HX8001 - PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

DEVELOPING A FLIGHT DELAY PREDICTION MODEL USING MACHINE LEARNING

SIVAKUMAR M (211719104131)

SABAREESVARAN S (211719101111)

SAI PRIYA S (211719104112)

SHIFI S J (211719104127)

Dissertation submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING

Branch: COMPUTER SCIENCE ENGINEERING

of Anna University

NOVEMBER 2022

RAJALAKSHMI INSTITUTE OF TECHNOLOGY

(Anna University Affiliated)

CHEMBARAMBAKKAM

– 600124

RAJALAKSHMI INSTITUTE OF TECHNOLOGY

CHEMBARAMBAKKAM -

600124

DEVELOPING A FLIGHT DELAY PREDICTION MODEL USING MACHINE LEARNING

Bonafide record of work done by

SIVAKUMAR M (211719104112)

SABAREESVARAN S (211719101111)

SAI PRIYA S(211719104112)

SHIFI S J (211719104127)

Project report submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING

Branch: COMPUTER SCIENCE ENGINEERING

of Anna University

NOVEMBER 2022

....	....
SUBHA S	**RAJENDRAN T**
Faculty Guide	Head of the Department

....	....
LALITHA GAYATHRI	**ANJANA DEVI**
Industry Mentor	Internal Evaluator

CONTENTS

CHAPTER 1
INTRODUCTION

In the present world, the major components of any transportation system include passenger airline, cargo airline, and air traffic control system. With the passage of time, nations around the world have tried to evolve numerous techniques of improving the airline transportation system. This has brought drastic change in the airline operations. Flight delays occasionally cause inconvenience to the modern passengers. Every year approximately 20% of airline flights are canceled or delayed, costing passengers more than 20 billion dollars in money and their time.

1.1. PROJECT OVERVIEW

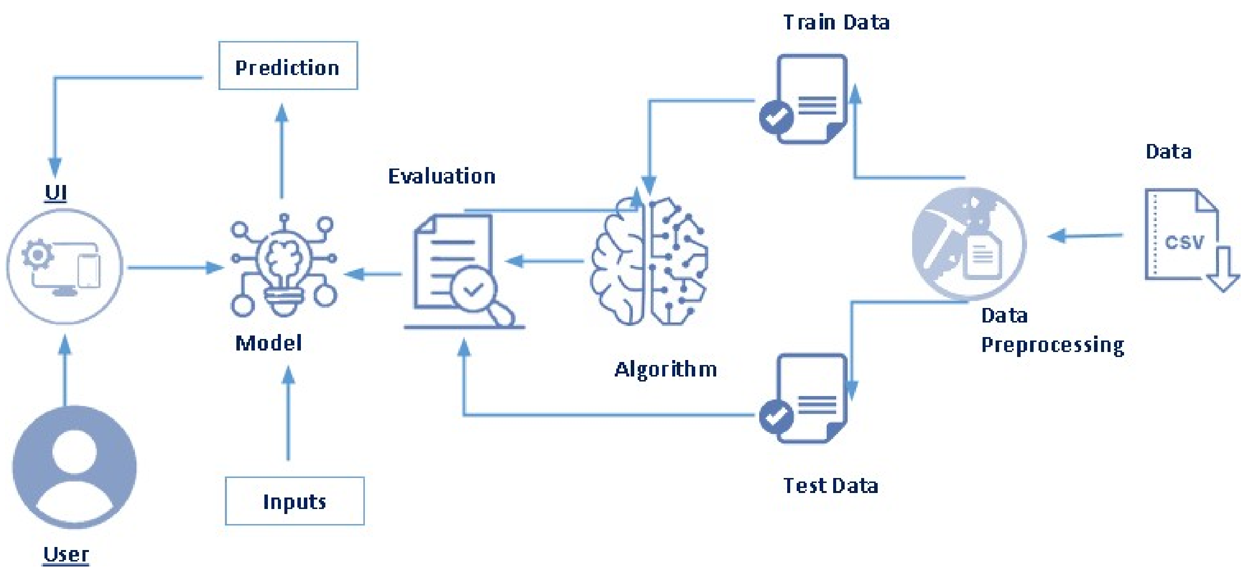


Figure 1.1. Technical Architecture

To collect the Passengers flight on-time performance data, pre-process the collected data, and apply some learning algorithms with data science to predict a delay of flight.Flight delays can be predicted using machine learning algorithm. The flight passengers need a way to predict the delay of flight so that they can plan their work accordingly. It also helps to manage the flight operations effectively.

