Prediction of flight delay using Exploratory Data Analysis incorporating Machine learning techniques

A PROJECT REPORT

Submitted by

DIVYASINDHU P – 312319205036
JOAN RESHMI L - 312319205057

of

BACHELOR OF TECHNOLOGY
in
INFORMATION TECHNOLOGY

St. JOSEPH'S COLLEGE OF ENGINEERING

(An Autonomous Institution)

St. Joseph's Group of Institutions

Jeppiaar Educational Trust

OMR, Chennai 600 119

ANNA UNIVERSITY: CHENNAI

March-2023

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"Prediction of flight delay using Exploratory Data Analysis incorporating Machine learning techniques**" is the bonafide work of **DIVYASINDHU P (312319205036)** and **JOAN RESHMI L (312319205057)** who carried out the project under my supervision.

SIGNATURE
Supervisor,
Ms. Divya J, M. Tech. ,(Ph.D),
Assistant Professor,
Department of IT,
St. Joseph's College of Engineering, OMR, Chennai-600119.

SIGNATURE
Head of the department,
Dr. V Muthulakshmi, M.E.,Ph.D,
Professor,
Department of IT,
St. Joseph's College of Engineering, OMR, Chennai-600119.

CERTIFICATE OF EVALUATION

COLLEGE NAME : St. Joseph's College of Engineering, Chennai-600119.

BRANCH : B.TECH., IT (Information Technology)

SEMESTER VIII

SL. NO	NAME OF THE STUDENT	TITLE OF THE PROJECT	NAME OF THE SUPERVISOR WITH DESIGNATION
1 2	DIVYASINDHU P (312319205036) JOAN RESHMI L (312319205057)	Prediction of flight delay using Exploratory Data Analysis incorporating machine learning techniques.	Ms. DIVYA J, M.Tech., (Ph.D.), ASSISTANT PROFESSOR

The report of the project work submitted by the above students in partial fulfillment for the award of Bachelor of Technology Degree in Information Technology of Anna University was confirmed to be reportofthe work done by the above students and then evaluated.

Submitted to Project and Viva Examination held on _____.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

At the outset we would like to express our sincere gratitude to the beloved **Chairman, Dr. Babu Manoharan, M.A.,M.B.A.,Ph.D.,** for his constant guidance and support.

We would like to express our heartfelt thanks to our respected **Managing Director Mrs. S. Jessie Priya, M.Com.,** for her kind encouragement and blessings.

We wish to express our sincere thanks to our **Executive Director Mr. B. ShashiSekar, M.Sc.,** for providing ample facilities in the institution.

We express our deepest gratitude and thanks to our beloved **Principal Dr.VaddiSeshagiri Rao, M.E., M.B.A., Ph.D., F.I.E.,** for his inspirational ideas during the course of the project.

We wish to express our sincere thanks and gratitude to **Dr.V.Muthulakshmi , M.E.,Ph.D.,** Head of the Department , Department of Information Technology, St. Joseph's College of Engineering for her guidance and assistance in solving the various intricacies involved in the project.

It is with deep sense of gratitude that we acknowledge our indebtness to our supervisor **Ms. J. Divya, M.Tech., (Ph.D.),** for her expert guidance and connoisseur suggestion.

Finally we thank our department staff members who helped us in the successful completion of this project.

ABSTRACT

One of the difficult situations in the business world, flight planning involves a lot of unpredictability. Such a circumstance exists when delays occur; they are caused by a variety of circumstances and come at a significant expense to airlines, operators, and passengers. Airlines, airports, and passengers all suffer considerable financial and other losses as a result of flight delays. In order to address this problem, precise forecasting of these aircraft delays helps airlines to respond to likely causes of the delays in advance to lessen the negative effects and lets passengers to be well prepared for the interruption caused to their journey. The prediction analysis gleaned from this research can serve as a prototype for identifying operational factors that cause delays in any situation. Data have been examined using exploratory data analysis to see what aspect (possible correlations or a lack thereof) might have a substantial impact on delays. In this project, we proposed a Deep Learning-based model for predicting flight delay. (DL). DL is one of the most recent approaches used to address issues with high levels of complexity and vast amounts of data. To complete the binary classification of flight delays, an Artificial Neural Network (ANN) was developed and evaluated. By contrasting the results of four measurements—accuracy, precision, recall, and f1-score—algorithm was evaluated. The ANN algorithm was improved using the Adam Optimization technique. Finally, in response to the critical characteristics identified for flight delay prediction, a novel artificial neural network (ANN) and genetic algorithm (GA)-based aviation delay technique was presented.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATON

ABBREVIATION	DEFINITION
CNN	Convolutional Neural Network
ANN	Artificial Neural Network
KNN	K nearest neighbour
SVM	Support Vector Machine
RF	Random Forest
GBM	Gradient Boosting Machine
MLP	Multilayer Perceptron
ML	Machine Learning
AI	Artificial Intelligence
IOT	Internet of things
RMSE	Root Mean Squared Error
WSN	Wireless Sensor Network
GUI	Graphical User Interface
ReLU	Rectified Linear Unit
GPU	Graphics Processing Unit

Chapter 1
INTRODUCTION

Airport capacity is sometimes described as being determined by the average aircraft delay. One of the most common problems in the globe is flight delays. It's quite difficult to justify a delay. There are a few uncommon causes of flight delays, such as runway work or heavy traffic, but inclement weather seems to be the most frequent one. As businesses rely on consumer loyalty to fund their frequent flier programs, it harms airports, and airlines, and impacts their marketing methods. The objective of this project is to create classification models for airline delays and predict the length of the delay period using various machine learning methods and various elements (controllable and uncontrolled variables). In order to address this problem, precise forecasting of these aircraft delays helps airlines to respond to likely causes of the delays in advance to lessen the negative effects and lets passengers be well prepared for the interruption caused to their journey. The goal is to examine the methods for creating forecasting models for aircraft delays brought on by adverse weather. Aircraft delays and their underlying causes were the primary targets for prediction using ML systems. The best approach for anticipating flight delays has been determined by comparing the performances of various algorithms. Artificial Neural Networks (ANN), have demonstrated their high accuracy when modeling sequential data. Adam Optimization algorithm has been used to optimize the ANN algorithm. Lastly, a novel aviation delay method is optimized using an artificial neural network (ANN) in conjunction with a genetic algorithm (GA) in response to the important parameters discovered for flight delay prediction. The ANN-GA model's performance has been confirmed, showing that it can accurately forecast delays with a score of 89%.

1.2. SYSTEM OVERVIEW

The predominant aim of this project is to propose a novel predictive model to predict flight delay using the dataset extracted from the Bureau of Transportation Statistics of the United States Department of Transportation. The dataset contains the data of all the arrived and departed US flights of 2022. The objectives of the proposed work are formulated below:

- To apply Exploratory Data analysis to infer patterns from the data collected.
- To predict flight delay using the generated patterns
- To create a novel model to implement and deploy on an application.

1.3. SCOPE OF THE PROJECT

The number of planes that fail to take off on time likewise rises as more people opt to travel by air. The airline industry experiences financial challenges as a result of this growth, which also worsens the overcrowding at airports. Delays in air travel are a sign of the aviation system's inefficiency. Both airline firms and their customers pay a hefty price for it. The Entire Delay Impact Study estimated that the total cost of air transportation delays to passengers and the airline sector in the US in 2007 was $32.9 billion, resulting in a $4 billion decline in GDP. As a result, anticipating delays can enhance airline operations and passenger pleasure, which will benefit the economy.

Chapter 2

LITERATURE SURVEY

[1] Seyedmirsajad , Mokhtarimousavi , ArminMehrabi , "Flight delay causality: Machine learning technique in conjunction with random parameter statistical analysis", International Journal Of Transportation Science And Technology, 2022

The consequences of flight delays can significantly impact airports' on-time performance and airline operations, which have a strong positive correlation with passenger satisfaction. Thus, an accurate investigation of the variables that cause delays is of main importance in decision-making processes. Although statistical models have been traditionally used in flight delay analysis, the presence of unobserved heterogeneity in flight data has been less discussed. An empirical analysis has been carried out to investigate the potential unobserved heterogeneity and the impact of significant variables on flight delay using two modeling approaches. First, preliminary insight into potential significant variables was obtained through a random parameter logit model (also known as the mixed logit model). Then, a Support Vector Machines (SVM) model trained by the Artificial Bee Colony (ABC) algorithm, was employed to explore the non-linear relationship between flight delay outcomes and causal factors. The data-driven analysis was conducted using three-month flight arrival data from Miami International Airport (MIA). A variable impact analysis was also conducted considering the black-box characteristic of the SVM and compared to the effects of variables indented through the random parameter logit modeling framework. While a large unobserved heterogeneity was observed, the

impacts of various explanatory variables were examined in terms of flight departure performance, geographical specification of the origin airport, day of month and day of the week of the flight, cause of delay, and gate information. The comprehensive assessment of the contributing factors proposed in this study provides invaluable insights into flight delay modeling and analysis.

[2] Desmond BalaBisandu, Irene Moulitsas & Salvatore Filippone, "Social ski driver conditional autoregressive-based deep learning classifier for flight delay prediction", Neural Computing And Applications, Vol. 34, 8777-8802, 2022

The importance of robust flight delay prediction has recently increased in the air transportation industry. This industry seeks alternative methods and technologies for more robust flight delay prediction because of its significance for all stakeholders. The most affected are airlines that suffer from monetary and passenger loyalty losses. Several studies have attempted to analyze and solve flight delay prediction problems using machine learning method. A novel alternative method has been proposed, namely social ski driver conditional autoregressive-based (SSDCA-based) deep learning. The Social Ski Driver algorithm was combined with Conditional Autoregressive Value at Risk by Regression Quantile. The most relevant instances has been created from the training dataset, which are the delayed flights. Data transformation has been applied to stabilize the data variance using Yeo-Johnson. The training and testing of the data has been performed using deep recurrent neural network (DRNN) and SSDCA-based algorithms. The SSDCA-based optimization algorithm helped us choose the right network architecture with better accuracy and less error than the existing literature.

The results of proposed SSDCA-based method and existing benchmark methods were compared. The efficiency and computational time of the proposed method are compared against the existing benchmark methods. The SSDCA-based DRNN provides a more accurate flight delay prediction with 0.9361 and 0.9252 accuracy rates on both dataset-1 and dataset-2, respectively. To show the reliability of the method, it has been compared with other meta-heuristic approaches. The result is that the SSDCA-based DRNN outperformed all existing benchmark methods tested in the experiment.

