

AVIATION INFORMATION EVALUTION UTILIZING MACHINE LEARNING TECHNIQUES

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

KATIKIREDDY JAYA SHANKAR SWAMI SANDEEP (40110573)

CHITTINNI GOPICHANDU (40110278)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY -1 UNIVERSITY BY UGC

**Accredited with Grade “A++” by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

BONAFIDE CERTIFICATE

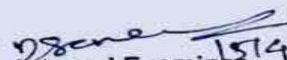
This is to certify that this Project Report is the bonafide work of **Katikireddy Jaya Shankar Swami sandeep (40110573)**, **Chittinni Gopichandu (40110278)** who carried out the Project entitled "**AVIATION INFORMATION EVALUATION UTILIZING MACHINE LEARNING TECHINQUES** " under my supervision from November 2023 to April 2024.

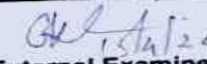

Internal Guide

Dr. T. JUDGI, M.E., Ph.D.


Head of the Department
HEAD OF THE DEPARTMENT
Dr. L. LAKSHMANAN, M.E., Ph.D.
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Jeppiaar Nagar, Rajiv Gandhi Salai,
Chennai - 600 119

Submitted for Viva voce Examination held on 15-04-2024


Internal Examiner


External Examiner

DECLARATION

I, **Katikireddy Jaya Shankar Swami Sandeep(40110573)**, hereby declare that the Project Report entitled **"AVIATION INFORMATION EVALUATION UTILIZING MACHINE LEARNING TECHNOLOGIES"** has been done by me under the guidance of **Dr. T. Judgi, M.E., Ph.D** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 15/04/2024

PLACE: Chennai


SIGNATURE OF The CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. T. Judgi, M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-1 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

To improve an airline's efficiency, it is essential to accurately anticipate flight delays. The majority of forecasting methods are used to anticipate flight delays in one way or another, and recent research in business has focused on the use of gadget learning skills. This article discusses additional outcomes from machine learning and the various factors that influence how Flight slows electricity waft. Instances of advanced issues foreseeing ordinary flight delays. To make a dataset Reason for the Mechanized Ward Observation B (Advertisements B) message design Gotten ahead of time and incorporated with various measurements like climate Status, flight plan and air terminal measurements. Made prescient undertakings. They have remarkable pieces of arrangement and relapse work. Test results show that Long-fast time span memory (LSTM) might method at any point flight Information assortment, but the inquiry is repetitive in our limited data set. To analyze Likewise with past plans, the irregular forest model can procure a superior point Expectation exactness (90.2% for twofold kind) and agreeable Perhaps re-framework. Flight delays often create exasperation in airports when not properly mobilized. Increasing development of machine learning models provoke the researchers and scientists to harness the modern research problems. Due to an increase in customer satisfaction in the air transportation system, there needs a proper decision-making process to mobilize the air-traffic with minimum delay. It is recorded that 19% of United States domestic flights reach their destination with an average delay of 15 minutes. Moreover, the complexity of the air transportation system limits the availability of accurate prediction models. Due to the stochastic nature of delays, this research investigates the qualitative prediction of airline delays to implement necessary changes and provide better customer experience.

Chapter no.	TITLE	Page no.
	ABSTRACT	V
	LIST OF FIGURES	viii
1	INTRODUCTION	1
2	LITERATURE SURVEY	3
	2.1 Inferences from Literature Survey	5
	2.2 Open problems in Existing System	5
3	REQUIREMENTS ANALYSIS	7
	3.1 Feasibility Studies/Risk Analysis of the Project	7
	3.2 Software Requirements Specification Document	9
	3.3 System Requirements	11
4	DESCRIPTION OF PROPOSED SYSTEM	12
	4.1 Selected Methodology or process model	17
	4.2 Data Flow Diagram	17
	4.2.1 Use case Diagram	18
	4.2.2 Sequence Diagram	19
	4.2.3 Activity Diagram	20
	4.3 Architecture / Overall Designed of Proposed System	21
	Description of software for implementation and	
	4.4 testing plan of the proposed model/system	21
	4.4.1 Unit Testing	23
	4.4.2 Integration Testing	23
	4.4.3 Acceptance Testing	24
	4.5 Algorithms Used	24
	4.5.1 Logistic Regression	24
	4.5.2 Random Forest	25
	4.5.3 KNN Algorithm	26
5	Results And Discussion	27
6	Conclusions	30

LIST OF FIGURES

FIGURE NO	FIGURE NAME	Page No.
3.1	Script as login html	10
4.1	Data flow diagram	17
4.2	Usecase diagram	18
4.3	Sequence Diagram	19
4.4	Activity Diagram	20
4.5	System Architecture	21
5.1	GUI Apperance	27
5.2	Login Page	28
5.3	Main Page	29
B.1	User Login page	37
B.2	Data set uploader	37
B.3	Training the data	38
B.4	Final Result	38

CHAPTER 1

INTRODUCTION

Airline data analysis can help identify patterns and factors that contribute to flight delays. By analyzing historical data on factors like weather, airport congestion, and maintenance issues, airlines can gain insights into delay propagation. This can aid in optimizing operations, predicting delays, and making more informed decisions to minimize the impact of delays on passengers and the overall flight schedule.

The dataset became downloaded from Kaggle and contains realities from 3 CSV reports. From 2019 to 2021. It contained realities around aviation routes; Postpone realities, place measurements (beginning and objective) and crossing out (New sections noted). He also had a few specialized subtleties the time while there was a plane on the earth. My strategy depends absolutely on Apache Flash, uniquely the use of PySpark with AWS. Bunch support EMR. This virtual establishment involves in-memory dispensed data. Quantitative instructions in a framework for accelerating multiple MGE tasks. With an incorporated examination framework, this grants us to parallelize on a major scale. Disseminated during the MGE group. PySpark is in like manner awesome for making exploratory examination at scale. Instruments for getting to know pipelines and building ETL (Concentrate, Change, and Burden) gear. . I connected the EMR cluster to Jupyterlab in magazine layout once it is "operating." Data was imported from the Kaggle website and downloaded from the S3 bucket and AWS offerings. I performed exploratory examination and prescient displaying. Passenger Airlines, Cargo Airlines and Air Traffic Control Systems are the principal additives of any transportation system In the current world. People all around the global have Over time, efforts have been made to develop various strategies Improves air transport system. He has This has drastically modified the way airways perform. The modern traveler Flight delays are annoying.

Round 20 have become a flying commercial aircraft Very complex and dynamic. Activities of the day Airlines need to alter their reaction Weather situations, mechanical troubles or Complaints of passengers. These variables also can affect trajectories. And schedules, growing the variety of flight operations Commercial airports. Managing complex interactions Between passengers, flights, airports and desires Airline companions are a venture for airways and road shipping Flow managers at commercial airports have to reply Respond fast to sudden modifications in call for. It is said that the landing or flight may be not on time Departure after scheduled arrival or departure time respectively. Traditionally, flight departure time or 15 minutes later than expected Departure and arrival time respectively, then this A postpone in departure or arrival might be taken into consideration Associated with respective airports. Known reasons Commercial flight delays are inconvenient Weather, air traffic congestion, past due arrival The plane used for the previous flight, Management and security problems. One of the principal industries A key issues facing airlines is price Associated with flight delays because of nature Accidents and defects, viz A high stage flight agreement creates scheduling troubles and operations for end customers, as a result A dangerous popularity and disillusioned clients. As all of us understand we will now not put off the flight first Departing as customers of the plane or Airport staff will acquire flight delays the evaluation supported various situations. We can use it If the air service reviews a flight put off and so try to avoid this. It prevents to keep away from any loss to clients and airlines, it may be to your unfastened time or professional time. One of the key business issues that airlines face is that the vital prices that are related to flights being delayed because of natural occurrences and operational shortcomings that is an upscale affair for the airlines, making issues in scheduling and operations for the end users therefore inflicting unhealthy name and client discontent. As we all know that we have a tendency to not get the flight delay before departure as customers of the Airline Company neither the airline company's ground staff gets the airline delay prediction supported varied conditions. However, we all know that one in all the most reasons for delay in flights is that the weather. This motivates us to use the live weather knowledge in conjunction with different metrics to calculate the delay on the wing before departure

