Enhancing Air Travel Efficiency: Study of ML & DL Techniques for Flight Delay Classification, Prediction and Recommendations

Swapnil Kapale
Dept. of Computer Engineering
PCCOE
Pune, India
swapnil.kapale21@pccoepune.org

Aabid Kachhi
Dept. of Computer Engineering
PCCOE
Pune, India
mohammadaabid.kachhi21@pccoepune.org

Sanket Kadam
Dept. of Computer Engineering
PCCOE
Pune, India
sanket.kadam21@pccoepune.org

Soham Joshi
Dept. of Computer Engineering
PCCOE
Pune, India
soham.joshi21@pccoepune.org

***Abstract*—Air travel delays have significant economic and environmental impacts, frustrating passengers and airlines alike. This paper explores the application of machine learning (ML) and deep learning (DL) techniques to address these challenges by developing models for flight delay classification, prediction, and recommendation generation. Leveraging historical flight data and relevant features, we propose a multi-faceted approach to enhance air travel efficiency. First, we present a supervised learning model for classifying flight delays into different categories based on causal factors. Next, we introduce a DL-based forecasting model that predicts departure delays using sequential data.Through extensive experiments on real-world datasets, we demonstrate the effectiveness of our methods in accurately categorizing delays, anticipating disruptions, and providing actionable insights to airlines and passengers. The proposed techniques have the potential to mitigate the adverse effects of flight delays, leading to improved resource allocation, customer satisfaction, and sustainability in the aviation industry.**

Keywords—Machine Learning, KNN, Regression

I. Introduction

Air travel has become an integral part of modern society, facilitating global connectivity and economic growth. However, the aviation industry faces significant challenges in the form of flight delays, which have far-reaching consequences for airlines, passengers, and the environment. Flight delays can result from various factors, including weather conditions, air traffic control issues, airline operations, and airport infrastructure constraints. These disruptions lead to increased operational costs, passenger dissatisfaction, and substantial carbon emissions due to prolonged aircraft idling and rerouting.

Addressing flight delays is a complex problem that requires a comprehensive understanding of the underlying causes and the ability to anticipate and mitigate disruptions proactively. Traditional approaches, such as rule-based systems and manual decision-making, often fall short in dealing with the dynamic and multifaceted nature of air travel operations. In recent years, machine learning (ML) and deep learning (DL) techniques have demonstrated remarkable potential in tackling complex problems across various domains, offering promising solutions for enhancing air travel efficiency.

This paper presents a comparative analysis of traditional methods such as linear regression, polynomial regression, Lasso regression, Ridge regression, Elastic net regression and K-Nearest Neighbors (KNN) with deep learning (DL) models based on their performance on the flight-delay-and-cancellation-dataset-2019-2023. By harnessing the power of historical flight data and relevant features, our approach aims to provide a comprehensive solution for airlines, air traffic controllers, and passengers.

II. Literature Review

Several studies have been conducted on modelling and predicting flight delays using various machine learning and deep learning models. Yazdi et al proposed a Deep Learning (DL) model using the Levenberg-Marquart algorithm for predicting flight delays.[1] They utilized a stack denoising autoencoder (SDA) to handle noisy flight delay data and improve model accuracy.

Fujun Wang et al discussed the use of machine learning and deep learning models, such as random forest, gradient boosted decision tree, recurrent neural network, and long short-term memory, for predicting flight delays.[2] They highlighted the challenges associated with multi-airport delay prediction and analyzing individual flights.

Yi Ding proposed a method to model arriving flights using multiple linear regression and predicted flight delays based on departure delay and route distance.[3] The proposed model showed an accuracy of approximately 80% and outperformed Naive-Bayes and C4.5 approaches in terms of accuracy, precision, recall, and F-score.

A study focused on predicting flight delays using supervised machine learning models and evaluated seven algorithms for binary classification of flight delays. The Decision Tree algorithm performed the best with an accuracy of 0.9777, while the KNN algorithm had the worst performance with an f1-score of 0.8039. Tree-based ensemble classifiers generally outperformed other base classifiers.[4]

Zámková et al aimed to analyse the causes of flight delays for a selected European airline and identify potential risks and reasons for delays in air transport. They used data from the years 2013-2019, including information on the duration and causes of delays and the characteristics of individual flights. Multidimensional statistics methods, such as tests of independence and correspondence analysis, were applied for data processing [5]

Overall, these studies demonstrate the application of various machine learning and deep learning models for predicting flight delays and analysing their causes. The results indicate that accurate prediction of flight delays can help airlines optimize their operations and improve passenger satisfaction.