[3] W. Shao, A. Prabowo, S. Zhao, P. Koniusz, F.D. Salim, "Predicting flight delay with spatio-temporal trajectory convolutional network and airport situational awareness map", Neurocomputing, Volume 472, Pages 280-293, 2022

To model and forecast flight delays accurately, it is crucial to harness various vehicle trajectories and contextual sensor data on airport tarmac areas. These heterogeneous sensor data, if modeled correctly, can be used to generate a situational awareness map. Existing techniques that apply traditional supervised learning methods to historical data, contextual features, and route information among different airports to predict flight delay are inaccurate and only predict arrival delay but not departure delay, which is essential to airlines. A vision-based solution was developed to achieve a high forecasting accuracy, applicable to the airport. The solution leverages a snapshot of the airport situational awareness map, which contains various trajectories of aircraft and contextual features such as weather and airline schedules. An end-to-end deep learning architecture, Trajectory CNN, was proposed which captures both the spatial and temporal

information from the situational awareness map. Additionally, the situational awareness map of the airport was revealed which has a vital impact on estimating flight departure delays. The proposed framework obtained a good result (around 18 min error) for predicting flight departure delay at Los Angeles International Airport.

[4] Stefan Reitmann and Michael Schultz, "An Adaptive Framework for Optimization and Prediction of Air Traffic Management (Sub-)Systems with Machine Learning", Application Of Data Science To Aviation , Vol. 9, Issue 2, 2022

Evaluating the performance of complex systems, such as air traffic management (ATM), is a challenging task. When regarding aviation as a time-continuous system measured in value-discrete time series via performance indicators and certain metrics, it is important to use sufficiently targeted mathematical models within the analysis. A consistent identification of system dynamics at the evaluation level, without dealing with the actual physical events of the system, transforms the analysis of time series into a system identification process, which ensures control of an unknown (or only partially known) system. The requirements for mathematical modeling are presented in the form of a step-by-step framework, which can be derived from the formal process model of ATM. The framework is applied to representative datasets based on former experiments and publications, for whose prediction of boarding times and classification of flight delays with machine learning (ML) the framework presented here was used. While the training process of neural networks was described in detail there, the paper shown here focuses on the control options and optimization possibilities based on the trained models. Overall, the discussed framework represents a

strict guideline for addressing data and machine learning (ML)-based analysis and metaheuristic optimization in ATM.

[5] Lucas Giusti, Leonardo Carvalho, Antonio Tadeu Gomes, Rafaelli Coutinho, Jorge Soares & Eduardo Ogasawara, "Analyzing flight delay prediction under concept drift", Evolving System, Vol. 13, 723-736, 2022

Flight delays impose challenges that impact any flight transportation system-predicting when they will occur in a meaningful way to mitigate this issue. However, the distribution of the flight delay system variables changes over time. This phenomenon is known in predictive analytics as concept drift. This paper investigates the prediction performance of different drift handling strategies in aviation under different scales (models trained from flights related to a single airport or the entire flight system). Specifically, two research questions have been answered: (1) how do drift-handling strategies influence the prediction performance of delays? (2) Do different scales change the results of drift handling strategies? In the analysis, drift handling strategies are relevant, and their impacts vary according to scale and machine learning models.

[6] Ilinka Ivanoska, Luisina Pastorino, Massimiliano Zanin, "Assessing Identifiability in Airport Delay Propagation Roles Through Deep Learning Classification", Journals And Magazines IEEE, Volume 10, 2022

Delays in air transport can be seen as the result of two independent contributions, respectively stemming from the local dynamics of each airport and from a global propagation process; yet, assessing the relative importance

of these two aspects in the final behavior of the system is a challenging task. The use of the score has been proposed and obtained in a classification task, performed over vectors representing the profiles of delays at each airport, as a way of assessing their identifiability. Deep Learning models are able to recognize airports with high precision, thus suggesting that delays are defined more by the characteristics of each airport than by the global network effects. This identifiability is higher for large and highly connected airports, constant through the years, but modulated by season and geographical location. Some operational implications of this approach has been discussed finally.

[7] Hatıpoğlu, Irmak, Tosun, Ömür, Tosun, Nedret, "Flight Delay Prediction Based With Machine Learning", LogForum, Vol. 18 , Issue 1, 2022

The delay of a planned flight causes many undesirable situations such as cost, customer satisfaction, and environmental pollution. There is only one way to prevent these problems before they occur, and that is to know which flights will be delayed. The aim of is to predict delayed flights. For this, the use of machine learning techniques is preferred. Methods: Estimations are made with three up-to-date techniques XGBoost, LightGBM, and CatBoost techniques based on Gradient Boosting from machine learning techniques. The bayesian technique is used for hyper-parameter settings. In addition, the Synthetic Minority Over-Sampling Technique (SMOTE) technique is also used, as the majority of flights are on time and delayed flights, which constitute a minority class, may adversely affect the results. The results are analyzed and shared with and without SMOTE. Results: As a consequence of the application, which was run on a data set containing all of an

international airline's flights [18148 flights] for a year, it was discovered that flights may be predicted with high accuracy. Conclusions: The application of machine learning techniques to anticipate flight delays is new, but it has a lot of potential. Companies will be able to avert problems before developed if delays are correctly estimated, which can generate plenty of issues. As a result, concrete advantages such as lower costs and higher customer satisfaction will emerge. Improvements will be made at the most vulnerable place in the aviation business.

[8] Micha Zoutendijk and Mihaela Mitici, "Probabilistic Flight Delay Predictions Using Machine Learning and Applications to the Flight-to-Gate Assignment Problem", Application of Data Science to Aviation, Vol. 8, Issue 6, 152, May 2021.

The problem of flight delay prediction is approached most often by predicting a delay class or value. However, the aviation industry can benefit greatly from probabilistic delay predictions on an individual flight basis, as these give insight into the uncertainty of the delay predictions. Therefore, two probabilistic forecasting algorithms, Mixture Density Networks and Random Forest regression, are applied to predict flight delays at a European airport. The algorithms estimate well the distribution of arrival and departure flight delays with a Mean Absolute Error of less than 15 min. To illustrate the utility of the estimated delay distributions, these probabilistic predictions was integrated into a probabilistic flight-to-gate assignment problem. The objective of this problem is to increase the robustness of flight-to-gate assignments. Considering probabilistic delay predictions, the proposed flight-to-gate assignment model reduces the number of conflicted aircraft by up to 74% when compared to a deterministic flight-to-gate assignment

model. In general, the results illustrate the utility of considering probabilistic forecasting for robust airport operations' optimization.

[9] D. Truong, "Using causal machine learning for predicting the risk of flight delays in air transportation", Journal of Air Transport Management, Volume 91, 101993, March 2021.

Delays in air transportation are a major concern that has negative impacts on the airline industry and the economy. Given the complexity of the National Air Space system, predicting the risk of flight delays and identifying significant predictors is vital to risk mitigation. The purpose is to perform data mining using causal machine learning algorithms in the USELEI process to predict the probability of flight delays in air transportation using data collected from different sources. The findings indicated significant effects of predictors, including reported arrivals and departures, arrival and departure demands, capacity, efficiency, and traffic volume at the origin and destination airports on the risk of flight delays. More importantly, how these predictors interact with one another and how these interactions lead to delay incidents. Finally, sensitivity analysis and causal inference can be performed to evaluate various what-if scenarios and form effective strategies to mitigate the risk of delays.

[10] Waqar Ahmed Khan, Hoi-Lam Ma, Sai-Ho Chung, Xin Wen, "Hierarchical integrated machine learning model for predicting flight departure delays and duration in series", Tranportation Part C : Emerging Technologies, Vol. 129 , 103225 , 2021

Flight delays may propagate through the entire aviation network and are becoming an important research topic. A novel hierarchical integrated

machine learning model has been proposed for predicting flight departure delays and duration in series rather than in parallel to avoid ambiguity in decision making. The proposed model has been analyzed using various machine learning algorithms in combination with different sampling techniques. The highly noisy, unbalanced, dispersed, and skewed historical high dimensional data provided by an international airline operating in Hong Kong was used to demonstrate the practical application of the model. The result shows that for a 4-h forecast horizon, a constructive neural network machine learning algorithm with the Synthetic Minority Over Sampling Technique-Tomek Links (SMOTETomek) sampling technique was able to achieve better average balanced recall accuracies of 65.5%, 61.5%, 59% for classifying delay status and predicting delay duration at thresholds of 60 min and 30 min, respectively. Similarly, for minority labels, the precision-recall and area under the curve showed that the proposed model achieved better results of 32.44% and 35.14% compared to the parallel model of 26.43% and 21.02% for thresholds of 60 min and 30 min, respectively. The effect of different sampling techniques, sampling approaches, and estimation mechanisms on prediction performance is also studied.

[11] K. Cai, Y. Li, Y. Fang, Y. Zhu, "A deep learning approach for flight delay prediction through time-evolving graphs", IEEE Transactions on Intelligent Transportation Systems, Volume 23 Issue 8, Aug 2021.

Flight delay prediction has recently gained growing popularity due to the significant role it plays in efficient airline and airport operation. Most of the previous prediction works consider the single-airport scenario, which overlooks the time-varying spatial interactions hidden in airport networks.

The flight delay prediction problem is investigated from a network perspective (i.e., multi-airport scenario). To model the time-evolving and periodic graph-structured information in the airport network, a flight delay prediction approach based on the graph convolutional neural network (GCN) is developed. More specifically, regarding that GCN cannot take both delay time-series and time-evolving graph structures as inputs, a temporal convolutional block based on the Markov property is employed to mine the time-varying patterns of flight delays through a sequence of graph snapshots. Moreover, considering that unknown occasional air routes under emergency may result in incomplete graph-structured inputs for GCN, to expose spatial interactions hidden in airport networks. Through extensive experiments, it has been shown that the proposed approach outperforms benchmark methods with a satisfying accuracy improvement at the cost of acceptable execution time. The obtained results reveal that deep learning approach based on graph-structured inputs have great potentials in the flight delay prediction problem.

[12] H. Alla, L. Moumoun and Y. Balouki, "A multilayer perceptron neural network with selective-data-training for flight arrival delay prediction", Scientific Programming, Article No. 5558918, Jun 2021.