1.2. PURPOSE

Flight Delay Prediction aims to predict the delay in the aircrafts due to increasing number of travelers in the recent times. An aircraft arrival is considered to be delay if the aircraft is late by over 15 minutes between the scheduled time and the arrival time. Flight Delay Prediction takes into consideration various attributes of the delay which includes scheduled time, source and destination of the flight, arrival time of the flight and departure time of the flight and many more attributes to predict the delay in the flight arrivals. These flight delays help the user massively to select the airlines, to select the source station and other economical aspects of the travelers. At the same time, Flight Delay prediction also helps the airlines to focus on the major reasons of the flight delay and minimize delay time on future occasions. Aviation industry are also benefitted with the help of the Flight Delay Prediction.

CHAPTER 2

LITERATURE SURVEY

[1] H. Khaksar and A. Sheikholeslami, “Airline delay prediction by machine learning algorithms”, Scientia Iranica, Transactions A: Civil Engineering 26 (2019) 2689-2702.

Proposed work: This paper proposes a flight delay prediction model through different methods which includes Bayesian modeling, decision tree, cluster classification, random forest, and hybrid methods. These methods were applied to estimate the occurrences and magnitude of delay in a network.

[2] Miguel Lambelho, Mihaela Mitici, Simon Pickup, Alan Marsden,”Assessing strategic flight schedules at an airport using machine learning-based flight delay and cancellation predictions”, Journal of Air Transport Management, Volume 82, 2020, 101737, ISSN 0969-6997.

Proposed work: This paper provides a machine learning- based approach to assess the strategic flight schedules in terms of potential arrival/departure flight delays and cancellations. This paper also provides an approach that supports an integrated strategic flight schedule assessment, where strategic flight schedules are evaluated with respect to flight delays and cancellations.

[3] Navoneel Chakrabarty, ”A Data Mining Approach to Flight Arrival Delay Prediction for American Airlines“, The 9th Annual Information Technology, Electromechanical and Microelectronics Conference (IEMECON 2019).

Proposed work: This paper aims at analyzing flight information of US domestic flights operated by American Airlines, covering top 5 busiest airports of the US and predicting possible arrival delay of the flight using Data Mining and Machine Learning Approaches.

[4] Kaiquan Cai, Yue Li, Yiping Fang, Yanbo Zhu,” A Deep Learning Approach for Flight Delay PredictionthroughTime-EvolvingGraphs”.IEEE Transactions on Intelligent Transportation Systems,IEEE,In press,pp.1-11.ff10.1109/TITS.2021.3103502ff. ffhal-03428046f.

Proposed work: This paper is about the flight delay prediction problem is investigated from a network perspective (i.e., multi-airport scenario). To model the time-evolving and periodic graph-structured information in the airport network, a flight delay prediction approach based on the graph convolutional neural network (GCN) is developed in this paper.

[5] Yi Ding,” Predicting flight delay based on multiple linear regression” ,2017 IOP Conf. Ser.: Earth Environ. Sci. 81 012198

Proposed work: This paper proposes a method to model the arriving flights and a multiple linear regression algorithm to predict delay, comparing with Naive-Bayes and C4.5 approach.

[6] Qu, J., Zhao, T., Ye, M. et al. “Flight Delay Prediction Using Deep Convolutional Neural Network Based on Fusion of Meteorological Data.”, Neural Process Lett 52, 1461–1484 (2020).

Proposed work: This paper provides two flight delay prediction models using deep convolutional neural networks based on fusion of meteorological data. The first model is DCNN (Dual- channel Convolutional Neural Network), which refers to the ResNet network structure. The second model is SE- DenseNet (Squeeze and ExcitationDensely Connected Convolutional Network).

[7] G. Gui, F. Liu, J. Sun, J. Yang, Z. Zhou and D. Zhao, "Flight Delay Prediction Based on Aviation Big Data and Machine Learning," in IEEE Transactions on Vehicular Technology, vol. 69, no. 1, pp. 140-150, Jan. 2020, doi: 10.1109/TVT.2019.2954094.

Proposed work: This paper 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 weather condition, flight schedule, and airport information.

[8] Yu, Bin; Guo, Zhen; Asian, Sobhan; Wang, Huaizhu; Chen, Gang (2019),”Flight delay prediction for commercial air transport: A deep learning approach.” Transportation Research Part E: Logistics and Transportation Review.

