A

Major Project

On

FLIGHT DELAY PREDICTION USING MACHINE LEARNING ALGORITHMS

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

BY

P. Rajeshwar Reddy (187R1A05N0)

R. Shivani (187R1A05N3)

G. Suryateja (187R1A05J9)

Under the Guidance of

Dr. K. Maheswari

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)
Recognized Under Section 2(f) & 12(B) of the UGC Act.1956,
Kandlakoya (V), Medchal Road, Hyderabad-501401.

2018-2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**FLIGHT DELAY PREDICTION USING MACHINE LEARNING ALGORITHMS**" being submitted by **P.RAJESHWAR REDDY (187R1A05N0), R.SHIVANI (187R1A05N3), G.SURYATEJA (187R1A05J9)** in partial fulfillment of the requirements for the award of the degree of B. Tech in Computer Science and Engineering of the Jawaharlal Nehru Technological University Hyderabad, during the year 2021-2022. It is certified that they have completed the project satisfactorily. The results embodied in this thesis have not been submitted to any other University or institute for the award of any degree or diploma.

INTERNAL GUIDE
Dr. K. Maheswari
Associate Professor

DIRECTOR
Dr. A. Raji Reddy

HOD
Dr. K. Srujan Raju

EXTERNAL EXAMINER

Submitted on viva voce Examination held on _________________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide, **Dr. K. Maheswari** for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by her shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) Coordinators: **Dr. Mr. J. Narasimha Rao, Dr. T.S. Mastan Rao, Mr. A. Uday Kiran, Mr. A. Kiran Kumar, Mrs. G. Latha** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to the Head of the Department **Dr. K. Srujan Raju** for providing excellent infrastructure and a nice atmosphere for completing this project successfully.

We are obliged to our Director **Dr. A. Raji Reddy** for being cooperative throughout the course of this project. We would like to express our sincere gratitude to our Chairman Sri. **Ch. Gopal Reddy** for his encouragement throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus, HYDERABAD** who contributed and who are contributing to this project, was vital for the success of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement without which this assignment would not be possible. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project.

P. RAJESHWAR REDDY (187R1A05N0)
R. SHIVANI (187R1A05N3)
G. SURYATEJA (187R1A05J9)

ABSTRACT

Flight Planning is one of the challenges in industrial world which faces many uncertain conditions. One such condition is delay occurrence, which stems from various factors and imposes considerable costs on airlines, operators, and travelers. Delays in departure can occur due to bad weather conditions, seasonal and holiday demands, airline policies, technical issue such as problems in airport facilities, luggage handling and mechanical apparatus, and accumulation of delays from preceding flights. In flight delay prediction system based on the weather parameters which can result in delays. The system considers the temperature, humidity, rain in mm, visibility and month number as important parameters for prediction of 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. Collection of historical weather data and operational data during departure and arrival at airports serves as the source for building prediction models.

LIST OF FIGURES

LIST OF SCREENSHOTS

TABLE OF CONTENTS — Page No's

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

The project titled ”Flight delay prediction using machine learning algorithms” is an flight delay prediction system which is used to predict flight delays and provides an easier way to know and save the time of the passengers in predicting the flight delay. Here the system is given the input dataset and then data is pre processed then by using algorithms the system will able to predict the flight delay.

1.2 PROJECT PURPOSE

The purpose is to design a system which overcomes the struggles faced by the customers in flight journey. Using this system the customers can directly know the delay of flights and can able to manage their time. Here the system predicts the delay of flights based on different parameters taking into consideration such as weather conditions etc.,.

1.3 PROJECT FEATURES

The core features of this project are it provides the output of the predicted flight delay by using the dataset given to the system. It also enables the user customer to get an idea of the particular flight is available at particular time or not. Here the project overcomes the disadvantages of existing system and provides better flight delay predictions. In this system the prediction is very accurate in delay of flights.

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive at a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the user faces. The model is built in such a way that it addresses the difficulties faced by the user. Consider the example, there are two users the first one who needs a model which also allows him to move around in public and the second one who needs a model to be trained only to function inside his house, based on these requirements a model is generated and tested to see if the user is satisfied with application.

2.2 EXISTING SYSTEM

In Existing System, It is not possible to predict the flight delays and it was very difficult to know the that there was a delay of flight or not.Hence it also leads to financial loss to the airlines and the customers. Here the customers also loss their time and it may lead to problems.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

Following are the limitations of the existing system.

- This cause inconvenience to the airlines and to the passengers.
- It causes heavy expenses to the airlines.
- It also wastes the quality time of the passengers.

2.3 PROPOSED SYSTEM

In the proposed system, To make the system more scalable it is necessary to choose an algorithm which considers all the parameters to be independent. Supervised learning as the name indicates a presence of supervisor as teacher. After that, machine is given new set of data so algorithmic rule analyses and produces an correct outcome from tagged data Using supervised machine learning approach. Naïve bayed model is one of the algorithm which is proven to be efficient for real time prediction. It considers every attribute to be independent from each other makes it an apt algorithm for the concerned project.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

The following are the advantages of the proposed system:

- User friendly interface.
- It is able to predict the delay of the flights with the reason.
- It becomes very easy for passengers to know whether there is delay or not.
- It also saves lot of money for the airlines and quality time of passengers

2.4 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 user.

Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on a project, which will give the user the best quality of life possible. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication that the system is ecomically possible for development.

2.4.2 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. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- Processor : 800MHz Intel Pentium III or equivalent
- RAM : 512MB and above.
- Hard Disk : 20 GB`
- Input Devices : Mouse, Keyboard

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

• Operating Systems :Windows XP/Windows 7/10

• Programming Languages :Python 3.6

• Tools : Anaconda Navigator, Jupytor Notebook

.

3. ARCHITECTURE

3.ARCHITECTURE

3.1 PROJECT ARCHITECTURE

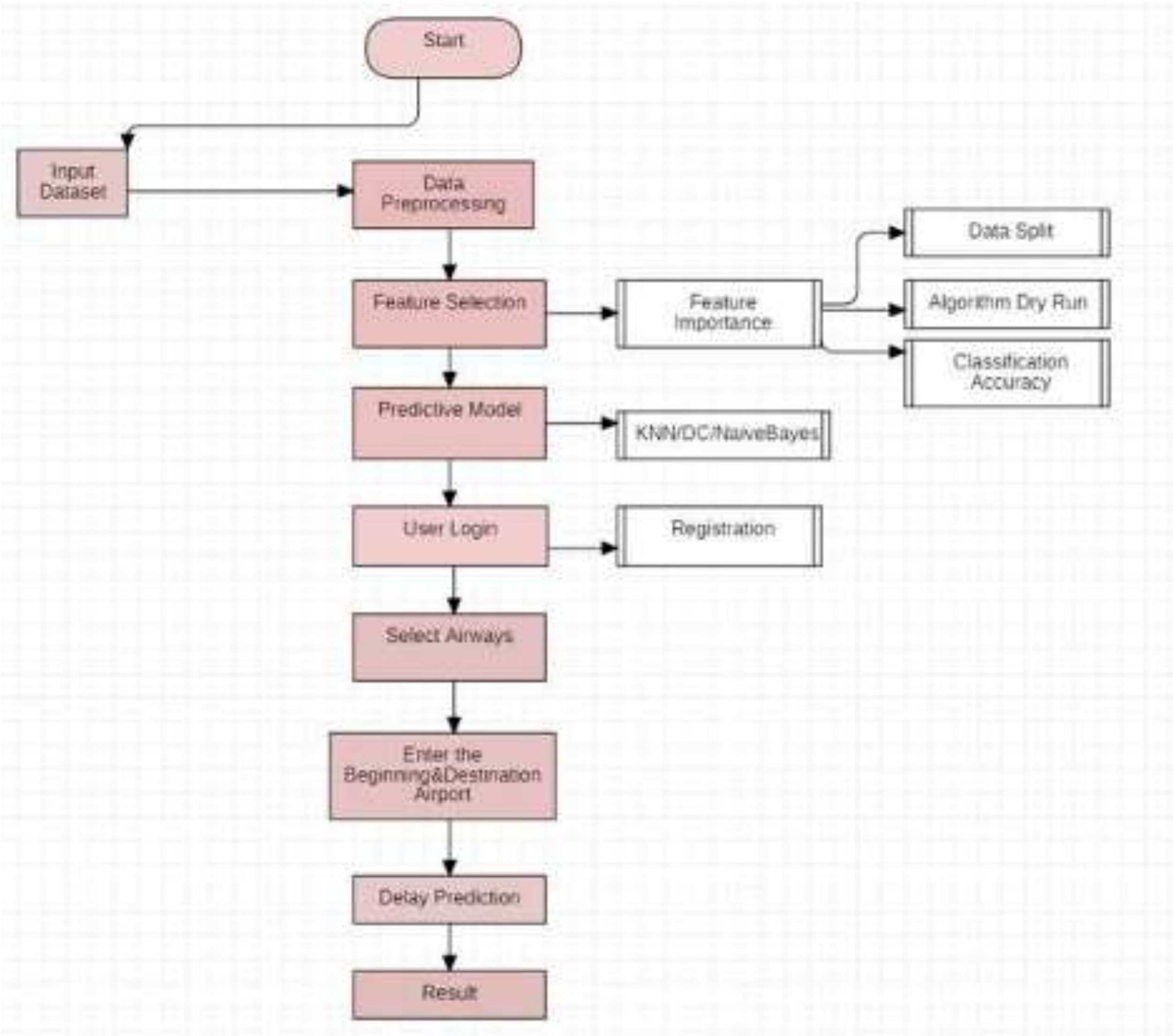


Fig. 3.1 Project Architecture of Flight Delay prediction using Machine Learning

3.2 DESCRIPTION

Input Data: Input data is given to the system to find the desired predicted output.

Data preprocessing: It is the process in which the collected data is processed for the execution of the data.