Flight delay is the most common preoccupation of aviation stakeholders around the world. Airlines, which suffer from a monetary and customer loyalty loss, are the most affected. Various studies have attempted to analyze and solve flight delays using machine learning algorithms. It aims to predict flights' arrival delay using Artificial Neural Network (ANN). A MultiLayer Perceptron (MLP) was applied to train and test the data. Two approaches have been adopted in the work. In the first one, historical flight data was

extracted from Bureau of Transportation Statistics (BTS). The second approach improves the efficiency of the model by applying selective-data training. It consists of selecting only most relevant instances from the training dataset which are delayed flights. According to BTS, a flight whose difference between scheduled and actual arrival times is 15 minutes or greater is considered delayed. Departure delays and flight distance proved to be very contributive to flight delays. An adjusted and optimized hyperparameters using grid search technique helped us choose the right architecture of the network and have a better accuracy and less error than the existing literature. The results of both traditional and selective training were compared. The efficiency and time complexity of the second method are compared against those of the traditional training procedure. The neural network MLP was able to predict flight arrival delay with a coefficient of determination of 0.9048, and the selective procedure achieved a time saving and a better score of 0.9560. To enhance the reliability of the proposed method, the performance of the MLP was compared with that of Gradient Boosting (GB) and Decision Trees (DT). The result is that the MLP outperformed all existing benchmark methods.

[13] Z. Guo, B. Yu, M. Hao, W. Wang, Y. Jiang, F. Zong (May 2021), "A novel hybrid method for flight departure delay prediction using random forest regression and maximal information coefficient", Aerospace Science and Technology, 116:106822

Flight departure delay prediction is one of the most critical components of intelligent aviation systems. The accurate prediction of flight departure delays can provide passengers with reliable travel schedules and enhance the service performance of airports and airlines. A hybrid method of Random

Forest Regression and Maximal Information Coefficient (RFR-MIC) has been proposed for flight departure delay prediction. Random Forest Regression and Maximal Information Coefficient are inherently fused in terms of Information Consistency. Furthermore, it focuses on utilizing flight information on multiple air routes for flight departure delay prediction. To validate the proposed flight departure delay prediction model, a numerical study is conducted using flight data collected from Beijing Capital International Airport (PEK). The proposed RFR-MIC model exhibits good performance compared with linear regression (LR), k-nearest neighbors (k-NN), artificial neural network (ANN), and standard Random Forest Regression (RFR). The results also show that flight information on multiple air routes can certainly improve the accuracy of flight departure delay prediction.

[14] Michael Schultz, Stefan Reitmann, Sameer Alam, "Predictive classification and understanding of weather impact on airport performance through machine learning", Transportation Research Part C: Emerging Technologies, Vol. 131, 2021

Efficient airport operations depend on appropriate actions and reactions to current constraints. Local weather events and their impact on airport performance may have network-wide effects. The classification of expected weather impacts enables efficient consideration in airport operations on a tactical level. The airport performance has been classified with recurrent and convolutional neural networks considering weather data. London–Gatwick Airport has been used to apply the developed approach. The weather data is derived from local meteorological reports and airport performance is derived from both flight plan data and reported delays. The application of machine

learning that has been shown approaches is an appropriate method to quantify the correlation between decreased airport performance and the severity of local weather events. The developed models could achieve prediction accuracy higher than 90% for departure movements. The approach is one key element for a deeper understanding of interdependencies between local and network operations in the air transportation system.

[15] Xinting Zhu, Lishuai Li, "Flight time prediction for fuel loading decisions with a deep learning approach", Transportation Research Part C: Emerging Technologies, Vol. 128, 2021

Excess fuel is loaded by dispatchers and (or) pilots to handle fuel consumption uncertainties, primarily caused by flight time uncertainties, which cannot be predicted by current Flight Planning Systems (FPS). A novel spatial weighted recurrent neural network model was developed to provide better flight time predictions by capturing air traffic information at a national scale based on multiple data sources, including Automatic Dependent Surveillance - Broadcast (ADS-B), Meteorological Aerodrome Reports (METAR), and airline records. A new training procedure associated with the spatial weighted layer is introduced to extract OD-specific spatial weights and then integrate into one model for a nationwide air traffic network. Long short-term memory (LSTM) networks are used after the spatial weighted layer to extract the temporal behavior patterns of network delay states. With the improved flight time prediction, fuel loading can be optimized and resulting reduced fuel consumption by 0.016%–1.915% without increasing the fuel depletion risk.

Chapter 3

SYSTEM ANALYSIS

3.1. EXISTING SYSTEM

For airlines, estimating flight delays correctly is essential since the data may be used to boost customer happiness and revenue for airline agencies. There have been numerous studies on modeling and predicting flight delays, and the majority of them have sought to do so by identifying key traits and closely related elements. However, due to the vast amounts of data, dependencies, and parameters, the majority of the offered approaches are not precise enough.

3.1.1 DISADVANTAGES OF THE EXISTING SYSTEM

- It lacks the necessary characteristics for determining flight delay, which results in less accurate results when determining aircraft delays.
- The majority of earlier studies compared the delay predictions of no more than five machine learning models to investigate aircraft delays.

3.2. PROPOSED SYSTEM

Predicting flight delays is the focus, because it produces the most income for many nations and industries. This mode of transportation is the quickest and most comfortable, making it possible to discover and reduce airline delays and so save a significant amount of time. Based on the results of this simulation, which took into account the time of day, the weather, and other factors that could cause delays in large airports, the number of delays should be kept to a minimum. Different ML techniques/algorithms has been looked at to try to

predict if a flight will be delayed or not before it is even announced on the departure boards. If you go into a plane knowing already that there is a departure delay, chances are that the flight will be late upon arrival. The same happens if you already know that the plane has an arrival delay. So this information was looked at as part of the Exploratory Data Analysis (EDA) but was taken out of the main models. The dataset for flight delay analysis has been considered, where the analytics preparation of the dataset began in order to make it machine-learning format-capable. With the new dataset, the selective teaching technique was applied, which limits itself to accounting for delayed traffic. Then, 20% of the dataset was utilized for validation and 80% for learning. The prediction is made using the test set. With the help of an artificial neural network, the model is trained. The model is optimized using the Adam optimizer technique.

3.2.1. ADVANTAGES OF THE PROPOSED SYSTEM

- Because delays are stochastic, this study looks into the qualitative prediction of airline delays in order to make the required adjustments and improve customer service.
- The suggested approach leverages Principal Component Analysis (PCA), a dimensionality reduction algorithm that speeds up other machine learning models, to optimize the model.

3.3. REQUIREMENTS SPECIFICATION

3.3.1. HARDWARE REQUIREMENTS

- Intel or ARM-based processor, minimum Pentium or equivalent
- Minimum 512MB RAM

- Minimum 4GB disk space

3.3.2. SOFTWARE REQUIREMENTS

JUPYTER NOTEBOOK:

Jupyter notebook is an open-source collaboration technology that can be used to edit and run code using many of the most popular programming languages. Within the realm of programming, notebooks act as a one-stop-shop that allows you to work on all stages of the data analysis process on one interactive page. It offers a simple, streamlined, document-centric experience. Jupyter Notebook also gives the user access to a community of fellow users and open-source programming libraries. Once you begin using it, it is easy to find additional information and instructions on how to use the technology and integrate it into other components that may interest you. Divided into front end and back end interfaces, Jupyter Notebook not only gives users access to the outcome of their code but also assists in the process of tweaking and editing the code before it is executed.

NUMPY:

Numpy is a general-purpose array-processing package. It provides a high- performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. Besides its obvious scientific uses, Numpy can also be used as an efficient multi- dimensional container of generic data.

PANDAS:

Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library. Pandas is fast and it has high performance & productivity for users.

TENSORFLOW/KERAS:

TensorFlow is an open source library created for Python by the Google Brain team. TensorFlow compiles many different algorithms and models together, enabling the user to implement deep neural networks for use in tasks like image recognition/classification and natural language processing. TensorFlow is a powerful framework that functions by implementing a series of processing nodes, each node representing a mathematical operation, with the entire series of nodes being called a "graph".

ANACONDA NAVIGATOR :

Normally Navigator is used online so that it can download and install packages. If Navigator detects that internet access is not available, it automatically enables offline mode. Using Navigator in offline mode is equivalent to using the command line conda commands create, install, remove, and update with the flag --offline so that conda does not connect to the internet. The Home page displays all of the available applications that you can manage

with Navigator. The Environments page allows you to manage installed environments, packages and channels. With Navigator, like with conda, you can create, export, list, remove, and update environments that have differentversions of Python and/or other packages installed. Switching or moving between environments is called activating the environment. Only one environment is active at any point in time.

FLASK:

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2.

PyGAD:

PyGAD is an open-source Python library for building the genetic algorithm and optimizing machine learning algorithms. It works with Keras and PyTorch. PyGAD supports different types of crossover, mutation, and parent selection operators. PyGAD allows different types of problems to be optimized using the genetic algorithm by customizing the fitness function. Besides building the genetic algorithm, it builds and optimizes machine learning algorithms. Currently, PyGAD supports building and training (using genetic algorithm) artificial neural networks for classification problems.

The library is under active development and more features added regularly. Please contact us if you want a feature to be supported.

3.4. LANGUAGE SPECIFICATION

The python programming language is chosen for this project because it offers concise and readable code. Python is an interpreted high-level programming language, it offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive library. Python is a multi-paradigm programming language. Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited too many tasks. Python is a programming language that sets itself apart from others by offering the adaptability, clarity, and dependable tools necessary to develop contemporary software. Python is best suited for machine learning since it is reliable and based on simplicity. Due to its independent platform and widespread use in the programming community, the Python programming language is the most suitable for machine learning.