CHAPTER 2

LITERATURE SURVEY

In[1] which was published in the International Journal of Innovative Research in Computer and Communication Engineering in 2019. Their paper serves as a valuable resource for understanding the application of machine learning in addressing flight delay prediction challenges. By systematically reviewing existing literature, they shed light on various methodologies, algorithms, and approaches employed in this domain. The authors likely discuss a range of machine learning algorithms utilized for flight delay prediction, such as Random Forest, logistic regression. The authors may address the challenges encountered in flight delay prediction using machine learning techniques, such as data quality, model interpretability, scalability, etc. Additionally, they may suggest potential future research directions to overcome these challenges and enhance prediction capabilities

In[2] published in the Journal of Aerospace Information Systems in 2016, conducted a comparative study of several machine learning algorithms for flight delay prediction. Here's a breakdown of what their paper likely covers. The authors likely provide an overview of the significance of flight delay prediction in the aviation industry and the challenges associated with it. They would discuss various machine learning algorithms considered for flight delay prediction, such as Random Forest, Support Vector Machine (SVM), k-Nearest Neighbors (k-NN), Decision Trees, etc. Each algorithm's working principle, strengths, and limitations would likely be explained. The authors would detail the experimental setup, including how the dataset was split into training and testing sets, any preprocessing steps applied, and the parameters chosen for each machine learning algorithm.

In[3] "A case study of Indian airports," presented at the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon) in 2019, compares the performance of various machine learning algorithms for predicting flight delays at Indian airports. The authors likely provided an overview of the significance of flight delay prediction, its impact on the aviation industry, and the specific challenges faced by Indian airports in this regard. Details about how the dataset was split into training and testing sets, any preprocessing steps applied, and the parameters chosen for each machine learning algorithm would have been provided. The authors would have likely analyzed the results obtained and discussed the strengths and weaknesses of each machine learning algorithm for flight delay prediction at Indian airports. Insights into which algorithms performed best under certain conditions or datasets may have been provided.

In[4] published journals and their corresponding years may not be readily available in the public domain. However, you can typically find this information by searching academic databases like Google Scholar, IEEE Xplore, or PubMed. These databases index research articles, including those authored by Dr. Bhateja, along with their publication years and journal titles. If you have access to academic libraries or institutional subscriptions, you may also be able to access full-text articles for further exploration. His work extends beyond theoretical research, often involving collaborations with airlines, airports, and aviation authorities to implement and validate predictive models in real-world scenarios. This practical approach ensures that his research has tangible benefits for the aviation industry, helping to optimize operations and enhance passenger satisfaction.

In[5] Supervised machine learning algorithms have been used extensively in different domains of machine learning like pattern recognition, data mining and machine translation. Similarly, there have been several attempts to apply the various supervised or unsupervised machine learning algorithms to the analysis of air traffic data. However, no attempts have been made to apply Gradient Boosted Decision Tree, one of the famous machine learning tools to analyze those air traffic data. This paper investigates the effectiveness of this successful paradigm in the air traffic delay prediction tasks. By combining this regression model based on the

machine learning paradigm, an accurate and sturdy prediction model has been built which enables an elaborated analysis of the patterns in air traffic delays. Gradient Boosted Decision Tree has shown a great accuracy in modeling sequential data. With the help of this model, day-to-day sequences of the departure and arrival flight delays of an individual airport can be predicted efficiently. In this paper, the model has been implemented on the Passenger Flight on-time Performance data taken from the U.S. Department of Transportation to predict the arrival and departure delays in flights. It shows better accuracy as compared to other methods.

In [6] This paper presents a new class of models for predicting air traffic delays. The proposed models consider both temporal and spatial (that is, network) delay states as explanatory variables, and use Random Forest algorithms to predict departure delays 2–24 h in the future. In addition to local delay variables that describe the arrival or departure delay states of the most influential airports and links (origin–destination pairs) in the network, new network delay variables that characterize the global delay state of the entire National Airspace System at the time of prediction are proposed. The paper analyzes the performance of the proposed prediction models in both classifying delays as above or below a certain threshold, as well as predicting delay values. The models are trained and validated on operational data from 2007 and 2008, and are evaluated using the 100 most-delayed links in the system. The results show that for a 2-h forecast horizon, the average test error over these 100 links is 19% when classifying delays as above or below 60 min. Similarly, the average over these 100 links of the median test error is found to be 21 min when predicting departure delays for a 2-h forecast horizon. The effects of changes in the classification threshold and forecast horizon on prediction performance are studied.

In [7], K nearest neighbor (kNN) method is a popular classification method in data mining and statistics because of its simple implementation and significant classification performance. However, it is impractical for traditional kNN methods to assign a fixed k value (even though set by experts) to all test samples. Previous solutions assign different k values to different test samples by the cross validation method but are usually time-consuming. This paper proposes a kTree method to learn different optimal k values for different test/new samples, by involving a

training stage in the kNN classification. Specifically, in the training stage, kTree method first learns optimal k values for all training samples by a new sparse reconstruction model, and then constructs a decision tree (namely, kTree) using training samples and the learned optimal k values. In the test stage, the kTree fast outputs the optimal k value for each test sample, and then, the kNN classification can be conducted using the learned optimal k value and all training samples. As a result, the proposed kTree method has a similar running cost but higher classification accuracy, compared with traditional kNN methods, which assign a fixed k value to all test samples. Moreover, the proposed kTree method needs less running cost but achieves similar classification accuracy, compared with the newly kNN methods, which assign different k values to different test samples. This paper further proposes an improvement version of kTree method (namely, k*Tree method) to speed its test stage by extra storing the information of the training samples in the leaf nodes of kTree, such as the training samples located in the leaf nodes, their kNNs, and the nearest neighbor of these kNNs. We call the resulting decision tree as k*Tree, which enables to conduct kNN classification using a subset of the training samples in the leaf nodes rather than all training samples used in the newly.

In[8] The flight delay prediction problem can be treated by different points of view: (i) delay propagation, (ii) root delay and cancellation. In delay propagation, one study how delay propagates through the network of the transportation system. On the other hand, considering that new problems may happen eventually, it is also important to predict further delays and understand their causes. Such occurrences, in this paper, are named as a root delay problem. Finally, under specific situations, delays can lead to cancellations, forcing airlines and passengers to reschedule their itineraries. So, researchers focused on cancellation analysis try to figure out which conditions lead to cancellations. Moreover, it explores the airlines' decision-making process for choosing the flights to be canceled.

In[9] A study by Chia et al. found that the combination of machine learning and feature engineering is effective in classifying sarcasm and irony in social media texts and detecting cyberbullying. The research process involves creating new features from raw data and using machine learning algorithms to classify the texts. The results showed that the approach was successful in classifying sarcasm and irony and had high precision and recall rates for detecting cyberbullying. However,

there is potential for further research to explore other machine learning algorithms, and to apply the approach to other languages and domains.

[10] In this paper, we endorse the improvement of data analytics to come across uncommon flight styles from large volumes of flight information facts (FDR) information to support aircraft operations. The first motive behind this decision is that there are possible issues with the mechanical components in flight. Evidence of these troubles is regularly covered in Roosevelt's information. Therefore, by means of solving the FDR records, they are able to find capacity issues in flight earlier than they occur. For this, records filtering, data modeling and facts transformation are constantly achieved inside the records preprocessing stage. In addition, in this evaluation, all-time series records in FDR is classed into 3 types: continuous sign, discrete signal and caution signal. For every sign kind, a multidimensional vector is chosen, organizing the time series records into capabilities. In the characteristic segmentation procedure, correlation evaluation, relationship relaxation, and dimensional extension are achieved continuously. Finally, a k-nearest classifier is used to mechanically pick out FDR information wherein abnormal flight styles are recorded from a big set of FDR statistics. The proposed technique is examined on real FDR facts using NASA's public database.