Feature Selection: The features required for the prediction of flight delay are considered

3.3 USE CASE DIAGRAM

In the use case diagram,we have two actors who are the user,and the application

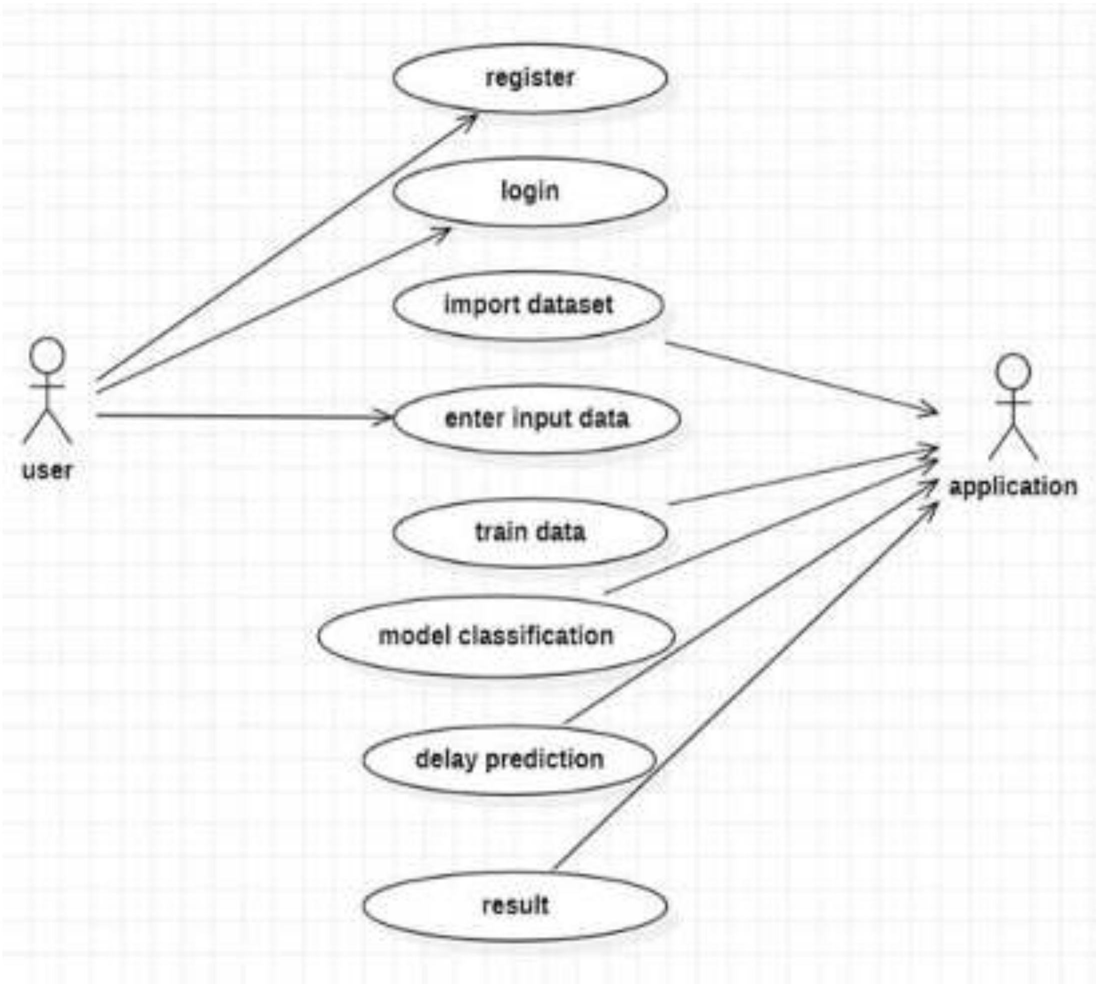


Fig 3.3 Usecase Diagram for Flight Delay prediction using Machine Learning Algorithms

3.4 CLASS DIAGRAM

Class Diagram is a collection of classes and objects.

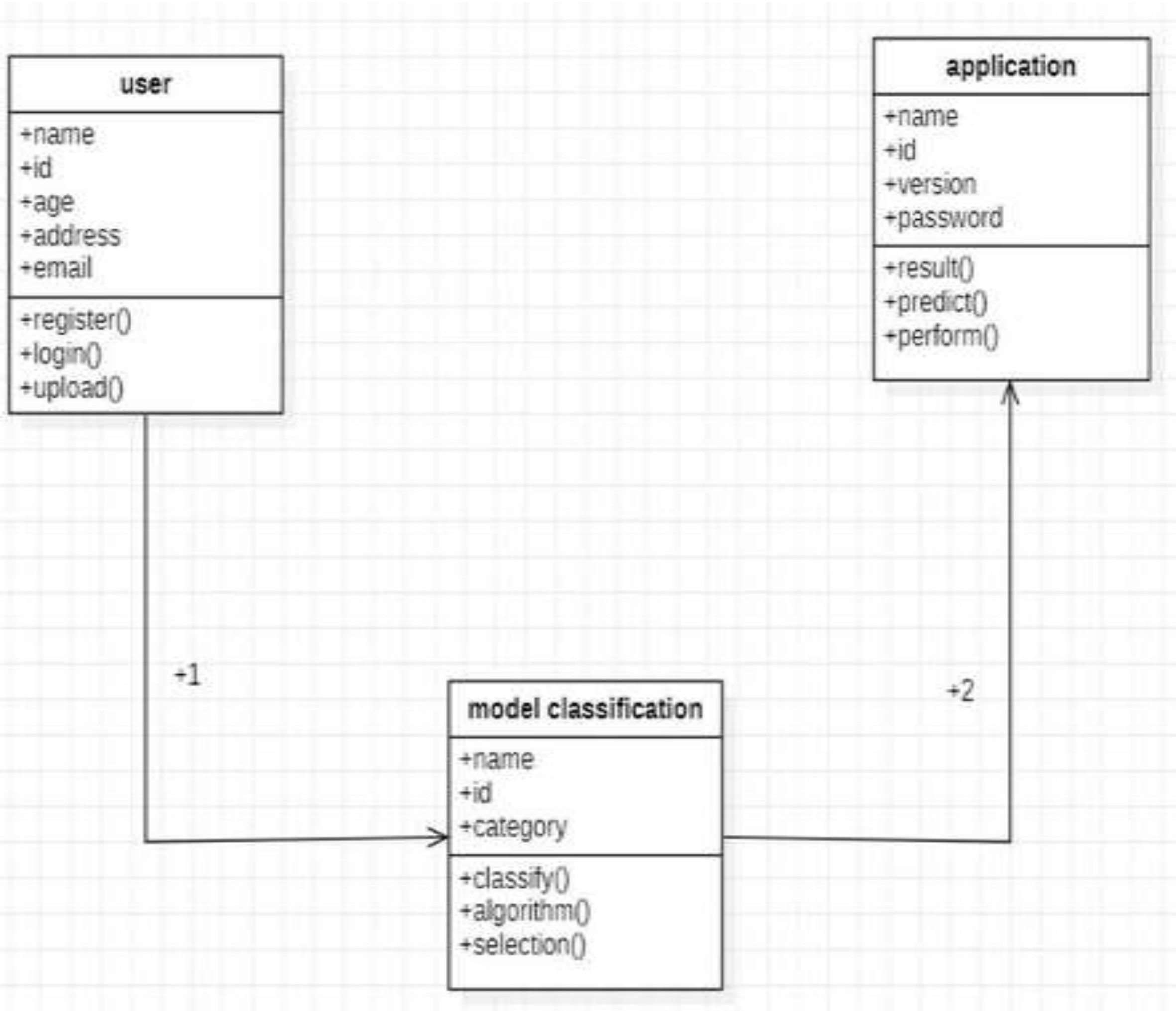


Fig. 3.4 Class Diagram for Flight Delay prediction using Machine Learning Algorithms

3.5 SEQUENCE DIAGRAM

The sequence diagram shows the sequence in which different tasks are being carried out by the actors.

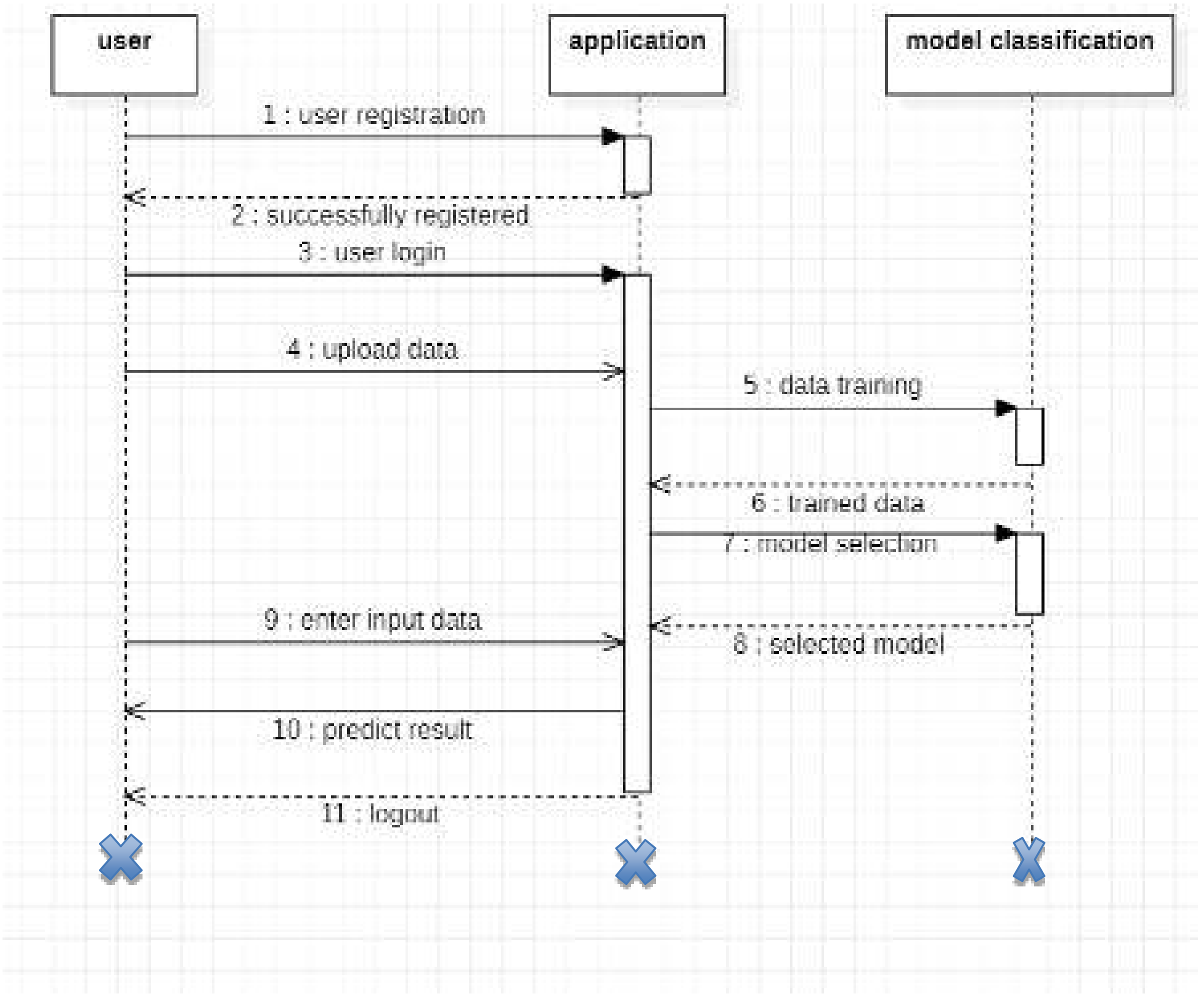


Fig.3.5 Sequence Diagram for Flight Delay prediction using Machine Learning Algorithms

3.6 ACTIVITY DIAGRAM

It describes the flow of activity states.