Chapter 4

SYSTEM IMPLEMENTATION

4.1. SYSTEM ARCHITECTURE

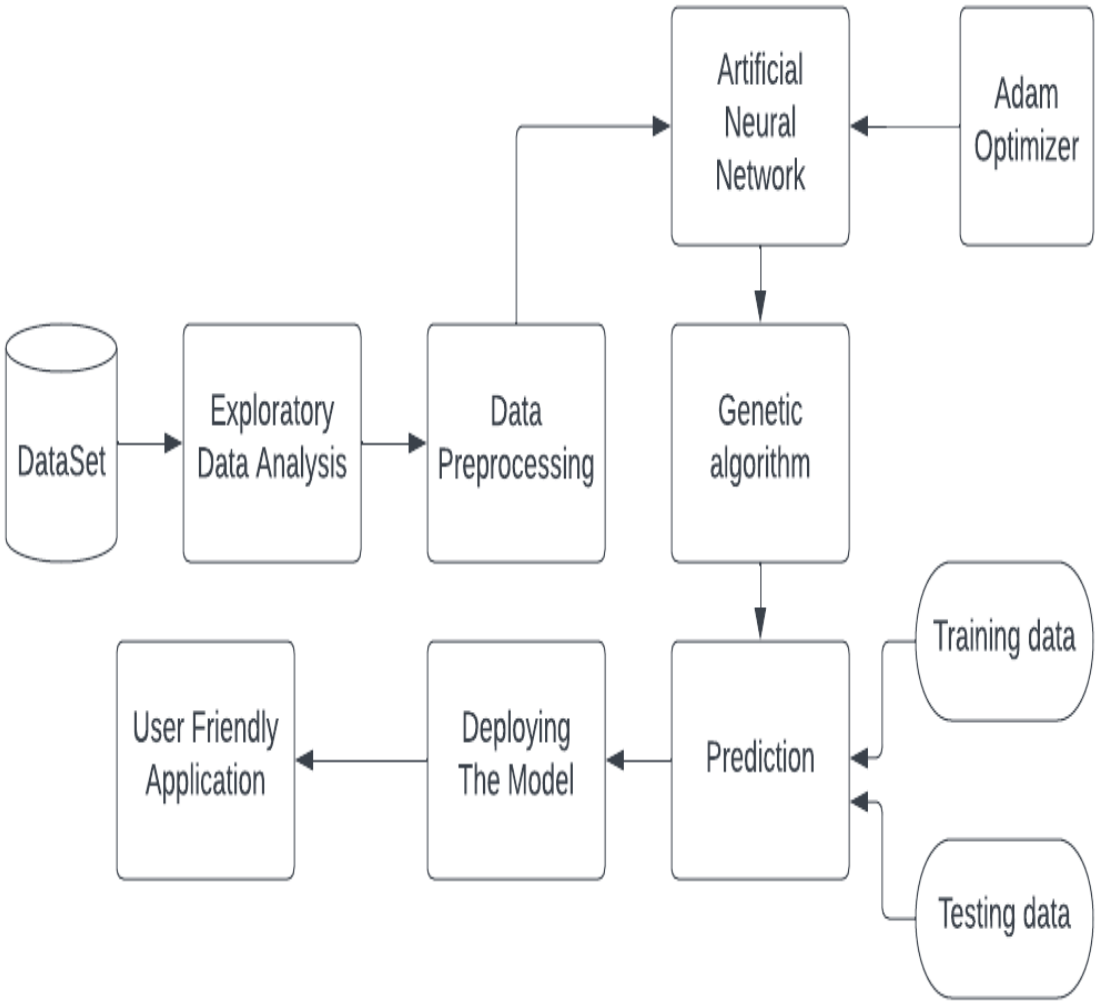


Figure 4.1. System Architecture

a) Dataset Collection:

The process of gathering data depends on the type of project they desire to make, if the project uses real-time data, then we can build a system that uses different sensor data. The data set can be collected from various sources such as a file, database, sensor, and many other such sources but the collected data cannot be used directly for performing the analysis process as there might be a lot of missing data, extremely large values, unorganized text data or noisy data. Therefore, to solve this problem Data Preparation is done.

b) Data pre-processing:

Data pre-processing is one of the most important steps in machine learning. It is the most important step that helps in building machine learning models more accurately. In machine learning, there is an 80/20 rule. Every data scientist should spend 80% time on data pre-processing and 20% time actually performing the analysis. Data pre-processing is a process of cleaning the raw data i.e. the data is collected in the real world and is converted to a clean data set. In other words, whenever the data is gathered from different sources it is collected in a raw format and this data isn't feasible for analysis. Therefore, certain steps are executed to convert the data into a small clean data set, this part of the process is called as data pre-processing.

c) Data Cleaning:

Data pre-processing is a process of cleaning the raw data into clean data, so that can be used to train the model. So, data pre-processing is needed to achieve good results from the applied model in machine learning and deep learning projects.

Most of the real-world data is messy, some of these types of data are:

1. **Missing data:** Missing data can be found when it is not continuously created or due to technical issues in the application (IOT system).

2. **Noisy data:** This type of data is also called outliners, this can occur due to human errors (humans manually gathering the data) or some technical problem of the device at the time of collection of data.

3. **Inconsistent data:** This type of data might be collected due to human errors (mistakes with the name or values) or duplication of data.

Data transformation is the process of converting, cleansing, and structuring data into a usable format that can be analyzed to support decision making processes, and to propel the growth of an organization. Data transformation is used when data needs to be converted to match that of the destination system

d) Exploratory Data Analysis:

Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods. It helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions. EDA is primarily used to see what data can reveal beyond the formal modeling or hypothesis testing task and provides a provides a better understanding of data set variables and the relationships between them. It can also help determine if the statistical techniques you are considering for data analysis are appropriate.

EDA techniques continue to be a widely used method in the data discovery process today.

e) Feature Extraction:

Feature extraction refers to the process of transforming raw data into numerical features that can be processed while preserving the information in the original data set. It yields better results than applying machine learning directly to the raw data. Feature extraction can be accomplished manually or automatically.

f) Building and Training the model:

Building an ML Model requires splitting of data into two sets, such as a 'training set' and 'testing set' in the ratio of 80:20 or 70:30; A set of supervised (for labeled data) and unsupervised (for unlabeled data) algorithms are available to choose from depending on the nature of input data and business outcome to predict. You train the classifier using 'training data set', tune the parameters using 'validation set' and then test the performance of your classifier on an unseen 'test data set'. An important point to note is that during training the classifier only the training and/or validation set is available. The test data set must not be used during the training of the classifier. The test set will only be available during testing of the classifier.

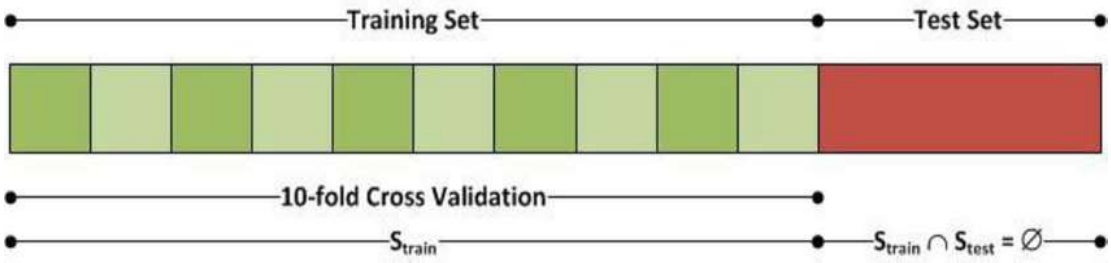


Figure 4.1. Training and testing split

Training set: The training set is the material through which the computer learns how to process information. Machine learning uses algorithms to perform the training part. A set of data used for learning, that is to fit the parameters of the classifier.

Validation set: Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. A set of unseen data is used from the training data to tune the parameters of a classifier.

Test set: A set of unseen data used only to assess the performance of a fully-specified classifier.

g) Model Deployment:

The process of deploying a model is thought to be difficult for data scientists. This is because it is frequently not regarded as their primary duty and because model creation, training, and the organizational tech stack, including versioning, testing, and scaling, make deployment challenging, differ technologically and psychologically from one other. With the appropriate model deployment frameworks, tools, and procedures, these technological and organizational silos can be broken down.

The models can be deployed in production environments to perform prediction either in batch inference mode or on-line inference mode. The batch inference can be achieved by scheduling as a job to run at a time interval and send the results via email to the intended users. The on-line inference can be achieved by exposing the model as a web service using frameworks such as python flask library or streamlit library to develop interactive web applications and invoke the model using its HTTP endpoint.

Chapter 5
MODULE DESCRIPTION

5.1. MODULES

- Dataset Preparation
- Data Preprocessing
- Exploratory Data Analysis
- Model Building
- Comparison of models
- Deploying the model

5.1.1. Dataset Preparation

The dataset that will be examined contains information on every domestic flight that took place in the United States year 2022, across all airports and carriers. The Bureau of Transportation Statistics of the United States Department of Transportation gathered and released the information.

There are over more than 4 million samples in the entire set. Every example comes with 61 characteristics by default, albeit not all of the details are available for every example. Despite this, the dataset is enormous and thorough, which might allow for precise statistical inferences.

5.1.2. Data Pre-processing/Cleaning

It is a routine cleaning with little feature engineering, driven after the 20 most typical arrival destinations were determined based on the number of flights, and it is the one that has the most feature engineering completed. Before beginning the data cleansing, the first step was to specify what would consider a delayed flight. This is crucial since it will affect whether one can eliminate other columns or not and how predictive features will be chosen to utilize.

In terms of engineered features, the first one to be calculated as the target (FLIGHT_STATUS) was whether the flight was delayed or not. This is a binary column, with a 0 for flights arriving on time, and a 1 for flights arriving late, calculated from the "Arrival Delay" (ARR_DELAY) column. With this column ready, the next step was a quick check for the data distribution, meaning, checking if the data is balanced or not. Results are plotted in the below figure and they suggest a severe imbalance dataset with an almost 2:1 ratio, this means right away that looking at accuracy on its own will not be enough to evaluate the models, but I will also need to look at other metrics such as Precision, Recall, and F1.

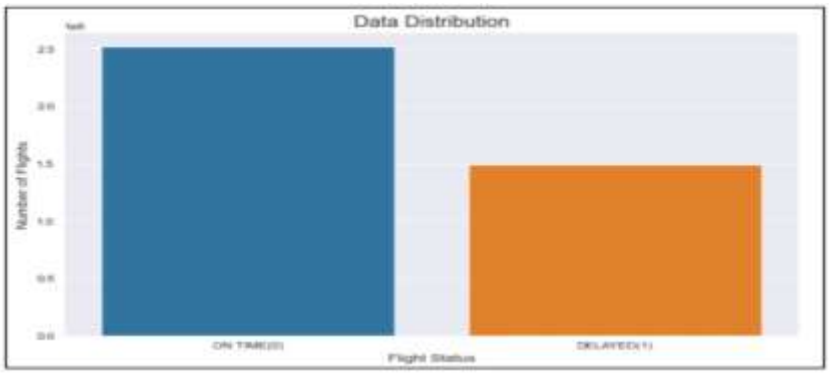


Figure 5.1.2.a) Data distribution

Other features were engineered mainly to perform the EDA. Among those, some of the most relevant were:

- Calculating the total number of flights and the total number of delayed flights (from departure and arrivals separately) by airline
- Extracting the "weekday" from the date using the "DateTime" function from Pandas. Using the same function, the "month" and "day of the month" were also extracted
- Calculating percentages of delayed departures and arrivals by airlines and cities
- Extracting the top destinations with average delays and arrivals
- Calculating the best weekday to travel in terms of delays (departures and arrivals)
- Impact of late departure on arrival time (with the difference between both)

5.1.3. Exploratory Data Analysis

We have to analyze the data in order to find out what component (possible correlations or a lack thereof) would have a substantial impact on delays. Even though each sample in the collection contained 31 items of data, not all of them were timely as in the flight number.

Almost couldn't affect any delay. Additionally, because variables like taxi wait time and wheels-off time are directly tied to departure time (they are a set of

procedures with a generally defined duration), these qualities shouldn't have any bearing on our data set. We chose a number of variables for the data analysis that may have some correlations with our aim and could improve our forecasts.

