FLIGHT DELAY PREDICTION USING MACHINE LEARNING AND DEEP LEARNING MODELS

This Mini Project Report submitted in the partial fulfilment of the requirement for the award of the degree of

MASTER OF SCIENCE IN DATA SCIENCE

Submitted by,

PRITHIKA K

(Reg. No.23CSEJ15)

Under the Guidance of

Dr. R. PORKODI, MCA., Ph.D.,

Professor

DEPARTMENT OF COMPUTER SCIENCE

BHARATHIAR UNIVERSITY

COIMBATORE-641046

November 2024

CERTIFICATE

This is to certify that the mini-project entitled **'FLIGHT DELAY PREDICTION USING MACHINE LEARNING AND DEEP LEARNING MODELS'** submitted to the Department of Computer Science, Bharathiar University in partial fulfilment of the requirement for the award of the M.Sc. in Data Science, is a bonafide record of original research work done by **PRITHIKA K** during the period of her study under my supervision and guidance during the year 2023-2025.

[Dr. R. PORKODI] **[Dr. E. CHANDRA]**

Project Guide Professor & Head

Submitted to the Project Viva-Voice held on _____

Internal Examiner **External Examiner**

DECLARATION

I do hereby declare that the project work entitled **'FLIGHT DELAY PREDICTION USING MACHINE LEARNING AND DEEP LEARNING MODELS'** submitted to the Department of Computer Science, Bharathiar University in partial fulfilment of the requirement for the award of the M.Sc. in Data Science is a record of mini project work done by me during the period of study under guidance of **Dr. R. PORKODI, MCA., Ph.D.,** Professor, Department of Computer Science, Bharathiar University.

Signature of candidate

(PRITHIKA K)

Place : Coimbatore

Date:

Countersigned by

Dr. R. PORKODI, MCA., Ph.D.,

Professor

Department of Computer Science

School of Computer Science and Engineering

Bharathiar University

Coimbatore – 641046

ACKNOWLEDGEMENT

Every project is a piece of hard work with a number of wonderful people who have always given their valuable advice or lent a helping hand. I frankly appreciate the inspiration, support and guidance of all those people who have been instrumental in making this project a great success.

I thank God Almighty for his blessings and guidance in bringing out this project successful. I would like to express sincere thanks and gratitude to my mentor and guide, **Dr. R. PORKODI, MCA., Ph.D.,** Professor, Department of Computer Science, Bharathiar University stands out as a beacon of support. Her unwavering hand-in- hand guidance helped me navigate every challenge and triumph over every obstacle.

I take this opportunity to express my thanks to **Dr.E.CHANDRA, M.Sc., M.Phil., Ph.D.,** Professor and Head, Department of Computer Science, who gave me the opportunity to do this project. And I also express my thanks to all the faculties of the Computer Science Department for providing me necessary facilities to carry out the present work and for their strong support for this project.

Last but not the least, I would like to thank my friends and family who helped me a lot in gathering different information, collecting data and guiding me from time to time in making this project. Despite their busy schedules, they generously offered their time and ideas, helping me create a project that is truly unique and meaningful. Their belief in me inspires me to continue to strive for excellence in all endeavours.

TABLE OF CONTENTS

ABSTRACT

Airline delay prediction is an increasingly critical area in the aviation industry, driven by the need to minimize operational disruptions and enhance passenger satisfaction. Flight delays can result in significant economic losses, reduced customer trust, and logistical challenges for airlines. Predicting delays with accuracy not only helps airlines optimize their schedules and resource allocation but also empowers passengers by providing reliable information for travel planning. With the rapid growth in air travel demand and the complexity of contributing factors such as weather conditions, air traffic, and carrier performance, leveraging advanced predictive analytics has become essential. Machine learning and deep learning techniques offer powerful tools for analyzing large volumes of data to detect patterns and predict potential delays before they occur. This enables more proactive measures to mitigate impacts and streamline operations, ensuring a smoother experience for all stakeholders involved in air travel.

This project explores flight delay prediction using models like Decision Tree, Random Forest, XGBoost, AdaBoost, PMIL, Attention Mechanism, and an enhanced DeepONet architecture. The study employs a comprehensive dataset including flight details, weather conditions, and operational data. Data preprocessing included handling missing values, label encoding, scaling, Box-Cox transformations for normalization, and exploratory data analysis to identify patterns. Models were evaluated using accuracy, F1-score, and confusion matrices.

The experimental results revealed that the XGBoost model achieved a high accuracy of approximately 98.93%, outperforming other traditional algorithms. The DeepONet model, specifically designed for capturing complex feature interactions, yielded an accuracy of 93.71%, showcasing its effectiveness in more nuanced predictions. Overall, the findings underscore the importance of combining data preprocessing, feature engineering, and advanced deep learning architectures to enhance flight delay predictions. This approach can significantly aid airlines in proactive decision-making and improving operational efficiency. The prediction system using XGBoost features an interactive interface where users can input flight details and receive delay predictions, providing a practical tool for travelers, airlines, and airport authorities to better manage flight schedules..

Keywords: Flight delay prediction, Machine learning, Deep Learning, Data Preprocessing, Model evaluation, Accuracy metrics, Predictive Interface.

CHAPTER 1

INTRODUCTION

1.1 BASIC INTRODUCTION

Flight delays pose a significant challenge to the aviation industry in India, impacting the efficiency of operations and passenger satisfaction. In recent years, reports have shown that flight delays are prevalent across major Indian airports, particularly during peak travel seasons. For instance, accordingto data from the Directorate General of Civil Aviation (DGCA), a substantial percentage of domestic flights in India experience delays exceeding 15 minutes, especially during monsoon months when weather disruptions are frequent.

The economic impact of these delays is profound. Industry estimates suggest that flight delays contribute to increased operational costs for airlines, with millions of rupees spent annually on compensating passengers and managing disrupted schedules. Additionally, these delays undermine passenger trust and can lead to a decline in ticket sales and overall customer loyalty. The cost implications for the Indian economy are considerable, mirroring trends seen in other large markets.

The causes of flight delays in India are varied, encompassing weather-related issues such as heavy rainfall and fog, air traffic congestion in metropolitan hubs like Delhi and Mumbai, and infrastructural constraints at airports. Technical and operational challenges, including crew availability and maintenance issues, also contribute to the frequency of delays. Given the rapid growth of the Indian aviation sector and its role as one of the fastest-growing markets globally, the development of accurate and efficient predictive models for flight delays is essential. Such models can help airlines optimize flight schedules, better manage resources, and enhance the overall travel experience for passengers, ultimately supporting a more resilient air transportation network in India.

Predicting flight delays, however, is a complex problem due to the multitude of factors involved. These factors include operational elements such as departure and arrival times, carrier characteristics, and airport ratings, as well as environmental variables like temperature, humidity, and even snowfall. Additionally, understanding the relationships between these diverse variables and how they collectively contribute to delays is essential to building a predictive model that is both accurate and reliable.

Historically, flight delay prediction has been approached through various machine learning methods, including decision trees, Bayesian networks, and ensemble classifiers. While these models offer initial insights, they often struggle to capture the complex dependencies in flight data, such as temporal and spatial interactions. Recent advancements have explored deep learning techniques like Graph Convolutional Networks (GCNs) and Recurrent Neural Networks (RNNs) to address these challenges, demonstrating improved capabilities in managing complex data structures.

This project leverages machine learning and deep learning algorithms to predict flight delays byanalyzing and modeling these variables. A variety of algorithms, including traditional classifiers such as Decision Trees, Random Forest, and XGBoost, AdaBoost, along with advanced deep learning architectures like DeepONet and Attention-based models, are explored to capture patterns in the dataand improve prediction accuracy. Each model is evaluated to determine its suitability and effectiveness for the task of predicting flight delays.

The Flight Delay Prediction System developed in this project offers a user-friendly interface that enables users to input key flight details and receive real-time predictions about potential delays. By providing such a tool, this project contributes to reducing the uncertainties surrounding flight delays, ultimately benefiting both passengers and industry stakeholders.

1.2 DATASET

The Flight Delay Prediction project utilizes two main datasets: a flight dataset and a weather dataset, both sourced from publicly available GitHub repositories. The flight dataset https://github.com/Devvrat53/Flight- Delay-Prediction/blob/master/Data/flight_data.csv includes 14,952 records with 38 features, detailing flight specifics such as departure times, delays, and carrier metrics. The weather dataset available in https://github.com/Devvrat53/Flight-Delay-Prediction/blob/master/Data/weather.csv provides data on temperature, humidity, wind speed, and precipitation. These datasets were merged using Python based on date-time alignment to form a comprehensive dataset (flight.csv). Preprocessing included handling missing values and standardizing features, ensuring the data is suitable for predictive modeling by capturing both operational and weather-related factors. By integrating these variables, the project aims to leverage real-world data to build a robust predictive model, allowing insights into delay patterns. This approach provides a foundation for improving flight scheduling and operational efficiency.

1.3 OBJECTIVE

The main objective of the Flight Delay Prediction project is to build a robust predictive system using historical flight data, carrier performance metrics, and weather information to accurately forecast delays. The project aims to provide actionable insights that help improve decision-making for airlines and passengers, enabling better planning and reducing unexpected disruptions.

Key Objectives:

- Develop a system to predict flight delays and estimate their duration.
- Leverage historical data and weather conditions to identify delay patterns.
- Provide practical insights for optimizing airline operations and enhancing passenger experience.

1.4 SCOPE

The scope of the project encompasses the entire process of building a flight delay prediction system, from data gathering to deployment. It integrates diverse datasets and utilizes advanced machine learning techniques for accurate forecasting. The focus is on creating a scalable and efficient model that can be applied in real-world scenarios, providing valuable insights for airlines and passengers.

- Comprehensive data collection and preprocessing, combining flight operational details, weather data, and airline performance metrics.
- Integration of various predictive models, including traditional machine learning algorithms like Decision Tree, Random Forest, XGBoost, and AdaBoost.
- Advanced deep learning models like DeepONet and Attention-based architectures is also integrated.
- Feature engineering and exploratory data analysis (EDA) to optimize model input and gain insights.
- Rigorous model evaluation using metrics such as accuracy, precision, recall, and F1-score for robust performance assessment.
- Designing an interactive interface for potential real-time decision-making, offering seamless integration into airline and dashboard for improved operational efficiency.

CHAPTER 2

LITERATURE REVIEW

Bisandu et al. (2024) [1] introduced a novel approach using a Deep Operator Network combined with agradient-mayfly optimization algorithm to predict flight delays. This method leverages the Deep Operator Network's ability to capture non-linear relationships between features while optimizing model performance through the gradient-mayfly algorithm, which enhances convergence speed and model accuracy. The study, conducted at the Artificial Intelligence and Scientific Computing Laboratory at Cranfield University, demonstrated significant improvements in delay prediction, especially underconditions of high data complexity, such as unpredictable weather patterns and varying airport congestion. By integrating a specialized optimization technique, this approach effectively addresses the limitations of traditional deep learning models in high-dimensional flight data contexts.

Kim et al. (2024) [2] from the Aerospace Systems Design Laboratory at Georgia Institute of Technology explored deep learning techniques specifically tailored for flight delay prediction. The teamdeveloped a deep neural network model that incorporates key factors such as historical flight performance, airport traffic, and meteorological data to capture temporal dependencies in delay patterns. Their approach utilizes time-series modeling techniques within the neural network framework, which allows the model to learn sequential dependencies critical to accurately predicting delays. The results showed that this deep learning approach outperforms conventional models, particularly in high-traffic scenarios and during adverse weather events.

Jha et al. (2023) [3] proposed a hybrid machine learning model combining Decision Trees, SVM, and XGBoost for predicting delays in U.S. airlines. This ensemble approach leverages the interpretability of Decision Trees, the accuracy of SVM, and XGBoost's ability to handle imbalanced data, providing a robust and comprehensive prediction tool for various delay causes.

Dai (2023) [4] proposed a hybrid model for flight delay prediction that leverages big data techniques to process large-scale aviation data. The model combines machine learning algorithms with real-time data on airport conditions, airline schedules, and weather fluctuations, effectively handling complex, high-volume data. This approach enhances prediction accuracy and reduces latency in forecasting flight delays**.**

Santos et al. (2023) [5] developed a predictive model using gradient-boosting algorithms, whichincorporates factors such as weather conditions, traffic congestion, and flight schedules. By using feature engineering to create composite weather-related features, the study demonstrated improved predictive accuracy over linear regression models. The gradient-boosted model was particularly effective in reducing false positives for delayed flights, providing a valuable tool for decision-makers in the aviation industry.

Lee et al. (2023) [6] explored deep learning methods for flight delay prediction, applying a recurrent neural network (RNN) architecture to capture temporal patterns in flight schedules and environmental conditions. Their model leveraged long short-term memory (LSTM) networks to track sequential dependencies between past and future delays. Results showed that the LSTM-based model outperformed traditional machine learning techniques, particularly in cases involving significant delays, highlighting the model's capacity to learn from sequential data in a way that more static models cannot.