Proposed work: This paper analyzes high-dimensional data from Beijing International Airport and presents a practical flight delay prediction model. Following a multifactor approach, a novel deep belief network method is employed to mine the inner patterns of flight delays. Support vector regression is embedded in the developed model to perform a supervised fine-tuning within the presented predictive architecture

[9] Esmaeilzadeh, Ehsan; Mokhtarimousavi, Seyedmirsajad (2020).“Machine Learning Approach for Flight Departure Delay Prediction and Analysis”. Transportation Research Record: Journal of the Transportation Research Board.

Proposed work: This paper employs a support vector machine (SVM) model to explore the non- linear relationship between flight delay outcomes. Individual flight data were gathered from 20 days in 2018 to investigate causes and patterns of air traffic delay at three major New York City airports

[10] Etani, Noriko (2019),”Development of a predictive model for on-time arrival flight of airliners by discovering correlation between flight and weather data.”, Journal of Big Data,2019.

Proposed work:This paper aims to discover the correlation between fight data and weather data. A predictive model of on-time arrival flight is proposed using flight data and weather data. The feasibility of the predictive model is evaluated by developing a tool of on-time arrival fight prediction.

CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1. EMPATHY MAP CANVAS

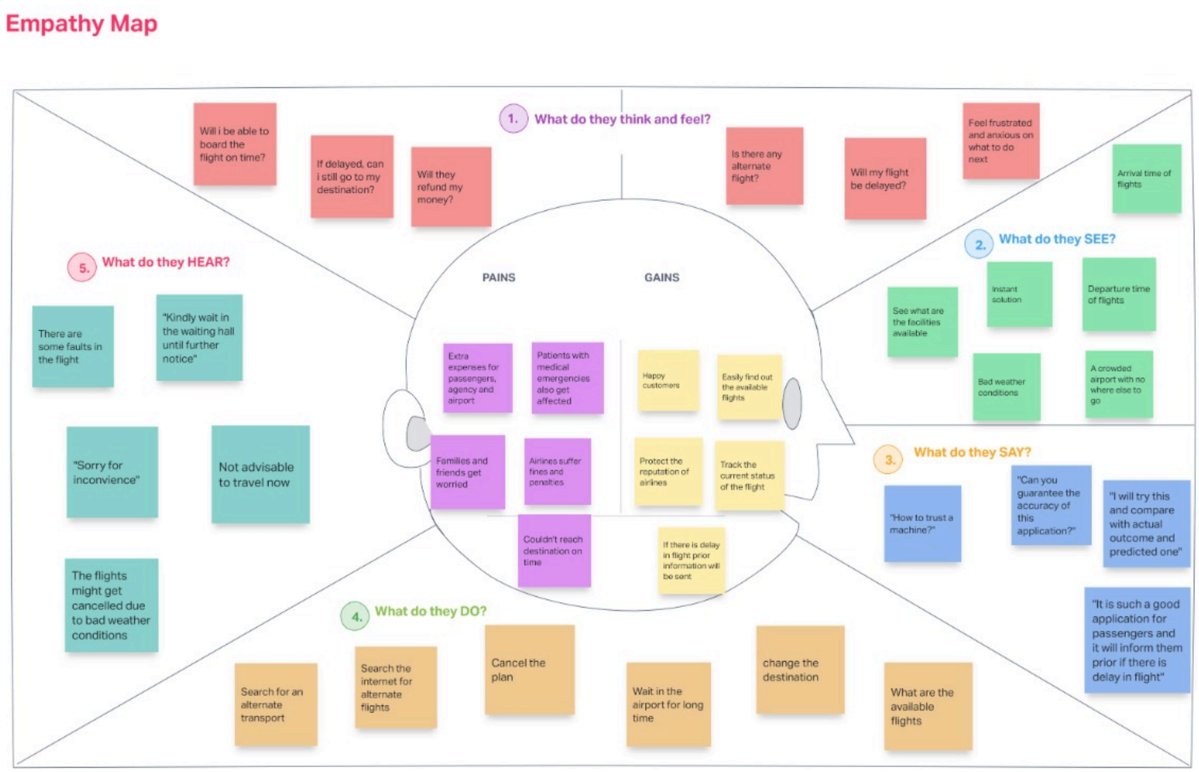
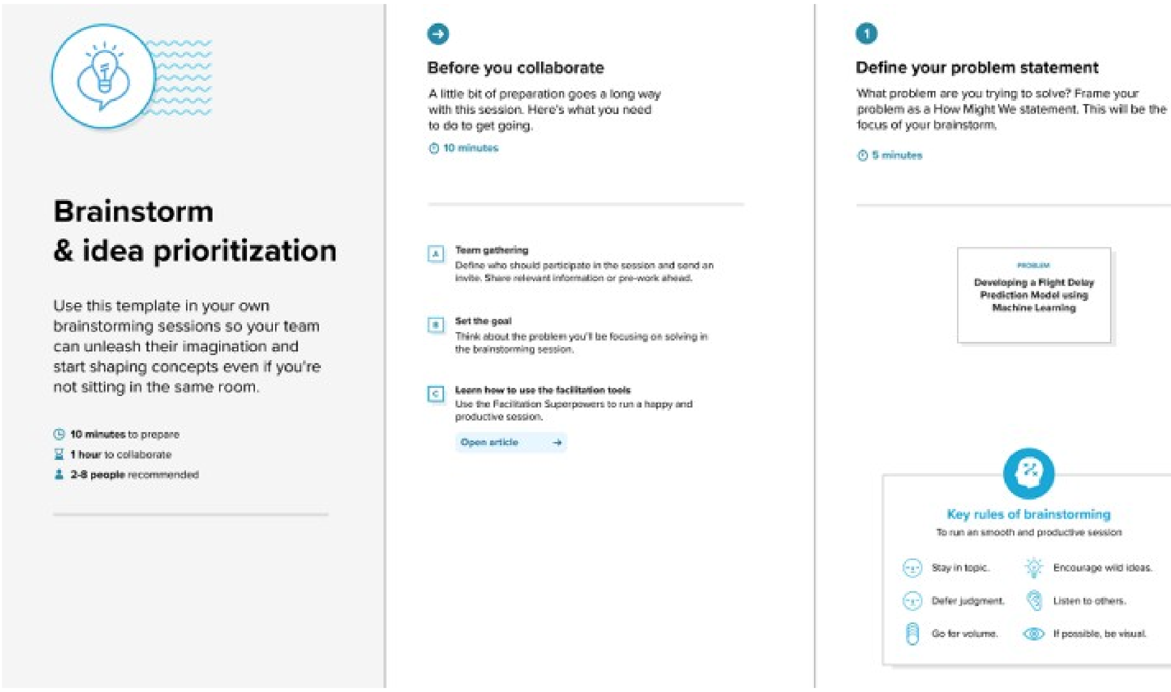