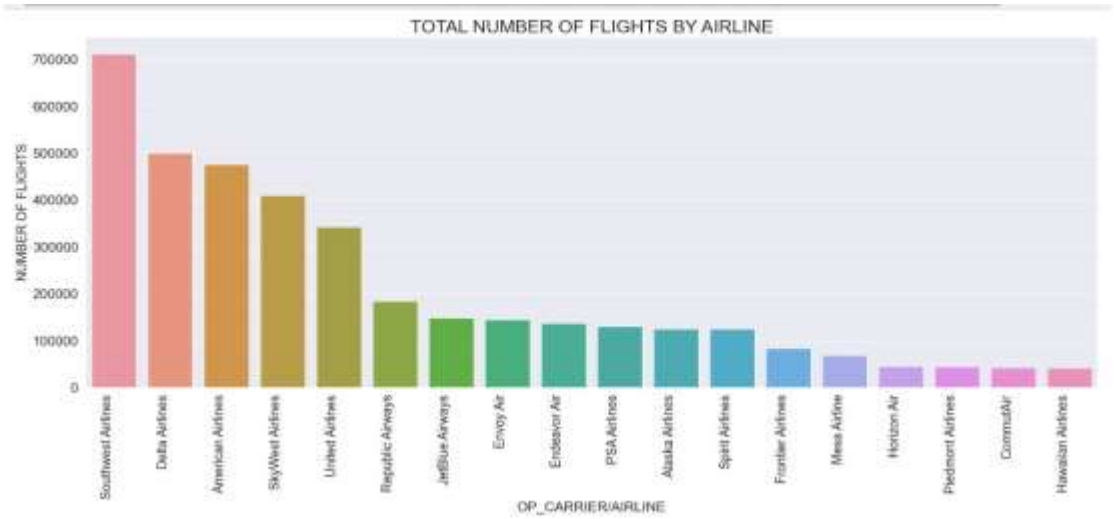


Figure 5.1.3.a) Total number of flights by airline

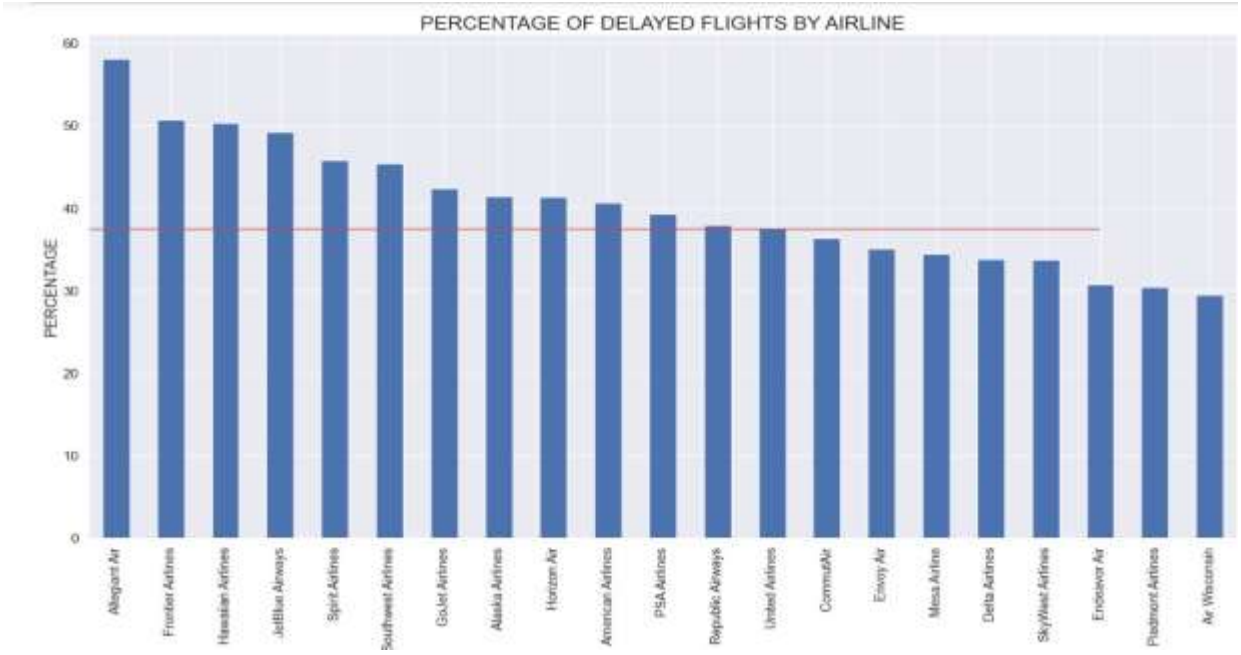


Figure 5.1.3.b) Percentage of delayed flights by airline

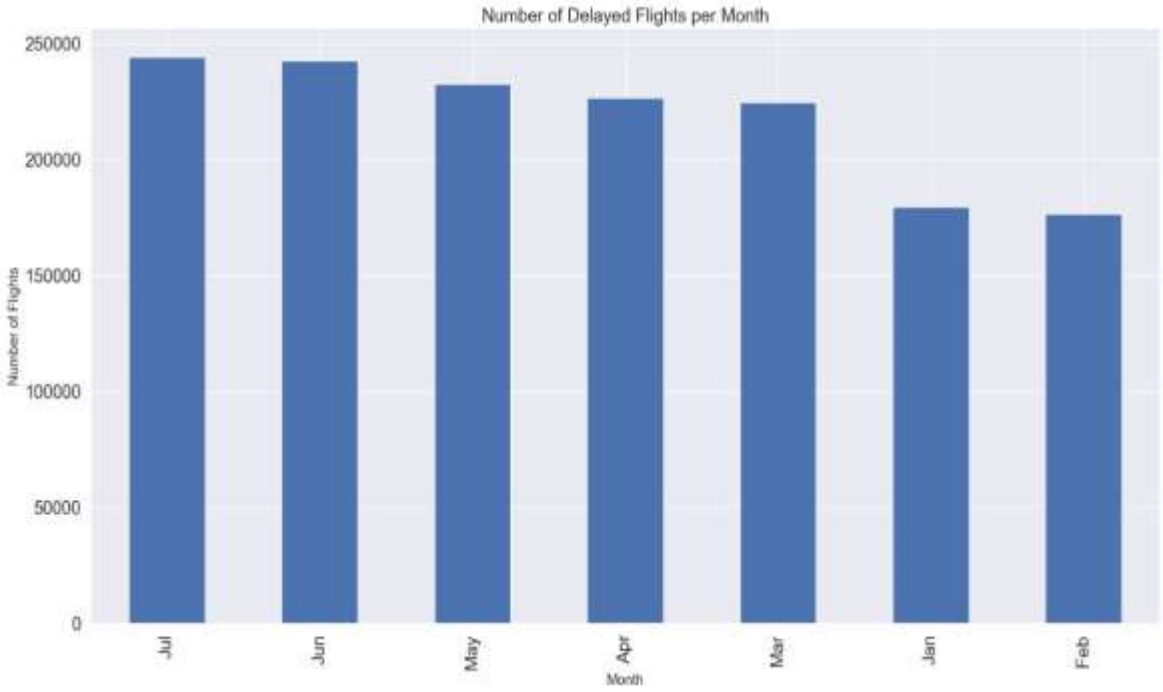


Figure 5.1.3. c) Number of Delayed Flights per Month

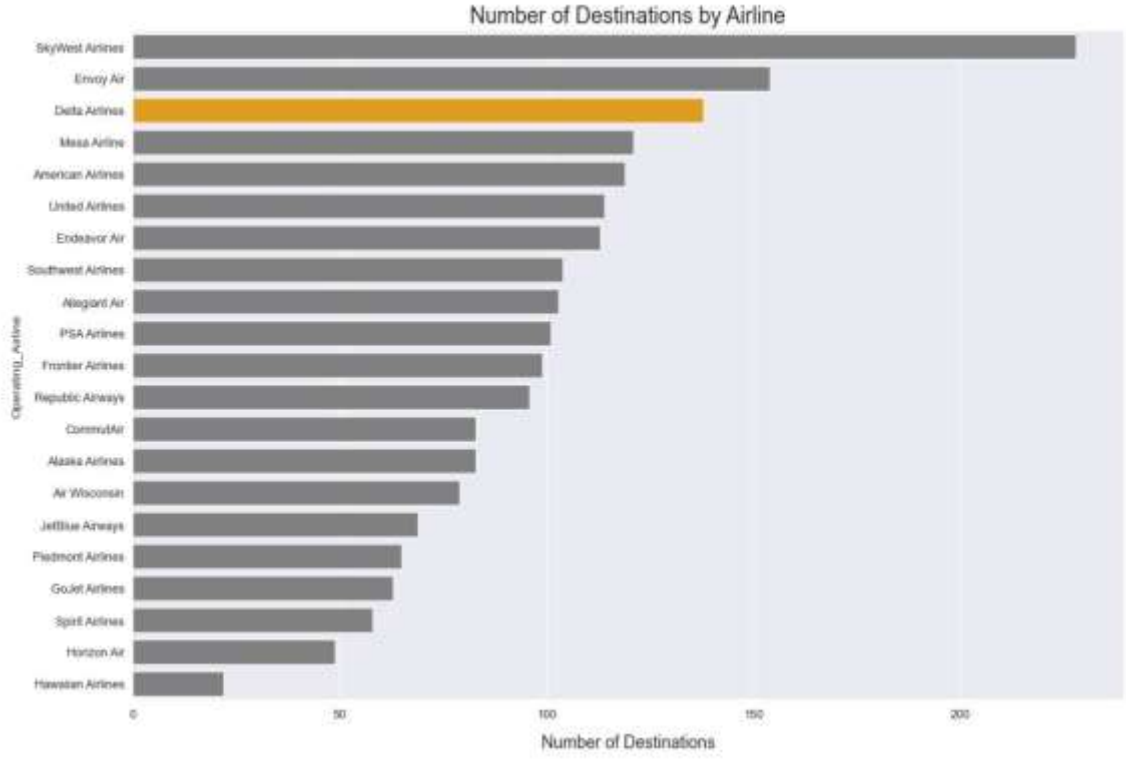


Figure 5.1.3. d) Number of Destinations by Airline

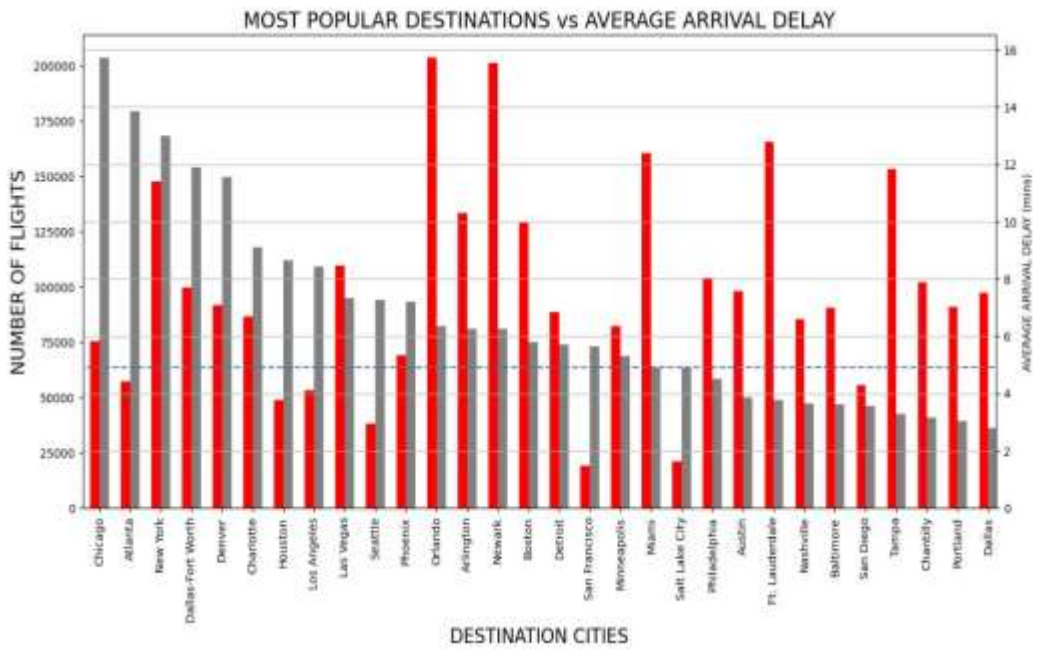


Figure 5.1.3. e) Most popular Destinations vs Average Arrival Delay

The data allows for the following interpretations:

1. There are two seasonal rises in aircraft delays. The first one occurs in the winter, perhaps as a result of the weather and winter breaks. The second is when many individuals travel for a summer vacation in the summer, particularly in June and July.
2. The three largest US airlines—American, United, and Delta—are all less likely to experience delays, with Delta receiving the highest rating in this regard.
3. It is interesting to see how Atlanta, having such a high number of landings, has a very low average delay, whereas Newark, a not-so-popular destination, has such a high minute average delay. San Francisco is another destination that stands out with a high average delay as well as Orlando and Boston.

4. The best months to travel are February, January and September, clearly after the standard holidays (summer and Christmas)
5. Delta Airlines is the second-largest airline by the number of flights and fourth-largest by the number of flights that are delayed, but it has the best proportion of delayed flights at 32%, making it the top airline overall. That amounts to 7% under the cutoff.

After extracting features from the dataset we will generate our algorithms on that dataset.

5.1.4. Model Building

Feature Scaling: Standardisation

For many machine learning algorithms, feature scaling through standardization (or Z-score normalization) can be a crucial preprocessing step. Rescaling the features to give them the characteristics of a standard normal distribution with a mean of zero and a standard deviation of one is what standardization entails.

Algorithms used:

When deep neural networks are trained using gradient-based optimization techniques, disappearing gradients can happen. That happens because of how the neural network is trained using the backpropagation technique. These techniques change each neural network's weight during each training iteration proportionally to the partial derivative of the error function with respect to the current weight. Many optimization issues have been solved using the genetic algorithm (GA), a metaheuristic algorithm influenced by the process of natural

selection in evolutionary algorithms. A population of potential solutions will be started in GA and improved over time. Moreover, some attempts have been made to replace gradient descent-based techniques with GA while training deep neural network models. GA has proven to perform exceptionally well in a variety of learning scenarios, including those with non-convex objective functions with many local optima, non-smooth objective functions, a high number of parameters, and noisy environments. Due to the ease with which the learning process can be implemented on parallel/distributed computing platforms, GA also suits the parallel/distributed computing environment very well. The evolution of the suggested model is shown in the below figure.

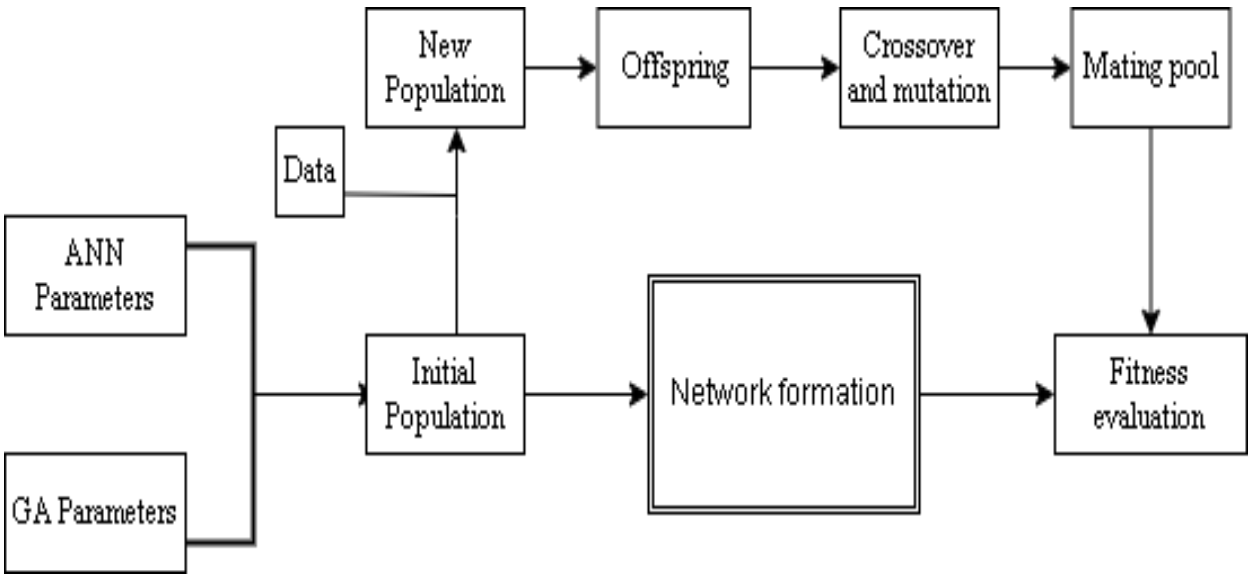


Figure 5.1.4. a) Flow of the algorithm