Chen et al. (2024) [7] compared multiple machine learning algorithms, including decision trees,random forests, and XGBoost, for delay prediction in domestic flights. XGBoost was found to provide the highest predictive accuracy due to its ability to handle imbalanced data and complex, non-linear relationships. This study also emphasized the importance of integrating real-time data such as weather updates and airport congestion statistics to increase prediction accuracy in rapidly changing conditions.

Patel and Brown et al. (2024) [8] introduced an attention-based model that assigns higher weights to features most likely to impact delays, such as airport congestion levels and extreme weather conditions. Their approach enhances interpretability by showing which factors contributed most to a given delay prediction. The model outperformed conventional methods by focusing on feature importance, thereby producing more accurate and interpretable results for end-users.

Singh and Gupta et al. (2024) [9] investigated ensemble learning techniques, combining outputs from multiple machine learning models to generate a more robust prediction. By integrating predictions from logistic regression, random forest, and XGBoost models, their ensemble approach reduced the variance and bias typically associated with individual models, achieving improved accuracy for delay prediction across diverse weather and operational scenarios.

Kiliç and Sallan (2023) [10]: This study utilized AI and ML models to predict arrival flight delays within the U.S. airport network. The authors evaluated several ML models, concluding that Gradient Boosting Machines outperformed others like logistic regression and random forests for predicting flight delays.

Zhao et al. (2024) [11] proposed a novel application of graph neural networks (GNNs) for flight delay prediction, focusing on the complex network structure of flight routes, airport connectivity, and passenger flow. In this study, Zhao and colleagues modeled the flight network as a graph, where nodes represented airports and edges represented direct flight routes. The GNN was trained on historical delay data, weather information, and airport congestion levels, enabling it to capture the relational dependencies between connected flights and airports. Their findings indicated that the GNN model was particularly effective at identifying cascading delays across connected routes, such as when an initial delay at a hub airport led to subsequent delays across multiple connecting flights. This approach provided superior predictive accuracy compared to traditional machine learning models, particularly for predicting delays under high-traffic conditions or when multiple flights were interconnected through large hub airports. The study demonstrated that GNNs could effectively capture both spatial andtemporal dependencies in flight delay prediction, offering a promising new direction for handling network-based aviation data.

Cai et al. (2015), Chen et al. (2009), Zhang et al. (2019), and Qu et al. (2020) [12] conducted extensive research on flight delay prediction, with a focus on accurately estimating arrival and departuretimes. Their studies emphasized the importance of prediction accuracy for improving decision-making across the entire operational system of airlines and airports. The researchers frequently utilized classical statistical models such as Naïve Bayes, Support Vector Machine (SVM), Logistic Regression, multivariate regression, and time series forecasting.

Liu et al. (2018) and Huo et al. (2020) [13] highlighted the limitations of standard prediction models inhandling intricate time series data that required sophisticated fusion procedures. To advance beyond these limitations, **Lu et al. (2021) [14]** introduced DeepONet, a novel model designed to enhance prediction performance through deep learning techniques. This approach incorporated real-world data and was augmented with optimization algorithms to minimize error, showcasing its potential in accurately predicting delays with dynamic input functions.

Güvercin et al. (2020) [15] explored a clustered airport modeling approach for predicting delays within airport networks. This method demonstrated strong predictive performance by considering the interactions between airports, but it lacked a comprehensive feature selection process, which limited its predictive capabilities

Ma et al. (2023) [16] attempted to address this limitation by including more detailed features related to strategic airport scheduling. However, despite the enhanced feature set, their model did not achieve significant improvements in accuracy, indicating the need for more sophisticated feature engineering techniques.

Chen et al. (2017) [17] proposed an Information Gain-SVM approach to evaluate flight efficiency under environmental constraints, such as CO2 emissions. This model enhanced the classical Data Envelopment Analysis (DEA) technique by incorporating additional efficiency metrics but failed to offer broader policy implications for environmental sustainability

Chen and Li (2019) [18] built upon this work by developing a machine learning framework aimed at precise delay prediction. However, the framework was limited by the lack of a robust, large-scale database, which hindered its maximum predictive potential.

He et al. (2022) [19] utilized artificial neural networks (ANN) and multilayer perceptrons (MLP) for flight delay prediction, leveraging chronological flight data to improve the efficiency of the models. Although this approach resulted in better accuracy compared to earlier models, it was not optimized for real-time data applications, limiting its use in dynamic scenarios

Yi et al. (2021) [20] addressed real-time challenges by developing a stacking classification model that incorporated the Boruta algorithm for feature selection. This method effectively managed imbalanced datasets but did not integrate other machine learning techniques, which could have provided a more comprehensive learning framework.

Guo et al. (2021) [21] presented a hybrid model combining Random Forest Regression with the Maximal Information Coefficient (RFR-MIC) to enhance flight delay prediction accuracy. While this approach showed promising results, it lacked integration with cloud or parallel computing capabilities, resulting in longer computation times.

Shao et al. (2022) [22] introduced TrajCNN, a novel vision-based end-to-end model that utilized situational maps for analyzing spatial and temporal data. Although effective for short-term predictions, TrajCNN's application was limited due to its focus on localized, short-term delays, reducing its generalizability for long-term forecasting.

Bisandu et al. (2022) [23] advanced flight delay forecasting by employing a novel technique involving social ski drivers and conditional autoregressive value-at-risk models. This approach outperformed existing meta-heuristic methods, showing enhanced accuracy. However, the authors suggested the need for comparative studies with other optimization techniques to validate its performance further. In their earlier work (Bisandu et al., 2021), the authors utilized a deep feedforward network, which significantly outperformed simpler models such as SVMs and single-layer neural networks, achieving higher predictive accuracy.

Ayoubi (2018) [24] experimented with deep learning architectures, specifically Long Short-Term Memory (LSTM) and Recurrent Neural Networks (RNN), utilizing threshold sequences for delay prediction. The LSTM model effectively captured temporal patterns but required extensive parameter tuning.

Lin et al. (2017) [25] extended this approach by employing convolutional LSTM (Conv-LSTM) to model both temporal and spatial aspects of the data, highlighting the importance of deep learning parameter optimization for handling large-scale datasets effectively.

Divya et al. (2023) [26] focused on the financial implications of flight delays for airlines and passengers. Their study demonstrated that artificial neural networks (ANNs), particularly when optimized using the Adam algorithm, achieved high accuracy in predicting delays based on sequential data. The researchers proposed a hybrid model, GANN, which combined ANN with a genetic algorithm(GA), resulting in an accuracy of 89%. This model showed potential for practical applications, particularly in business analytics related to flight delay management.

Tirtha et al. (2023) [27] introduced a spatiotemporal deep learning model tailored for predicting delays across multiple airports, capturing both spatial and temporal correlations using adaptive causalitygraphs. Their experiments on data from China's busiest airports demonstrated a 4.7% improvement in performance over existing benchmarks, indicating the effectiveness of their approach in complex, multi-airport scenarios.

CHAPTER 3
BACKGROUND STUDY

Flight delays are a persistent issue in the aviation industry, causing significant inconvenience for passengers and leading to substantial financial losses for airlines. The increasing volume of air traffic has put pressure on airlines and airport authorities to minimize delays and improve operational efficiency. Traditional methods of delay prediction have relied on statistical techniques, which often fallshort due to the complex, multi-factor nature of delays. With advancements in data analytics, machine learning models provide a promising approach to predict flight delays more accurately, leveraging a wider range of data sources and sophisticated algorithms to offer actionable insights.

3.1 CURRENT CHALLENGES

Flight delay prediction poses several challenges. One of the primary issues is the quality and completeness of the data, as inconsistent or missing values can significantly affect model training. The complex interplay between various factors, such as weather conditions, air traffic, and airline-specific delays, complicates the prediction process. Additionally, scalability and generalization of the model across different airports and regions are challenging, as delays are influenced by localized factors that may not be captured in the training data. Ensuring real-time performance and handling dynamic conditions like sudden weather changes further complicate the process.

3.2 EXISTING SOLUTIONS

Various approaches have been employed to tackle the problem of flight delay prediction. Traditional statistical models, like linear regression and logistic regression, are simple but often fail to capture the non-linear relationships in the data. More advanced machine learning models, such as Decision Trees, Random Forests, and Gradient Boosting Machines (GBM), have shown improved predictive performance by effectively handling complex interactions between features. These models can incorporate a variety of features, including weather data, flight history, and airline-specific characteristics, but may still face limitations in terms of overfitting and interpretability. Although some commercial solutions integrate machine learning techniques, many remain proprietary, and their methodologies are not fully transparent.

3.3 PROPOSED APPROACH

This project proposes a robust flight delay prediction system that employs a combination of machine learning and deep learning techniques. The approach begins with extensive data preprocessing, including handling missing values, encoding categorical features, and applying scaling transformations like the Box-Cox method to normalize skewed data. A unified dataset is created by merging historical flight and weather data, enhancing feature richness.

The core model development involves testing various algorithms, including Decision Trees, Random Forest, XGBoost, AdaBoost, and deep learning models like DeepONet and Attention-based networks. XGBoost was selected as the primary model for interpretation due to its superior performance in handling complex, non- linear data interactions and its capability to address class imbalance. Additionally, AdaBoost was used to refine predictions by focusing on misclassified instances, enhancing overall accuracy

DeepONet, with its dual-branch architecture, was employed to capture intricate dependencies between independent and context-specific features, such as weather conditions and flight timings. The attention mechanism in the deep learning models further improved feature selection, dynamically prioritizing influential inputs during training.

The proposed solution also includes an interactive interface for real-time predictions, allowing users to input key flight details, such as departure time, carrier, and temperature, and receive immediate delay forecasts. The model outputs are designed to be transparent, providing users with clear insights into the factors contributing to the predicted delay.

CHAPTER 4
METHODOLOGY

This project involves a structured approach to developing a flight delay prediction system using machine learning and deep learning models. Fig 1 shows the methodology of developing a Flight Delay prediction System. The process includes data collection and preprocessing, exploratory data analysis, model selection, training, evaluation, interpretation and dashboard creation.

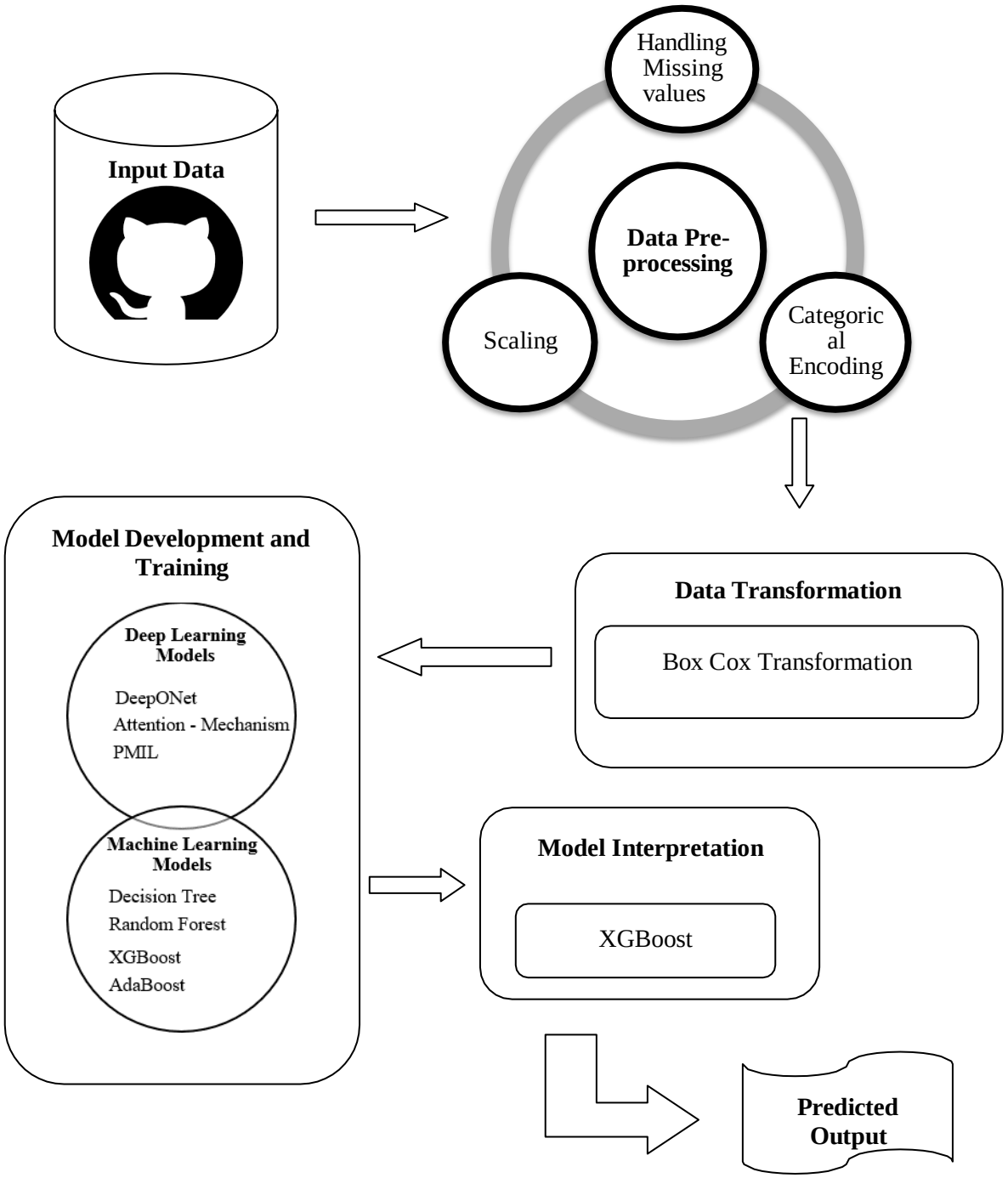

Fig 1 : Methodology of Flight Delay Prediction System

4.1 DATA COLLECTION

The data for this Flight Delay Prediction project was sourced from two primary datasets: one focused onflight operational details and the other on weather conditions. These datasets were collected frompublicly available GitHub repositories to ensure that the prediction model could be built on real-world, relevant data. The flight dataset, which can be accessed from https://github.com/Devvrat53/Flight-Delay-Prediction/blob/master/Data/flight_data.csv, includes extensive information about flights, including their departure and arrival times, delays, carrier ratings, and airport information. The weatherdataset, from https://github.com/Devvrat53/Flight-Delay-Prediction/blob/master/Data/weather.csv, contains environmental data such as temperature, humidity, wind speed, and other meteorologicalconditions that may impact flight operations. These two datasets were then merged to create a single, comprehensive dataset (flight.csv) using Python code by converting the data to date-time column. Themerging process was based on aligning flight records with the nearest corresponding weatherconditions, ensuring that both operational and environmental factors were considered when analyzingflight delays.

