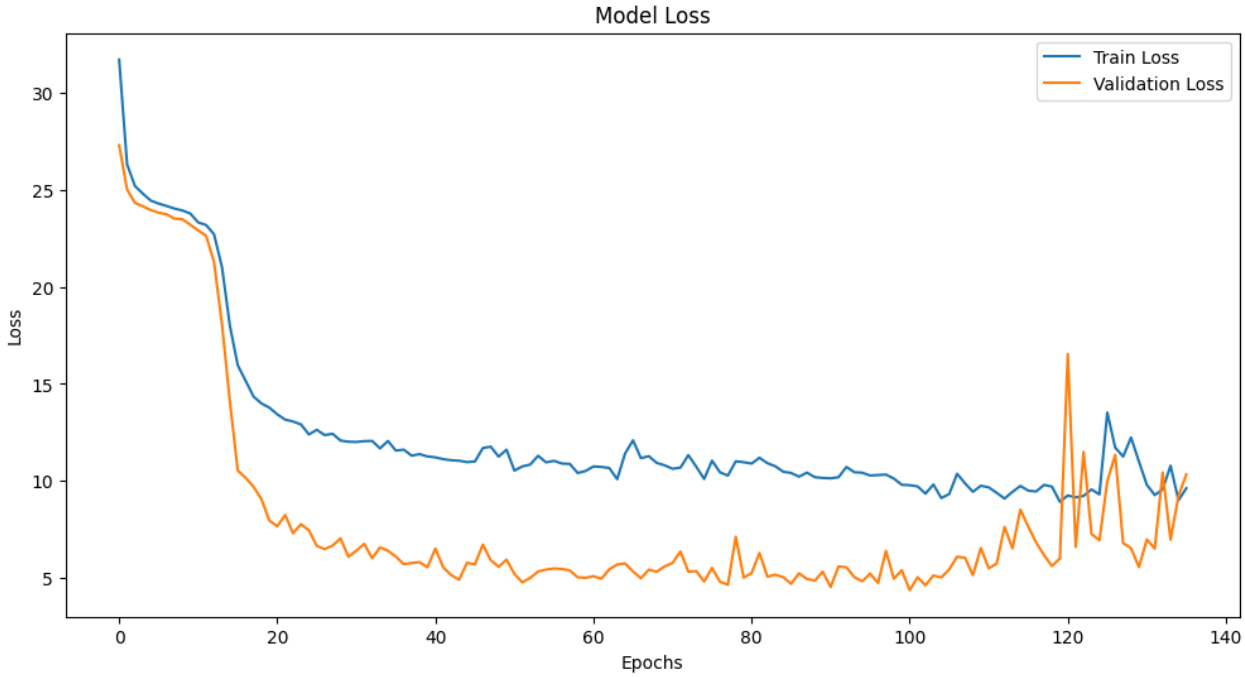

Figure 12: Train & Validation Loss (ANN)

The model performs better than Linear Regression model. However, it still is an underperformer compared to the Random Forest model.