i) The evolution strategy:

The combination of two germ cells, an egg (female) and a sperm (male), known as fertilization (sexual reproduction) in animals, results in the restoration of the somatic chromosome number and the development of children that exhibit traits from both parents.

Male and female gametes unite and recombine to produce children with unique gene combinations that inherit characteristics from both parents. There are many benefits to sexual reproduction over asexual reproduction, which results in children that are genetically identical to the parent organism. As a result, it has been embraced by the vast majority of plants and animals.

ii) Parent Selection and Mating Pool

The process of choosing parents who will mate and recombine to produce offspring for the following generation is known as parent selection. A loss of diversity results from the solutions being too similar to one another in the solution space, which can be avoided by taking precautions to prevent one exceptionally fit solution from displacing the entire population within a few generations. For a GA to be successful, maintaining good diversity in the population is essential. Premature convergence is the term for this situation in a GA where one exceptionally fit solution absorbs the entire population.

iii) Steady-state selection (SSS, default)

A genetic algorithm's run is typically divided into generations, where the outcome of selection and reproduction replaces all (or at least most) of the population, and only children survive. Yet, only a small number of individuals are replaced at a time in a steady-state genetic algorithm, which means that the majority of the individuals will live to the next generation; there is no such

thing as a generation per se. There are numerous applications. In one variation, for instance, parents are chosen using, for instance, tournament selection. Then, only the best parents and kids are reintroduced into the population after comparison. In a different variation, the best chromosomes are chosen for reproduction and the best individuals are replaced by their progeny.

iv) Roulette wheel selection (RWS)

The casino roulette wheel and the roulette wheel method for parent selection are identical except that in the casino roulette wheel, each pot has an equal chance of containing the ball when the wheel comes to a complete stop. But in this case, we specify the likelihood of each pot (individual of the population). The fitness of an individual is the probability of each person.

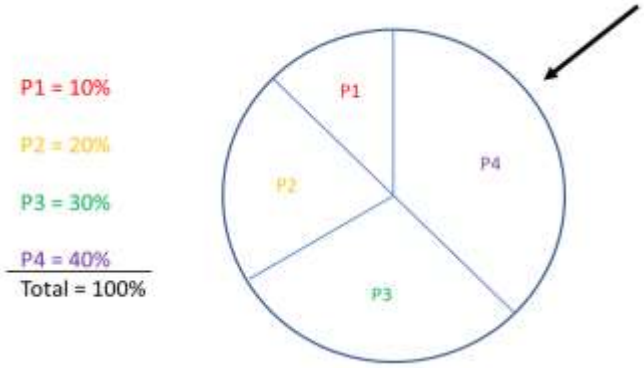
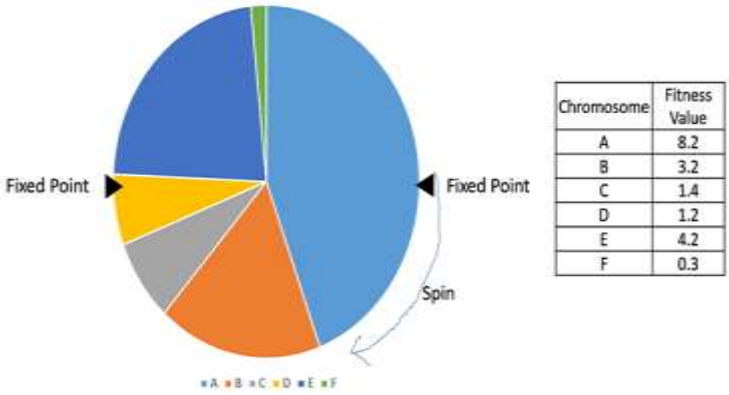


Figure 5.1.4.b) Roulette Wheel

We have four parents, P1, P2, P3, and P4, with respective selection probabilities of 0.1, 0.2, 0.3, and 0.4. The wheel is turned while the arrow is fixed in place. The parent that the arrow points to gets picked for breeding when the wheel stops turning; the larger the region on the wheel, the higher the probability of selection.

v) Stochastic universal sampling selection (sus)

Another fitness proportionate selection variant that shows no bias and little dissemination is SUS. The same roulette wheel is used as in RWS, with the same proportions, but here all the parents are selected at once rather than using a single selection point and rolling the roulette wheel again until all necessary individuals have been selected. In order to do that, the wheel is only spun once, and several selection points distributed evenly around the wheel determine who is drawn.