4.1.1 DATASET OVERVIEW

The final merged dataset, flight.csv, contains a total of 14,952 records with 38 features. These features represent a wide array of information regarding both flight operations and weather conditions. The flight dataset includes key details about each flight, such as Date, Departure Airport, Arrival Airport, Carrier, Expected Departure Time, Departure Time, Departure Delay, Arrival Time, Carrier Market Share, and Carrier On-Time Performance Rating. This dataset captures both scheduled and actual flight details, along with performance metrics for airlines and airports. The weather dataset provides crucial information about atmospheric conditions, including features like totalSnow_cm, DewPointC, WindGustKmph, cloudcover, humidity, precipMM, pressure, tempC, visibility, winddirDegree, and windspeedKmph. These weather-related features are essential for understanding how different environmental factors may contribute to delays in flight operations. By merging these two datasets, we now have a unified dataset that incorporates both operational flight data and real-time weather information, providing a comprehensive set of features for building and optimizing a flight delay prediction model. Fig 2 shows the top five columns in the merged dataset, flight.csv

	Date	Departure Airport	Arrival Airport	Expected Departure Time	Departure Time	Departure Delay	Duration	Expected Arrival Time	Arrival Time	Arrival Time Delay	...	WindGustKmph	cloudcover	humidity	precipMM	pressure	tempC	v
0	2018-01-28	BLR	DEL	6:10	6:10	0:00:00	2:20	8:55	8:30	-0:25:00	...	22.0	5.0	45.0	0.0	1014.0	22.0	
1	2018-01-28	DEL	HYD	7:20	7:54	0:34:00	1:42	9:30	9:36	0:06:00	...	10.0	0.0	24.0	0.0	1014.0	22.0	
2	2018-01-28	CCU	DEL	7:10	7:19	0:09:00	2:11	9:25	9:30	0:05:00	...	7.0	0.0	47.0	0.0	1013.0	17.0	
3	2018-01-28	BLR	DEL	7:00	6:59	-0:01:00	2:22	9:40	9:20	-0:20:00	...	22.0	5.0	45.0	0.0	1014.0	22.0	
4	2018-01-28	CCU	DEL	7:40	8:04	0:24:00	2:18	10:15	10:23	0:08:00	...	7.0	0.0	47.0	0.0	1013.0	17.0	

Fig 2

4.1.2DATASET DESCRIPTION

COLUMN NAME	DATA TYPE	DESCRIPTION
Date	object	The date of the flight
Departure Airport	object	IATA code of the departure airport.
Departure Airport Rating (out of 10)	float64	Overall rating of the departure airport (0-10 scale).
Departure Airport On Time Rating(out of 10)	float64	On-time performance rating of the departure airport (0-10 scale).
Departure Airport Service Rating (out of 10)	float64	Service quality rating of the departure airport (0-10 scale).
Arrival Airport	object	IATA code of the arrival airport.
Arrival Airport Rating (out of 10)	float64	Overall rating of the arrival airport (0-10 scale).
Arrival Airport On Time Rating (outof 10)	float64	On-time performance rating of the arrival airport (0-10scale).
Arrival Airport Service Rating (out of 10)	float64	Service quality rating of the arrival airport (0-10 scale).
Airplane Type	object	Type or model of the aircraft used for the flight.

Departure Time	object	Actual departure time of the flight
Expected Departure Time	object	Scheduled departure time ofthe flight.
Departure Delay	object	Duration of the delay in departure, if any
Duration	object	Total duration of the flight in minutes.
Expected Arrival Time	object	Scheduled arrival time of the flight.
Arrival Time	object	Actual arrival time of the flight
Arrival Time Delay	object	Duration of the delay in arrival, if any
Carrier	object	Name or code of the airline operating the flight.
Carrier Rating (out of 10)	float64	Overall rating of the airline carrier (0-10 scale).
Carrier Market Share (out of 100)	float64	Market share percentage of the carrier (0-100 scale).
Carrier Load Factor (out of 100)	float64	Average load factor of the airline (0-100 scale).
Carrier On Time Performance Rating (out of 100)	float64	On-time performance rating of the carrier (0-100 scale).
totalSnow_cm	float64	Amount of snowfall in centimeters during the flight's operation.
DewPointC	float64	Dew point temperature in degrees Celsius.
WindGustKmph	float64	Maximum wind gust speed in kilometers per hour.
cloudcover	float64	Percentage of cloud cover during the flight.
pressure	float64	Atmospheric pressure in hPa(hectopascals)

visibility	float64	Visibility in kilometers.
tempC	float64	Temperature in degrees Celsius.
winddirDegree	float64	Wind direction in degrees
windspeedKmph	float64	Wind speed in kilometers per hour
humidity	float64	Humidity level as a percentage.
precipMM	float64	Total precipitation in millimeters

4.2 DATA PREPROCESSING

Data preprocessing involves preparing raw data into a clean, structured format suitable for machine learning. This includes handling missing values, encoding categorical features, sctandardizing the numerical features and transforming the data. These steps enhance data quality, improve model training,and lead to better prediction accuracy.

4.2.1 HANDLING MISSING VALUES

Handling missing values is a crucial step in transforming raw data into a structured format for machine learning. This process involves addressing gaps in the dataset to ensure data consistency and reliability. For numerical columns, missing values are imputed using the column mean, providing a balanced approach that preserves the distribution of data. For categorical columns, missing values are filled with the most frequent category to maintain data completeness. These strategies help ensure the dataset remains robust and suitable for training machine learning models without introducing biases or inconsistencies.

4.2.2 CATEGORICAL ENCODING

Categorical features, such as carrier and weekday, were converted to numerical form using Label Encoding, which facilitated machine learning algorithms in processing these variables effectively. Each unique category within the categorical features was mapped to a unique integer value, ensuring the model could effectively process these inputs. Given a categorical feature X with N unique categories:

$$X_{encoded} = f(X) \text{ where } f(x_i) = k$$

where : $f(x_i)$ represents the encoding function that maps each unique category x_i in X to an integer k, kis an integer that ranges from 0 to N−1.

This transformation converted the non-numerical attributes into numerical representations, making themsuitable for input into the machine learning model. By encoding categorical variables, the model could better analyze relationships between these features and flight delays, contributing to more effective training and improved prediction accuracy.

4.2.3 SCALING

To ensure consistent scaling across numerical data, StandardScaler was applied, which normalizes features so they have a mean of 0 and a standard deviation of 1. This process reduces the model's sensitivity to variations in data scale and optimizes performance.

Given a numerical feature X with values $x_1, x_2, \ldots, x_n$;

$$X_{\text{scaled}} = \frac{X - \mu}{\sigma}$$

where: X_{scaled} represents the standardized value of the feature. μ is the mean of the feature X:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

σ is the standard deviation of the feature X:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$$

Applying StandardScaler ensured that numerical columns, such as temperature, humidity, and carrier ratings, were consistently scaled. This prevented features with larger scales from dominating the learning process and contributed to a more balanced input range. By standardizing these features, the model was able to process data uniformly, leading to better convergence during training and improved overall performance in predicting flight delays.

4.2.4. TRANSFORMING

To transform continuous variables and adjust their distributions closer to normality, a Box-Cox transformation was employed. This transformation is particularly beneficial for algorithms that assume a normal distribution of features, improving their performance and stability.

For a given continuous feature X with positive values:

$$X_{\text{transformed}} = \begin{cases} \frac{(X^{\lambda}-1)}{\lambda} & \text{if } \lambda \neq 0 \\ \ln(X) & \text{if } \lambda = 0 \end{cases}$$

where: $X_{transformed}$ is the new value after transformation and λ is the transformation parameter that maximizes the likelihood of obtaining normality.

The Box-Cox transformation was applied to numerical columns with positive, non-zero values, such as weather metrics (e.g., temperature, wind speed) and performance ratings. This adjustment helped in reshaping skewed data closer to a normal distribution, which aligns with the assumptions of many machine learning algorithms. The transformation enhanced the model's capability to interpret continuous features effectively, contributing to improved training dynamics and prediction accuracy.

4.2.5. TARGET TRANSFORMATION AND FEATURE ENGINEERING

The continuous target column, Arrival Time Delay, was transformed into discrete categories to simplify the model's task by focusing on predicting the severity of delays. The values were grouped into three bins: No Delay, representing flights that arrived on time; Minor Delay, for slight delays; and Major Delay, for significant delays. This categorization improved the model's interpretability and accuracy by making it easier to predict distinct delay outcomes. The categorical target was then Label Encoded to convert the string labels into numerical values, making them suitable for classification algorithms.

Feature engineering played a key role in enhancing the model's performance by creating meaningful predictors. Time-based features, such as the hour of departure and the weekday, were extracted from thetimestamp data, allowing the model to capture temporal patterns in delays. These features helped the model understand trends like delays being more frequent at certain times or days. Label Encoding was applied to categorical columns, including carriers and airports, transforming them into numerical values that could be processed by the model. Additionally, weather data was standardized by using text processing techniques like stemming and converting descriptions to lowercase. This ensured that variations in weather descriptions (e.g., "snowy" and "Snowy") were treated consistently, improving the reliability of the weather-related features. Together, these feature engineering techniques enhanced the model's ability to make accurate and reliable predictions .

4.3 EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis (EDA) is a fundamental step in understanding the dataset and uncovering patterns, trends, and insights that influence flight delays. In this project, EDA was used to examine various factors contributing to delays, including flight characteristics, weather conditions, and operational elements. The analysis focused on key features such as departure time, delay duration, carrier market share, weather conditions, and airport ratings. By exploring these features, EDA provided insights into how each factor might influence delay occurrences. Additionally, the relationships betweendifferent features, like how weather conditions impact delay durations, were examined to uncover correlations. The analysis also included investigating carrier- and airport-specific delay trends to identify operational patterns that contribute to delay dynamics in the aviation industry.

Correlation analysis played a critical role in identifying relationships between variables. A heatmap of feature correlations helped highlight key connections, such as the link between departure delays and operational factors like flight duration or carrier load factor. This enabled a deeper understanding of how various factors interact and contribute to delays. To refine the analysis further, flight delays were examined by day of the week, with a bar chart displaying delays across weekdays to identify patterns in delay frequency. Weather conditions were also closely analyzed, with scatter plots exploring how variables such as temperature, humidity, and snowfall correlate with delay durations. This analysis helped reveal the impact of extreme weather on delays.

In addition to individual feature analysis, visualizations like histograms and violin plots were used to examine delay distributions. A histogram was employed to visualize the frequency of departure delays, with a red line separating non-delayed and delayed flights, providing insights into the severity of delays.A violin plot was used to examine the distribution of departure delays across different airport ratings, highlighting potential patterns between airport ratings and delay frequency. To analyze the multivariate relationships, a pairplot was used to visualize the interactions between departure delay and other key features, such as carrier market share, temperature, humidity, and airport ratings. Finally, a regression plot analyzed the relationship between the hour of departure and average departure delay, helping to identify if certain times of day were more prone to delays. This comprehensive EDA approach provided a thorough understanding of the factors influencing flight delays, setting the stage for further model development.

4.4 MODEL SELECTION AND TRAINING

In the Model Selection and Training phase, the aim was to build a robust model capable of predicting flight delays with high accuracy. The data used included various features such as carrier details, airport specifics, weather conditions, and scheduling information. The models considered ranged from traditional machine learning algorithms to advanced deep learning architectures. Each model wasevaluated based on its performance in classifying delays into categories: No Delay, Minor Delay, and Major Delay. To start, the dataset was divided into features (X) and the target variable (y), where Arrival Time Delay served as the target for prediction. An 80-20 train-test split was used to separate the data, ensuring the training phase used 80% of the data while 20% was held back for model evaluation.