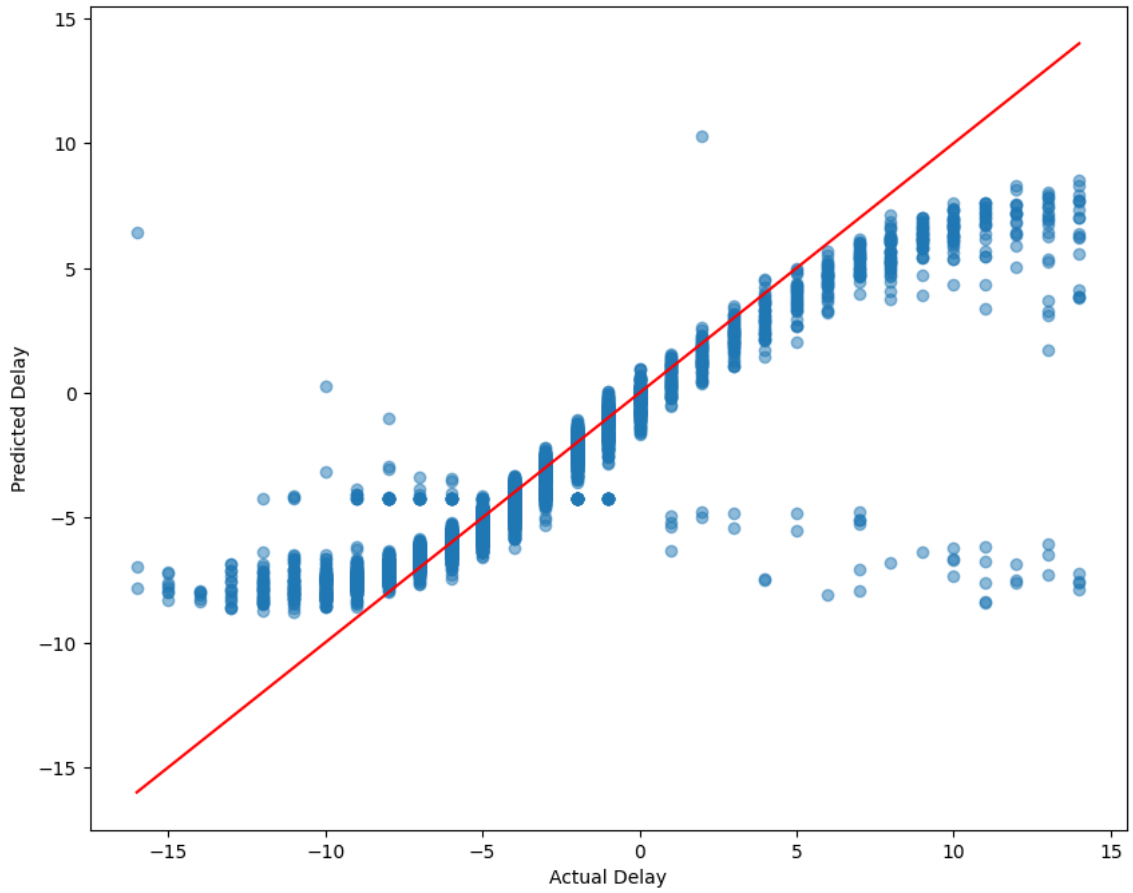

Figure 13: Actual vs. Predicted (ANN)

Metrics	Score
Mean Absolute Error	1.0344
Mean Squared Error	5.0704
Root Mean Squared Error	2.2518
R2-Score	0.7941

- **Extreme Gradient Boosting Regressor**

  It is a powerful machine learning model, designed for **regression tasks**. It is based on an optimized version of Gradient Boosting that enhances performance through **regularization (L1 & L2), parallel processing, and tree pruning.** This is best model so far in this study, attaining the lowest error rates compared to the other models employed.