Figure 3.1. Empathy Map

3.2. IDEATION & BRAINSTORMING

Step 1 - Team Gathering, Collaboration and Selecting the Problem Statement



Step 2 - Brainstorm, Idea Listing and Grouping

Step 3 - Idea Prioritization

3.3. PROPOSED SOLUTION

3.4. PROBLEM SOLUTION FIT

CHAPTER 4

REQUIREMENT ANALYSIS

4.1. FUNCTIONAL REQUIREMENT

4.2. NON-FUNCTIONAL REQUIREMENTS

CHAPTER 5

PROJECT DESIGN

5.1. DATA FLOW DIAGRAMS

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Figure 5.1. Data flow diagram

5.2. SOLUTION & TECHNICAL ARCHITECTURE

Figure 5.2. Solution Architecture

5.3. User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook, Instagram, other social media	I can register & access the dashboard with Facebook/Instagram Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register and access the dashboard	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access the dashboard	High	Sprint-1
	Dashboard	USN-6	As a user, I can navigate through different pages using the dashboard	I can access various pages	High	Sprint-1
	Search	USN-7	As a user, I can search for flights for different locations	I can receive information on different flights for various locations	High	Sprint-2
	View	USN-8	As a user, I can view the details of flights	I will get the information such as flight no, departure and arrival time, etc.,	High	Sprint-2
	Receive notifications	USN-9	As a user, I will receive notifications about the flight	I will get frequent updates of the flight's location	Low	Sprint-3
	Track	USN-10	As a user, I can track the location of my flight	I can track my flight	Medium	Sprint-3,4
Admin	GPS	USN-11	As an admin, I will need the location of flights	I can track my flight	High	Sprint-3,4
	Analyze	USN-12	As an admin, I will analyze the given dataset	I can analyze the dataset	High	Sprint-2
	Predict	USN-13	As an admin, I will predict the delays	I can predict the flight delays	High	Sprint-2