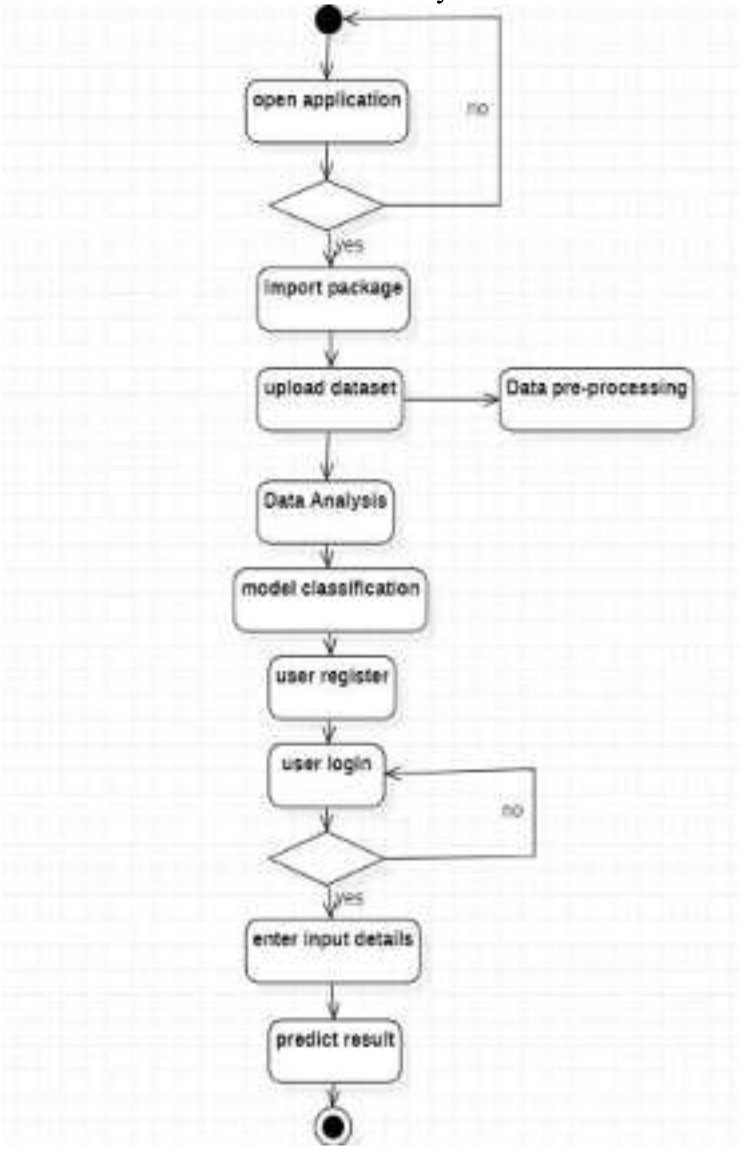


Fig. 3.6 Activity Diagram for Flight Delay prediction using Machine Learning Algorithms

3.6 DATAFLOW DIAGRAM

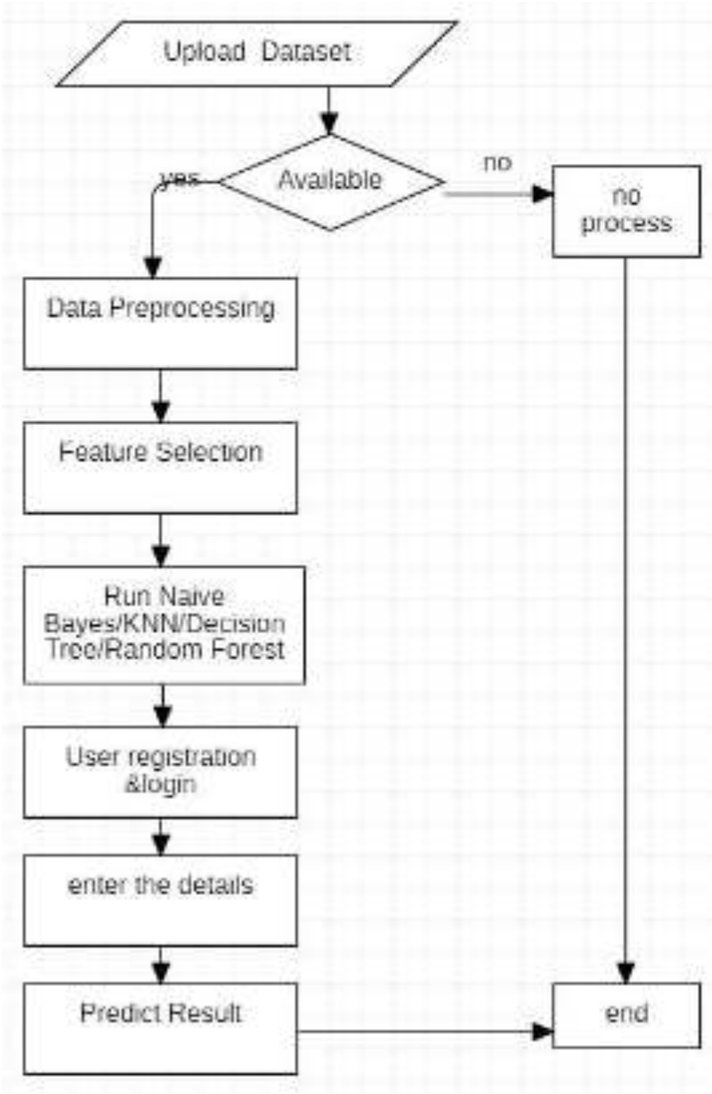


Fig. 3.6 Dataflow Diagram for Flight Delay prediction using Machine Learning Algorithms

4.IMPLEMENTATION

4.IMPLEMENTATION

4.1 SAMPLE CODE

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib as mpl
import matplotlib.patches as patches
from matplotlib.patches import ConnectionPatch
from collections import OrderedDict
from matplotlib.gridspec import GridSpec
%matplotlib inline
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
Exploring the data
df = pd.read_csv("Data/flight_data.csv")
planes = pd.read_csv("Data/planes.csv")
airports = pd.read_csv("Data/airports.csv")
carriers = pd.read_csv("Data/carriers.csv")
df.head(5)
# Checking the dimensions of the 'flight_data' dataset
df.shape
# Now checking whther the dataset contains the NULL values or not.
df.isnull().sum()
# Dropping the rows that have NaN i.e. NULL values in them
df = df.dropna()
df.head(5)
df.isnull().sum()
# Before type casting of 'dep_time', 'dep_delay', 'arr_time', 'arr_delay'
df.info()
# Type casting
df['dep_time'] = df['dep_time'].astype('int64')
df['dep_delay'] = df['dep_delay'].astype('int64')
df['arr_time'] = df['arr_time'].astype('int64')
df['arr_delay'] = df['arr_delay'].astype('int64')
# After type casting of 'dep_time', 'dep_delay', 'arr_time', 'arr_delay'
df.info()
df.head(10)
plt.figure(figsize = (18, 6))
sns.countplot(df['month'])
plt.title('Month Distribution', size = 25)
plt.xticks(size = 15)
plt.yticks(size = 15)
```

```
plt.xlabel("Months", size = 20)
plt.ylabel("Frequency", size = 20)
plt.show()
plt.figure(figsize = (20, 6))
sns.countplot(df['carrier'])
plt.title('Various Carriers in US')
plt.xticks(size = 15)
plt.yticks(size = 15)
plt.xlabel("Carriers", size = 20)
plt.ylabel("Frequency", size = 20)
plt.show()# function that extract statistical parameters from a grouby objet:
def get_stats(group):
    return {'min': group.min(), 'max': group.max(),
            'count': group.count(), 'mean': group.mean()}
# Creation of a dataframe with statitical infos on each airline:
global_stats = df['dep_delay'].groupby(df['carrier']).apply(get_stats).unstack()
global_stats = global_stats.sort_values('count')
global_stats
# graphs on flights, airports & delays
global_stats1 = global_stats
global_stats = global_stats1.head(14)
codes = global_stats.index.tolist()
carriers1 = carriers[carriers['IATA_CODE'].isin(codes)]
abbr_companies = carriers1.set_index('IATA_CODE')['AIRLINE'].to_dict()

font = {'family' : 'DejaVu Sans', 'weight' : 'bold', 'size'   : 15}
mpl.rc('font', **font)
import matplotlib.patches as mpatches
#__________________________________________________________________
# I extract a subset of columns and redefine the airlines labeling
df2 = df.loc[:, ['carrier', 'dep_delay']]
df2['carrier'] = df2['carrier'].replace(abbr_companies)
#________________________________________________________________________
colors = ['royalblue', 'grey', 'wheat', 'c', 'firebrick', 'seagreen', 'lightskyblue',
          'lightcoral', 'yellowgreen', 'gold', 'tomato', 'violet', 'aquamarine', 'chartreuse']
#_________________________________
fig = plt.figure(1, figsize=(22,17))
gs=GridSpec(2,2)
ax1=fig.add_subplot(gs[0,0])
ax2=fig.add_subplot(gs[0,1])
ax3=fig.add_subplot(gs[1,:])
#------------------------------
# Pie chart nº1: nb of flights
#------------------------------
labels = [s for s in  global_stats.index]
sizes  = global_stats['count'].values
explode = [0.3 if sizes[i] < 20000 else 0.0 for i in range(len(abbr_companies))]
patches, texts, autotexts = ax1.pie(sizes, explode = explode,
                                labels=labels, colors = colors,  autopct='%1.0f%%',
                                shadow=False, startangle=0)
```