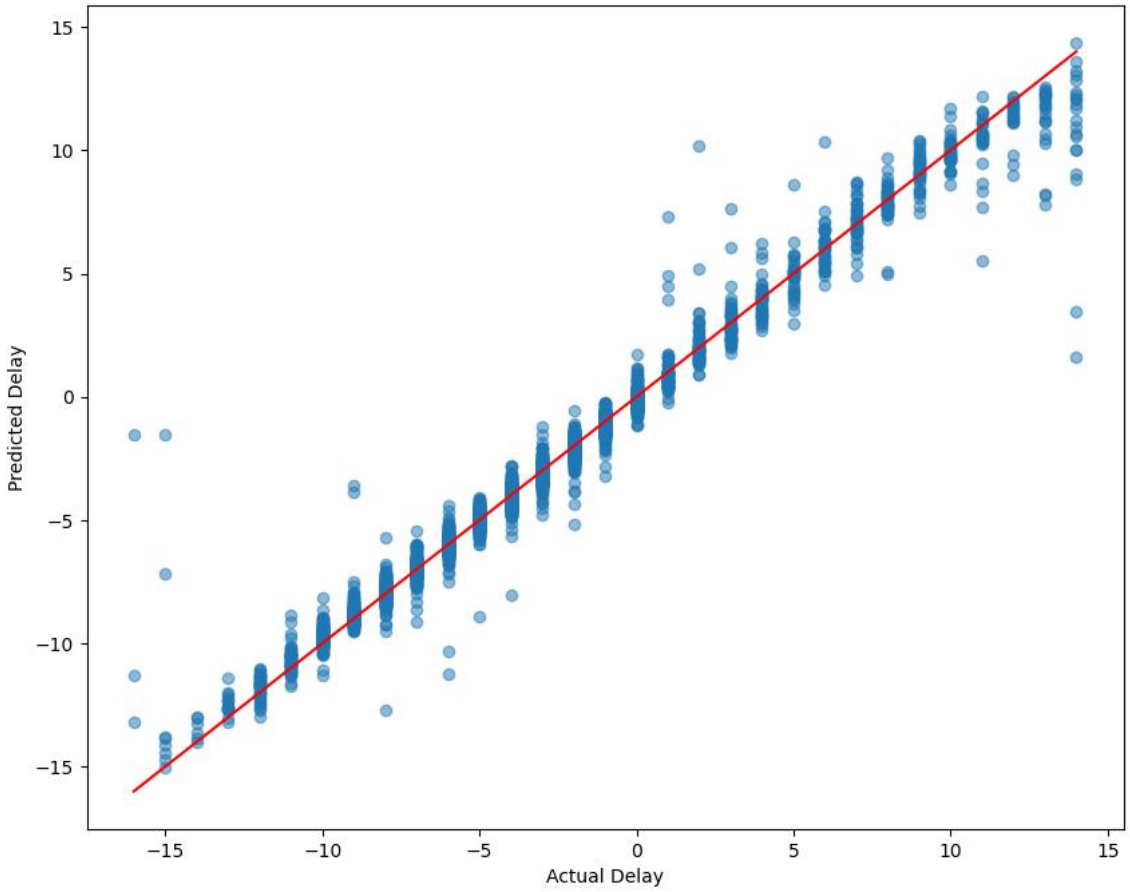

Figure 14: Actual vs. Predicted (XG-Boost)

Metrics	Score
Mean Absolute Error	0.3813
Mean Squared Error	0.5700
Root Mean Squared Error	0.7550
R2-Score	0.9768

This graph shows that how the model performance enhances as we increase the number of estimators in XGB-Regressor model.