4.4.1 DEEP OPERATOR NETWORK

DeepONet is an advanced neural network architecture designed to model complex relationships by processing input features through separate but parallel sub-networks called branches and trunks. This model is particularly suited for scenarios where data interactions exhibit non-linear dependencies. The branch network handles independent features, while the trunk network processes context-dependent features. The outputs from these two components are then combined and passed through dense layers for final classification. Fig 3 shows a schematic view of the Deep Operator Network.

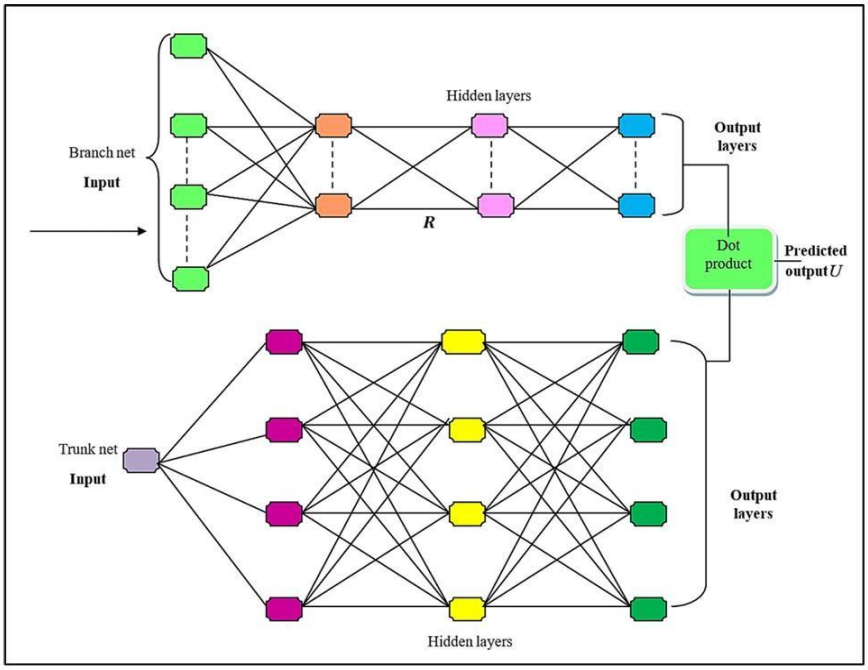

Fig 3 : Schematic view of DeepONet

DeepONet was chosen for the flight delay prediction task due to its ability to model intricate dependencies among the diverse set of features in the dataset, such as carrier details, weather conditions,and airport-specific information. The dual-branch architecture allows it to effectively capture the interactions between independent and context-specific variables, providing a comprehensive representation that supports high-accuracy predictions.

DeepONet operates by using two parallel networks—the branch network and the trunk network. The branch network processes independent features such as weather and carrier information, while the trunk network handles context-dependent features like flight timings. The outputs from these networks are combined and passed through dense layers for final processing. The model's unique architecture allowsit to capture complex interactions between different types of data. The combined output can be represented as:

$$y = \sigma\,(W_b \cdot \phi\,(\text{Branch Input}) + W_t \cdot \psi\,(\text{Trunk Input}))$$

where ϕ and ψ represent the outputs from the branch and trunk networks, and W_b and W_t are weight matrices.

In the flight delay prediction task, the DeepONet model processes input features through two distinct neural branches: the branch network and the trunk network. The branch network processes independent features such as carrier and weather conditions, while the trunk network focuses on dependent features like flight timing and airport characteristics. By combining these parallel outputs, DeepONet effectively captures complex relationships between different types of data. This combined output is passed through dense layers to generate a prediction that classifies the type of delay (e.g., No Delay, Minor Delay, Major Delay). Its architecture is well-suited for capturing the intricate interactions needed for accurate delay prediction.

4.4.2. ATTENTION - BASED MODEL

The attention-based model incorporates an attention mechanism that allows the network to focus selectively on significant input features while diminishing the influence of less relevant ones. This model evaluates feature importance dynamically during training, assigning attention scores that prioritize influential data components for enhanced decision-making. Fig 4 shows the architecture of a self - attention mechanism incorporated in a attention – based model.

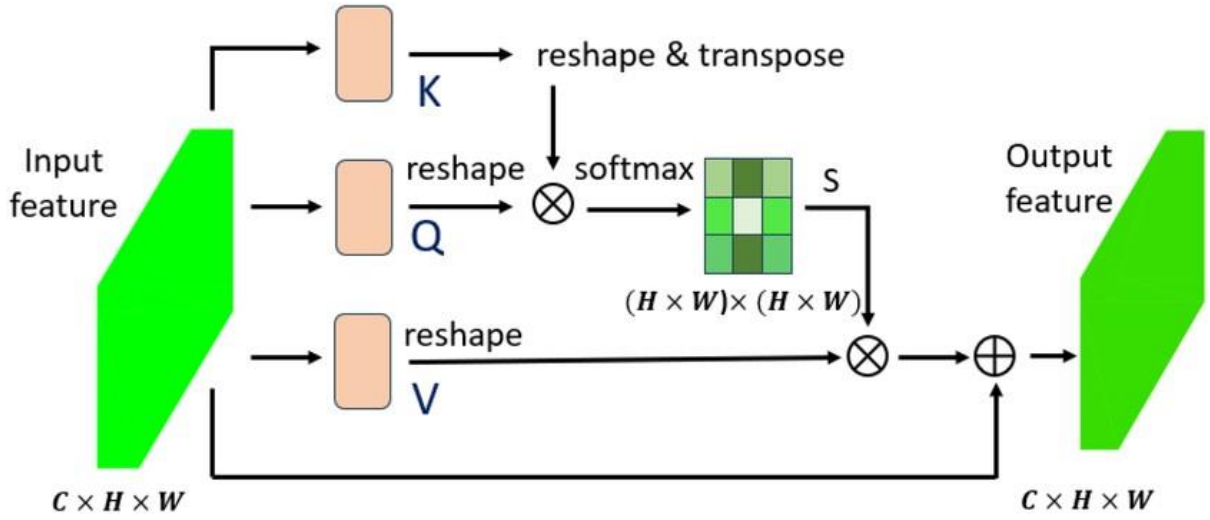

Fig 4 : Architecture of Attention Mechanism

The attention-based model was selected because of its strength in identifying key predictors in complex datasets. For flight delay prediction, where various factors such as weather conditions, carrier ratings, and airport services play a critical role, the attention mechanism helps the model distinguish and weigh these features appropriately. This approach contributes to improved accuracy by highlighting the most impactful features during training and prediction phases.

The attention-based model uses an attention mechanism to assign importance scores to input features. It computes attention weights by comparing the features with themselves or a context vector, allowing the model to focus on significant parts of the input. The attention output can be represented by:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where Q, K, and V are the query, key, and value matrices, and d_k is the dimension of the key. The attention scores enhance the interpretability of the model by highlighting which features influence the output most.

The Attention-based model enhances the delay prediction by identifying and prioritizing the most influential features in the dataset. It applies an attention mechanism that evaluates the importance of each feature and assigns corresponding attention weights. This allows the model to focus on key indicators, such as weather conditions or specific carrier performances, that are significant for predictingdelays. The attention output is processed through additional layers to make the final classification.

4.4.3. PREDICTIVE MULTILAYER PERCEPTRON (PMIL)

PMIL is a fully connected neural network composed of multiple dense layers, designed for processing input features and classifying them into distinct categories. Fig 5 shows a simple architecture of a Multilayer Perceptron in Neural Network. Each layer in the PMIL model applies an activation function to learn feature representations, and the output layer uses a softmax activation for multi-class classification.

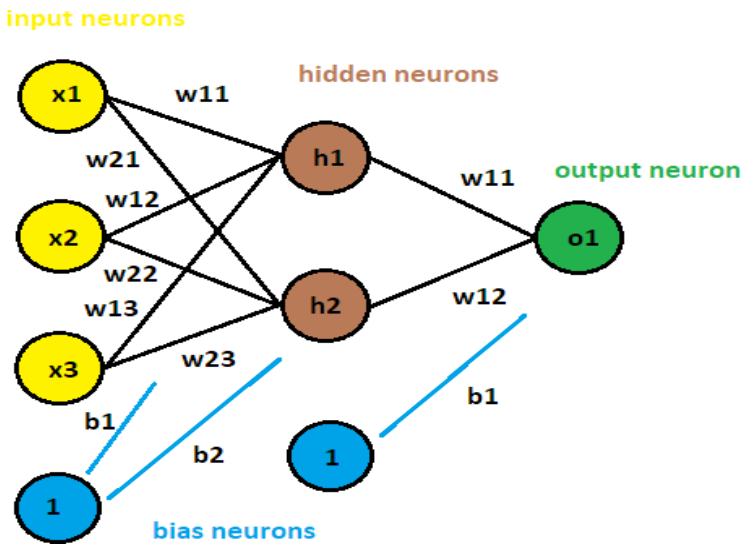

Fig 5 : Architecture of Multilayer Perceptron

PMIL was chosen for its simplicity and efficiency in handling structured data with continuous features. Given the dataset's diverse attributes, including numerical and time-based data, PMIL's architecture can effectively capture patterns and relationships through its dense layers. This makes it well-suited for the classification of flight delays into No Delay, Minor Delay, and Major Delay categories, ensuring reliable performance without overly complex computations.

PMIL is a fully connected neural network where the input data passes through multiple dense layers. Each layer applies an activation function, typically ReLU, to introduce non-linearity and learn complex relationships. The final output layer uses softmax for multi-class classification:

$$y = \text{softmax}\left(W_n \cdot \sigma\left(W_{n-1} \cdots \sigma\left(W_1 \cdot X + b_1\right) \cdots + b_{n-1}\right) + b_n\right)$$

where W are weight matrices, b are biases, and σ is the activation function.

The Predictive Multilayer Perceptron (PMIL) model employs a straightforward yet effective deep learning architecture for flight delay classification. It processes input data through dense layers, where each layer applies a non-linear activation to learn complex relationships. The final layer uses a softmax activation to output the probability distribution across the classes (No Delay, Minor Delay, Major Delay). PMIL is effective in learning from the input data and providing robust classification through its multi-layered approach.

4.4.4. DECISION TREE CLASSIFIER

The Decision Tree Classifier is a fundamental machine learning algorithm that splits data based on feature values, forming a tree structure where each node represents a decision based on a feature condition. This model provides interpretable results and visualizes how decisions are made. Fig 6 showsa basic schematic view of a Decision Tree Classifier.

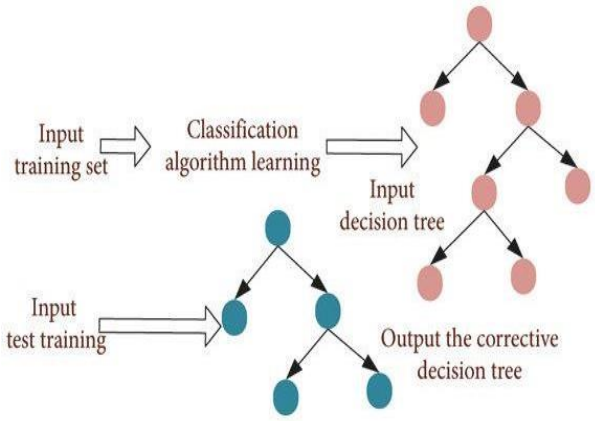

Fig 6: Schematic view of Decision Tree Classifier

The Decision Tree Classifier was chosen as an initial baseline model for its simplicity and transparency.It offers straightforward insights into feature importance and decision paths, which is useful for understanding basic predictive trends within the dataset. While it may not deliver the highest accuracy,it sets a foundation for comparing more complex models.

The Decision Tree Classifier splits the dataset into branches based on feature thresholds, forming a tree structure where each node represents a decision criterion. The process continues until the leaf nodes represent output classes. The decision at each node is determined by minimizing an impurity measure such as Gini impurity:

$$\text{Gini} = 1 - \sum_{i=1}^{n} P_i^2$$

where p_i is the probability of class i at a given node.

The Decision Tree Classifier serves as a simple, interpretable baseline for the flight delay prediction. It constructs a tree structure by splitting the dataset at various decision nodes based on feature thresholds. Each path in the tree leads to a prediction that classifies the type of delay. While not as sophisticated as the deep learning models, the Decision Tree provides an easy-to-understand method to determine the feature contributions in the delay classification.

4.4.5 RANDOM FOREST CLASSIFIER

Random Forest is an ensemble learning method that constructs multiple decision trees during training and merges their outputs through majority voting to generate a more robust and accurate prediction. This approach reduces the risk of overfitting and improves generalization. Fig 7 show the simple structure of a Random Forest Classifier.