```
for i in range(len(abbr_companies)):
    texts[i].set_fontsize(14)
ax1.axis('equal')
ax1.set_title('% of flights per company', bbox={'facecolor':'midnightblue', 'pad':5},
            color = 'w',fontsize=18)
#___________________________________________________
# I set the legend: abreviation -> airline name
comp_handler = []
for i in range(len(abbr_companies)):
    comp_handler.append(mpatches.Patch(color=colors[i],
            label = global_stats.index[i] + ': ' + abbr_companies[global_stats.index[i]]))
ax1.legend(handles=comp_handler, bbox_to_anchor=(0.2, 0.9),
           fontsize = 13, bbox_transform=plt.gcf().transFigure)
#----------------------------------------
# Pie chart nº2: mean delay at departure
#----------------------------------------
sizes  = global_stats['mean'].values
sizes  = [max(s,0) for s in sizes]
explode = [0.0 if sizes[i] < 20000 else 0.01 for i in range(len(abbr_companies))]
patches, texts, autotexts = ax2.pie(sizes, explode = explode, labels = labels,
                                    colors = colors, shadow=False, startangle=0,
                                    autopct = lambda p :  '{:.0f}'.format(p * sum(sizes) / 100))
for i in range(len(abbr_companies)):
    texts[i].set_fontsize(14)
ax2.axis('equal')
ax2.set_title('Mean delay at origin', bbox={'facecolor':'midnightblue', 'pad':5},
            color='w', fontsize=18)
#------------------------------------------------------
# striplot with all the values reported for the delays
#________________________________________________________________________
# I redefine the colors for correspondance with the pie charts
codes = global_stats1.index.tolist()
carriers1 = carriers[carriers['IATA_CODE'].isin(codes)]
abbr_companies = carriers1.set_index('IATA_CODE')['AIRLINE'].to_dict()

colors = ['firebrick', 'gold', 'lightcoral', 'aquamarine', 'c', 'yellowgreen', 'grey',
          'seagreen', 'tomato', 'violet', 'wheat', 'chartreuse', 'lightskyblue', 'royalblue',
          'black', 'grey', 'white', 'silver', 'black', 'pink']
#________________________________________________________________________
ax3 = sns.stripplot(y="carrier", x="dep_delay", size = 4, palette = colors,
                    data=df2, linewidth = 0.5,  jitter=True)
plt.setp(ax3.get_xticklabels(), fontsize=14)
plt.setp(ax3.get_yticklabels(), fontsize=14)
ax3.set_xticklabels(['{:2.0f}h{:2.0f}m'.format(*[int(y) for y in divmod(x,60)])
                         for x in ax3.get_xticks()])
plt.xlabel('Departure delay', fontsize=18, bbox={'facecolor':'midnightblue', 'pad':5},
           color='w', labelpad=20)
ax3.yaxis.label.set_visible(False)
#_________________________
plt.tight_layout(w_pad=3)
```

```
#plotting mean delays by airlines
carrier_code=carriers.set_index('IATA_CODE')['AIRLINE'].to_dict()
mpl.rc('patch', edgecolor = 'dimgray', linewidth = 1)
mpl.rcParams.update(mpl.rcParamsDefault)
mpl.rcParams['hatch.linewidth'] = 2.0

fig = plt.figure(1, figsize = (11, 6))
ax = sns.barplot(x = 'dep_delay', y = 'carrier', data = df, color = 'lightskyblue', ci = None)
ax = sns.barplot(x = 'arr_delay', y = 'carrier', data = df, color = 'r', hatch = '///', alpha = 0.0, ci
= None)
labels = [carrier_code[item.get_text()] for item in ax.get_yticklabels()]
ax.set_yticklabels(labels)
ax.yaxis.label.set_visible(False)
plt.xlabel("Mean delay [min] (@departure: blue, @arrival: hatch lines)", fontsize = 15,
weight = 'bold', labelpad = 10)
mpl.rc('patch', edgecolor = 'dimgray', linewidth = 1)
mpl.rcParams.update(mpl.rcParamsDefault)
mpl.rcParams['hatch.linewidth'] = 2.0

fig = plt.figure(1, figsize = (10, 6))

#Subset 4 major airlines
ax = sns.barplot(x = 'dep_delay', y = 'carrier', data = df, order = ['AA', 'DL', 'F9', 'HA', 'B6'],
color = 'lightskyblue', ci = None)
ax = sns.barplot(x = 'arr_delay', y = 'carrier', data = df, order = ['AA', 'DL', 'F9', 'HA', 'B6'],
color = 'r', hatch = '///', alpha = 0.0, ci = None)
labels = [carrier_code[item.get_text()] for item in ax.get_yticklabels()]
ax.set_yticklabels(labels)
ax.yaxis.label.set_visible(False)
plt.xlabel("5 Major Carrier's Mean Delay [min] (@departure: blue, @arrival: hatch lines)",
fontsize = 12, weight = 'bold', labelpad = 10)
plt.pie(df['origin'].value_counts(),
    labels = df['origin'].value_counts().index,
    explode = (0.1, 0, 0),
    startangle = 90,
    autopct = '%1.1f%%',
    colors = ['#52D017', '#F62217', '#43C6DB'])

plt.tight_layout()
plt.title("New York City Airport Market share")
plt.show()
fig = plt.figure(1, figsize = (12, 6))
df[df['origin'] == 'EWR']['month'].value_counts().sort_index().plot(kind = 'line', color =
'#52D017')
df[df['origin'] == 'JFK']['month'].value_counts().sort_index().plot(kind = 'line', color =
'#F62217')
df[df['origin'] == 'LGA']['month'].value_counts().sort_index().plot(kind = 'line', color =
'#43C6DB')

plt.title("Flights in US", size = 15)
```

```
plt.xticks(range(1, 13), size = 12)
plt.yticks(size = 12)
plt.xlabel("Month", size = 17)
plt.ylabel("Frequency", size = 17)
plt.legend(['EWR', 'JFK', 'LGA'])
def map_labels(delays):
    if delays > 15:
        return 1
    else:
        return 0

df['delayed'] = ((df['dep_delay'].map(map_labels) + df['arr_delay'].map(map_labels)) !=
0).astype(int)
df['delayed'].value_counts(normalize = True)
# feature omission
columns_to_remove = ['dep_time', 'sched_dep_time', 'dep_delay', 'arr_time', 'sched_arr_time',
'arr_delay', 'flight', 'tailnum', 'air_time', 'distance', 'hour', 'minute', 'time_hour']
df.drop(columns_to_remove, axis = 1, inplace = True)
df['delayed'].value_counts().to_frame()
df['dest'].value_counts().to_frame()
saving_data = df.to_csv("Data/Processed_data.csv", index = False)
### Imports
import seaborn as sns
from sklearn.model_selection import cross_val_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import jaccard_score
from sklearn import metrics
from sklearn import preprocessing
import matplotlib.pyplot as plt
# Import dataset
data = pd.read_csv('Data/Processed_data.csv')
columns= ['carrier','dest', 'origin']
le=LabelEncoder()
for i in columns:
     data[i]=le.fit_transform(data[i])
data['carrier'].unique()
data['origin'].unique()
data['dest'].unique()
X = data.iloc[:, 0:6].values # from column(years) to column(distance)
X[0:5]
y = data['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)
from sklearn.ensemble import RandomForestClassifier
RF = RandomForestClassifier(n_jobs=-1,random_state=1000)
RF.fit(X_train, y_train)
predictions = RF.predict(X_test)
print("*Confusion Matrix for RF: ")
print(confusion_matrix(y_test, predictions))
print("*Classification Report for RF: ")
print(classification_report(y_test, predictions))
confusion = confusion_matrix(y_test, predictions)
TP = confusion[0, 0]
TN = confusion[0, 1]
FP = confusion[1, 0]
FN = confusion[1, 1]
classification_error = (FP + FN) / float(TP + TN + FP + FN)
print(classification_error)
RF_ACC = accuracy_score(y_test, predictions)
print("Accuracy: ",RF_ACC)
from sklearn import metrics
RF_sensitivity = (TP / float(FN + TP))
print("Sensitivity: ",RF_sensitivity)
RF_specificity = (TN / (TN + FP))
print("Specificity: ",RF_specificity)
from sklearn import tree
DT = tree.DecisionTreeClassifier()
DT.fit(X_train, y_train)
predictions = DT.predict(X_test)
print("*Confusion Matrix for DT: ")
print(confusion_matrix(y_test, predictions))
print("*Classification Report for DT: ")
print(classification_report(y_test, predictions))
confusion = confusion_matrix(y_test, predictions)
TP = confusion[0, 0]
TN = confusion[0, 1]
FP = confusion[1, 0]
FN = confusion[1, 1]
classification_error = (FP + FN) / float(TP + TN + FP + FN)