III. Analysis of dataset

Preprocessing is a crucial step in data analysis, particularly when dealing with large and complex datasets. In our study, we applied several data cleaning and transformation techniques to prepare the dataset for predicting flight delays. The dataset used is Flight Delay and Cancellation Dataset (2019-2023) from Kaggle which has 32 attributes and 3 million samples. The attributes contain flight date, airline, airline dot, airline code, dot code, flight number, origin, origin city, destination, destination city, crs departure time, departure time, departure delay, taxi out, wheels off, wheels on, taxi in, crs arrival time ,arrival time, arrival delay, cancelled, cancellation code, diverted, crs elapsed time, elapsed time, air time, distance, delay due carrier, delay due weather, delay due NAS(National Airline System), delay due security and delay due late aircraft.

The dataset had around 70k null values for attributes such as departure delay, wheels off time, Arrival time, etc. The cancelled flights were removed from dataset as we are only interested in predicting delay for non-cancelled flights. Most of these attributes are mathematically related and can be inferred from available values of other attributes. These mathematical relations are given below.

To handle missing arrival delay values, we replaced them with zeros. However, when the arrival delay was zero, we used the CRS_ARR_TIME instead to fill missing arrival times.

Missing TAXI_IN values were replaced with the average TAXI_IN where ARR_DELAY was zero.

To calculate WHEELS_ON, we subtracted TAXI_IN from ARR_TIME for all records except those with missing values. For missing values, we used ARR_TIME and TAXI_IN instead. Mathematically,

$$WHEELS\_ON = ARR\_TIME - TAXI\_IN$$

Similarly, to calculate AIR_TIME, we subtracted WHEELS_OFF from WHEELS_ON for all records except those with missing values. For missing values, we used WHEELS_ON and WHEELS_OFF instead.

$$AIR\_TIME = WHEELS\_ON - WHEELS\_OFF$$

To calculate ELAPSED_TIME, we summed TAXI_OUT, AIR_TIME, and TAXI_IN for all records except those with missing values. For missing values, we used the calculated ELAPSED_TIME instead.

$$ELAPSED\_TIME = TAXI\_OUT + AIR\_TIME \\ + TAXI\_IN$$

Lastly, we replaced missing values in DELAY_DUE_CARRIER, DELAY_DUE_WEATHER, DELAY_DUE_NAS, DELAY_DUE_SECURITY, and DELAY_DUE_LATE_AIRCRAFT with zeros. A column TOTAL_DELAY was created with binary values 0 or 1, representing if a flight was delay or not.

We applied Min-Max normalization to all time-related and distance features to ensure that they were on a similar scale. This normalization technique transforms each feature to a range between 0 and 1, making it easier to compare and analyse the data. By normalizing both time and distance columns, we ensured that all features were on a similar scale, making it easier to analyse the data and extract meaningful insights.

We also converted categorical columns to numerical values using label encoding. This technique was used to convert non-numerical categories into numerical values, making it easier for machine learning algorithms to process. To reduce memory usage, the categorical columns were processed in chunks, where the dataframe was divided into equal parts, and label encoding was applied to each chunk separately. The label encoder and the mapping between the original categories and their corresponding numerical values were saved for future use. This process was implemented using the LabelEncoder class from the Sklearn library. The resulting numerical dataframe was then used for further analysis and modeling.

We performed data cleaning and reduction by dropping columns that contained redundant or similar information. Specifically, we dropped the columns 'FL_DATE', 'AIRLINE', 'AIRLINE_CODE', 'DOT_CODE', 'ORIGIN', and 'DEST' as they were either already represented by other columns or not relevant to the analysis. The 'AIRLINE_DOT' column already contained information about the airline and its corresponding code, and the 'ORIGIN_CITY' and 'DEST_CITY' columns contained information about the origin and destination of the flights. By removing these columns, we reduced the dimensionality of the data and focused on the most relevant features for our analysis. This process resulted in a final dataframe with 25 columns.

By applying these preprocessing steps, we ensured that all missing values were handled, and the dataset was ready for further analysis. The following correlation matrix demonstrates the relation between various features from our dataset.