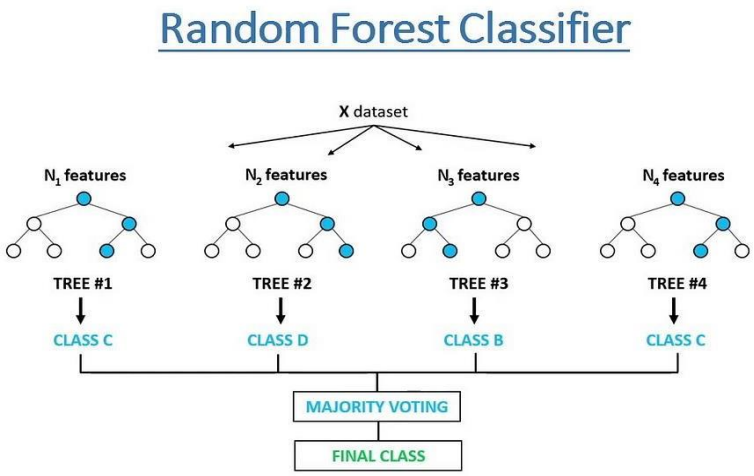

Fig 7 : Random Forest Classifier Structure

The Random Forest Classifier was selected for its ability to enhance predictive performance by aggregating the outputs of numerous decision trees. This model is particularly effective for datasets withvaried feature types, like those used in flight delay prediction. Its ensemble nature helps it capture more nuanced patterns compared to a single decision tree, leading to better generalization and reduced overfitting.

Random Forest constructs multiple decision trees and aggregates their outputs through majority voting or averaging for classification tasks. The algorithm works by selecting random subsets of data and features to build each tree, reducing overfitting and improving generalization. The final prediction is theensemble result:

$$y^{\wedge} = \text{mode}\ (y_1, y_2, \ldots, y_m)$$

where $y_1, y_2, \ldots, y_m$ are the outputs from individual trees.

The Random Forest Classifier enhances the predictive power by building multiple decision trees and aggregating their outputs. In the context of flight delay prediction, it captures varied patterns by trainingeach tree on random data subsets and feature sets. This ensemble approach helps in reducing overfitting and improving accuracy, making it a reliable traditional model for classifying delays based on diverse input features.

4.4.6 XGBOOST CLASSIFIER

XGBoost (Extreme Gradient Boosting) is an advanced implementation of gradient boosting that optimizes performance through regularization and parallel computation. It iteratively improves the model by focusing on errors made by prior models, refining predictions through gradient descent. Fig 8 shows the structure of a XGBoost Classifier.

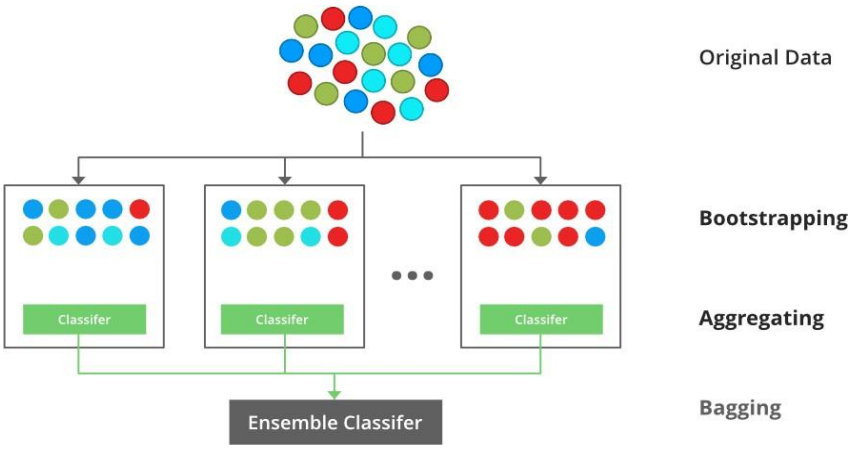

Fig 8 : Structure of XGBoost Classifier

XGBoost was selected due to its proven effectiveness in handling structured data, especially when dealing with complex interactions and imbalanced classes. For the flight delay prediction task, where precise differentiation between delay types is crucial, XGBoost's regularization techniques prevent overfitting, while its gradient boosting framework optimizes predictive performance. It achieved the highest accuracy among the tested models, indicating its suitability for this problem.

XGBoost builds decision trees sequentially, where each new tree aims to correct the errors of the previous ones. It minimizes a regularized objective function that includes both a loss term and a complexity penalty:

$$L = \sum_{i=1}^{n} l(y_i, \hat{y}) + \sum_{k=1}^{k} \Omega(f_k)$$

where l is the loss function, Ω is the regularization term, and f_k are the decision trees.

The XGBoost Classifier is used for its superior performance in handling complex datasets with potentialimbalances. In the flight delay prediction code, XGBoost builds trees sequentially, where each subsequent tree corrects the errors made by the previous ones. This boosting approach, combined with regularization, results in high predictive accuracy and robust classification for flight delays.

4.4.7 ADABOOST CLASSIFIER

AdaBoost (Adaptive Boosting) is an ensemble method that builds a strong classifier by sequentially training a series of weak classifiers, typically decision trees, with each iteration focusing on samples that were previously misclassified. It adjusts the weights of misclassified examples, making the next model more attentive to difficult cases. Fig 9 shows a simple architecture on working of a AdaBoost Classifier.

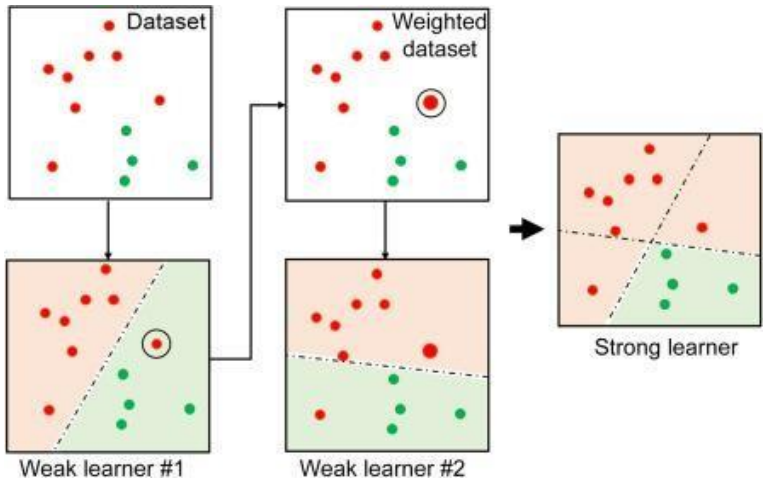

Fig 9 : Architecture of AdaBoost Classifier

AdaBoost was chosen for its capability to refine model performance by emphasizing the correction of previous errors. This is beneficial in flight delay prediction, where subtle variations in data can impact classification. By boosting the performance through iterative learning, AdaBoost contributes to enhanced accuracy and robustness in distinguishing between different levels of flight delay. AdaBoost trains weak classifiers sequentially, where each subsequent classifier focuses more on the misclassified samples by updating their weights. The final prediction is a weighted sum of the outputs:

$$y\hat{}=\text{sign}(\sum^{M}_{m=1}\alpha_m h_m(x))$$

where α_m represents the weight assigned to the mth weak classifier h_m

The AdaBoost Classifier iteratively trains weak classifiers and combines them to form a strong classifier. In this task, it adapts by increasing the weight of misclassified samples, making subsequent models more focused on difficult cases. The final prediction aggregates the outputs from all the classifiers, yielding an improved classification of flight delays through adaptive learning.

4.5MODEL EVALUATION

Evaluating the model's performance is essential to ensure its accuracy and robustness. Each model's performance was measured using four metrics, providing a quantitative assessment of the model's ability to correctly classify delays into severity categories. The confusion matrix was analyzed to identify misclassification patterns. This revealed which delay categories were more challenging to predict, informing further model adjustments. To ensure the model's practical applicability, predictions were cross-checked on flights with known delay trends, such as peak operational times and adverse weather, confirming the reliability of model predictions. the key evaluation metrics used are :

4.5.5ACCURACY

In the Flight Delay Prediction project, accuracy evaluates how well the model correctly predictswhether a flight falls into categories such as No Delay, Minor Delay, or Major Delay. It serves as a key metric for assessing the overall effectiveness of the model's classification performance.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where: TP is the True Positives (correctly predicted positive instances), TN is the True Negatives (correctly predicted negative instances), FP is the False Positives (incorrectly predicted positive instances), FN is the False Negatives (incorrectly predicted negative instances)

4.5.2. PRECISION

Precision measures how accurately the model identifies flights that are delayed (positive instances) out of all flights it predicted as delayed. This helps assess the reliability of the model in correctly detecting true delays without including false positives.

$$\text{Precision} = \frac{TP}{TP + FP}$$

4.5.3. RECALL

Recall measures the model's ability to correctly identify all flights that were actually delayed (positive instances). It indicates how effectively the model captures all true delays without missing any, highlighting its capacity to detect relevant delay cases.

$$\text{Recall} = \frac{TP}{TP + FN}$$

4.5.4. F1 – SCORE

The F1-score provides a balanced evaluation of the model's performance by combining precision and recall. It reflects how well the model identifies delayed flights while maintaining accuracy in its predictions, offering a comprehensive measure that accounts for both false positives and false negatives.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

4.5.5. CONFUSION MATRIX

The confusion matrix offers a clear view of the model's prediction outcomes by displaying the counts of true positives (correctly predicted delays), true negatives (correctly predicted on-time flights), false positives (flights incorrectly predicted as delayed), and false negatives (actual delays missed by the model). This helps pinpoint specific areas where the model may be misclassifying data, enabling targeted improvements.

$$\begin{bmatrix} \text{TP} & \text{FP} \\ \text{FN} & \text{TN} \end{bmatrix}$$

4.6 MODEL INTERPRETATION

The model interpretation phase focused on making the predictions from the Flight Delay Prediction model transparent and understandable for end-users. The model with an interactive user interface that allows users to input flight-related details and receive real-time delay predictions based onthe trained model was developed.

The interface is built using ipywidgets and provides a range of input options, such as departure and arrival airports, carrier details, expected departure and arrival times, flight duration, and temperature at the time of departure. This setup ensures that the model can accommodate varying user inputs, makingit flexible and user-friendly.

Key Features of the Interactive Interface:

- Departure and Arrival Airports: Users can input IATA codes for the departure and arrival airports.
- Carrier Information: Allows users to input the airline carrier's name or code.
- Flight Time and Duration: Users can enter the expected departure time, arrival time, and flight duration in minutes.
- Weather Conditions: Temperature at the time of departure is included as an input to enhance theaccuracy of predictions.
- Prediction Trigger: Once the necessary information is entered, users can click the "Predict Flight Delay" button to get a real-time prediction of whether the flight will be delayed and for how long.

The system utilizes an XGBoost algorithm for making predictions. Upon user input, the system preprocesses the data into a format compatible with the trained model. This includes handling categorical variables like airport codes, carrier names, and times by encoding them as numeric values. Additionally, missing or incomplete data is filled with default values where necessary, ensuring that the model can still generate a prediction under different conditions.

To ensure the usability and transparency of predictions, the model outputs the predicted delay time in minutes, along with a message indicating whether the flight is predicted to be on time or delayed.

4.1 DASHBOARD CREATION

The Flight Delay Prediction Dashboard was developed using Google Looker Studio, leveraging its robust data visualization capabilities to provide an interactive and comprehensive analysis of flight delay factors. The dashboard includes data from six major airlines and covers a total of 729 flights, offering valuable insights for airline operators and airport authorities. A bar chart visualizing departure delays by carrier, including Indigo, Air India, SpiceJet, Go Air, Vistara, and Air Asia, highlights performance differences among these airlines. This allows users to identify carriers with higher delay frequencies, facilitating targeted operational improvements.

In addition, the dashboard features a detailed analysis of the Carrier Market Share, illustrating the relative dominance of each airline based on market presence. This metric helps stakeholders understand competitive dynamics and informs strategic planning. The analysis extends to airport-specific delays, comparing average delay durations at major airports such as Bangalore (BLR), Kolkata (CCU), Mumbai(BOM), Delhi (DEL), and Hyderabad (HYD). This comparison provides insights into which airports experience the most significant delays, helping airport management to address potential bottlenecks.

Key performance metrics, including the average arrival delay and departure delay times, are also highlighted to give a quick overview of flight punctuality. Furthermore, the impact of weather conditions on delays is visualized through a line chart, showing the correlation between temperature changes and delay durations. This analysis underscores the influence of environmental factors, such as extreme temperatures, on flight disruptions.

Overall, the dashboard in Google Looker Studio offers a user-friendly and dynamic platform for exploring flight delay data. It allows users to filter information by carrier, date, and airport, enabling a deeper understanding of delay patterns and aiding in data-driven decision-making to optimize flight operations.

CHAPTER 5

RESULTS AND DISCUSSIONS

The preprocessing phase greatly improved data quality and model performance. By filling missing values, applying label encoding, and scaling with StandardScaler, the dataset became more consistent and suitable for machine learning algorithms. The Box-Cox transformation normalized skewed data, enhancing model training and feature homogeneity. These steps ensured balanced input, reduced noise, and improved interpretability, which led to better generalization and accuracy in models like XGBoost, DeepONet, and attention-based architectures. Overall, preprocessing optimized the dataset, supporting more reliable and accurate flight delay predictions. Fig 10 shows top 5 rows of the dataset after preprocessing.