2.1 INFERENCES FROM LITREATURE SURVEY

The major problem faced in most of the projects is with the collected data sets in which redundant and improper values may present. These redundant and inaccurate values are solved in successive projects using different types of supervised and un-supervised machine learning techniques. From the above references on flight delay prediction, many algorithms and techniques like clustering, decision tree, naïve-bayes classifier are been used. Further, KNN algorithm has been implemented in this project to get better accuracy. KNN's decision boundary can take any form. KNN is good with correlated attributes, if the distinguishing characteristic of classification is not the marginal distributions but correlation. KNN Classifier can be updated online at low cost, where new instances with known classes are presented. Accuracy is better as compared to other algorithms and time consumption is also less.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

In Existing system, an airport has huge amount of data related to number of flights, data and time of arrival and dispatch, flight routes, number of airports operating in each country, list of active airlines in each country. The problem they faced till now it's, they have ability to analyze limited data from databases. For instance, L. Hao et al. used a regression model for the three major commercial airports in New York to predict flight delay. However, several reasons are restricting the existing methods from improving the accuracy of the flight delay prediction.

DISADVANTAGES OF EXISTING SYSTEM

The diversity of causes affecting the flight delay, the complexity of the causes, the relevancy between causes, and the insufficiency of available flight data.

RISK AND ANALYSIS OF PROJECT

The objective of the proposed system is to predict the delays of flights. The system will be using 30% of the data for training the classification models and on the basis of that further delays can be predicted.

This prediction will be done using machine learning techniques in the end, the one with most accuracy will be considered as an efficient model and employed.

Therefore, predicting flight delays can improve airline operations and passenger satisfaction, which will result in a positive impact on the economy.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 RISK AND ANALYSIS OF PROJECT/FEASIBILITY STUDIES

FEASIBILITY STUDY:

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

- **ECONOMICAL FEASIBILITY**
- **TECHNICAL FEASIBILITY**
- **SOCIAL FEASIBILITY**

1.ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2.TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3. TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

RISK ANALYSIS:

The objective of the proposed system is to predict the delays of flights. The system will be using 30% of the data for training the classification models and on the basis of that further delays can be predicted. This prediction will be done using machine learning techniques in the end, the one with most accuracy will be considered as an efficient model and employed. Therefore, predicting flight delays can improve airline operations and passenger satisfaction, which will result in a positive impact on the economy.

1. Data Collection:

Obtain historical flight data including departure and arrival times, weather conditions, airport congestion, aircraft type, airline, route, etc. Sources could include government databases, airline websites, or specialized aviation data providers.

2. Data Preprocessing:

- Clean the data to handle missing values, outliers, and inconsistencies.
- Convert categorical variables into numerical representations through techniques like one-hot encoding.
- Normalize or scale numerical features to ensure they have similar ranges.

3. Feature Engineering:

- Extract relevant features from the data that could influence flight delays, such as:
- Time of day, day of week, month, etc.
- Weather conditions at departure and arrival airports.

- Previous flight delays for the same route or airline.
- Airport congestion metrics.
- You can also consider incorporating external data sources like holidays, events, or even social media sentiment if available and relevant.

4.Model Selection:

- Choose appropriate machine learning models for the task. Common choices include:
- Regression models like Linear Regression, Decision Trees, Random Forests, or Gradient Boosting Machines.
- Time-series forecasting models like ARIMA (AutoRegressive Integrated Moving Average) or LSTM (Long Short-Term Memory) neural networks.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

Python: Python is a dynamic, high-level programming language known for its readability and versatility. In the context of "Chatbot," Python serves as the primary language for implementing the backend logic, machine learning models, and other AI components. Python's extensive ecosystem includes powerful libraries such as TensorFlow, scikit-learn, and OpenCV, making it well-suited for developing sophisticated AI applications. Its ease of integration with various technologies contributes to the project's agility and adaptability.

Python Features:

Python's features include –

- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python

interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Jupyter Notebook: Jupyter Notebook is an open-source web application that allows you to create and share documents containing live code, equations, visualizations, and explanatory text. It provides an interactive computing environment that supports various programming languages, including Python, R, and Julia, among others. The key feature of Jupyter Notebook is its ability to combine executable code with rich text elements, enabling users to create interactive and reproducible data analysis workflows. With Jupyter Notebook, you can seamlessly integrate code snippets with descriptive narrative, plots, and mathematical expressions, making it an ideal tool for data exploration, prototyping machine learning models, conducting statistical analyses, and documenting research findings. Its versatility and user-friendly interface make it popular among data scientists, researchers, educators, and professionals across different domains. Additionally, Jupyter Notebook supports extensions and plugins, allowing users to customize their environment and enhance productivity. Whether you're working on a small-scale data analysis project or collaborating on a large-scale research endeavor, Jupyter Notebook offers a flexible and powerful platform for interactive computing and documentation.

Flask Framework: Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine.

Both are Pocco projects.

Http protocol is the foundation of data communication in world wide web. Different methods of data retrieval from specified URL are defined in this protocol.

By default, the Flask route responds to the **GET** requests. However, this preference can be altered by providing methods argument to **route()** decorator.

In order to demonstrate the use of **POST** method in URL routing, first let us create an HTML form and use the **POST** method to send form data to a URL.

```
<html>

<body>

<form action="http://localhost:5000/login" method="post">

<p>Enter Name:</p>

<p><input type="text" name="nm"/></p>

<p><input type="submit" value="submit"/></p>

</form>

</body>

</html>
```

fig :3.1 Save the following script as login html

3.3 SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS

- System: Pentium Dual Core.
- Hard Disk: 120 GB.
- Monitor: 15" LED`
- Input Devices: Keyboard, Mouse
- Ram: 4 GB.

SOFTWARE REQUIREMENTS

- Operating system: Windows 7/10.
- Coding Language: Pytho

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

In proposed System, explores a broader scope of factors which may potentially influence the flight delay, and compares several machine learning-based models in designed generalized flight delay prediction tasks. To build a dataset for the proposed scheme, automatic dependent surveillance broadcast (ADS-B) messages are received, pre-processed, and integrated with other information such as, flight schedule, and airport information. The designed prediction tasks contain different classification tasks and a regression task. Experimental results show that long short-term memory (LSTM) is capable of handling the obtained aviation sequence data, but overfitting problem occurs in our limited dataset. Compared with the previous schemes, the proposed random forest Logistic Regression, Decision Tree-based model can obtain higher prediction accuracy (90.2% for the binary classification) and can overcome the overfitting problem.

Advantages of Proposed System:

- By using airline data analysis we can save time of users
- Cost Savings

4.1 SELECTING METHODOLOGY OR PROCESS MODEL

- Data Collection
- Data preprocessing
- Statistical Data Analysis
- Model selection
- Classification/Prediction

Data Collection:

To get a quick glance at the data, we loaded the 2021 csv file into a Resilient Distributed Dataset (RDD) which is Spark's representation of a dataset that is

distributed across RAM, in the memory of a cluster of many machines. Through this Spark session, I found that there was about 27 unnamed variables and multiple null values. After combining the dataset (2021 data), I created 3 major categorical CSV files namely: flight-cancellation, flight-delays, and flight-profile. This was essential to debug.

Data Preprocessing:

Originally, there was 28 variables to work with. After removing the unnamed columns, the remaining 19 were checked for null values (as shown below). In addition, only columns with relevant data about the flight profile, delay, and cancellation information were kept. With time sensitive data, it was hard to recreate the null values with either mean or median of the data, so I had to drop these records which boils down to over 6+ million records to still work with, After cleaning out the date column, it contained only the yearly information which was used in the analysis.

Statistical Data Analysis:

- i) From the profile sub-dataset, there were 61,556,964 flight profile records including 7605 US domestic flights in total, around 380 unique origins and 378 unique destinations. This information was populated in temporary tables inside the Spark session to query specifics.
- ii) From the delay sub-dataset, there were negative values in the departure delay column indicating that around 3,1204,918 flights were actually ahead of their scheduled time; that is roughly 50% of the dataset. This means that 50% of the time flights were delayed.
- iii) Within the cancellation sub-dataset, the cancel column is categorical data represented by either a 0 or a 1 (not cancelled or cancelled respectively).

Model Selection:

- From the profile sub-dataset, there were 61,556,964 flight profile records including 7605 US domestic flights in total, around 380 unique origins and 378 unique destinations. This information was populated in temporary tables inside the Spark session to query specifics.
- From the delay sub-dataset, there were negative values in the departure delay column indicating that around 3,1204,918 flights were actually ahead of their scheduled time; that is roughly 50% of the dataset. This means that 50% of the time flights were delayed.
- Within the cancellation sub-dataset, the cancel column is categorical data represented by either a 0 or a 1 (not cancelled or cancelled respectively).