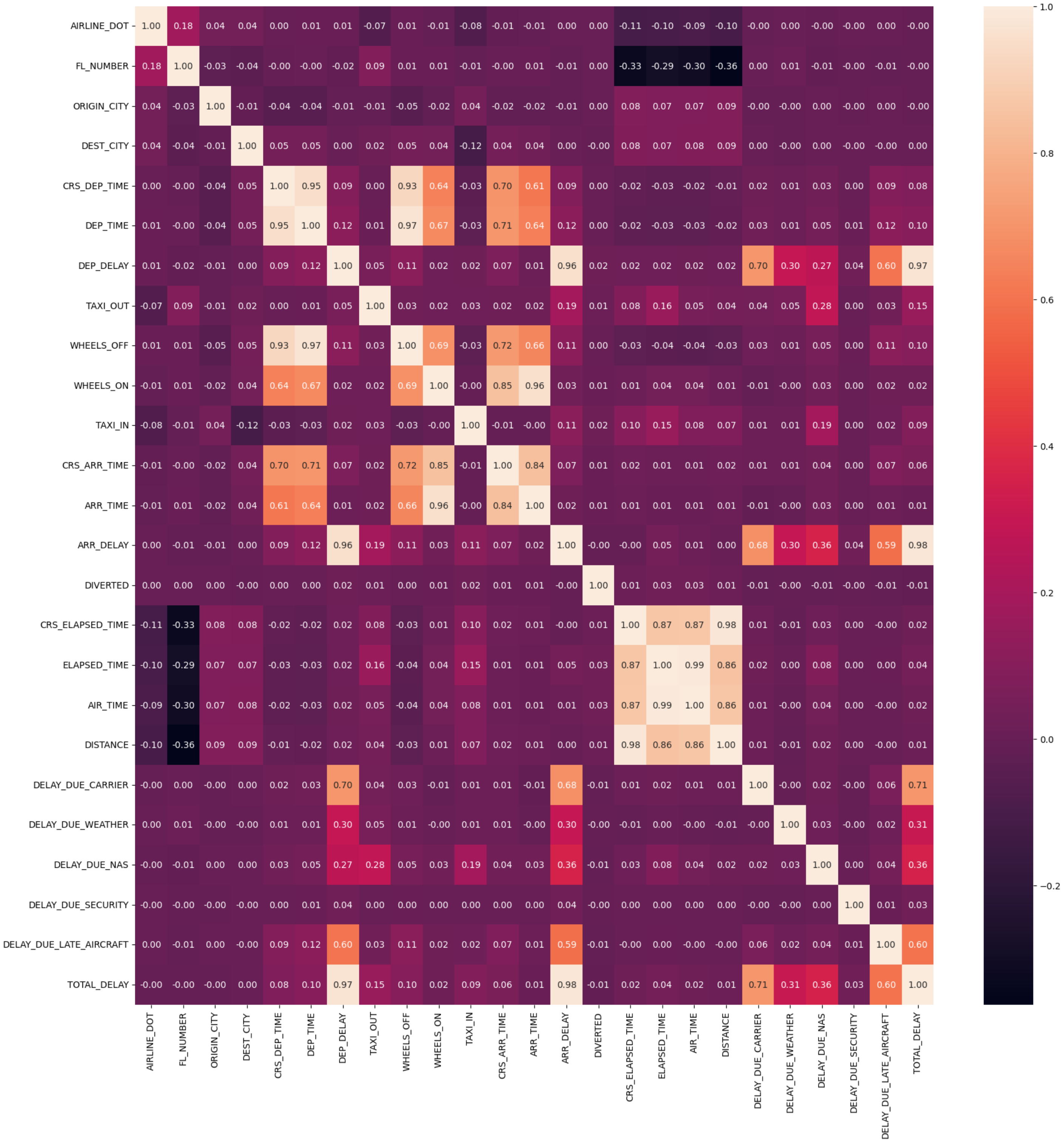

Figure 1. Correlation matrix of all features

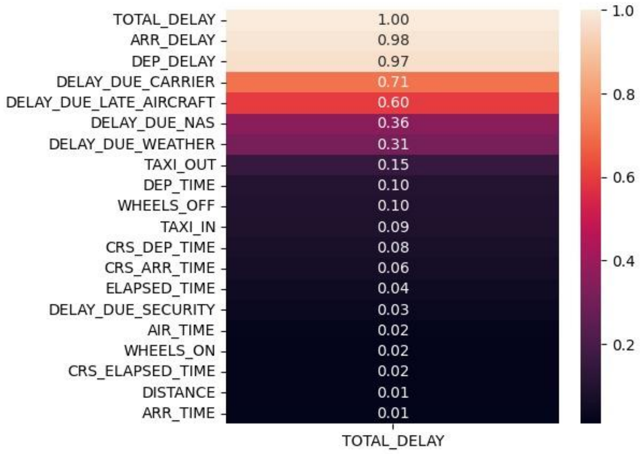

Figure 2. Correlation of TOTAL_DELAY with rest of the features

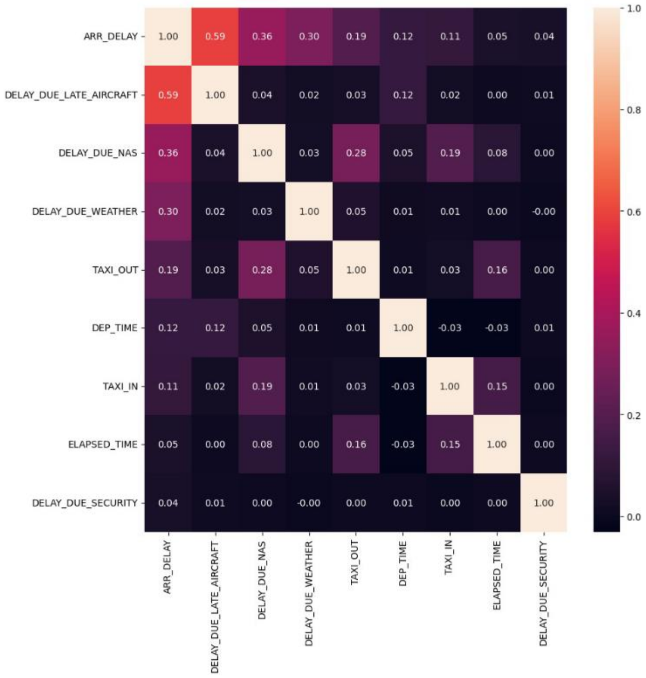

Figure 3. The selected features after removing multi collinear features

IV. PERFORMANCE METRICS

1] Accuracy: Accuracy is a popular indicator for gauging a classifier's efficacy on evenly distributed training data. In other words, it is the proportion of successful forecasts to total model predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Where:

- The count of observations the model wrongly interpreted as negative is denoted by FN (False Negative).
- True Negative (TN) is the fraction of data points for which the model made an accurate negative prediction.
- True Positive (TP) denotes the count of observations that were correctly identified as positive by the model,
- The number of times the model wrongly classifies a set of observations as positive is the number of FPs (False Positive).

2] Precision: Precision is a crucial metric in evaluating the performance of a binary classification model as it quantifies the accuracy of positive predictions made by the model relative to all positive predictions.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

3] Recall: Recall, also known as sensitivity or true positive rate, measures the model's ability to correctly identify all positive instances from the total instances of positive observations.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

4] F-Measure: This is a popular statistic for gauging the efficacy of a binary classification model since it provides an overall measure of performance by incorporating both precision and recall into a single value [18]. For formal purposes, we use equation 4 to determine the F-measure:

$$F - measure = \frac{2 * precision * recall}{precision + recall} \tag{4}$$

V. EXPERIMENTATION ENVIRONMENT

In this research, we utilized a variety of tools and environments for experimentation. We used both Google Colab and Jupyter Notebooks running locally on Visual Studio Code for our development and testing. The models were trained on a simple i5 CPU, which allowed us to test the performance and scalability of our algorithms on a limited computational resource.

We utilized several popular libraries and frameworks for our implementation, including PyTorch for deep learning models, NumPy for numerical computations, and scikit-learn for machine learning algorithms and utilities. These libraries provided a robust and flexible environment for our research, allowing us to quickly prototype and test our ideas.

Additionally, we used the pandas library for data manipulation and cleaning, matplotlib and seaborn for data visualization, and the LabelEncoder class from the sklearn.preprocessing module for preprocessing categorical data. These tools helped us to efficiently process and analyze the data, gaining insights and driving the direction of our research.

Overall, our experimentation environment provided a solid foundation for our research, allowing us to explore and test our ideas in a flexible and efficient manner.

VI. Experimentation

We used various feature selection techniques like SelectKBest and SelectPercentile for feature selection. Two values of k, 5 and 7 were used with SelectKBest and 50 and 70 percentile value for SelectPercentile method. Along with these 4 dataframes, we also used a data frame with removed multi collinearity as shown in Figure 3. Few important features to make dataset more practical were added like Flight number, Origin and Destination city and Distance.