Chromosome	Fitness Value
A	8.2
B	3.2
C	1.4
D	1.2
E	4.2
F	0.3

Figure 5.1.4. c) Stochastic Universal Sampling

vi) Rank Selection (rank)

When the population's members have relatively similar fitness values, rank selection is frequently employed. It also works with negative fitness values (this happens usually at the end of the run). This results in each person having a nearly equal slice of the pie (like in the case of fitness proportionate selection), as shown in the following image, and as a result, each person, regardless of how fit they are in comparison to one another, has a roughly equal chance of being chosen as a parent. As a result, the selection pressure for fitter people decreases, which forces the GA to choose unsuitable parents in such circumstances.

vii) Crossover and Mutation

The crossover_type defines how children are generated from the selected parents; in other words, how the reproduction works. At the time of writing, PyGAD supports 4 algorithms:

1. crossover_type="single_point": Type of the crossover operation. Supported types are single_point (for a single-point crossover),
2. two_points (for two points crossover),
3. uniform (for a uniform crossover)
4. scattered (for a scattered crossover).

The used crossover algorithm was a uniform crossover. We effectively toss a coin to determine whether each gene is taken from one parent or the other in a uniform crossover. In other words, there is an equal chance of selecting a gene from each parent. For each new offspring, the method is used once more.

viii) Random mutation

A random value from the set of permissible values is assigned to a randomly chosen gene. For random mutation, the random_mutation_min_val parameter specifies the start value of the range from which a random value is selected to be added to the gene. It defaults to -1. This parameter has no action if mutation_type is None. For random mutation, the random_mutation_max_val parameter specifies the end value of the range from which a random value is selected to be added to the gene. It defaults to +1. This parameter has no action if mutation_type is None.

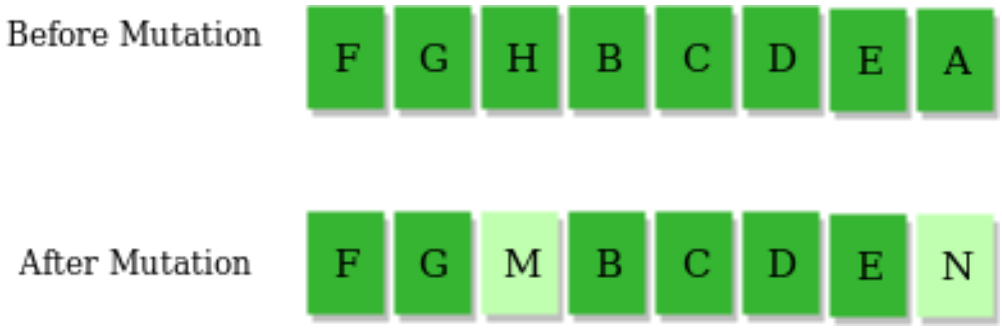


Figure 5.1.4. d) Random mutation

It is used to specify the possible values for each gene in case the user wants to restrict the gene values. It is useful if the gene space is restricted to a certain range or to discrete values. It accepts a list, tuple, range, or numpy.ndarray.

5.1.5. Evaluation of Models:

Performance measurements are required in order to compare several models against one another. Several metrics might be employed depending on the type of challenge. This is a discussion of the metrics most frequently used for each type of machine learning problem:

The different metrics used for classification problems can best be explained via a so-called confusion matrix. For a general binary classification, the confusion matrix is given in table 1. Where the definitions of True Positive (TP), False positive (FP), False Negative(FN), and True Negative (TN) are given, which correspond to what the model predicted and what the actual class was (True or False).

a) Accuracy

Accuracy is the number of correct predictions divided by the total number of made predictions or in terms of elements of the confusion matrix,

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (1)$$

b) Precision

Precision gives a measure of all the positive classes which are predicted, and how many are actually positive. It can be taught as the ability of the model to identify only relevant instances within the dataset. In terms of elements of the confusion matrix, it can be formulated as,

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

c) Recall

Recall gives a measure of all the actual positive classes, and how much was predicted correctly. It can be taught as the ability of a model to find all the relevant instances within the dataset. In terms of elements of the confusion matrix, it can be formulated as

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

d) F1-score

In order to compare models on a single metric, the F1-score is introduced, which is the harmonic mean of precision and recall. It can be computed using

$$F1 - score = \frac{2*Precision*Recall}{Precision+Recall} \quad \text{——————(4)}$$

5.1.6. Deployment of the model

Deploying a machine learning model simply refers to integrating the model into an already-existing production environment that can accept an input and produce a useful output for business decision-making. One component of the data is the model that is built/trained using different techniques on a large dataset. However, implementing machine learning in the actual world requires two steps, the first of which is employing these models in various applications.

The model needs to be deployed online so that users from the outside world can use it in order to forecast the new data. With the help of Flask, a web application was built utilizing the machine learning model ANN-GA .We will build a simple HTML webpage to accept the measurements as input and classify the variety based on the classification model. Although there are competing frameworks on the market, such as FastAPI, flask is still the most popular and well-respected framework among machine learning experts for deploying models.

CHAPTER 6

6.1. CONCLUSION

This initiative and the data gathered from it are helpful for all aviation sector decision-makers, not just from the perspective of the passengers. In addition to the financial losses suffered by the business, flight delays also damage the airlines' reliability and create a bad reputation for them. It contributes to a number of sustainability problems, such as a rise in gas emissions and fuel use. In order to make conclusions about arrival and departure delays and to find associations between flight timings and delays, we conducted Exploratory Data Analysis (EDA) on the flight dataset for this project. The analysis done here not only forecasts delays based on previously known data, but also provides statistical descriptions of airlines, their ranks based on their performance in terms of on-time arrivals, and delays with respect to time, displaying the peak hours of delay using EDA. We have looked into the classification of aircraft delays using neural networks. The findings on identifying flights as delayed or not-delayed over a period of 6 months show an average accuracy of 89% using a delay threshold of 15 minutes. The principal value of the departure delay and the scheduled arrival and departure times of the agents (flights) are used in this method, as opposed to other machine learning methods that call for various features, training sets, and test sets.

6.2. FUTURE ENHANCEMENTS

Data analysis from the first half of 2022 was the foundation for this project. There is a sizable dataset accessible from 2008 through 2022, but managing a larger dataset necessitates extensive data pre-treatment and cleaning. Therefore, adding a larger dataset is a part of this project's future effort. Preprocessing a bigger dataset can be done in a variety of methods, such as establishing a Spark cluster on a computer or using cloud services like AWS and Azure. The project's focus is mostly on aircraft and meteorological data for the United States, but we can also incorporate data from other nations like China, India, and Russia. We may broaden the project's reach by including flight information from international flights rather than simply domestic ones.

APPENDIX I

Deployment:

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="utf-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <title>Flight Delay Prediction</title>
   <link href="https://fonts.googleapis.com/css?family=Lato:400,700" rel="stylesheet">
   <link type="text/css" rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}" />
   <link type="text/css" rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}" />
</head>
<body>
   <div id="booking" class="section">
      <div class="section-center">
```

```
        <div class="container">
            <div class="row">
                <div class="col-md-4">
                    <div class="booking-cta">
                        <h1>Flight Delay Prediction</h1>
                        <p></p>
                        <div class="container">
                            <!-- {{ prediction_text }} -->
{% if prediction_text == 0 %}
                            <h2>The flight is not delayed</h2>
                            {% elif prediction_text == 1 %}
                            <h2>The flight is delayed</h2>
                            {% endif %}
                        </div>
                    </div>
                </div>
                <div class="col-md-7 col-md-offset-1">
                    <div class="booking-form">
                        <form action={{ url_for("predict") }} method="post">
```

```
<div class="row">
  <div class="col-md-4">
    <div class="form-group">
      <span class="form-label">Year</span>
        <input type="text" class="form-control" name="year" placeholder="Enter year" required="true">
        <span class="select-arrow"></span>
    </div>
  </div>
    <div class="col-md-4">
      <div class="form-group">
        <span class="form-label">Month</span>
        <input type="text" class="form-control" name="month" placeholder="Enter month" required="true">
        <span class="select-arrow"></span>
      </div>
    </div>
      <div class="col-md-4">
        <div class="form-group">
          <span class="form-label">Date</span>
```

```
                                <input type="text" class="form-control" name="day" placeholder="Enter date" required="true">
                                    <span class="select-arrow"></span>
                                </div>
                            </div>
                        <div class="row">
                            <div class="col-md-6">
                                <div class="form-group">
                                    <span class="form-label">Select an Airline</span>
                                    <!-- <input class="form-control" type="date" required> -->
                                    <select class="form-control" name="carrier">
                                        <option value="UA">United Air Lines Inc.(UA)</option>
                                        <option value="AA">American Airlines Inc.(AA)</option>
                                        <option value="US">US Airways Inc.(US)</option>
                                        <option value="F9">Frontier Airlines Inc.(F9)</option>
```

<option value="B6">JetBlue Airways(B6)</option>

</select>

Flying from

<!--<input class="form-control" type="text" placeholder="City or airport">-->

<select class="form-control" name="origin">

<option value="EWR">Newark Liberty International Airport(EWR)</option>

<option value="JFK">John F. Kennedy International Airport(New York International Airport)(JFK)</option>

<option value="LGA">LaGuardia Airport(Marine Air Terminal)(LGA)</option>

</select>

```
                </div>
                    <div class="col-md-6">
                        <div class="form-group">
                            <span class="form-label">Flying to</span>
                            <!-- <input class="form-control" type="text" placeholder="City or airport"> -->
                            <select class="form-control" name="dest">
                                <option value="ATL">Hartsfield-Jackson Atlanta International Airport(ATL)</option>
                                <option value="ORD">Chicago O'Hare International Airport(ORD)</option>
                                <option value="LAX">Los Angeles International Airport(LAX)</option>
                                <option value="BOS">Gen. Edward Lawrence Logan International Airport(BOS)</option>
                                <option value="MCO">Orlando International Airport(MCO)</option>
                            </select>
                        </div>
                    </div>
```

```
                    </div>
                    <div class="form-btn">
                        <button class="submit-btn">Predict</button>
                    </div>
                    <!Button>
                  <form>
                </div>
              </div>
            </div>
          </div>
        </div>
    </body>
</html>
```

Styles.css

```
.section {
  position: relative;
  height: 100vh;
```

```
}
.section .section-center {
  position: absolute;
  top: 50%;
  left: 0;
  right: 0;
  -webkit-transform: translateY(-50%);
  transform: translateY(-50%);
}
.booking-form {
  position: relative;
  background: #fff;
  max-width: 642px;
  width: 100%;
  margin: auto;
  padding: 45px 25px 25px;
  border-radius: 4px;
  -webkit-box-shadow: 0px 0px 10px -5px rgba(0, 0, 0, 0.4);
  box-shadow: 0px 0px 10px -5px rgba(0, 0, 0, 0.4); }
```