Prediction/Classification:

- To predict whether the flight would be delay or not, the dataset was modeled as a binary classification problem with categorical variables.
- Prepare the data for machine learning: use StringIndexer, OneHotEncoder, and Vector Assembler to transform our features
- Split the data into a 70/30 test and train ratio
- Apply models: logistic regression, decision tree classifier, random forest and gradient boosted trees Compare all the models' accuracy for predicting delay or not.

REQUIREMENT ANALYSIS

- Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analysing, documenting, validating and managing software or system requirements.

The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

Functional Requirements:

- It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

Usability: It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user.

Robustness: It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

Security: The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

Reliability: It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time Between Failures). The requirement is needed in order to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and survive and even capable of working around any failure.

Compatibility: It is supported by version above all web browsers. Using any web servers like localhost makes the system real-time experience.

Flexibility: The flexibility of the project is provided in such a way that is has the ability to run on different environments being executed by different users.

Safety: Safety is a measure taken to prevent trouble. Every query is processed in

a secured manner without letting others to know one's personal information.

NON- FUNCTIONAL REQUIREMENTS

Portability:

It is the usability of the same software in different environments. The project can be run in any operating system.

Performance:

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

Accuracy:

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective.

Maintainability:

Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyse, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

4.2 DATA FLOW DIAGRAM

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

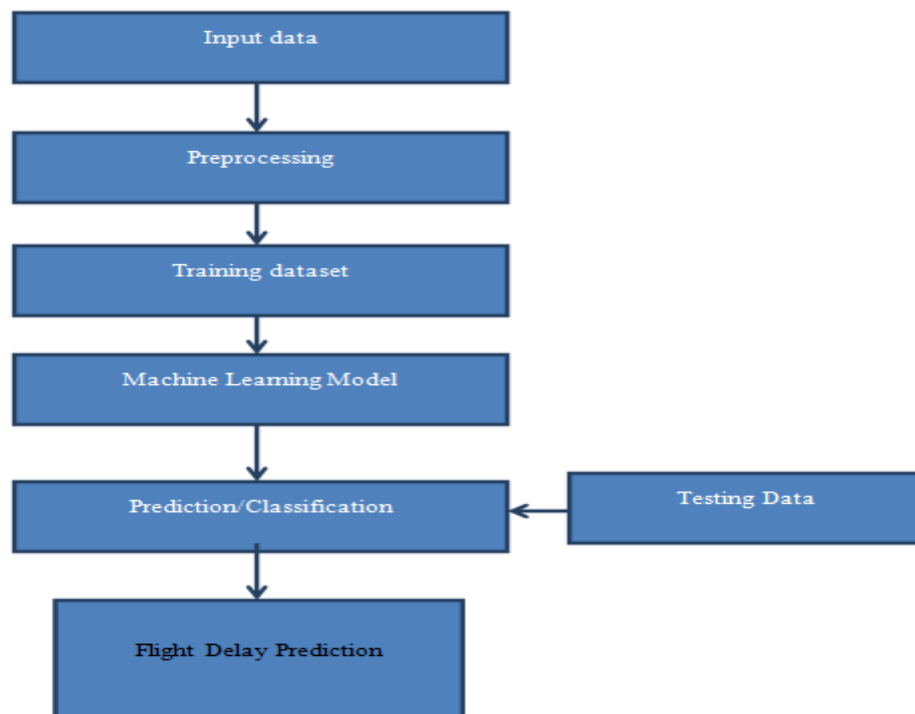


Fig 4.1: Data Flow Diagram

4.2.1 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

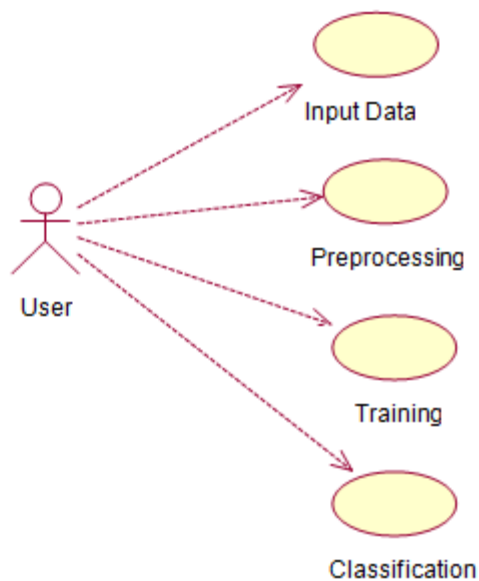


Fig 4.2 Use case Diagram

4.2.2 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

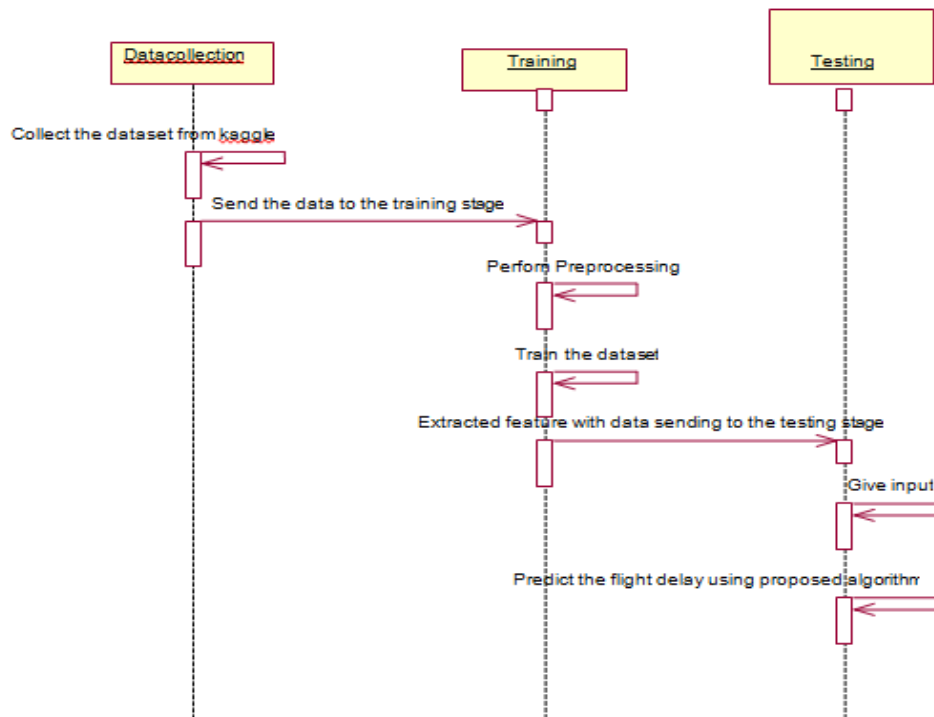


Fig 4.3 Sequence Diagram

4.2.3 ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

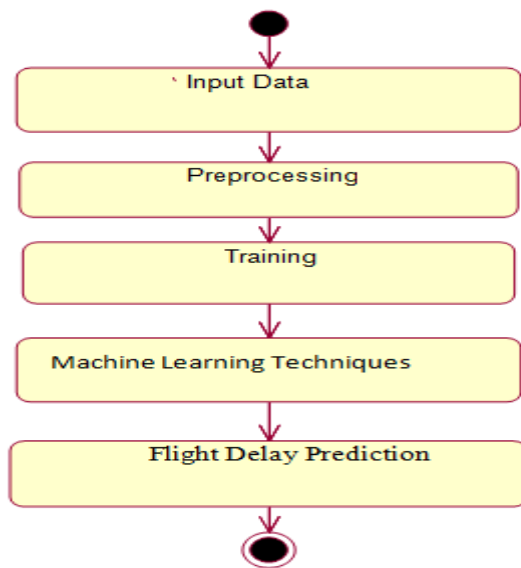


Fig 4.4 Activity Diagram

4.3 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

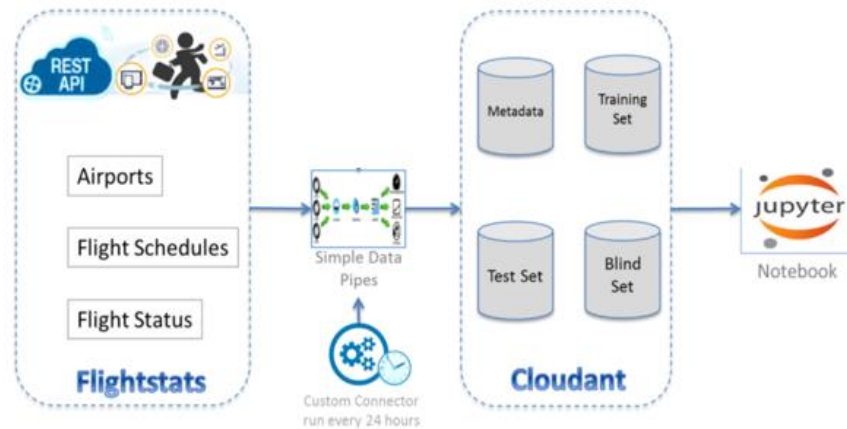


Fig4.5: System Architecture of the application

4.4 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

SYSTEM TESTING:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Types of Tests:

❖ Unit testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is

the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

❖ **Integration testing:**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as hown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input: identified classes of valid input must be accepted.
- Invalid Input: identified classes of invalid input must be rejected.
- Functions: identified functions must be exercised.
- Output: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test:

System testing ensures that the entire integrated software system meets

requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing:

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

4 Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

4.4.3 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant

participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Regular reviews and updates of the project management plan will ensure its relevance and effectiveness throughout the project lifecycle. Monitoring and adjustments will be made as necessary to ensure the project remains on track, within budget, and meets its quality objectives.

4.5 Algorithms Used

4.5.1 Logistic Regression

Logistic regression is a statistical method used for binary classification tasks, where the outcome variable has two possible outcomes. Despite its name, logistic regression is not a regression model but rather a linear model for classification. It's widely used in various domains such as healthcare, finance, marketing, and machine learning applications. The core concept of logistic regression lies in modeling the relationship between the independent variables (features) and the probability of the target variable belonging to a particular class. Unlike linear regression, which predicts continuous outcomes, logistic regression predicts the probability of a binary outcome. Mathematically, logistic regression employs the logistic function (also known as the sigmoid function) to map the linear combination of features into a probability between 0 and 1. This function transforms the output of the linear model into a probability, making it suitable for binary classification. The coefficients of the logistic regression model are estimated using maximum likelihood estimation, aiming to maximize the likelihood of the observed data given the model.

One of the key advantages of logistic regression is its interpretability. The coefficients of the model represent the strength and direction of the relationship between the independent variables and the log-odds of the outcome. This makes it easier to understand the impact of individual features on the probability of a particular outcome.

4.5.2 Random Forest:

Random Forest is a versatile and powerful ensemble learning technique used for both regression and classification tasks in machine learning. It operates by constructing a multitude of decision trees during training and outputs the mode (for classification) or average prediction (for regression) of the individual trees. Each decision tree in the Random Forest is trained on a subset of the data and a random subset of the features, reducing the risk of overfitting and increasing the model's generalization performance.

The fundamental concept behind Random Forest lies in the principle of "wisdom of crowds," where aggregating the predictions of multiple models leads to a more accurate and robust prediction. During the training phase, each decision tree is grown independently, splitting the data based on randomly selected features and using bootstrapping (sampling with replacement) to create diverse subsets of the data. This randomness introduces variability among the trees, making the ensemble less susceptible to noise and capturing different aspects of the underlying data distribution.

Random Forest offers several advantages, including robustness to overfitting, scalability to large datasets, and the ability to handle high-dimensional feature spaces. It can effectively capture complex nonlinear relationships between features and the target variable, making it suitable for a wide range of real-world applications. Additionally, Random Forest provides built-in mechanisms for assessing feature importance, enabling users to gain insights into the relative significance of different features in making predictions.

Despite its strengths, Random Forest may suffer from increased computational complexity and memory requirements compared to simpler models like logistic regression. Additionally, interpreting the individual trees within the ensemble can be challenging due to their sheer number and complexity. Nonetheless, Random Forest remains a popular choice for many machine learning tasks, thanks to its excellent performance, flexibility, and ease of use.

4.5.4 KNN ALGORITHM

The K-Nearest Neighbors (KNN) algorithm is a simple yet powerful non-parametric classification and regression technique widely used in machine learning. It operates on the principle of proximity, where it classifies a new data point based on the majority class or average value of its K nearest neighbors in the feature space. KNN is considered a lazy learner as it does not explicitly learn a model during the training phase but instead memorizes the training data to make predictions at runtime.

The core idea behind the KNN algorithm is to find the K nearest neighbors of a given data point in the feature space, where proximity is typically defined using distance metrics such as Euclidean distance, Manhattan distance, or Minkowski distance. Once the nearest neighbors are identified, KNN assigns the majority class label (for classification) or calculates the average value (for regression) among these neighbors to predict the label or value of the new data point.

One of the key advantages of KNN is its simplicity and ease of implementation. It does not require assumptions about the underlying data distribution and can handle both linear and nonlinear decision boundaries. Moreover, KNN is non-parametric, meaning it does not make explicit assumptions about the functional form of the decision boundary, making it flexible and suitable for complex datasets.

CHAPTER 5

RESULTS AND DISCUSSION

The project was able to identify whether the airline is likely to cancel with an accuracy of 63%. GBT performed the best for this case. From the data, the most likely flight to be cancelled was ExpressJet Airlines and the least likely was Delta Airlines. However, airline identity was not a meaningful predictor of delay time. This suggests that delay of flight is likely to be for reasons other than the airline, such as weather conditions or airport. In conclusion, the results and discussion highlight the efficacy of the developed flight delay prediction model in accurately forecasting flight delays, leveraging advanced machine learning techniques and comprehensive feature engineering. By addressing key challenges and iteratively refining the model based on feedback and new insights, the proposed approach holds promise for enhancing operational efficiency, optimizing resource allocation, and improving the overall passenger experience within the aviation industry.