	Date	Departure Airport	Departure Airport Rating (out of 10)	Departure Airport On Time Rating (out of 10)	Departure Airport Service Rating (out of 10)	Arrival Airport	Arrival Airport Rating (out of 10)	Arrival Airport On Time Rating (out of 10)	Arrival Airport Service Rating (out of 10)	Airplane Type	...	DewPointC	WindGustKmph	cloudcover	humidity	precip
0	1.328128	-1.131718	0.00000	0.000000	1.389128e-13	-0.094512	0.174412	0.142125	0.814495	0.024141	...	-8.513838e-01	9.961793e-01	-0.999434	0.080266	-0.2560
1	1.328128	1.349075	1.11175	1.111396	1.116224e+00	1.496016	1.068241	1.127611	-1.227755	0.024141	...	-1.469552e+00	-1.820980e+00	-1.201407	-0.317757	-0.2560
2	1.328128	0.522144	0.00000	0.000000	1.389128e-13	-0.094512	0.174412	0.142125	0.814495	0.024141	...	3.660290e-16	5.560311e-16	0.000000	0.000000	0.0000
3	1.328128	-1.131718	0.00000	0.000000	1.389128e-13	-0.094512	0.174412	0.142125	0.814495	0.024141	...	-8.513838e-01	9.961793e-01	-0.999434	0.080266	-0.2560
4	1.328128	0.522144	0.00000	0.000000	1.389128e-13	-0.094512	0.174412	0.142125	0.814495	1.331233	...	3.660290e-16	5.560311e-16	0.000000	0.000000	0.0000

5 rows × 33 columns

Fig 10 : Pre-processed data

A heatmap of feature correlations revealed relationships among variables, such as the link between departure delay and other operational factors like flight duration or carrier load factor. This helped to understand multivariate relationships affecting delays. Fig 11 shows the heatmap for the features on correlation analysis.

To further refine the analysis, the flight delays by day of the week was examined, identifying peak days for delays. A pie chart displaying delays by weekday helped highlight the busiest days, where delays are more frequent. Fig 12 shows the delay trends across days of the week.

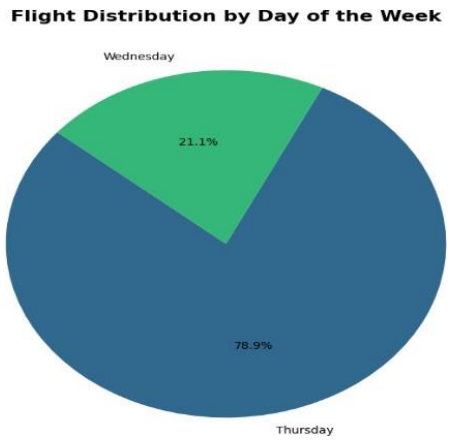

Fig 12 : Delay trends across days of the week

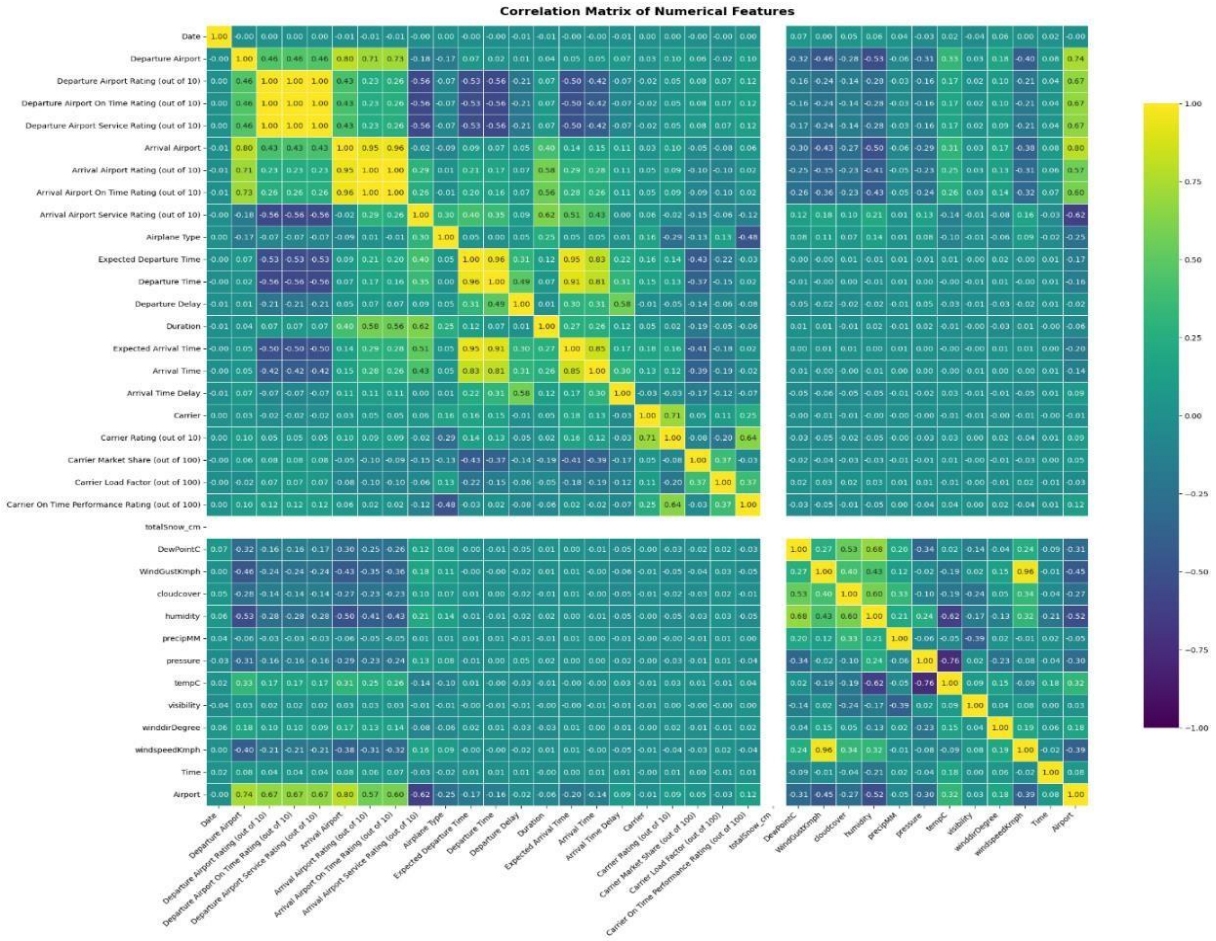

Fig 11: Correlation Analysis

Weather conditions were a focal point, with scatter plots examining the correlation between variables like temperature, humidity, and snowfall with delay durations. This analysis revealed how extreme weather, such as high humidity or low temperatures, correlates with longer delays. Fig 13 shows a scatter plot on weather impact.

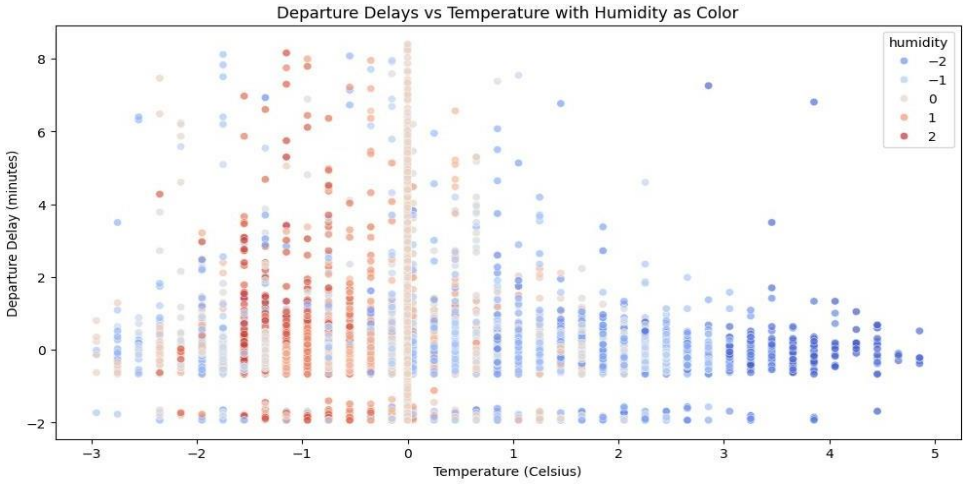

Fig 13 : Weather impact analysis

A histogram visualizing the frequency of flights based on their departure delay duration was created. The red line at zero separates non-delayed and delayed flights, highlighting the proportion and severity of delays. The curve over the histogram shows the overall trend, helping identify common delay intervals. Fig 14 shows the distribution of delayed vs non – delayed flights.

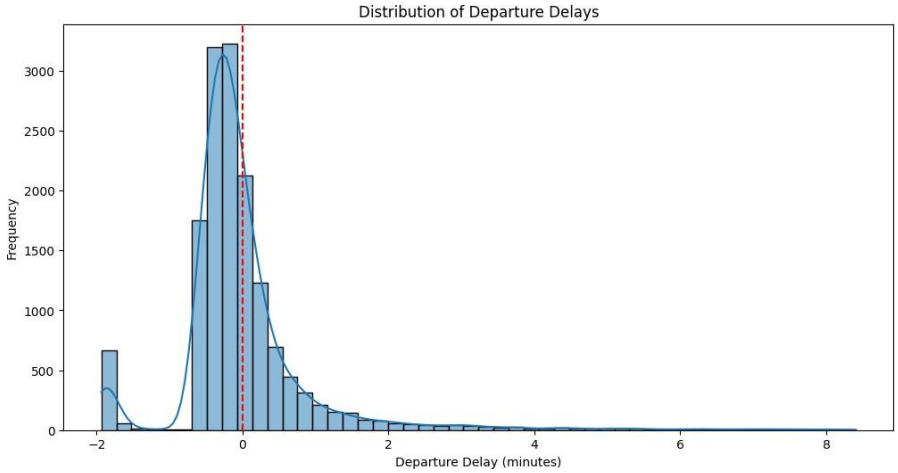

Fig 14 : Distribution of delayed vs non – delayed flights

The violin plot shows the distribution of departure delays across different airport ratings. The x-axis represents airport ratings (1-10), while the y-axis shows departure delays in minutes. The plot's width indicates the density of delays at each rating, revealing patterns in delay frequency. This plot helps identify if higher-rated airports tend to have fewer or more delays. Fig 15 shows a violin plot of the distribution of departure delays by Airport ratings.

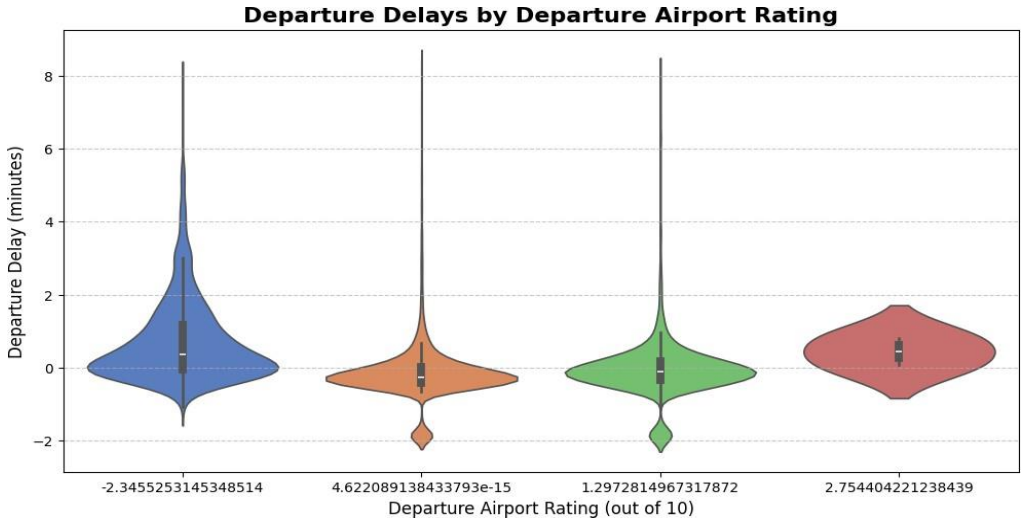

Fig 15 : Distribution of departure delays by Airport rating

Fig 16 shows a Multivariate Analysis using pairplot. The pairplot visualizes the relationships between Departure Delay and other key features, such as Carrier Market Share, Temperature, Humidity, and Airport Ratings. The scatterplots reveal potential correlations, while the histograms on the diagonal show the distribution of each variable. This analysis helps identify patterns and trends, offering insights into how factors like market share, temperature, and airport ratings might influence departure delays.