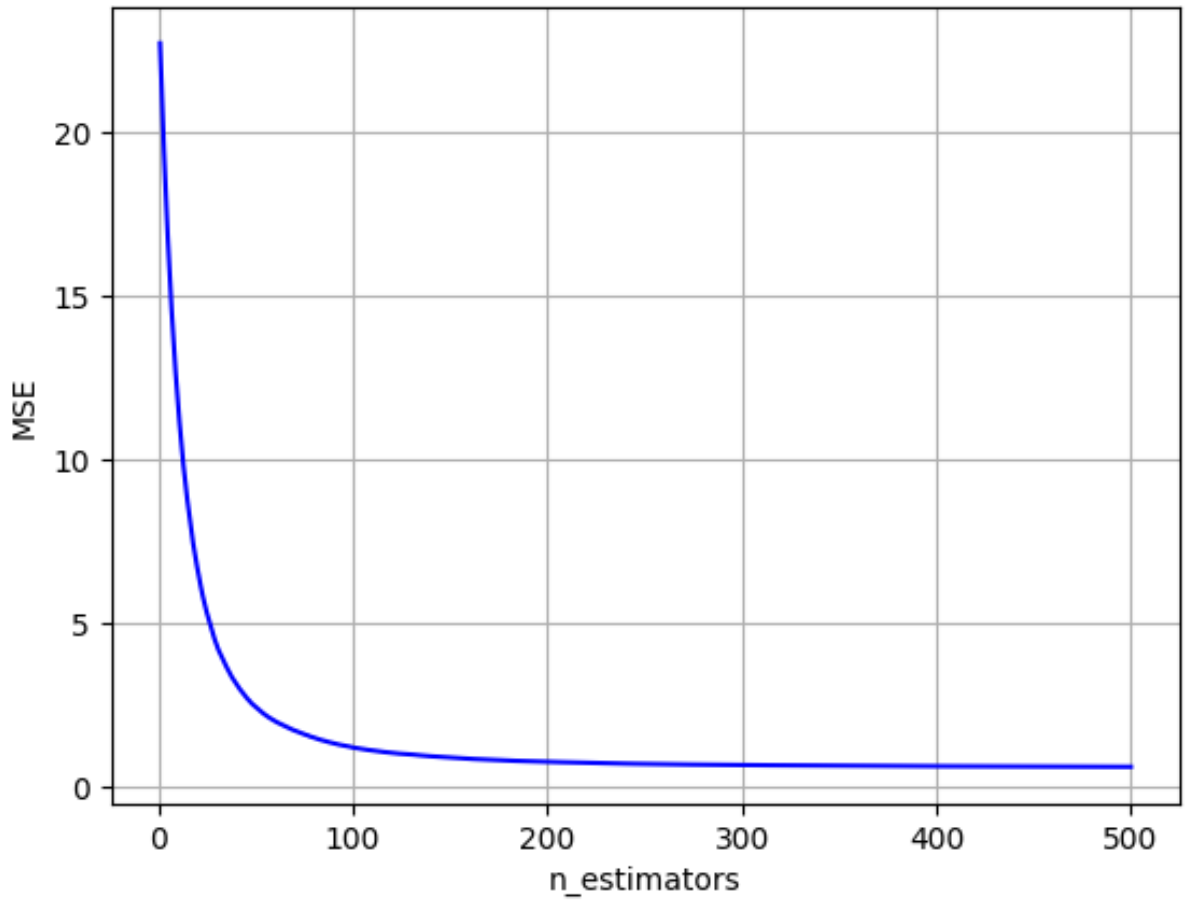

Figure 15: No. of Estimators vs. MSE in XGB-Regressor

- **Results Summary**

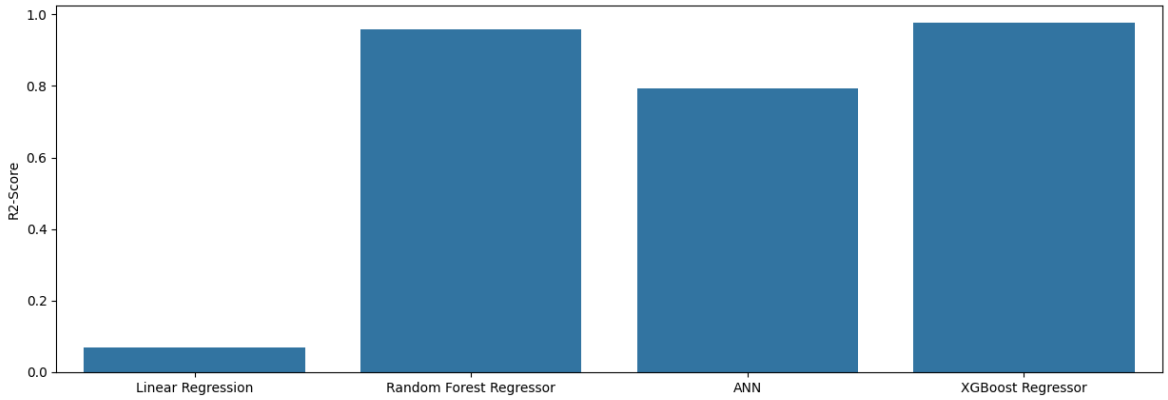

Figure 16: Comparison of R2-Score

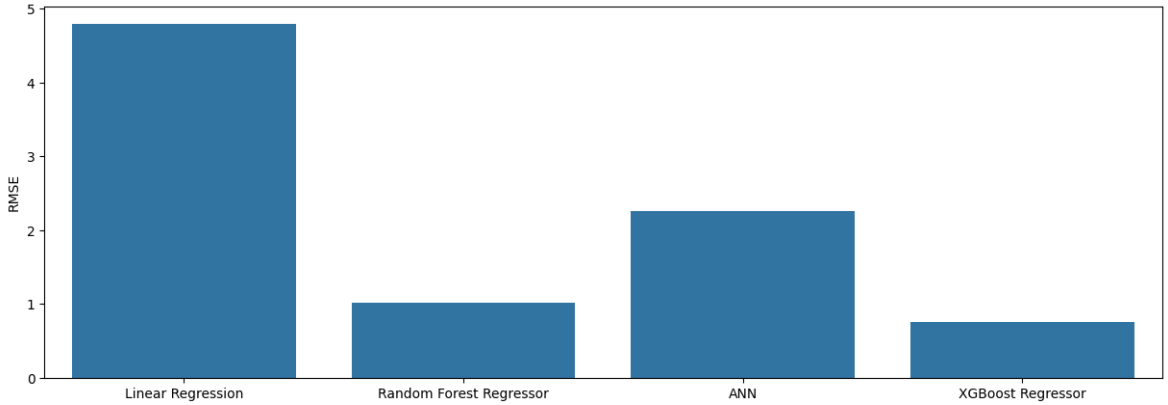

Figure 17: Comparison of RMSE

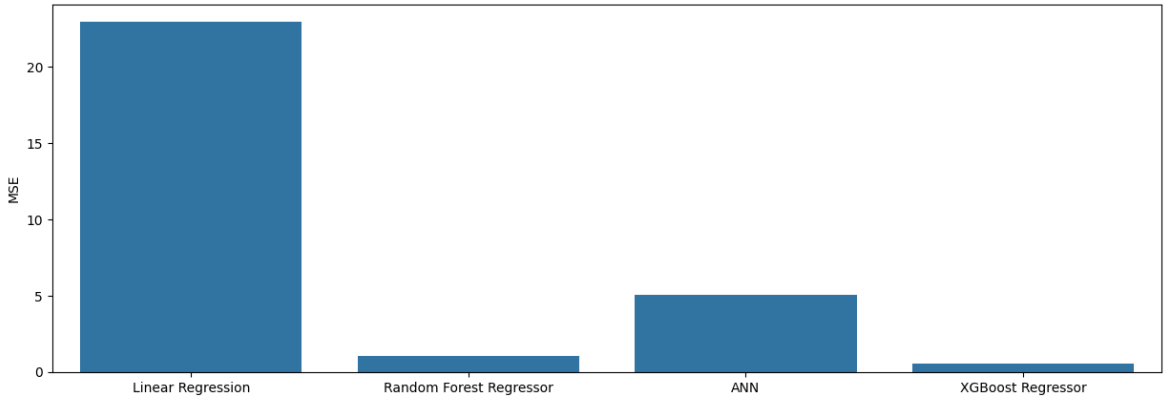

Figure 18: Comparison of MSE

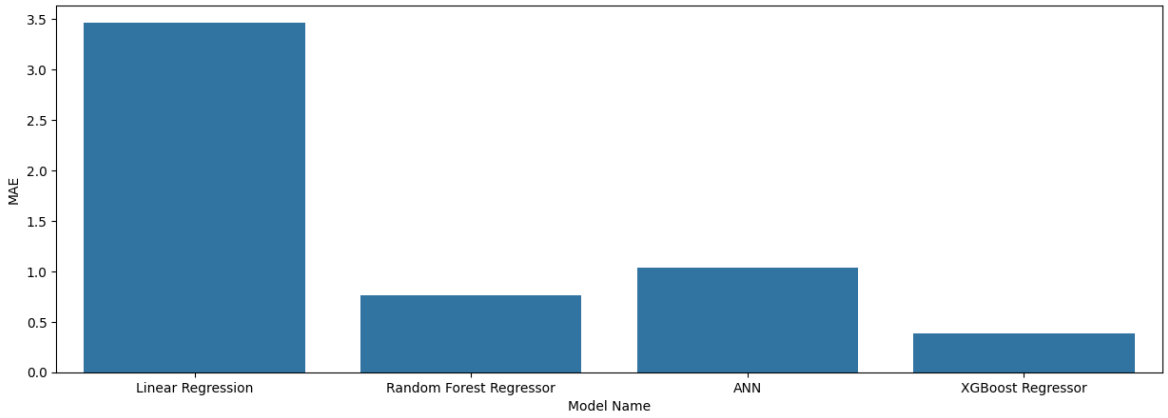

Figure 19: Comparison of MAE

Chapter 6: Conclusion

The analysis of flight delays using data-driven techniques has provided valuable insights into the factors contributing to delays and the potential for predictive modeling to enhance airline operations. Through Exploratory Data Analysis (EDA), visualization, and machine learning techniques, we identified key delay patterns, assessed their impact, and developed models to forecast delays effectively. Data visualization techniques further helped in understanding temporal and geographical delay trends. The removal of outliers, encoding of categorical features, scaling of numerical features and Principal Component Analysis (PCA) on the dataset helped to reduce the complexity of the data. Hence, the models employed in this study were able to converge faster than expected and generalize better.

Out of the four models trained namely – Linear Regression, Random Forest, ANN and XG-Boost, the two best models obtained are **Random Forest** with a RMSE of 1.0097 and R2-Score of 0.9586, and **XG-Boost** with a RMSE of 0.7550 and R2-Score of 0.9768.