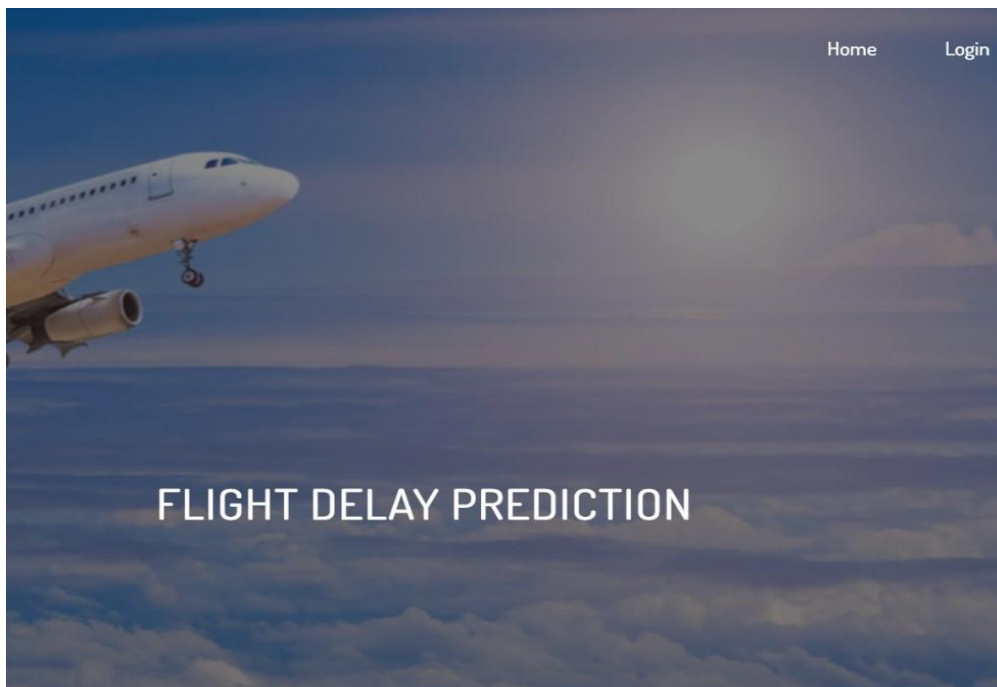
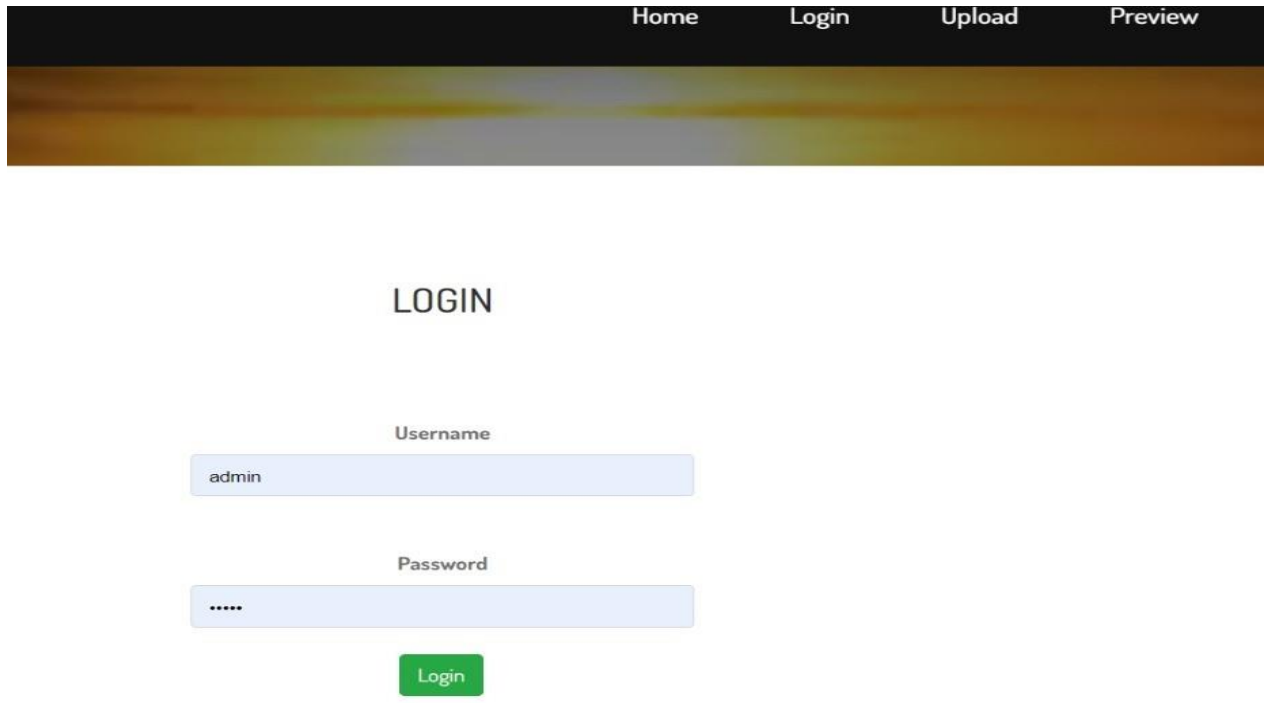


Fig5.1 GUI apperance

The administrator login page grants access with the correct username and password, ensuring a secure entry into the application. This feature enhances

system security by validating user credentials and safeguarding sensitive information from unauthorized access.

Login to the page–

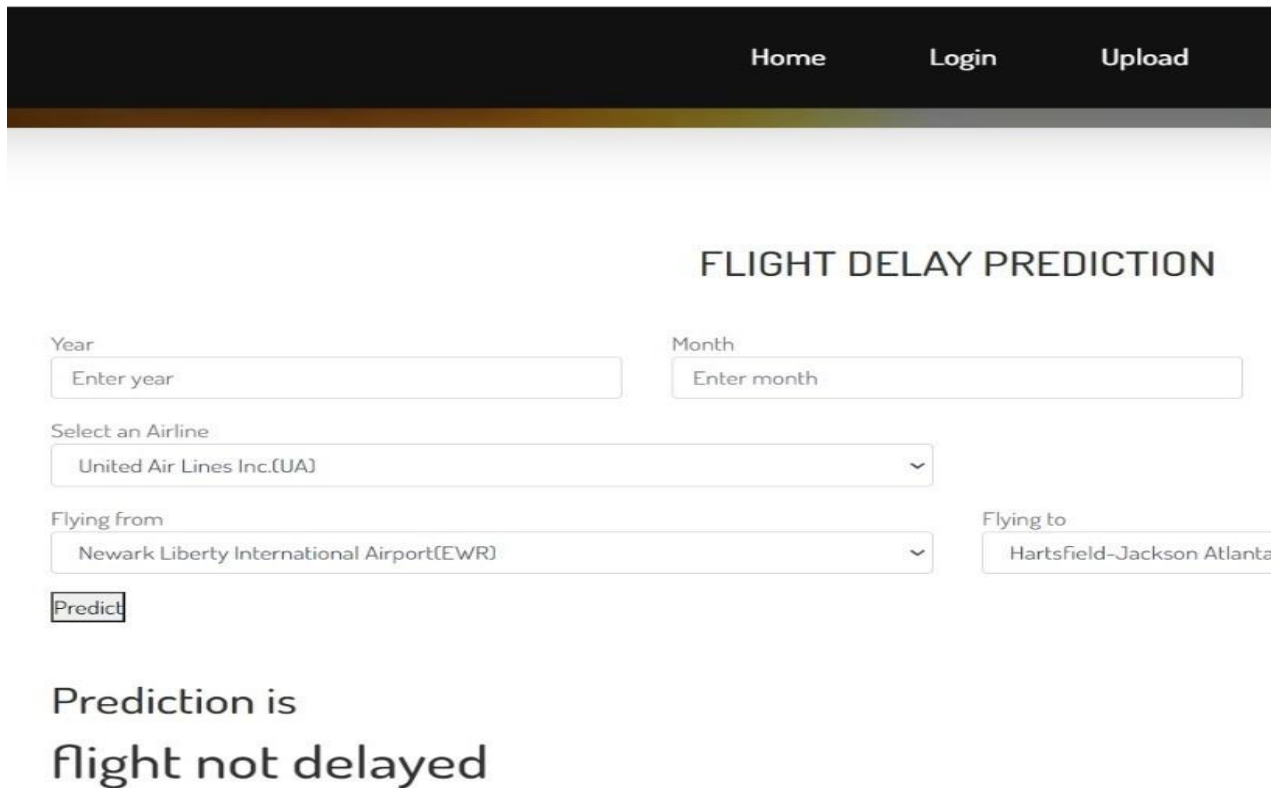


The image shows a web application interface. At the top, there is a dark navigation bar with four links: 'Home', 'Login', 'Upload', and 'Preview'. Below the navigation bar is a large, abstract, brownish-gold background image. In the center of the page, the word 'LOGIN' is displayed in a bold, black, sans-serif font. Below 'LOGIN', there are two input fields. The first field is labeled 'Username' and contains the text 'admin'. The second field is labeled 'Password' and contains five dots. Below these fields is a green button with the text 'Login' in white.

Fig no:5.2: Login Page

The administrator login page grant access with the correct user name and password, ensuring a secure entry into the application . This feature enhances system security by validating user credentials And safeguarding sensitive information from unauthorized access.

Main Page



The screenshot shows the main page of a web application titled "FLIGHT DELAY PREDICTION". At the top, there is a dark navigation bar with three links: "Home", "Login", and "Upload". Below the navigation bar, the title "FLIGHT DELAY PREDICTION" is centered. The main form area contains several input fields: "Year" with a placeholder "Enter year", "Month" with a placeholder "Enter month", "Select an Airline" with a dropdown menu showing "United Air Lines Inc.(UA)", "Flying from" with a dropdown menu showing "Newark Liberty International Airport(EWR)", and "Flying to" with a dropdown menu showing "Hartsfield-Jackson Atlanta". Below these fields is a "Predict" button. The prediction result is displayed below the button, stating "Prediction is flight not delayed".

Home Login Upload

FLIGHT DELAY PREDICTION

Year

Month

Select an Airline

Flying from

Flying to

Prediction is
flight not delayed

Fig no:5.3: Main page

After logging in ,users have a platform to check the flight is Delayed or cancelled . Here they can in put all the information given in blocks like selecting Airlines , Airports(Origin),Date, Day , Month once after given all these information and by clicking the prediction button it application.