print(classification_error)
DT_ACC = accuracy_score(y_test, predictions)
print("Accuracy: ",DT_ACC)
from sklearn import metrics
DT_sensitivity = (TP / float(FN + TP))
print("Sensitivity: ",DT_sensitivity)
DT_specificity = (TN / (TN + FP))
print("Specificity: ",DT_specificity)
from sklearn.neural_network import MLPClassifier
MLP = MLPClassifier()
MLP.fit(X_train, y_train)
```

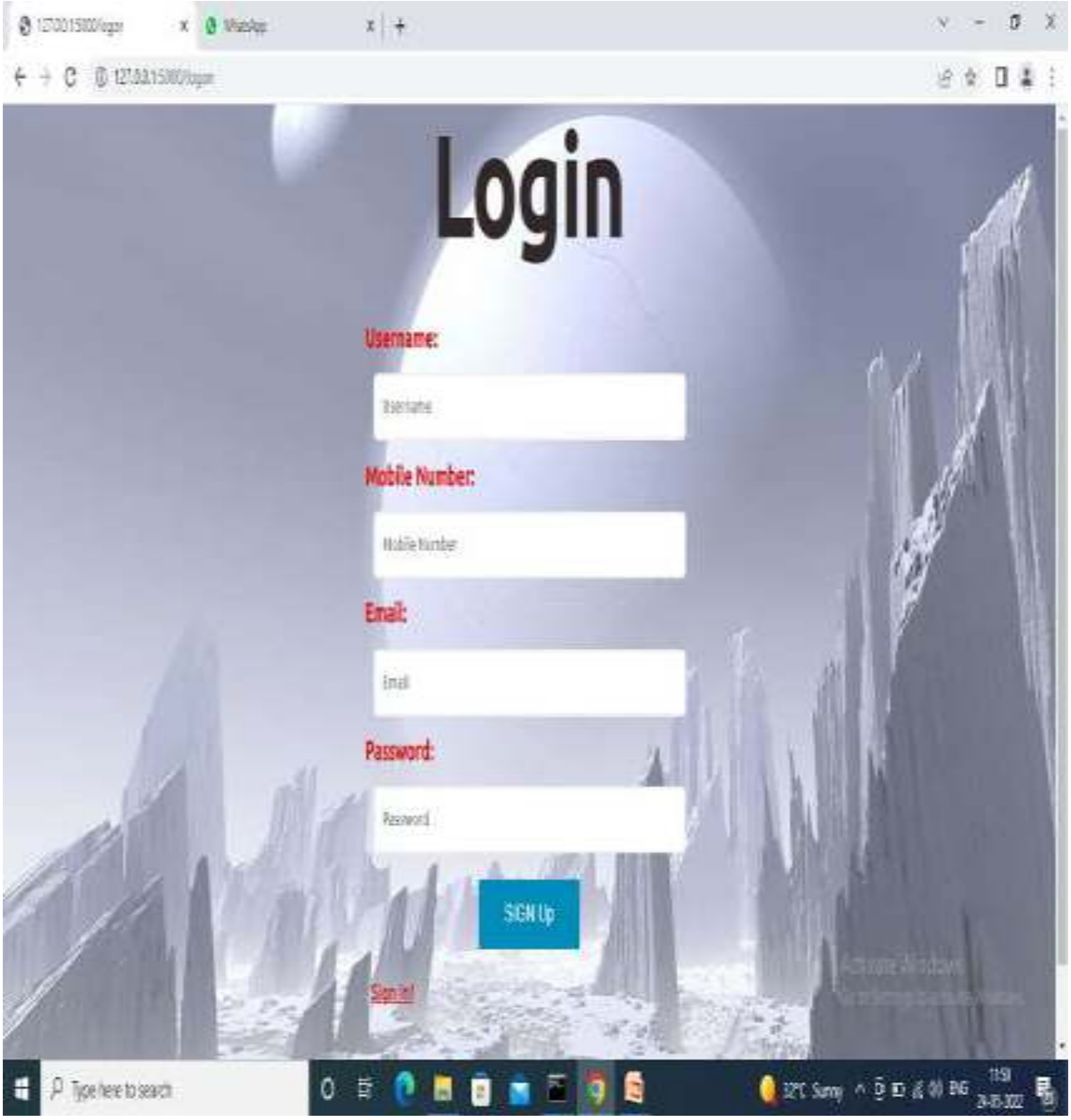
```
predictions = MLP.predict(X_test)
print("*Confusion Matrix for MLP: ")
print(confusion_matrix(y_test, predictions))
print("*Classification Report for MLP: ")
print(classification_report(y_test, predictions))
confusion = confusion_matrix(y_test, predictions)
TP = confusion[0, 0]
TN = confusion[0, 1]
FP = confusion[1, 0]
FN = confusion[1, 1]
classification_error = (FP + FN) / float(TP + TN + FP + FN)

print(classification_error)
MLP_ACC = accuracy_score(y_test, predictions)
print("Accuracy: ",MLP_ACC)
from sklearn import metrics
MLP_sensitivity = (TP / float(FN + TP))

print("Sensitivity: ",MLP_sensitivity)
MLP_specificity = (TN / (TN + FP))

print("Specificity: ",MLP_specificity)
from sklearn.naive_bayes import BernoulliNB
BNB = BernoulliNB()
BNB.fit(X_train, y_train)
predictions = BNB.predict(X_test)
print("*Confusion Matrix for BNB: ")
print(confusion_matrix(y_test, predictions))
print("*Classification Report for BNB: ")
print(classification_report(y_test, predictions))
confusion = confusion_matrix(y_test, predictions)
TP = confusion[0, 0]
TN = confusion[0, 1]
FP = confusion[1, 0]
FN = confusion[1, 1]
classification_error = (FP + FN) / float(TP + TN + FP + FN)
print(classification_error)
BNB_ACC = accuracy_score(y_test, predictions)
print("Accuracy: ",BNB_ACC)
from sklearn import metrics
BNB_sensitivity = (TP / float(FN + TP))
print("Sensitivity: ",BNB_sensitivity)
BNB_specificity = (TN / (TN + FP))
print("Specificity: ",BNB_specificity)
from sklearn.neighbors import KNeighborsClassifier
KNN = KNeighborsClassifier()
KNN.fit(X_train, y_train)
predictions = KNN.predict(X_test)
print("*Confusion Matrix for KNN: ")
print(confusion_matrix(y_test, predictions))
```

```
print("*Classification Report for KNN: ")
print(classification_report(y_test, predictions))
confusion = confusion_matrix(y_test, predictions)
TP = confusion[0, 0]
TN = confusion[0, 1]
FP = confusion[1, 0]
FN = confusion[1, 1]
classification_error = (FP + FN) / float(TP + TN + FP + FN)

print(classification_error)
KNN_ACC = accuracy_score(y_test, predictions)
print("Accuracy: ",KNN_ACC)
from sklearn import metrics
KNN_sensitivity = (TP / float(FN + TP))

print("Sensitivity: ",KNN_sensitivity)
KNN_specificity = (TN / (TN + FP))

print("Specificity: ",KNN_specificity)
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Bring some raw data.
frequencies = [RF_ACC,DT_ACC,MLP_ACC,BNB_ACC,KNN_ACC]

# In my original code I create a series and run on that,
# so for consistency I create a series from the list.
freq_series = pd.Series(frequencies)

x_labels = ['RF','DT','MLP','BNB','KNN']