We implemented various machine learning algorithms, including linear regression, lasso regression, and polynomial regression, on our dataset. Each algorithm produced different results, and we evaluated their performance using metrics like the Mean Absolute Error (MAE). Upon analysis, we observed that polynomial regression yielded the highest accuracy among them. Specifically, polynomial regression was able to fit the curve of the dataset more effectively compared to other algorithms. Therefore, we conclude that polynomial regression is the most suitable model for this dataset.

For predicting flight delays, we applied standard classification algorithms like K-nearest neighbors (KNN) and logistic regression. After examining the outcomes, we concluded that logistic regression consistently provided superior performance compared to KNN across all datasets. This indicates that logistic regression is more effective in accurately classifying whether a flight will experience a delay or not.

Our analysis revealed that logistic regression is the preferred choice for our classification task of predicting flight delays. Its ability to outperform KNN across all datasets demonstrates its effectiveness in accurately determining the likelihood of flight delays.

After our analysis, we realized that traditional classification algorithms didn't provide sufficient accuracy for our needs. Therefore, we decided to explore deep learning as an alternative. We developed a simple deep learning model, as illustrated in the figure, specifically for our classification task. This model yielded higher accuracy compared to the traditional machine learning algorithms we initially employed. Our findings suggest that deep learning offers promising results for improving the accuracy of our classification task.

VII. Results and evaluation

Table 1 compares the performance of various traditional regression techniques, including Linear Regression, Polynomial Regression, Ridge Regression, Lasso Regression, and Elastic Net Regression. The performance is evaluated using Mean Squared Error (MSE) and R-squared (R2) metrics for different feature selection methods.

The Linear Regression and Ridge Regression models generally perform better than the other techniques, with higher R-squared values and lower MSE values across different feature selection methods. Polynomial Regression also shows comparable performance in terms of MSE, but its R-squared values are slightly lower.

Table 2 shows the accuracy of the logistic regression model for different feature selection techniques. The accuracy remains constant at 0.66 across all feature selection methods, including SelectPercentile50, SelectPercentile70, SelectKBest5, SelectKBest7, and without addressing multicollinearity. This suggests that the feature selection method does not significantly impact the accuracy of the logistic regression model for this particular dataset.

Table 3 presents the performance metrics of the K-Nearest Neighbours (KNN) model under different feature selection techniques. The metrics reported are accuracy, precision, recall, and F1-score. Similar to the logistic regression table, the values remain constant across all feature selection methods, including SelectPercentile50, SelectPercentile70, SelectKBest5, SelectKBest7, and without addressing multicollinearity.

The accuracy of the KNN model is relatively lower at around 0.595, which may indicate that the model is not performing well on this particular dataset. Additionally, the precision, recall, and F1-score values are also quite low, suggesting that the model might be struggling to classify the instances correctly.

Overall, the provided tables offer a comprehensive comparison of different machine learning models and feature selection techniques, allowing for an evaluation of their performance on the given dataset.

Deep Learning model gives an accuracy of 97% on same dataset with subset of samples . Thus we can infer that deep learning model performs better from classical approaches on this dataset

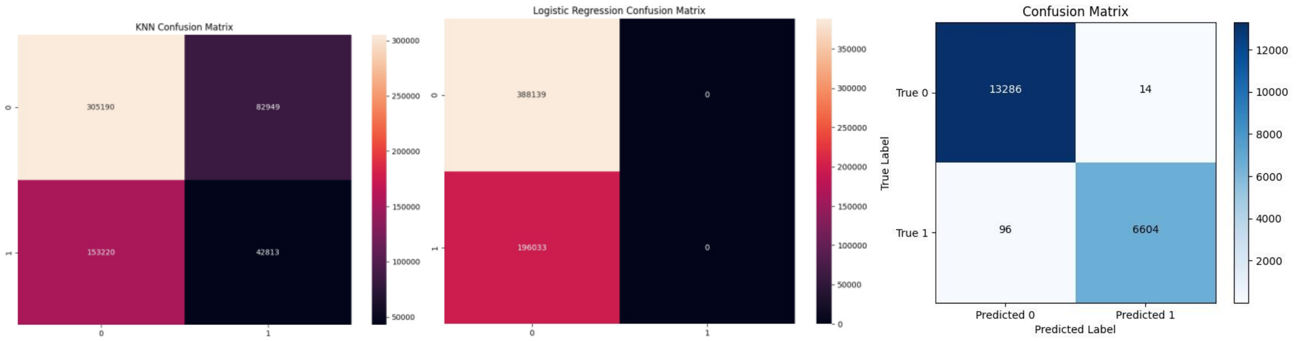

Figure 4. shows confusion matrices for KNN, Logistic Regression and Deep Learning approaches respectively