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1. SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration and Login	USN-1	As a new user, I can register for the application by entering my email and my password.	2	High	Sivakumar M, Saipriya S
Sprint-2	Confirmation email	USN-2	As a user, I will receive confirmation email once I have registered for the application	2	Medium	Saipriya S, Shifi S j
Sprint-1	User login	USN-3	As a user, I can login into the application by entering the registered email-id and password	2	High	Sabareesvaran S, Sivakumar M
Sprint-2	Admin Panel	USN-4	As an admin, I can authenticate the registration and login credentials of the passengers	2	High	Shifi S J, Sabareesvaran S
Sprint-3	Arrival and Departure time of flights	USN-5	As a user, I can find all the details of a specific flight with its number or name	2	High	Sivakumar M ,Shifi S J
Sprint-3		USN-6	As a user, I can find exactly how long the flight will be delayed	2	High	Saipriya S,Sivakumar M
Sprint-4	Helpdesk	USN-7	As a customer care executive, I can provide the contact details of the airlines	1	Medium	Sabareesvaran S

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-4		USN-8	As a passenger, I can find alternative flights to the destination that are available	1	High	Sivakumar M Saipriya.S sabareesvaran S Shifi S J
Sprint-4	Feedback	USN-9	As a user, I can provide my suggestions and feedback for the improvement of the application	2	Medium	Shifi S J

6.2. SPRINT DELIVERY SCHEDULE

Burn down Chart:

Figure 6.1 - Burn down Chart

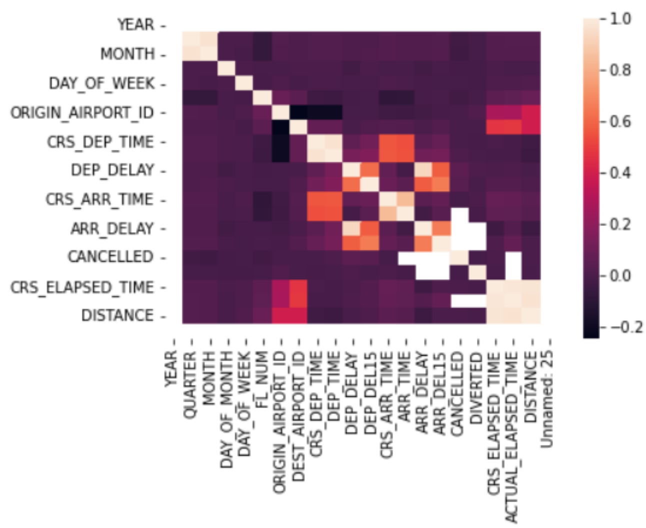
CHAPTER 7

CODING AND SOLUTIONING

7.1. FEATURE 1 - CORRELATION BETWEEN THE VARIABLES IN THE DATASET

In [ ]:
```
sns.heatmap(dataset.corr())
```

Out[ ]:



To find out the correlation between the variables in the dataset which would help us to find out the columns that are unnecessary and hence to be dropped.

7.2. FEATURE 2 - ONE HOT ENCODING

In [ ]:
```
dataset =pd.get_dummies(dataset,columns=['ORIGIN','DEST'])
dataset.head()
```

Out[ ]:

	FL_NUM	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	CRS_ARR_TIME	DEP_DEL15	ARR_DEL15	ORIGIN_0	ORIGIN_1	ORIGIN_2	ORIGIN_3	ORIGIN_4	DEST_0	DEST_1	DEST_2
0	1399	1	1	5	2143	0.0	0.0	1	0	0	0	0	0	0	0
1	1476	1	1	5	1435	0.0	0.0	0	1	0	0	0	0	0	0
2	1597	1	1	5	1215	0.0	0.0	1	0	0	0	0	0	0	0
3	1768	1	1	5	1335	0.0	0.0	0	0	0	0	1	0	0	0
4	1823	1	1	5	607	0.0	0.0	0	0	0	0	1	0	1	0

The cities in both Origin and Destination are one-hot encoded using the above code.