# Plot the figure.
plt.figure(figsize=(12, 8))
ax = freq_series.plot(kind='bar')
ax.set_title('Evaluation of ML & DL')
ax.set_xlabel('Classifier!')
ax.set_ylabel('Accuracy Range')
ax.set_xticklabels(x_labels)
```

5.SCREENSHOTS

5.1 REGISTRATION PAGE

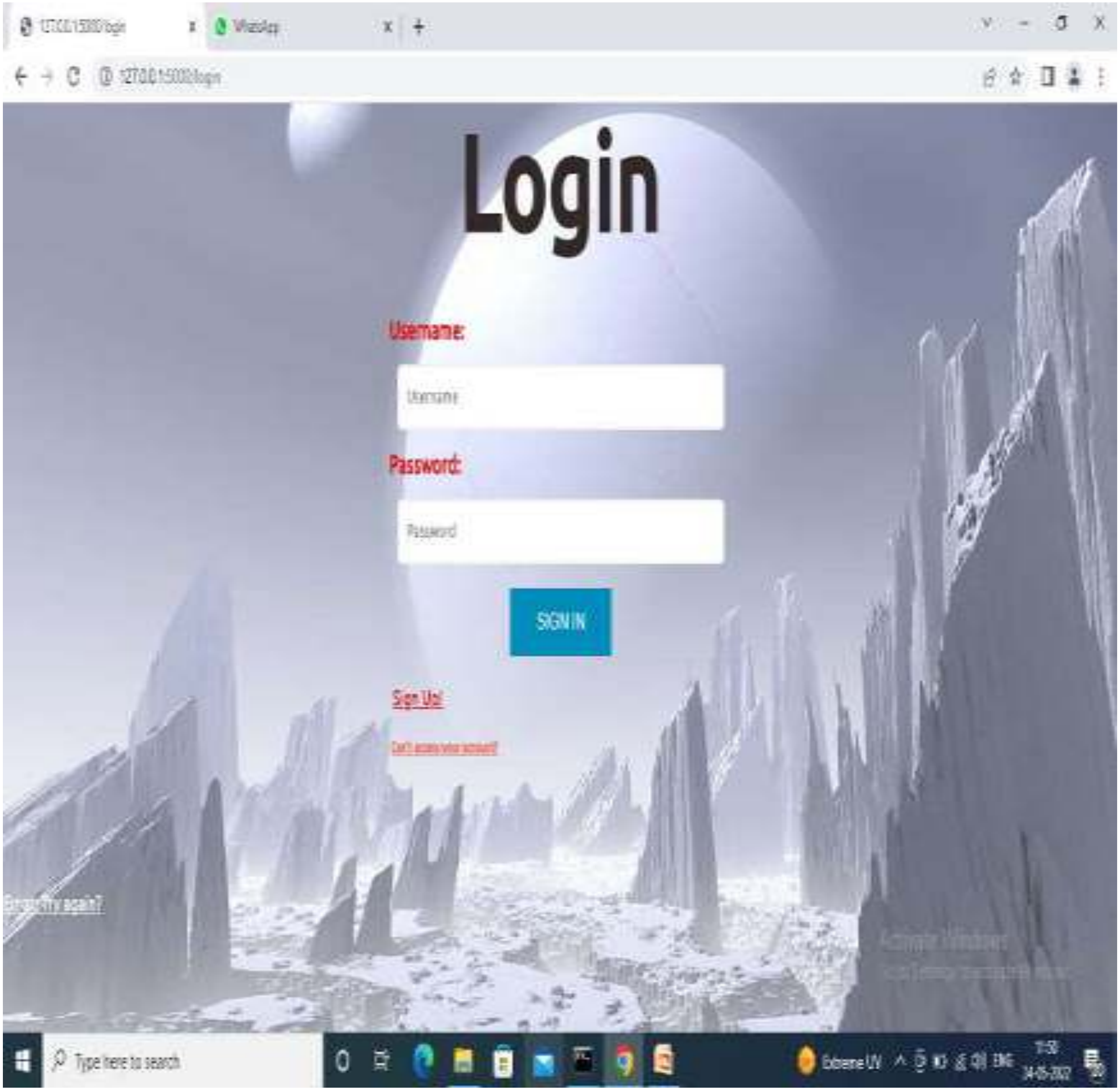
In the registration page the user will register into the portal and then the user will be able to login by the username and password



Screenshot 5.1 Registration Page

5.2 LOGIN PAGE

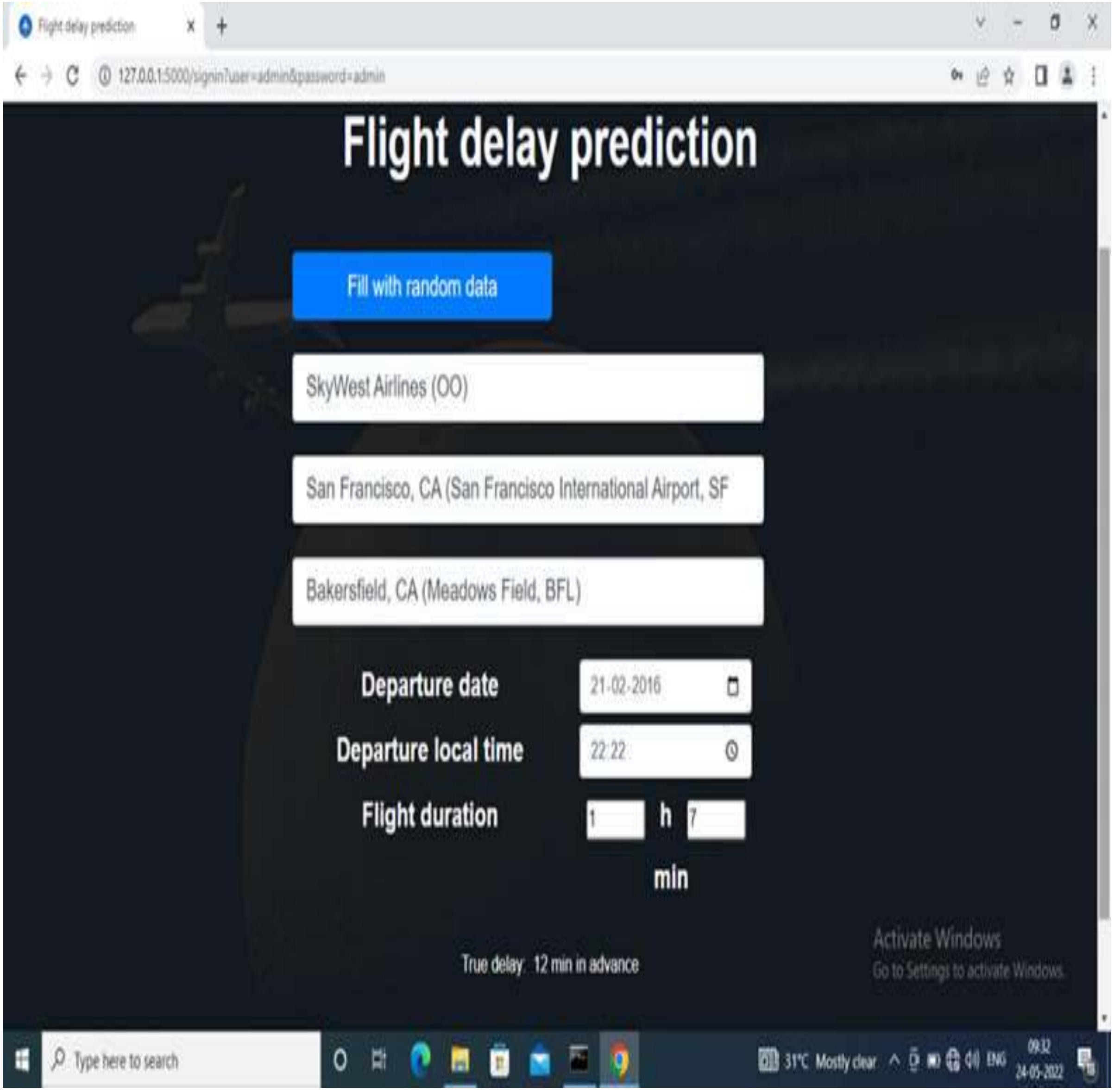
Here the user will be able to login by using the username and password



Screenshot 5.2 Login Page

5.3 PREDICTION OF FLIGHT DELAY

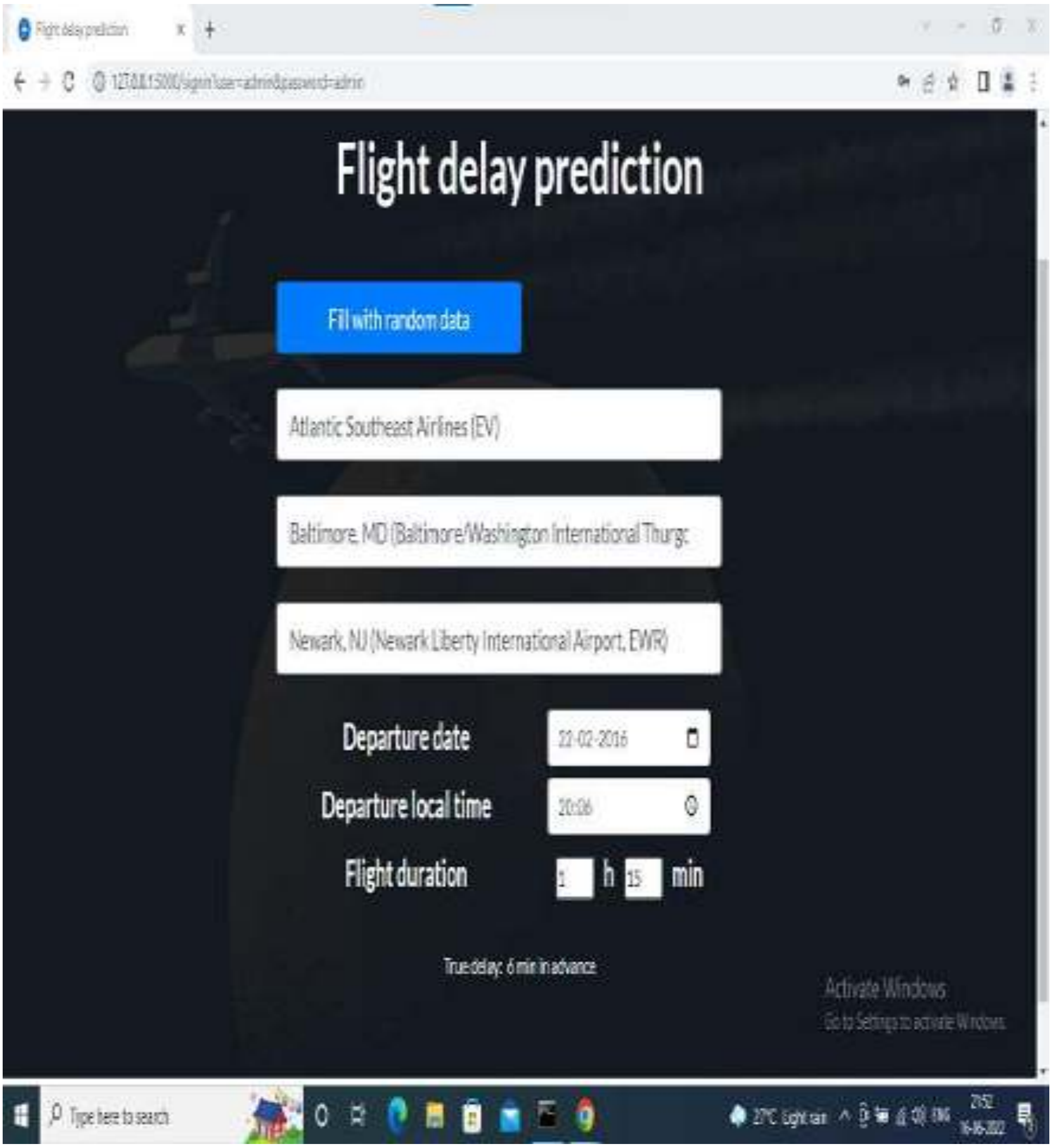
Here in this page we predict the Flight Delays by filling the required essentials in the given fields.



Screenshot 5..3 Prediction of Flight Delay

5.4 PREDICTION OF FLIGHT DELAY

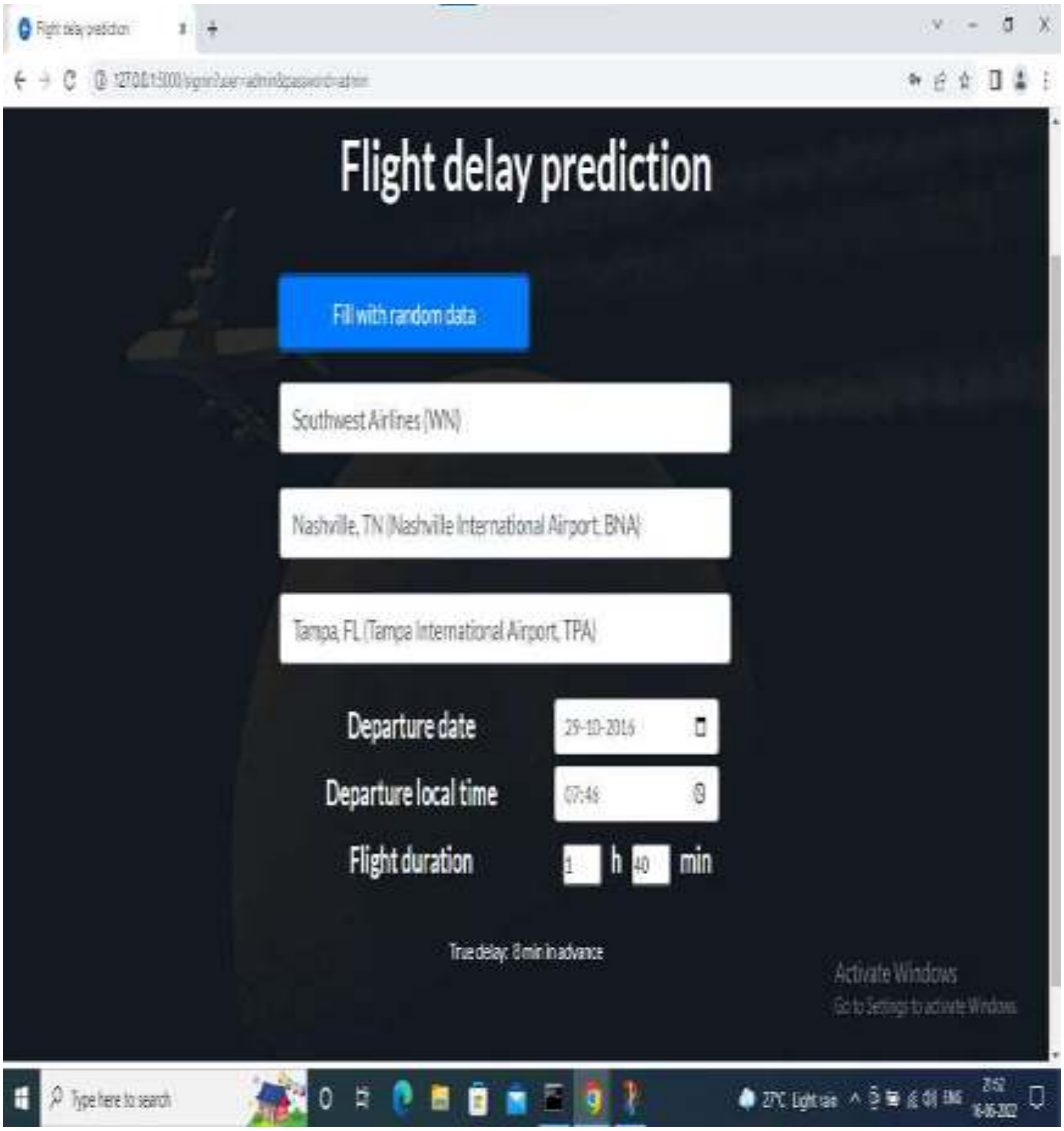
Here in this page we predict the Flight Delays by filling the required essentials in the given fields



Screenshot 5.4 Prediction of Flight Delay

5.5 PREDICTION OF FLIGHT DELAY

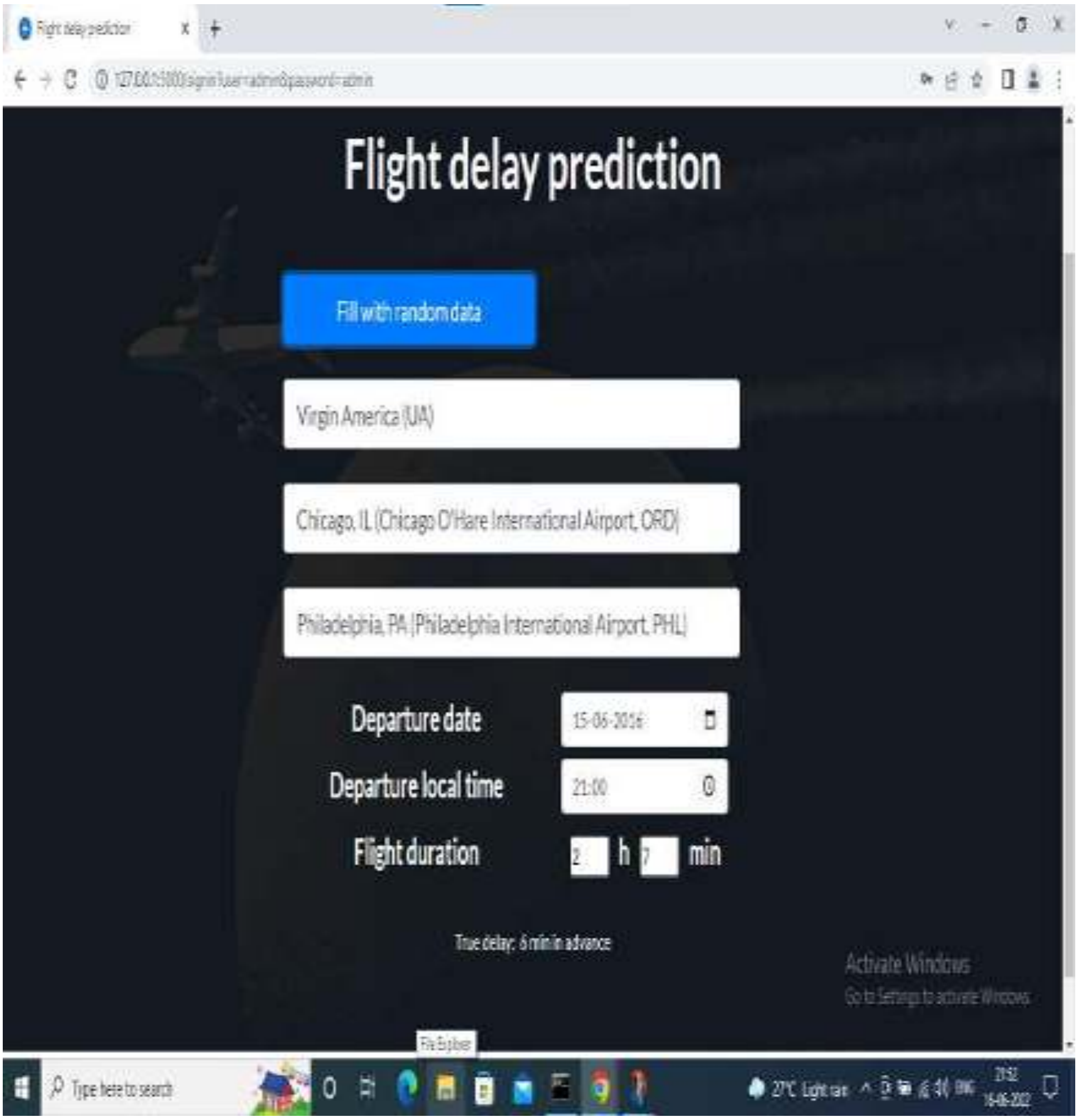
Here in this page we predict the Flight Delays by filling the required essentials in the given fields



Screenshot 5.5 Prediction of Flight Delay

5.6 PREDICTION OF FLIGHT DELAY

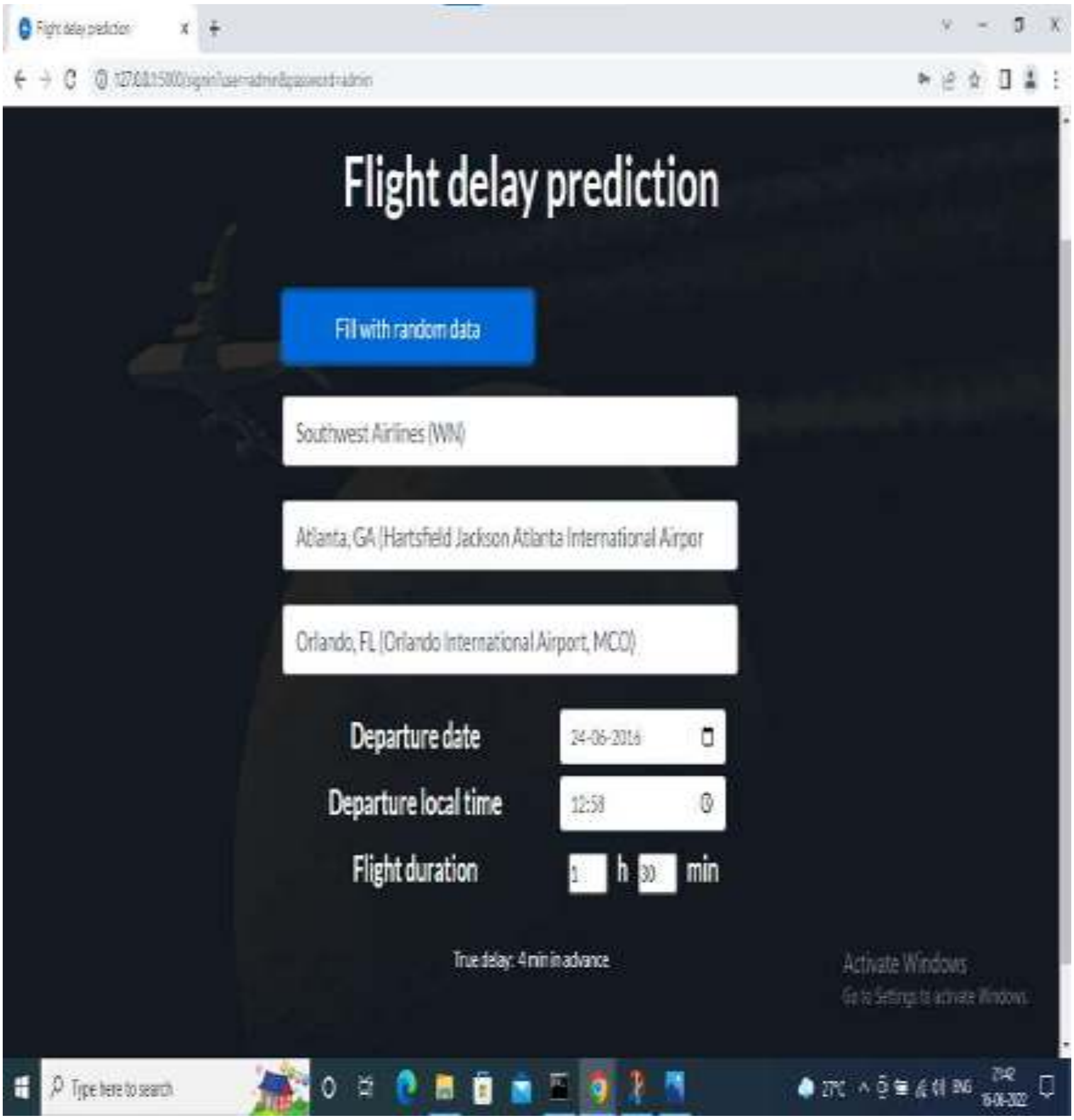
Here in this page we predict the Flight Delays by filling the required essentials in the given fields



Screenshot 5.6 Prediction of Flight Delay

5.7 PREDICTION OF FLIGHT DELAY

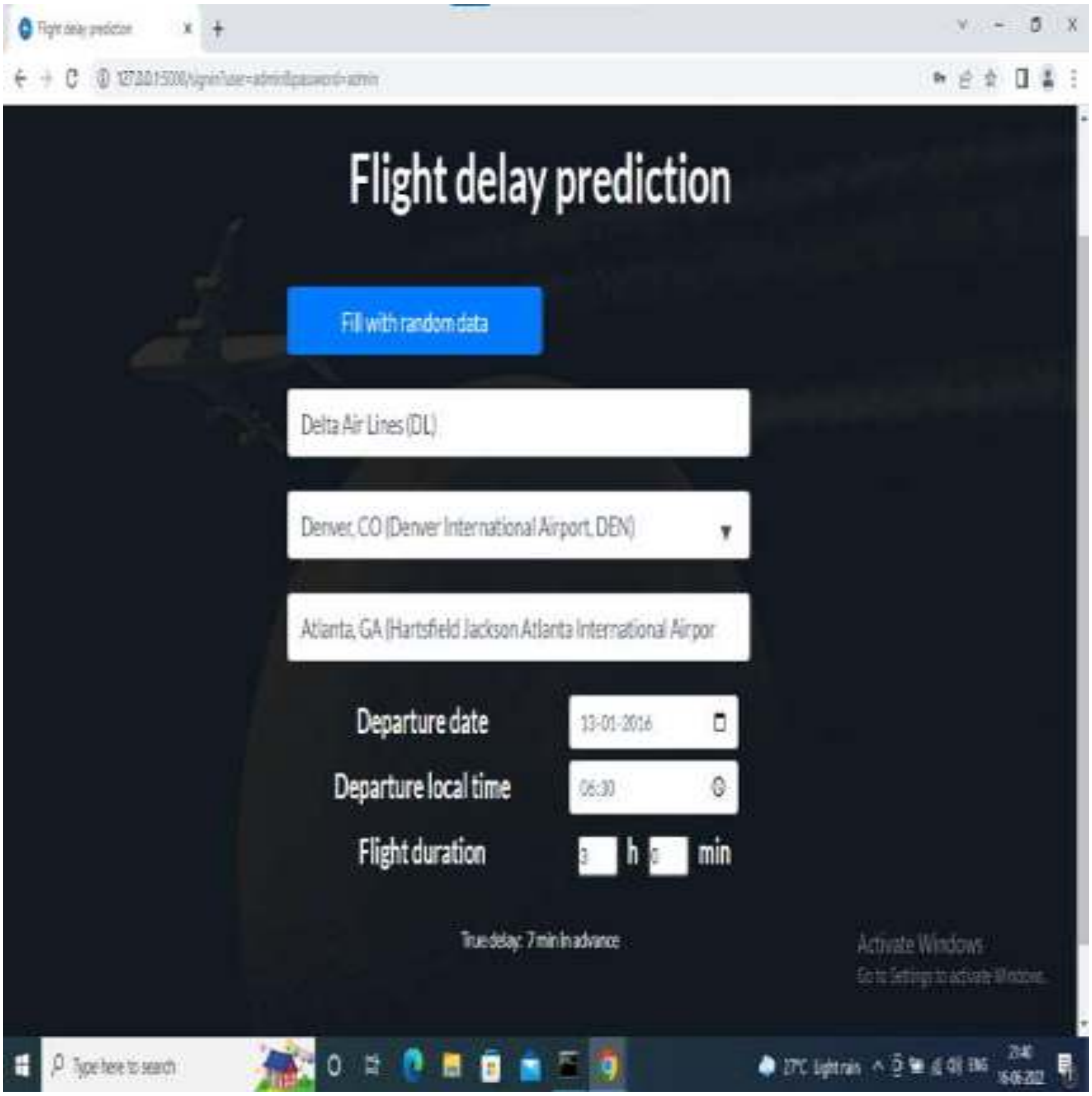
Here in this page we predict the Flight Delays by filling the required essentials in the given fields



Screenshot 5.7 Prediction of Flight Delay

5.8 PREDICTION OF FLIGHT DELAY

Here in this page we predict the Flight Delays by filling the required essentials in the given fields



Screenshot 5.8 Prediction of Flight Delay

6. TESTING

6. TESTING

6.1 INTRODUCTION TO 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, subassemblies, 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 tests. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 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.

6.2.2 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 satisfied, as shown 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.

6.2.3 FUNCTIONAL TESTING

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.

6.3 TEST CASES

6.3.1 USER REGISTRATION

Test case ID	Test case name	Purpose	Test Case	Output
1	User Interface	Check all the text boxes,radio buttons, buttons, etc	Click on Radio buttons, buttons and dropdowns	UI is perfect.