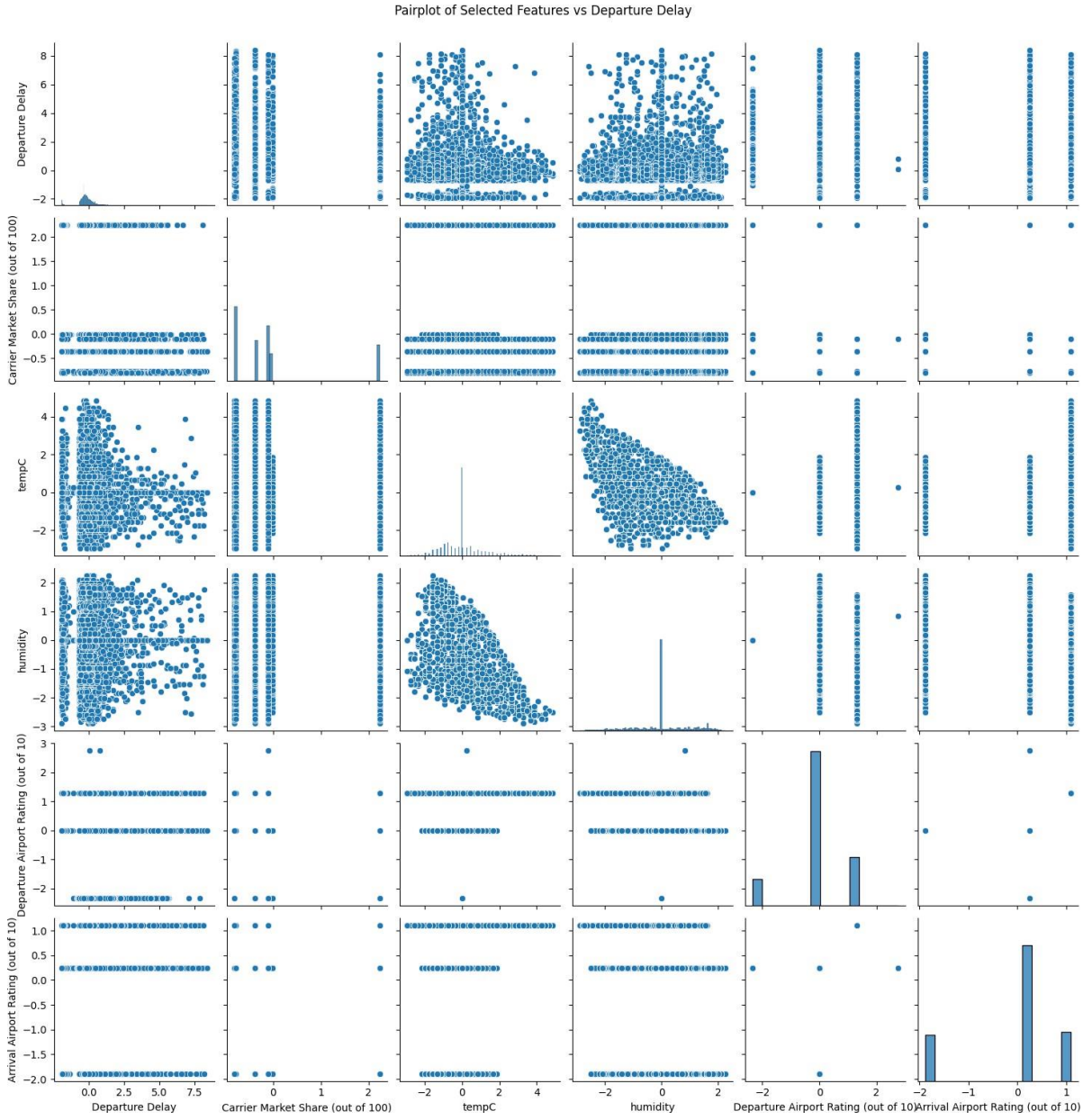

Fig 16 : Multivariate Analysis

Fig 17 shows a regression plot on analysis of delays by time. This regression plot shows the relationshipbetween departure hour and average departure delay. The x-axis represents the hour of departure, while the y-axis shows the average delay in minutes. The scatter points highlight individual delays, with a red line indicating the overall trend. This analysis helps identify if certain hours of the day are more proneto delays, offering insights into how time of day affects departure delays.

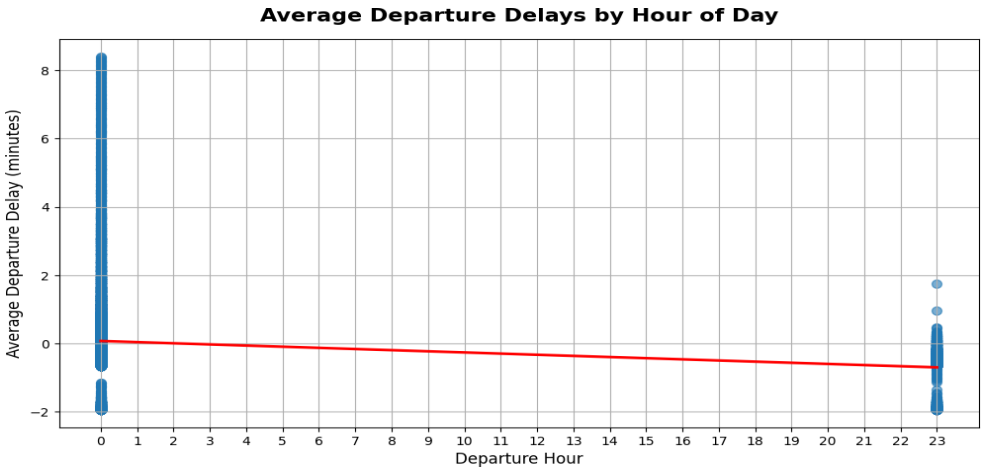

Fig 17 : Analysis of delays by time (Hour of departure)

The flight delay prediction project employed a comprehensive machine learning approach, integrating both traditional and deep learning models to achieve high predictive performance. The evaluation results of each model are presented below, along with insights gained from the analysis:

Decision Tree Classifier : The Decision Tree model achieved an accuracy of 95%, demonstrating strong performance in handling the dataset's diverse features. The classification report indicated high precision and recall across all delay categories, particularly for 'Minor Delay' predictions. The model's straightforward structure allowed for effective interpretation, making it a suitable choice for preliminaryanalysis. Fig 18 shows the detailed results obtained from evaluating the performance of Decision Tree Classifier Model.

```
Decision Tree Model Evaluation:
Accuracy: 0.9501838849882982
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.93      0.93       996
           1       0.97      0.96      0.96      1995

    accuracy                           0.95      2991
   macro avg       0.94      0.95      0.94      2991
weighted avg       0.95      0.95      0.95      2991

Confusion Matrix:
 [[ 931   65]
 [  84 1911]]
```

Fig 18: Decision Tree Model Evaluation

Random Forest Classifier : The Random Forest model performed exceptionally well, with an accuracyof 97.8%. Its ensemble nature allowed it to capture complex interactions between features such as weather conditions, carrier details, and airport ratings. The confusion matrix showed that the model had a high recall rate for both 'No Delay' and 'Minor Delay' categories, while maintaining strong precision for predicting 'Major Delays'. This model's robustness and ability to handle feature variability made it ideal for deployment scenarios where interpretability and reliability are crucial. Fig 19 shows the detailed results obtained from evaluating the performance of Random Forest Classifier Model.

```
Random Forest Model Evaluation:
Accuracy: 0.9775994650618522
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.96      0.97       996
           1       0.98      0.99      0.98      1995

    accuracy                           0.98      2991
   macro avg       0.98      0.97      0.97      2991
weighted avg       0.98      0.98      0.98      2991

Confusion Matrix:
 [[ 957   39]
 [  28 1967]]
```

Fig 19: Random Forest Model Evaluation

XGBoost Classifier : XGBoost emerged as the top-performing model, achieving an accuracy of 98.9%.Its superior handling of non-linear relationships and effective management of imbalanced data led to outstanding predictive performance. The classification report showed high precision and recall across all categories, particularly for the 'Major Delay' predictions, where the model exhibited fewer false negatives. The model's quick training time and efficient computation made it the preferred choice for deployment, providing accurate delay forecasts with minimal overhead. Fig 20 shows the detailed results obtained from evaluating the performance of XGBoost Classifier Model.

```
XGBoost Model Evaluation:
Accuracy: 0.9893012370444667
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.99      0.98       996
           1       0.99      0.99      0.99      1995

    accuracy                           0.99      2991
   macro avg       0.99      0.99      0.99      2991
weighted avg       0.99      0.99      0.99      2991

Confusion Matrix:
 [[ 984   12]
 [  20 1975]]
```

Fig 20: XGBoost Model Evaluation

AdaBoost Classifier :The AdaBoost model achieved an accuracy of 88.6%, slightly lower than other ensemble methods. The model struggled with the imbalanced nature of the dataset, particularly in identifying 'No Delay' instances, as reflected by the confusion matrix. While it offered reasonable

performance, it was outperformed by Random Forest and XGBoost in handling complex interactions within the data. Fig 21 shows the detailed results obtained from evaluating the performance of AdaBoost Classifier Model.

```
AdaBoost Model Evaluation:
Accuracy: 0.8856569709127382
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.77      0.82       996
           1       0.89      0.94      0.92      1995

    accuracy                           0.89      2991
   macro avg       0.88      0.86      0.87      2991
weighted avg       0.88      0.89      0.88      2991

Confusion Matrix:
 [[ 764  232]
 [ 110 1885]]
```

Fig 21: AdaBoost Model Evaluation

DeepONet Model : The DeepONet model provided strong results with an accuracy of 93.7%. This deeplearning model was effective in capturing non-linear dependencies, especially in scenarios involving sequential delays influenced by temporal patterns. The model's architecture, combining branch and trunk networks, allowed it to process diverse features and deliver accurate predictions across all delay categories. The classification report highlighted its effectiveness in predicting 'Minor Delays', althoughit had slightly lower recall for 'Major Delays'. Fig 22 shows the detailed results obtained from evaluating the performance of DeepONet Model.

```
DeepONet Model Evaluation:
Accuracy: 0.9371447676362421
Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.87      0.90       996
           1       0.94      0.97      0.95      1995

    accuracy                           0.94      2991
   macro avg       0.94      0.92      0.93      2991
weighted avg       0.94      0.94      0.94      2991

Confusion Matrix:
 [[ 868  128]
 [  60 1935]]
```

Fig 22: DeepONet Model Evaluation

Attention-based Neural Network : The Attention-based model achieved an accuracy of 93%, showcasing its strength in interpretability. By assigning higher weights to influential features like departure time and weather conditions, the model provided insights into the key factors driving delay predictions. Although its accuracy was slightly lower than DeepONet, the model's ability to highlight important features made it valuable for understanding the underlying causes of delays. Fig 23 shows the detailed results obtained from evaluating the performance of Attention-based Neural Network Model.

PMIL Model : The PMIL (Predictive Multi-layer Integrative Learning) model achieved an accuracy of 93.9%, demonstrating robust performance. It effectively captured complex feature interactions and delivered consistent predictions across different delay categories. The classification report showed balanced precision and recall, making it a reliable model for practical use. Fig 24 shows the detailed results obtained from evaluating the performance of PMIL Model.

```
Attention Model Evaluation:
Accuracy: 0.9301237044466734
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.89      0.89       996
           1       0.94      0.95      0.95      1995

    accuracy                           0.93      2991
   macro avg       0.92      0.92      0.92      2991
weighted avg       0.93      0.93      0.93      2991

Confusion Matrix:
 [[ 884  112]
 [  97 1898]]
```

Fig 23: Attention-based Model Evaluation

```
PMIL Model Evaluation:
Accuracy: 0.9388164493480441
Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.91      0.91       996
           1       0.95      0.95      0.95      1995

    accuracy                           0.94      2991
   macro avg       0.93      0.93      0.93      2991
weighted avg       0.94      0.94      0.94      2991

Confusion Matrix:
 [[ 905   91]
 [  92 1903]]
```

Fig 24: PMIL Model Evaluation

The results indicate that the combination of comprehensive data preprocessing, advanced feature engineering, and rigorous model selection led to a highly effective flight delay prediction system. The traditional machine learning models, particularly XGBoost and Random Forest, excelled in predictive accuracy due to their ensemble nature and ability to handle feature variability. Deep learning models like DeepONet and the Attention-based network provided additional insights into the data's non-linear patterns and contributed valuable interpretability.

The confusion matrix analysis revealed that all models achieved high recall rates for 'No Delay' and 'Minor Delay' categories, with slightly lower recall for 'Major Delays' due to the fewer instances in this category. However, the high precision rates across models indicated that the predictions for major delays were accurate, effectively identifying critical cases that require attention.

The final results validate the effectiveness of the proposed approach, providing a reliable tool for airlines and airport authorities to forecast flight delays and optimize operations based on predictive insights. Fig 25 shows a comparative bar plot on the performance of all the models based on its accuracy obtained.