7.3. FEATURE 3 - SAVING THE MODEL WEIGHTS FOR DEPLOYMENT

SAVING THE MODEL

```
In [63]: pickle.dump(classifier,open('flight_new.pk1','wb'))

In [64]: from sklearn.metrics import confusion_matrix
         confusion_matrix(predicted, y_test)

Out[64]: array([[1825,  129],
                [ 138,  117]], dtype=int64)

In [66]: from sklearn.metrics import classification_report
         print(classification_report(predicted, y_test, labels=[1, 2]))

              precision    recall  f1-score   support

           1       0.48      0.46      0.47       255
           2       0.00      0.00      0.00         0

   micro avg       0.48      0.46      0.47       255
   macro avg       0.24      0.23      0.23       255
weighted avg       0.48      0.46      0.47       255
```

The above code will save the model weights for further deployment in IBM Cloud and also measure the performance metrics.

7.4. FEATURE 4 - FLASK INTERFACE - UI

```
from flask import Flask, request, render_template
import numpy as np
import pandas as pd
import pickle
import os
model = pickle.load(open('flight_new.pk1','rb'))
app = Flask(__name__)
@app.route('/')
def home():
   return render_template("mainpage.html")

@app.route('/prediction',methods=['GET','POST'])
def predict():
   name = request.form['fname']
   month = request.form['month']
   dayofmonth = request.form['daymonth']
   dayofweek = request.form['dayweek']
   origin = request.form['origin']
```

```
    if(origin == "msp"):
        origin1, origin2, origin3, origin4, origin5 = 0,0,0,0,1
    if(origin == "dtw"):
        origin1, origin2, origin3, origin4, origin5 = 1,0,0,0,0
    if(origin == "jfk"):
        origin1, origin2, origin3, origin4, origin5 = 0,0,1,0,0,
    if(origin == "sea"):
        origin1, origin2, origin3, origin4, origin5 = 0,1,0,0,0
    if(origin == "atl"):
        origin1, origin2, origin3, origin4, origin5 = 0,0,0,1,0
    destination = request.form['destination']
    if(destination == "msp"):
        destination1,destination2,destination3,destination4,destination5 = 0,0,0,0,1
    if(destination == "dtw"):
        destination1,destination2,destination3,destination4,destination5 = 1,0,0,0,0
    if(destination == "jfk"):
        destination1,destination2,destination3,destination4,destination5 = 0,0,1,0,0
    if(destination == "sea"):
        destination1,destination2,destination3,destination4,destination5 = 0,1,0,0,0
    if(destination == "atl"):
        destination1,destination2,destination3,destination4,destination5 = 0,0,0,1,0
    dept = request.form['sdeparttime']
    arrtime = request.form['sarrivaltime']
    actdept = request.form['adeparttime']
    dept15 = int(dept)-int(actdept)
    total =
[[name,month,dayofmonth,dayofweek,arrtime,dept15,origin1,origin2,origin3,origin4,origi
n5,destination1,destination2,destination3,destination4,destination5]]
    y_pred = model.predict(total)
    print(y_pred)
    if(y_pred == [0.]):
        ans = "The Flight will be on time"
    else:
        ans = "The Flight will be delayed"
    return render_template("index.html",data = ans)

app.run(debug=True)
```

Explanation:

The above code will be able to get the details of the flight from the user in the respective text fields created using the HTML, scale the inputs and give the inputs to the model which has been developed already. The predictions are shown in another HTML page.

7.5. FEATURE 5 - HTML PAGES FOR FRONTEND DESIGN

```
<html>
<div align="center" class="logbg">
<head>
<meta charset="UTF-8">
<center>
<table>
<tr>
<td><h1><br>Prediction of Flight Delay<br><br></h1></td>
</tr>
</table>
</center>
</head>
<body background='C:\Users\Public\project\templates\flight_4.jpg'>
<form action="http://localhost:5000/prediction" method="POST" >
<center>
<table>
<tr>
<td>Enter the flight number:</td>
<td><input type="number" name="fname"><br></td>
</tr>
<tr>
<td>Month:</td>
<td><input type="number" name="month"><br></td>
</tr>
<tr>
<td>Day of Month:</td>
<td><input type="number" name="daymonth"><br></td>
</tr>
<tr>
<td>Day of Week:</td>
<td><input type="number" name="dayweek"><br></td>
</tr>
<tr>
<td>Origin:</td>
```

```html
<td><select name="origin">
<option value="atl">ATL</option>
<option value="dtw">DTW</option>
<option value="sea">SEA</option>
<option value="msp">MSP</option>
<option value="jfk">JFK</option>
</select></td>
<tr>
<tr>
<td>Destination:</td>
<td><select name="destination">
<option value="atl">ATL</option>
<option value="dtw">DTW</option>
<option value="sea">SEA</option>
<option value="msp">MSP</option>
<option value="jfk">JFK</option>
</select></td>
<tr>
<tr>
<td>Scheduled Departure Time:</td>
<td><input type="number" name="sdeparttime"><br></td>
</tr>
<tr>
<td>Scheduled Arrival Time:</td>
<td><input type="number" name="sarrivaltime"><br></td>
</tr>
<tr>
<td>Actual Departure Time:</td>
<td><input type="number" name="adeparttime"><br></td>
</tr>
<tr>
<td><br><input type="submit" class="btn" value="SUBMIT"></br>
</tr>
</table>
</center>
</form>
</body>
</div>
</html>
```