Dataframe	**Metrics**	**Traditional Methods**				
		Linear Regression	**Polynomial Regression**	**Ridge Regression**	**Lasso Regression**	**Elastic Net Regression**
SelectKBest 7	**MSE**	8E-06	8.7E-06	8E-06	0.00027	0.00027
	R2	0.96981	0.96737	0.9698	-3.2E-06	-3.2E-06
Select Percentile 50	**MSE**	8.6E-06	7E-06	8.6E-06	0.00027	0.00027
	R2	0.96746	0.97379	0.96745	-3.2E-06	-3.2E-06
SelectKBest 5	**MSE**	8.2E-06	6.3E-06	8.2E-06	0.00027	0.00027
	R2	0.96903	0.97633	0.96902	-3.2E-06	-3.2E-06
Select Percentile 70	**MSE**	8.2E-06	6.3E-06	8.2E-06	0.00027	0.00027
	R2	0.96924	0.97626	0.96923	-3.2E-06	-3.2E-06
No multicollinearity	**MSE**	7.99E-06	6.54E-06	7.99E-06	0.00027	0.00027
	R2	0.96993	0.9754	0.96992	#######	#######

Table 1. MSE and R squared values for various traditional machine learning approaches for different dataframes

Logistic Regression					
Dataframe	**Select Percentile 50**	**Select Percentile 70**	**SelectKBest 5**	**SelectKBest 7**	**No multicollinearity**
Accuracy	0.664425888	0.664425888	0.664425888	0.664425888	0.664425888

Table 2. Accuracy for Logistic regression for different dataframes

KNN						
Dataframe		**Select Percentile 50**	**Select Percentile 70**	**SelectKBest 5**	**SelectKBest 7**	**No multicollinearity**
Metrics	**Accuracy**	0.595726943	0.595720096	0.59572352	0.595720096	0.595720096
	Precision	0.340442112	0.340428746	0.34043416	0.340428746	0.340428746
	Recall	0.218402004	0.218396903	0.218396903	0.218396903	0.218396903
	F1	0.266096528	0.266088659	0.266090313	0.266088659	0.266088659

Table 3.Performance metrics of KNN for different dataframes

VIII. Conclusion

This study evaluated the performance of logistic regression, KNN, traditional regression techniques, and neural networks on a binary classification task using various feature selection methods. The logistic regression and KNN models showed consistent but modest accuracy, with KNN struggling to classify both classes accurately based on the confusion matrices. Polynomial regression generally outperformed other traditional techniques. However, the neural network model emerged as a strong contender, achieving high accuracy and low loss on both training and validation data, despite a relatively simple architecture with three dense layers and few trainable parameters. While some models faced challenges, the neural network's ability to learn and generalize effectively highlights its potential for this task. Future work could explore more advanced architectures, ensemble models, or techniques to handle class imbalance and feature importance for further performance improvements.

IX. REFERENCES

1. Yazdi, M.F., Kamel, S.R., Chabok, S.J.M. et al. Flight delay prediction based on deep learning and Levenberg-Marquart algorithm. J Big Data 7,106 (2020). https://doi.org/10.1186/s40537-020-00380-z

2. Wang, F., Bi, J., Xie, D., & Zhao, X. (2022). Flight delay forecasting and analysis of direct and indirect factors. IET Intelligent Transport Systems,16(7),890-907. https://doi.org/10.1049/itr2.12183

3. Predicting flight delay based on multiple linear regression Yi Ding 2017 IOP Conf. Ser.: Earth Environ. Sci. 81 012198 DOI 10.1088/1755-1315/81/1/012198

4. Airline Flight Delay Prediction Using Machine Learning Models Yuemin Tang,University of Southern California, DOI: https://doi.org/10.1145/3497701.3497725

5. Zámková, M., Rojík, S., Prokop, M., & Stolín, R. (2021). Factors Affecting the International Flight Delays and Their Impact on Airline Operation and Management and Passenger Compensations Fees in Air Transport Industry: Case Study of a Selected Airlines in Europe. Sustainability, 14(22), 14763. https://doi.org/10.3390/su142214763