```
.booking-form .form-group {
  position: relative;
  margin-bottom: 20px;
}
.booking-form .form-control {
  background-color: #fff;
  height: 65px;
  padding: 0px 15px;
  padding-top: 24px;
  color: #191a1e;
  border: 2px solid #dfe5e9;
  font-size: 16px;
  font-weight: 700;
  -webkit-box-shadow: none;
  box-shadow: none;
  border-radius: 4px;
  -webkit-transition: 0.2s all;
  transition: 0.2s all;
}
```

```
.booking-form .form-control::-webkit-input-placeholder {
  color: #dfe5e9;
}
.booking-form .form-control:-ms-input-placeholder {
  color: #dfe5e9;
}
.booking-form .form-control::placeholder {
  color: #dfe5e9;
}
.booking-form .form-control:focus {
  background: #f9fafb;
}
.booking-form input[type="date"].form-control:invalid {
  color: #dfe5e9;
}
.booking-form select.form-control {
  -webkit-appearance: none;
  -moz-appearance: none;
  appearance: none; }
```

```
.booking-form select.form-control+.select-arrow {
  position: absolute;
  right: 6px;
  bottom: 6px;
  width: 32px;
  line-height: 32px;
  height: 32px;
  text-align: center;
  pointer-events: none;
  color: #dfe5e9;
  font-size: 14px;
}
.booking-form select.form-control+.select-arrow:after {
  content: '\279C';
  display: block;
  -webkit-transform: rotate(90deg);
  transform: rotate(90deg);
}
```

```
.booking-form .form-label {
  position: absolute;
  top: 6px;
  left: 20px;
  font-weight: 700;
  text-transform: uppercase;
  line-height: 24px;
  height: 24px;
  font-size: 12px;
  color: #98c9ee;
}
.booking-form .form-checkbox input {
  position: absolute !important;
  margin-left: -9999px !important;
  visibility: hidden !important;
}
.booking-form .form-checkbox label {
  position: relative;
  padding-top: 4px;
```

```
  padding-left: 30px;
  font-weight: 700;
  color: #191a1e;
}
.booking-form .form-checkbox label+label {
  margin-left: 15px;
}
.booking-form .form-checkbox input+span {
  position: absolute;
  left: 2px;
  top: 4px;
  width: 20px;
  height: 20px;
  background: #fff;
  border: 2px solid #dfe5e9;
  border-radius: 50%;
}
.booking-form .form-checkbox input+span:after {
  content: '';
```

```
  position: absolute;
  top: 50%;
  left: 50%;
  width: 0px;
  height: 0px;
  border-radius: 50%;
  background-color: #4fa3e3;
  -webkit-transform: translate(-50%, -50%);
  transform: translate(-50%, -50%);
  -webkit-transition: 0.2s all;
  transition: 0.2s all;
}
.booking-form .form-checkbox input:not(:checked)+span:after {
  opacity: 0;
}
.booking-form .form-checkbox input:checked+span:after {
  opacity: 1;
  width: 10px;
  height: 10px; }
```

```
.booking-form .submit-btn {
  color: #fff;
  background-color: #4fa3e3;
  font-weight: 400;
  height: 65px;
  font-size: 18px;
  border: none;
  width: 100%;
  border-radius: 4px;
  text-transform: uppercase
}
.booking-cta {
  margin-top: 45px;
}
.booking-cta h1 {
  font-size: 52px;
  text-transform: uppercase;
  color: #4fa3e3;
  font-weight: 400; }
```

```
.booking-cta p {
  font-size: 22px;
  color: #191a1e;
```

App.py

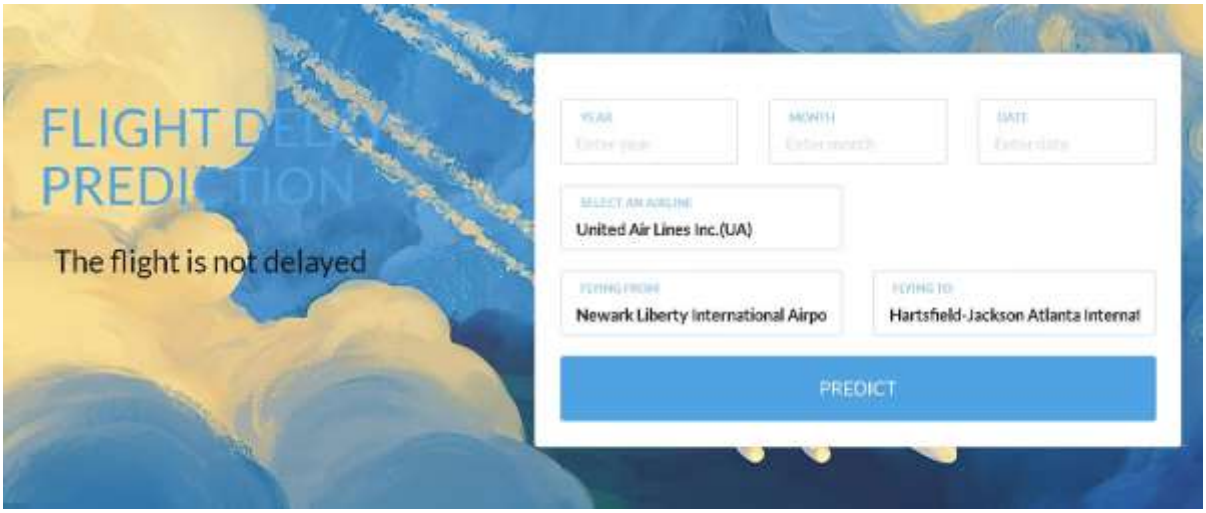
```
from flask import Flask, request, jsonify, render_template, url_for, request
import pickle
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
import pandas as pd
df = pd.read_csv('Data/Processed_data15.csv')
le_carrier = LabelEncoder()
df['carrier'] = le_carrier.fit_transform(df['carrier'])
le_dest = LabelEncoder()
df['dest'] = le_dest.fit_transform(df['dest'])
le_origin = LabelEncoder()
df['origin'] = le_origin.fit_transform(df['origin'])
X = df.iloc[:, 0:6].values
```

```
y = df['delayed']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=61)

app = Flask(__name__)

model = pickle.load(open('modelg.pkl', 'rb'))

@app.route('/')

def home():

    return render_template('index.html')

@app.route('/predict', methods=['POST'])

def predict():

    year = request.form['year']

    month = request.form['month']

    day = request.form['day']

    carrier = request.form['carrier']

    origin = request.form['origin']

    dest = request.form['dest']

    year = int(year)

    month = int(month)

    day = int(day)

    carrier = str(carrier)
```

```
    origin = str(origin)
    dest = str(dest)
    if year >= 2013:
        x1 = [year, month, day]
        x2 = [carrier, origin, dest]
        x1.extend(x2)
        df1 = pd.DataFrame(data=[x1], columns=['year', 'month', 'date', 'carrier',
'origin', 'dest'])
        df1['carrier'] = le_carrier.transform(df1['carrier'])
        df1['origin'] = le_origin.transform(df1['origin'])
        df1['dest'] = le_dest.transform(df1['dest'])
        x = df1.iloc[:, :6].values
        ans = model.predict(x)
        output = ans
    return render_template('index.html', prediction_text=output)
if __name__ == '__main__':
    app.run(debug=False)
```

APPENDIX II

OUTPUT:



REFERENCES

1. Flight delay causality: Machine learning technique in conjunction with random parameter statistical analysis ,Seyedmirsajad , Mokhtarimousavi , ArminMehrabi , International Journal Of Transportation Science And Technology, 2022

2. Social ski driver conditional autoregressive-based deep learning classifier for flight delay prediction, Desmond BalaBisandu, Irene Moulitsas& Salvatore Filippone, Neural Computing And Applications 8777-8802, 2022

3. Predicting flight delay with spatio-temporal trajectory convolutional network and airport situational awareness map, Wei Shaoa, Arian Prabowoab, SichenZhaoa, PiotrKoniuszbc, Flora D.Salima, NeuroComputing Volume 472, 2022

4. An Adaptive Framework for Optimization and Prediction of Air Traffic Management (Sub-)Systems with Machine Learning, Stefan Reitmann and Michael Schultz, Application Of Data Science To Aviation , 2022

5. Analyzing flight delay prediction under concept drift, Lucas Giusti, Leonardo Carvalho, Antonio Tadeu Gomes, Rafaelli Coutinho, Jorge Soares & Eduardo Ogasawara, Evolving System 13,723-736, 2022

6. Assessing Identifiability in Airport Delay Propagation Roles Through Deep Learning Classification, Ilinka Ivanoska; Luisina Pastorino; Massimiliano Zanin, Journals And Magazines IEEE Volume 10, 2022

7. Flight Delay Prediction Based With Machine Learning, Hatıpoğlu, Irmak; Tosun, Ömür; Tosun, Nedret, LogForum 18 (1), 2022

8. Probabilistic Flight Delay Predictions Using Machine Learning and Applications to the Flight-to-Gate Assignment Problem , MichaZoutendijkMihaelaMitici, AeroSpace 8(6) , 152, 2021

9. Using causal machine learning for predicting the risk of flight delays in air transportation ,Dothang Truong, Journal of Air Transportation Management , 2021

10. Hierarchical integrated machine learning model for predicting flight departure delays and duration in series , Waqar Ahmed Khan,Hoi-LamMa,Sai-HoChung,XinWen, Tranportation Part C : Emerging Technologies 129 , 103225 , 2021

11. A Deep Learning Approach for Flight Delay Prediction Through Time-Evolving Graphs , KaiquanCai, Yue Li, Yi-Ping Fang, YanboZhu,IEEE Transactions on Intelligent Transportation System , 2021

12. A Multilayer Perceptron Neural Network with Selective-Data Training for Flight Arrival Delay Prediction , HajarAlla,LahcenMoumoun,and Youssef Balouki , Scientific Programming , 2021

13. A novel hybrid method for flight departure delay prediction using Random Forest Regression and Maximal Information Coefficient , ZhenGuo,BinYu,MengyanHao,WensiWang,YuJiang,FangZong, Aerospace Science and Technology 116, 2021

14. Predictive classification and understanding of weather impact on airport performance through machine learning ,MichaelSchultz,StefanReitmann,SameerAlam, Transportation Research Part C: Emerging Technologies 131, 2021

15. Flight time prediction for fuel loading decisions with a deep learning approach ,XintingZhu,LishuaiLi, Transportation Research Part C: Emerging Technologies 128, 2021