CHAPTER 8
TESTING

8.1. TEST

User No	Flight No	Month	Day of month	Day of week	Origin	Destination	Scheduled Departure Time	Scheduled Arrival Time	Actual Departure Time	Actual Inputs
1	1232	1	1	1	ATL	MSP	1905	2305	1945	Delayed
2	1399	1	1	1	ATL	SEA	1805	2410	1855	Delayed
3	2351	1	2	3	ATL	DTW	1305	2305	1305	Not Delayed
4	2637	2	1	3	DTW	ATL	1500	2410	1505	Not Delayed

8.2. USER ACCEPTANCE TESTING

This report shows the number of test cases that have passed and failed

User No	Flight No	Month	Day Of Month	Day Of Week	Origin	Destin -ation	Scheduled Departure Time	Scheduled Arrival Time	Actual Departure Time	Actual Output	Predict -ed Output	Correct-ne ss
1	1232	1	1	1	ATL	MSP	1905	2305	1945	Delayed	Delayed	Correct
2	1399	1	1	1	ATL	SEA	1805	2410	1855	Delayed	Delayed	Correct
3	2351	1	2	3	ATL	DTW	1305	2305	1305	Not Delayed	Not Delayed	Correct
4	2637	2	1	3	DTW	ATL	1500	2410	1505	Not Delayed	Not Delayed	Correct

CHAPTER 9
RESULTS

9.1. PERFORMANCE METRICS

Training Accuracy

MODEL EVALUATION

```
acc=accuracy_score(predicted,y_test)
```

```
acc
```

```
0.8791308284291535
```

Confusion Matrix

```
from sklearn.metrics import confusion_matrix
confusion_matrix(predicted, y_test)
```

```
array([[1825,  129],
       [ 138,  117]], dtype=int64)
```

Classification Model

```
from sklearn.metrics import classification_report
print(classification_report(predicted, y_test, labels=[1, 2, 3]))
```

```
              precision    recall  f1-score   support

           1       0.48      0.46      0.47       255
           2       0.00      0.00      0.00         0
           3       0.00      0.00      0.00         0

   micro avg       0.48      0.46      0.47       255
   macro avg       0.16      0.15      0.16       255
weighted avg       0.48      0.46      0.47       255
```

CHAPTER 10
ADVANTAGES AND DISADVANTAGES

Advantages

- Customer satisfaction.
- The available flights are easily identified.
- In case the flight is delayed, prior information will be sent.
- The current status of the flight can be tracked.

Disadvantages

- Wrong prediction due to noise of input data.
- There will be extra expenses for the agencies,passengers and airport if the prediction is wrong.
- Passengers with medical emergencies gets affected.

CHAPTER 11
CONCLUSION

In this project, we use flight data, weather, and demand data to predict flight departure delay. In the end, our model correctly predicts the delayed and non-delayed flights correctly. As a result, there can be additional features related to the causes of flight delay that are not yet discovered using our existing data sources.

CHAPTER 12
FUTURE SCOPE

Further work we would like to improve our models with more datasets. The most interesting step would be to integrate such a model into a flight booking tool, to provide the delay prediction to future passengers, even this would require a strong confidence in the information provided, considering the possible impact in terms of reservations. These models can be used and applied in real world scenarios to make improvisation in airline industries.
Further analysis can be done by identifying the airline company in which the delays are occuring the most. With the new advancement in the field of deep learning, we can use Neural Networks algorithm on the flight and weather data. Neural Network works on the pattern matching methodology. It is divided into three basic parts for data modeling that includes feed forward networks, feedback networks, and self- organization network. Feed-forward and feedback networks are generally used in the areas of prediction, pattern recognition, associative memory, and optimization calculation, whereas self-organization networks are generally used in cluster analysis
.