2	Password Validation	Check the password when passing valid data	Enter value in alphanumeric in between 8-32 and click on register button	It should not show any validation message

Table:User Registration

6.3.2 LOGIN FORM

Test case ID	Test case name	Purpose	Test Case	Output
1	User Interface	Check all the Text boxes and buttons.	Check the Page	UI should be Perfect.
2	User Login	Check when passing a correct username and invalid Password.	Enter valid username and invalid Password and click Login button	User should not Login and Should show proper error message.

Table:User Login

7. CONCLUSION

7.CONCLUSION & FUTURE ENHANCEMENTS

7.1PROJECT CONCLUSION

This develops a system to predict the delay in flights and it gives the range of different methodology that is used to find out the delay in flights. As flight delay cost a lot to the airlines as well as passengers financial and environmental terms.This Flight Delay may increase prices to customers and operational prices to airlines.Hence there is a requirement to scale back monitory loss by having higher and smoother operation.

7.2 FUTURE ENHANCEMENTS

The future scope of this paper can include the application of more advanced, modern and innovative preprocessing techniques, automated hybrid learning and sampling algorithms, and deep learning models adjusted to achieve better performance. To evolve a predictive model, additional variables can be introduced our future work will focus on collecting or generating more training data, integrating more information like airport traffic flow, airport visibility into our dataset, and designing more delicate networks.

8.BIBLIOGRAPHY

8.BIBLIOGRAPHY

8.1 REFERENCES

- M. Leonardi, “Ads-b anomalies and intrusions detection by sensor clocks tracking,” IEEE Trans. Aerosp. Electron. Syst., to be published,doi:10.1109/TAES.2018.2886616.

- Y. A. Nijsure, G. Kaddoum, G. Gagnon, F. Gagnon, C. Yuen, and R. Mahapatra, “Adaptive air-to-ground secure communication system based onads-b and wide-area multilateration,” IEEE Trans. Veh. Technol., vol. 65,no. 5, pp. 3150–3165, 2015.

- J. J. Rebollo and H. Balakrishnan, “Characterization and prediction of air traffic delays,” Transp. Res. Part C Emerg. Technol., vol. 44, pp.231–241, 2014.

- W. Wang, R. Wu, and J. Liang, “ADS-B Signal Separation based on Blind Adaptive Beamforming,” IEEE Trans. Veh. Technol., vol. PP, no. c,pp. 1–1, 2019.

- Rebollo de la Bandera, and Juan José, “Characterization and Prediction of Air Traffic Delays”, DSpace@MIT, Massachusetts Institute of Technology, 2012, http://dspace.mit.edu/handle/1721.1/76107