The results emphasize the importance of integrating machine learning into airline decision-making systems. By leveraging predictive analytics, airlines can optimize resource allocation, reduce operational costs, and enhance passenger satisfaction. Future research should focus on real-time delay prediction using deep learning models and incorporating additional features such as passenger demand, airline policies, and external disruptions.

In conclusion, the combination of statistical analysis and advanced machine learning techniques provides a robust framework for understanding and mitigating flight delays. The ongoing advancements in AI and big data analytics will further refine predictive capabilities, paving the way for smarter and more efficient air travel systems.

References

[1] J. Pineda-Jaramillo, C. Munoz, R. Mesa-Arango, C. Gonzalez-Calderon, and A. Lange, “Integrating multiple data sources for improved flight delay prediction using explainable machine learning,” *Research in Transportation Business & Management*, vol. 56, pp. 101161–101161, Jun. 2024, doi: 10.1016/j.rtbm.2024.101161.

[2] Seyedmirsajad Mokhtarimousavi and A. Mehrabi, “Flight delay causality: Machine learning technique in conjunction with random parameter statistical analysis,” *International Journal of Transportation Science and Technology*, vol. 12, no. 1, pp. 230–244, Mar. 2022, doi: 10.1016/j.ijtst.2022.01.007.

[3] R. Henriques and Inês Feiteira, “Predictive Modelling: Flight Delays and Associated Factors, Hartsfield–Jackson Atlanta International Airport,” *Procedia Computer Science*, vol. 138, pp. 638–645, Jan. 2018, doi: 10.1016/j.procs.2018.10.085.

[4] D. Kansal, “Flight Take Off Data - JFK Airport,” *Kaggle.com*, 2019. https://www.kaggle.com/datasets/deepankurk/flight-take-off-data-jfk-airport (accessed Apr. 03, 2025).

[5] “NumPy documentation — NumPy v2.2 Manual,” *Numpy.org*, 2024. https://numpy.org/doc/stable/ (accessed Apr. 03, 2025).

[6] “Pandas documentation — pandas 2.2.3 documentation,” *Pydata.org*, 2024. https://pandas.pydata.org/docs/ (accessed Apr. 03, 2025).

[7] “Matplotlib documentation — Matplotlib 3.10.1 documentation,” *Matplotlib.org*, 2025. https://matplotlib.org/stable/index.html (accessed Apr. 03, 2025).

[8] “Seaborn: statistical data visualization — seaborn 0.13.2 documentation,” *Pydata.org*, 2024. https://seaborn.pydata.org/ (accessed Apr. 03, 2025).

[9] Scikit-learn, “scikit-learn: Machine Learning in Python,” *Scikit-learn.org*, 2024. https://scikit-learn.org/stable/

[10] “XGBoost Python Package — xgboost 3.0.0 documentation,” *Readthedocs.io*, 2022. https://xgboost.readthedocs.io/en/release_3.0.0/python/index.html (accessed Apr. 03, 2025).

[11] “TensorFlow,” *TensorFlow*, 2025. https://www.tensorflow.org/ (accessed Apr. 03, 2025).