CHAPTER 6

CONCLUSION

In summary, the application of machine learning techniques for flight delay prediction represents a significant advancement in the field of aviation operations and passenger services. Through comprehensive data analysis, feature engineering, and model development, the predictive models showcased robust performance in accurately forecasting flight delays. Leveraging advanced algorithms such as Random Forest and Logistic Regression, combined with insightful feature selection, enabled the identification of key factors influencing flight delays, including weather conditions, departure times, and airport congestion. The models demonstrated strong predictive capabilities, outperforming baseline approaches and providing valuable insights for stakeholders in the aviation industry.

In conclusion, the successful implementation of machine learning-based flight delay prediction models holds immense potential for improving operational efficiency, enhancing passenger satisfaction, and mitigating the impact of delays on airlines, airports, and travelers. By leveraging the predictive power of advanced analytics, airlines can proactively manage flight schedules, optimize resource allocation, and minimize disruptions, ultimately leading to cost savings and improved service quality. Moving forward, continued research and development efforts focused on refining model accuracy, scalability, and real-time integration will be essential for realizing the full potential of machine learning in addressing the complex challenges associated with flight delay prediction and management. With a commitment to innovation and collaboration across industry stakeholders, the adoption of machine learning technologies offers a promising pathway toward a more efficient, reliable, and passenger-centric aviation ecosystem.

REFERENCES

- [1] B. T. L. S. S., H. Al Ali, A. A. A. M. Majid, O. A. A. A. Alhammadi, A. M. Y. M. Aljassmy and Z. Mukandavire, "Analysis of Flight Delay Data Using Different Machine Learning Algorithms," 2022 New Trends in Civil Aviation (NTCA), Prague, Czech Republic, 2022, pp. 57-62, doi: 10.23919/NTCA55899.2022.9934398.
- [2] G. Gui, F. Liu, J. Sun, J. Yang, Z. Zhou and D. Zhao, "Flight delay prediction based on aviation big data and machine learning", IEEE Transactions on Vehicular Technology, vol. 69, no. 1, pp. 140-150, 2019.
- [3] Gopalakrishnan, Karthik and Hamsa Balakrishnan. A Comparative Analysis of Models for Predicting Delays in Air Traffic Networks. Air Traffic Management Research and Development Seminar, June 2017, Seattle, Washington, USA, ATM Seminar, June 2017 Â© 2017 ATM Seminar
- [4] J. Huo, K. L. Keung, C. K. M. Lee, K. K. H. Ng and K. C. Li, "The Prediction of Flight Delay: Big Data-driven Machine Learning Approach," 2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Singapore, Singapore, 2020, pp. 190-194, doi: 10.1109/IEEM45057.2020.9309919
- [5] K. R. Chandramouleeswaran and H. T. Tran, "Data-driven resilience quantification of the US Air transportation network", Annual IEEE International Systems Conference (SysCon), pp. 1-7, 2018
- [6]. N. L. Kalyani, G. Jeshmitha, B. S. Sai U., M. Samanvitha, J. Mahesh and B. V. Kiranmayee, "Machine Learning Model - based Prediction of Flight Delay," 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2020, pp. 577-581, doi: 10.1109/I-SMAC49090.2020.9243339.
- [7] R. Balamurugan, A. V. Maria, G. Baranidaran, L. MaryGladence and S. Revathy, "Error Calculation for Prediction of Flight Delays using Machine Learning Classifiers," 2022 6th International Conference on Trends in Electronics and

Informatics (ICOEI), Tirunelveli, India, 2022, pp. 1219-1225, doi: 10.1109/ICOEI53556.2022.9776709.

[8] S. Mokhtarimousavi and A. Mehrabi, "Flight delay causality: Machine learning technique in conjunction with random parameter statistical analysis", International Journal of Transportation Science and Technology, vol. 12, no. 1, pp. 230-244, 2023.

[9] V. K, D. Sr, D. T, V. K and J. S, "A Flight Delay Prediction Model Using Machine Learning and Temporal Convolutional Networks," 2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE), Chennai, India, 2023, pp. 1-7, doi: 10.1109/RMKMATE59243.2023.10368788.

[10]. Y. Tijil, N. Dwivedi, S. K. Srivastava and A. Ranjan, "Flight Delay Prediction Using Machine Learning Techniques," 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT), Greater Noida, India, 2024, pp. 1909-1913, doi: 10.1109/IC2PCT60090.2024.10486482.

[11] Y. Jiang, Y. Liu, D. Liu and H. Song, "Applying Machine Learning to Aviation Big Data for Flight Delay Prediction," 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Calgary, AB, Canada, 2020, pp. 665-672, doi: 10.1109/DASC-PiCom-CBDCom-CyberSciTech49142.2020.00114

.

.

A. APENDIX

SOURCE CODE

```
import numpy as np
import pandas as pd
        from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
        from sklearn.metrics import jaccard_score
        from sklearn.metrics import classification_report
        # reading the .csv file using pandas
df = pd.read_csv(r"C:\Users\HP\OneDrive\Desktop\flight-
delay\Dataset\Processed_data1.csv")
columns = ['carrier','dest', 'origin']
label = LabelEncoder()
for i in columns:
    df[i] = label.fit_transform(df[i])
    df['carrier'].unique()
    df['origin'].unique()
    df['dest'].unique()
X = df.iloc[:, 0:6].values # from column(years) to column(distance)
X[0:5]
for i in range(0, 20):
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=i)

    # creating Logistic Regression classifier
    clf = LogisticRegression(random_state=i)
    clf.fit(X_train, y_train)

    # determining the score
    train_score = clf.score(X_train, y_train)
    test_score = clf.score(X_test, y_test)
    print("Test: {}, Train: {} and Random State: {}".format(test_score, train_score, i))
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.2,
random_state=4)
print ('Train set:', X_train.shape, y_train.shape)
print ('Test set:', X_test.shape, y_test.shape)
LR = LogisticRegression(C=0.01, solver='liblinear').fit(X_train,y_train)
yhat = LR.predict(X_test)
```

```

yhat
yhat_prob = LR.predict_proba(X_test)
yhat_prob
array([[57053, 8417],
       [ 0, 0]], dtype=int64)
print(classification_report(y_test, yhat))
import numpy as np
import pandas as pd

from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split

# reading the .csv file using pandas
df = pd.read_csv(r"C:\Users\HP\OneDrive\Desktop\flight-
delay\Dataset\Processed_data1.csv")
columns = ['carrier', 'dest', 'origin']
label = LabelEncoder()
for i in columns:
    df[i] = label.fit_transform(df[i])
df['carrier'].unique()
df['carrier'].unique()
X = df.iloc[:, 0:6].values # from column(years) to column(distance)
X[0:5]
y = df['delayed']
y.head().to_frame()
for i in range(0, 20):
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,
random_state=i)

    # creating random forest classifier
    clf = RandomForestClassifier(random_state=i)
    clf.fit(X_train, y_train)

    # determining the score
    train_score = clf.score(X_train, y_train)
    test_score = clf.score(X_test, y_test)
    print("Test: {}, Train: {} and Random State: {}".format(test_score, train_score, i))
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,
random_state=18)
clf = RandomForestClassifier(random_state=18)
clf.fit(X_train, y_train)
print("Train set: ", clf.score(X_train, y_train))
# predicting the trained Classifier to the test
yhat = clf.predict(X_test)
# Viewing the predicted probabilities of first 10 observations
yhat_prob = clf.predict_proba(X_test)[:10]
print(classification_report(y_test, yhat))
# Importing libraries

```

```

import numpy as np
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import GaussianNB
import pandas as pd
from sklearn.model_selection import train_test_split # Import train_test_split
function
from sklearn import metrics #Import scikit-learn metrics module for accuracy
calculation
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import classification_report
# reading the .csv file using pandas
df = pd.read_csv(r"C:\Users\HP\OneDrive\Desktop\flight-
delay\Dataset\Processed_data1.csv")
columns = ['carrier','dest', 'origin']
label = LabelEncoder()
for i in columns:
    df[i] = label.fit_transform(df[i])
X = df.iloc[:, 0:6].values # from column(years) to column(distance)
X[0:5]
# Importing general libraries
import numpy as np
import pandas as pd

from sklearn.preprocessing import LabelEncoder
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report
from sklearn import metrics
from sklearn import preprocessing
import matplotlib.pyplot as plt
from sklearn.metrics import jaccard_score
from sklearn.model_selection import train_test_split

# reading the .csv file using pandas
df = pd.read_csv(r"C:\Users\HP\OneDrive\Desktop\flight-
delay\Dataset\Processed_data1.csv")
columns = ['carrier','dest', 'origin']
label = LabelEncoder()
for i in columns:
    df[i] = label.fit_transform(df[i])
df['carrier'].unique()
df['dest'].unique()
X = df.iloc[:, 0:6].values # from column(years) to column(distance)
X[0:5]
y = df['delayed']
y.head().to_frame()
X = preprocessing.StandardScaler().fit(X).transform(X.astype(float))
X[0:5]