- Shervin AhmadBeygi, Amy Cohn, and Yihan Guan, “Analysis of the Potential for Delay Propagation in Passenger Aviation Flight Networks”, April 1, 2007.

- Tu, Y., Ball, M. O., and Jank, W. S. “Estimating flight departure delay distributions - A statistical approach with long-term trend and short-term pattern”, Journal of the American Statistical Association, 103(481), pp. 112–125, 2008

- N. Kato, Z. M. Fadlullah, B. Mao, F. Tang, O. Akashi, T. Inoue, and K. Mizutani, “The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective,” IEEE Wireless Commun., vol. 24, pp. 146–153, June 2017.

- J. Wang, J. Liu, and N. Kato, “Networking and communications in autonomous driving: A survey,” IEEE Commun. Surveys Tuts., vol. 21, pp. 1243–1274, April 2019

- Z. Shi, M. Xu, Q. Pan, B. Yan, and H. Zhang, “LSTM-based flight trajectory prediction,” in Proc. Int. Jt. Conf. Neural Networks, pp. 1–8, IEEE, 2018.

- S. O. Sahin and S. S. Kozat, “Nonuniformly sampled data processing using lstm networks,” IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 5, pp. 1452–1461, 2018.

- T. Ergen and S. S. Kozat, “Online training of lstm networks in distributed systems for variable length data sequences,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 10, pp. 5159–5165, 2017.

- J. J. Robollo, Hamsa, Balakrishnan, "Characterization and Prediction of Air Traffic Delays".

- S. Sharma, H. Sangoi, R. Raut, V. C. Kotak, S. Oza, "Flight Delay Prediction System Using Weighted Multiple Linear Regression," International Journal of Engineering and Computer Science, vol. 4, no. 4, pp. 11668 - 11677, April 2015.

- A. M. Kalliguddi, Area K., Leboulluec, "Predictive Modelling of Aircraft Flight Delay," Universal Journal of Management, pp. 485 - 491, 2017.

- Noriko, Etani, "Development of a predictive model for on-time arrival fight of airliner by discovering correlation between fight and weather data," 2019.

-] N. Cheng, F. Lyu, W. Quan, C. Zhou, H. He, W. Shi, and X. Shen, "Space/aerial-assisted computing offloading for iot applications: A learning-based approach," IEEE J. Sel. Areas in Commun., vol. 37, no. 5, pp. 1117–1129, 2019.

- [28] H. Huang, Y. Peng, J. Yang, W. Xia, and G. Gui, "Fast beamforming design via deep learning," IEEE Trans. Veh. Technol., to be published, doi: 10.1109/TVT.2019.2949122.

- .M. Liu, T. Song, and G. Gui, "Deep cognitive perspective: Resource allocation for noma-based heterogeneous iot with imperfect sic," IEEE Internet Things J., vol. 6, no. 2, pp. 2885–2894, 2018.

8.2 WEBSITES

- www.tutorialspoint.com
- www.sciencedirect.com
- www.scribd.com

8.3 GITHUB LINK

https://github.com/PoudapallyRajeshwarReddy/Flight_Delay_Prediction