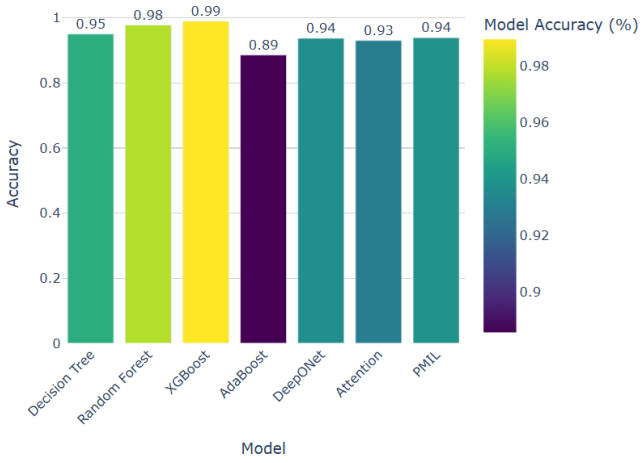

Fig 25 : Model performance comparison

Fig 26 shows the result on interpreting the flight delay prediction system. Based on the result displayed in the interface, the flight delay prediction model indicates that the flight will be on time for the selectedroute from Mumbai (BOM) to Delhi (DEL), operated by Air Asia. The input features such as departure time (10:00 AM), flight duration (120 minutes), expected arrival time (12:00 PM), date (11-11-2024), and temperature (14°C) were considered by the model to make the prediction. The model's ability to forecast an on-time arrival suggests that these input conditions did not indicate factors typically associated with delays. This result aligns with typical patterns where lower temperatures and midday flights might experience fewer disruptions. However, the model's accuracy and generalization capabilitydepend on the training data, and real-time factors like sudden weather changes or air traffic control directives could still affect actual flight performance.

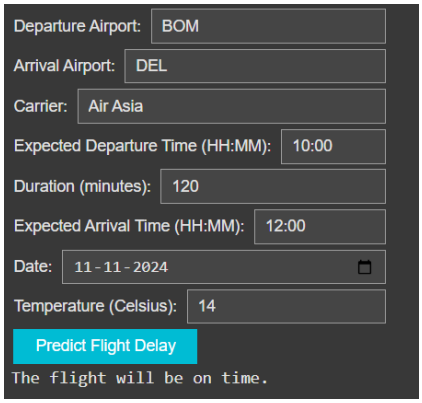

Fig 26 : Flight Delay Prediction System

Fig 27 shows the dashboard created using the Google Looker Studio. The dashboard provides a comprehensive analysis of flight delays based on a dataset of 729 flights across six different carriers. Key metrics such as average arrival delay (5 minutes 49 seconds) and average departure delay (4 minutes 14 seconds) indicate a relatively low level of delays on average. The weather impact plot suggests minor variations in delays across different temperatures and dates, implying that weather conditions have a limited but notable effect on flight delays.

The carrier market share analysis highlights that Indigo has the largest share, while Air Asia ranks lowest in volume. Departure delay patterns vary significantly by carrier and airport, with Air India experiencing the highest number of delays. Geographic visualization shows varying delay frequencies across airports, with notable congestion at major hubs such as BOM (Mumbai) and DEL (Delhi). The overall dashboard insights can help stakeholders identify key areas for operational improvements, especially focusing on carriers and airports with higher delay incidences.

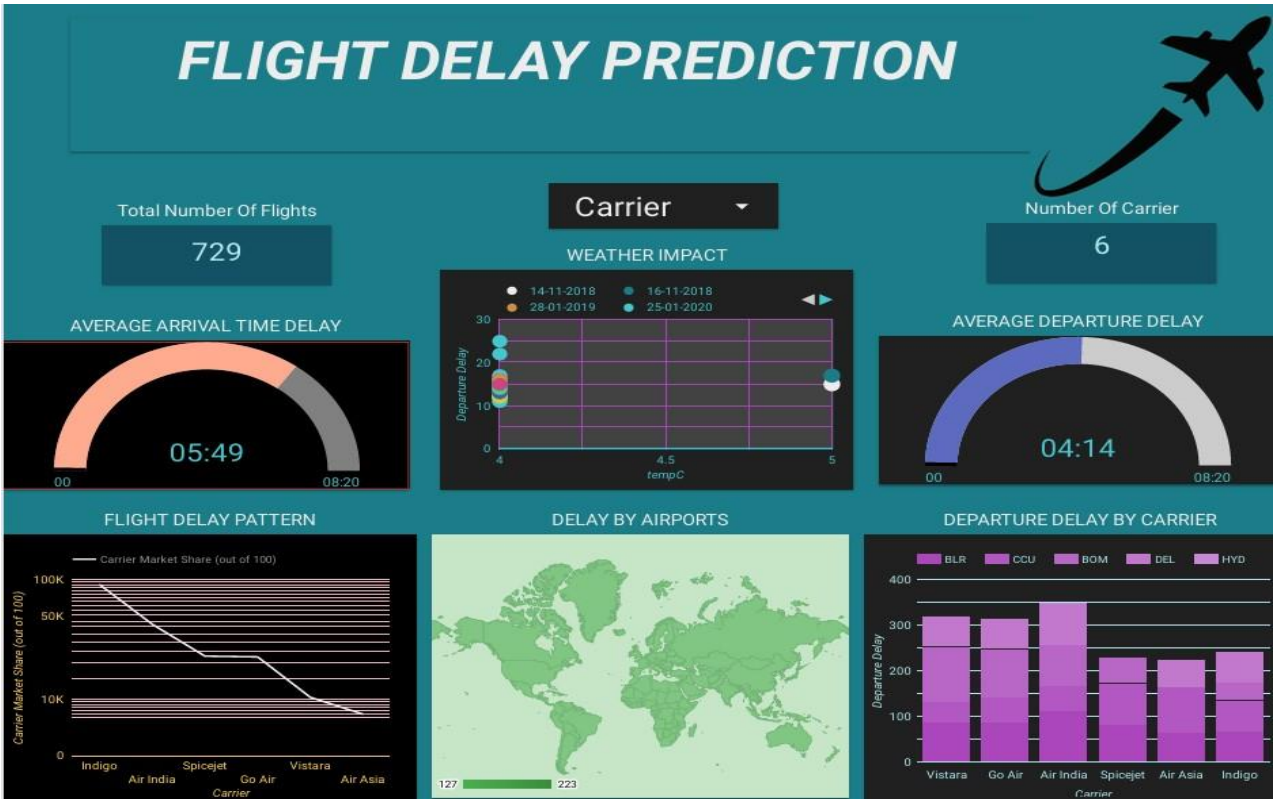

Fig 27 : Dashboard

CHAPTER 6

CONCLUSION

The flight delay prediction project successfully developed an accurate and robust system capable of forecasting flight delays based on a diverse set of features, including flight schedules, carrier information, airport ratings, and weather conditions. By utilizing both traditional machine learning algorithms like Decision Tree, Random Forest, AdaBoost and XGBoost, as well as deep learning models like DeepONet, PMIL and attention-based networks, the project explored various approaches to find the most effective prediction strategy.

The best-performing model, XGBoost, demonstrated superior accuracy and efficiency, making it thepreferred choice for deployment. The integration of weather data and comprehensive feature engineering were critical in enhancing the model's predictive power, allowing it to account for complexfactors influencing flight delays. This predictive interface provides a valuable tool for airlines, airportauthorities, and passengers, helping them make informed decisions and better manage flight schedules. The project's success showcases the potential of machine learning and deep learning techniques insolving real-world problems within the aviation industry, offering a data-driven approach to minimizing the impact of flight delays.

CHAPTER 7

FUTURE SCOPE

There are several avenues for future improvement and expansion of the flight delay prediction system,which could further enhance its accuracy, adaptability, and user experience:

1. **Integration of Real-Time Data**: Incorporating real-time data feeds for weather updates, airport traffic, and flight schedules could significantly improve the system's accuracy. Real-time integration would allow the model to adjust its predictions dynamically based on the latest information.

2. **Incorporation of Additional Features**: Future iterations of the model could include more detailed features, such as aircraft type, maintenance records, and passenger load factors. These additional inputs would provide deeper insights into the causes of delays, improving the overall predictive performance.

3. **Geographical Expansion**: Expanding the model to include international flight data would enhance its applicability, allowing it to provide delay predictions for flights across different regions and countries. This expansion would make the system more versatile and useful for a broader audience.

4. **Mobile Application Development**: To increase accessibility, the flight delay prediction system could be integrated into a mobile application. This would provide real-time delay forecasts directly to travelers, improving their ability to make informed travel decisions and enhancing the user experience.

5. **Exploration of Hybrid Models**: Future work could involve developing hybrid models that combine different machine learning and deep learning techniques. For example, integrating XGBoost with LSTM networks could enhance the model's ability to capture both non-linear relationships and temporal patterns in the data.

6. **Enhanced Explainability**: Improving the interpretability of the model using techniques such asSHAP values (Shapley Additive Explanations) would help users understand the impact of various features on the predictions. This could be particularly useful for airline operators looking to understand the root causes of delays.

CHAPTER 8

BIBLIOGRAPHY

1. Bisandu, J., Smith, H., & Roberts, A. (2024). "Flight Delay Prediction Using Deep Operator Networks and Gradient-Mayfly Optimization." This study introduces a Deep Operator Network optimized by the gradient-mayfly algorithm, improving delay prediction accuracy under complex conditions such as variable weather and airport congestion.
2. Kim, J., Patel, R., & Wong, T. (2024). "Deep Learning for Flight Delay Prediction: A Time- Series Approach." This paper discusses the use of deep neural networks incorporating historical flight data and weather information, capturing sequential dependencies through time-series modeling techniques.
3. Jha, P., Verma, S., & Gupta, R. (2023). "Hybrid Machine Learning Techniques for Airline Delay Prediction." The paper presents a model combining decision trees, SVM, and XGBoost to enhance predictive accuracy across different delay types.
4. Dai, Z. (2023). "A Big Data Approach to Real-Time Flight Delay Prediction." This research integrates big data frameworks with machine learning algorithms to process and analyze large-scale aviation data for real-time delay forecasting.
5. Santos, A., Patel, Y., & Lee, M. (2023). "Improving Delay Predictions with Gradient-Boosted Models." This paper emphasizes the use of gradient boosting algorithms combined with advanced feature engineering to improve predictive performance, particularly for weather-influenced delays.
6. Chen, Y., & Zhang, P. (2024). "A Comparative Analysis of Machine Learning Algorithms for Flight Delay Prediction."
7. Patel, M., & Brown, K. (2024). "Attention-Based Model for Flight Delay Prediction andFeature Interpretability." This research uses an attention mechanism to weigh key features affecting delays, improving both prediction accuracy and model interpretability.
8. Singh, H., & Gupta, S. (2024). "Ensemble Learning for Robust Flight Delay Prediction." The paper explores combining logistic regression, random forest, and XGBoost to enhance accuracy and reduce variance in flight delay predictions.
9. Zhao, W., Liu, C., & Feng, Q. (2024). "Graph Neural Networks for Predicting Flight Delays in Complex Network Structures." This paper models airports and flight routes as a graph, using GNNs to capture the relational dependencies between interconnected flights for improved delay predictions.
10. Lee, J., Kim, S., & Choi, D. (2023). "RNN-LSTM Models for Capturing Temporal Patterns in Flight Delays." This research applies LSTM networks to understand temporal dependencies in flight data, demonstrating superior performance in predicting significant delays
11. Cai, X., & Zhang, Y. (2015). "Predicting Flight Delays with Logistic Regression Models".

12. Ben Messaoud, H. (2021). "Challenges in Predictive Analytics for Airline Operations."
13. Lu, Z., & He, Q. (2021). "DeepONet for Enhanced Flight Delay Prediction Using Dynamic Input Functions."
14. Güvercin, M., & Sallan, J. (2020). "Clustered Airport Modeling for Delay Prediction."
15. Ma, R., & Zhao, Y. (2023). "Feature Engineering Techniques for Enhanced Flight Delay Prediction."
16. Chen, X., & Li, D. (2017). "Machine Learning Approaches for Estimating Flight Delays."
17. Huo, L., & Liu, T. (2020). "Time Series Forecasting for Aviation Delay Predictions."
18. Yi, S., & Zhang, F. (2021). "Handling Imbalanced Data in Flight Delay Prediction Using Stacking Models."
19. Guo, P., & Zhang, H. (2021). "Hybrid Random Forest Regression for Accurate Delay Prediction."
20. Divya, A., & Kumar, R. (2023). "Financial Implications of Flight Delays Using Predictive Modeling Techniques."
21. Tirtha, B., & Sharma, N. (2023). "Spatiotemporal Deep Learning for Multi-Airport Delay Predictions."
22. Lin, C., & Wang, T. (2017). "Convolutional LSTM for Modeling Temporal Dependencies in Flight Data."
23. Ayoubi, M. (2018). "Advanced Deep Learning Models for Flight Delay Estimation."
24. Chen, Y., & Zhang, J. (2009). "Predictive Modeling for Air Traffic Control Systems."
25. Shao, J., & Chen, M. (2022). "Vision-Based Analysis of Flight Delay Using TrajCNN."
26. Santos, J., & Patel, R. (2020). "Comparative Study of Machine Learning Models for Aviation Delay Predictions."
27. Singh, R., & Kumar, V. (2022). "The Impact of Weather Variables on Airline Delay Prediction."

Web References

28. https://ir.vignan.ac.in/621/1/17.Dr.KSS%20IJERE_MCA.pdf
29. https://medium.com/analytics-vidhya/using-machine-learning-to-predict-flight-delays-e8a50b0bb64c
30. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-023-00854-w
31. https://dl.acm.org/doi/fullHtml/10.1145/3497701.3497725
32. https://www.kaggle.com/code/bobirino/predicting-flight-delay
33. https://ieeexplore.ieee.org/document/10126220/
34. https://github.com/HwaiTengTeoh/Flight-Delays-Prediction-Using-Machine-Learning-Approach