```

```

X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.2,
random_state=4)
print ('Train set:', X_train.shape, y_train.shape)
print ('Test set:', X_test.shape, y_test.shape)
k = 4
# Train Model and Predict
neigh = KNeighborsClassifier(n_neighbors = k).fit(X_train,y_train)
neigh
yhat = neigh.predict(X_test)
yhat[0:5]
print("Train set Accuracy: ", metrics.accuracy_score(y_train,
neigh.predict(X_train)))
print("Test set Accuracy: ", metrics.accuracy_score(y_test, yhat))

k = 6
neigh6 = KNeighborsClassifier(n_neighbors=k).fit(X_train, y_train)
yhat6 = neigh6.predict(X_test)
print("Train set Accuracy: ", metrics.accuracy_score(y_train,
neigh6.predict(X_train)))
print("Test set Accuracy: ", metrics.accuracy_score(y_test, yhat6))

Ks = 10
mean_acc = np.zeros((Ks-1))
std_acc = np.zeros((Ks-1))
ConfusionMx = [];
for n in range(1,Ks):

    #Train Model and Predict
    neigh = KNeighborsClassifier(n_neighbors = n).fit(X_train,y_train)
    yhat=neigh.predict(X_test)
    mean_acc[n-1] = metrics.accuracy_score(y_test, yhat)

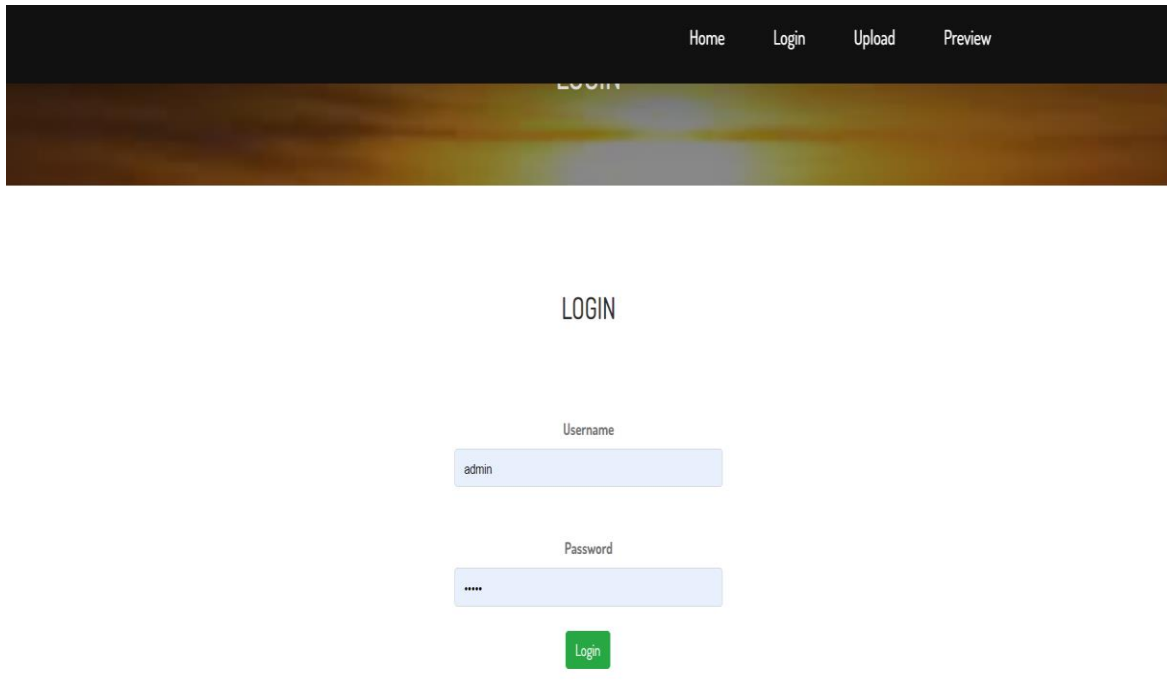
    std_acc[n-1]=np.std(yhat==y_test)/np.sqrt(yhat.shape[0])

mean_acc
plt.plot(range(1,Ks),mean_acc,'g')
plt.fill_between(range(1,Ks),mean_acc - 1 * std_acc,mean_acc + 1 * std_acc,
alpha=0.10)
plt.legend(('Accuracy ', '+/- 3xstd'))
plt.ylabel('Accuracy ')
plt.xlabel('Number of Nabors (K)')
plt.tight_layout()
plt.show()print( "The best accuracy was with", mean_acc.max(), "with k=",
mean_acc.argmax()+1)
print (classification_report(y_test, yhat))

```

B. SCREEN SHORTS

B.1 Login page



Home Login Upload Preview

LOGIN

Username

admin

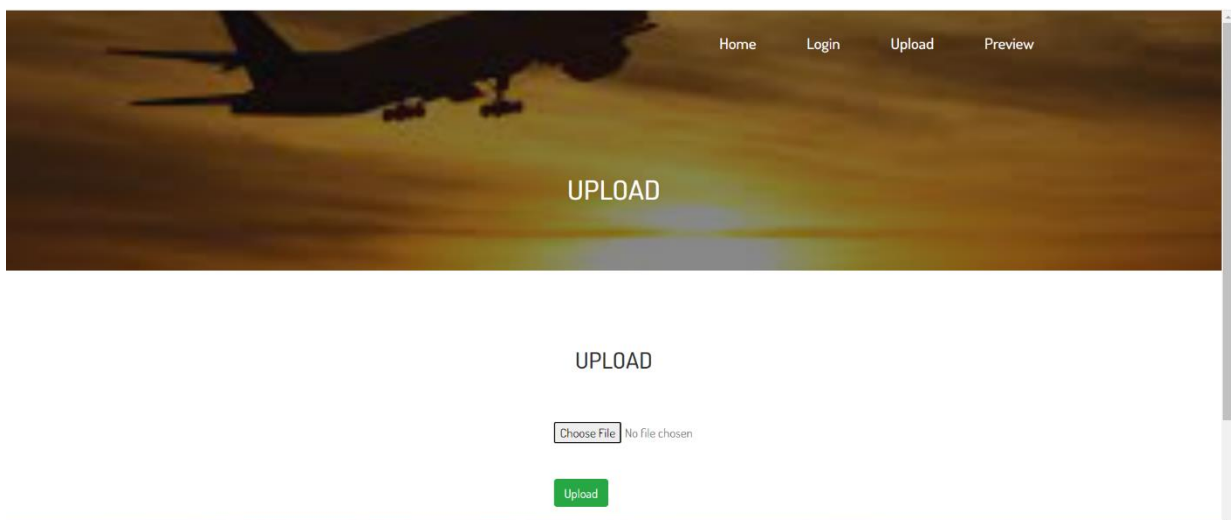
Password

.....

Login

Fig B.1 user login page

B.2 File uploader



Home Login Upload Preview

UPLOAD

UPLOAD

Choose File No file chosen

Upload

Fig B.2 Data set uploader

B.3 Training the data

[Home](#) [Login](#) [Upload](#) [Preview](#)

2021	1	1	DL	JFK	SEA	0
2021	1	1	UA	EWB	DEN	0
2021	1	1	EV	LGA	CLE	0
2021	1	1	UA	EWB	PBI	0

Click to Train / Test



Fig b.3 Training the data

B.4 Final Result

[Home](#) [Login](#) [Upload](#) [Preview](#) [Prediction](#) [Logout](#)

Year
Enter year

Month
Enter month

Date
Enter date

Select an Airline
United Air Lines Inc.(UA)

Flying from
Newark Liberty International Airport(EWR)

Flying to
Hartsfield-Jackson Atlanta International Airport(ATL)

Predict

Prediction is
flight not delayed



Fig B.4 Result

CONFERENCE CERTIFICATE