CHAPTER 13
APPENDIX

13.1 Source codes

13.1.2 Exploratory Data Analysis

```
#!/usr/bin/env python
# coding: utf-8

# **Importing all the libraries**

# In[1]:

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
get_ipython().run_line_magic('matplotlib', 'inline')
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
import sklearn.metrics as metrics

# **Importing the dataset**

# In[2]:

data=pd.read_csv("flightdata.csv")

# In[3]:

data.head()

# In[4]:

data.info()
```

```
# In[5]:

data=data.drop('Unnamed: 25',axis=1)

# In[6]:

data.info()

# In[7]:

data.describe()

# In[ ]:

# **Handling Missing Values**

# In[8]:

data=data.dropna()

# In[9]:

data.info()

# **Analysis**

# In[10]:

plt.scatter(data.index,data['ARR_TIME'])
plt.ylabel('Arrival Time')
plt.title('Distribution of the Arrival Time')

# In[11]:

plt.hist(data['FL_NUM'])

# In[12]:

columns=list(data.columns)
```

```
# In[13]:

sns.scatterplot(x='ARR_DELAY',y='ARR_DEL15',data=data)

# In[14]:

sns.catplot(x='ARR_DELAY',y='ARR_DEL15',data=data,kind='bar')

# In[15]:

data['ARR_DEL15'].nunique()

# In[16]:

# In[17]:

data.describe()

# **Dropping off unnecessary columns**

# In[18]:

data.corr()['ARR_DEL15']

# In[19]:

sns.heatmap(data.corr())

# In[20]:

new_data=data.drop(['ORIGIN_AIRPORT_ID','DEST_AIRPORT_ID','FL_NUM','YEAR','
CANCELLED','DIVERTED','DISTANCE','DAY_OF_MONTH','QUARTER','MONTH','DAY
_OF_WEEK','UNIQUE_CARRIER','TAIL_NUM'],axis=1)

# In[21]:

new_data.head()

# **Label Encoding**

# In[22]:
```

```
cities=new_data['ORIGIN'].unique()

# In[23]:

cities

# In[24]:

new_data['DEST'].unique()

# In[25]:

city_map={cities[i]:i for i in range(0,len(cities))}

# In[26]:

city_map

# In[27]:

def encode(c):
  return city_map[c]

# In[28]:

new_data['ORIGIN']=new_data['ORIGIN'].apply(encode)

# In[29]:

new_data['DEST']=new_data['DEST'].apply(encode)

# In[30]:

new_data.head()

# In[31]:

new_data.corr()['ARR_DEL15']
```

```
# In[32]:

#data=data.drop('Unnamed: 25',axis=1)
data.isnull().sum()

# In[33]:

data=data[["FL_NUM","MONTH","DAY_OF_MONTH","DAY_OF_WEEK","ORIGIN","DE
ST","CRS_ARR_TIME","DEP_DEL15","ARR_DEL15"]]
data.isnull().sum()

#

# In[34]:

data=data.fillna({'ARR_DEL15': 1})
data=data.fillna({'DEP_DEL15': 0})
data.iloc[177:185]

# In[35]:

import math
for index, row in data.iterrows():
  data.loc[index,'CRS_ARR_TIME'] = math.floor(row['CRS_ARR_TIME'] / 100)
data.head()

# In[36]:

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
data['DEST'] = le.fit_transform(data['DEST'])
data['ORIGIN'] = le.fit_transform(data['ORIGIN'])

# In[37]:

data.head()

# In[38]:
```

```
from sklearn.preprocessing import OneHotEncoder
oh = OneHotEncoder()
z=oh.fit_transform(data['ORIGIN'].values.reshape(-1,1)).toarray()
t=oh.fit_transform(data['DEST'].values.reshape(-1,1)).toarray()

# In[ ]:

# In[ ]:

# In[39]:

data=pd.get_dummies(data,columns=['ORIGIN','DEST'])

# In[40]:

data['ARR_DEL15'].value_counts()

# In[41]:

data.tail()

# **Split the data into dependent and independent variables**
#
# In[42]:

x=data[[i for i in data.columns if i!='ARR_DEL15']].values
y=data[[i for i in data.columns if i=='ARR_DEL15']].values

# In[43]:

x.shape

# In[44]:

y.shape

# In[ ]:
```

CHAPTER 13
APPENDIX

13.1. SOURCE CODE

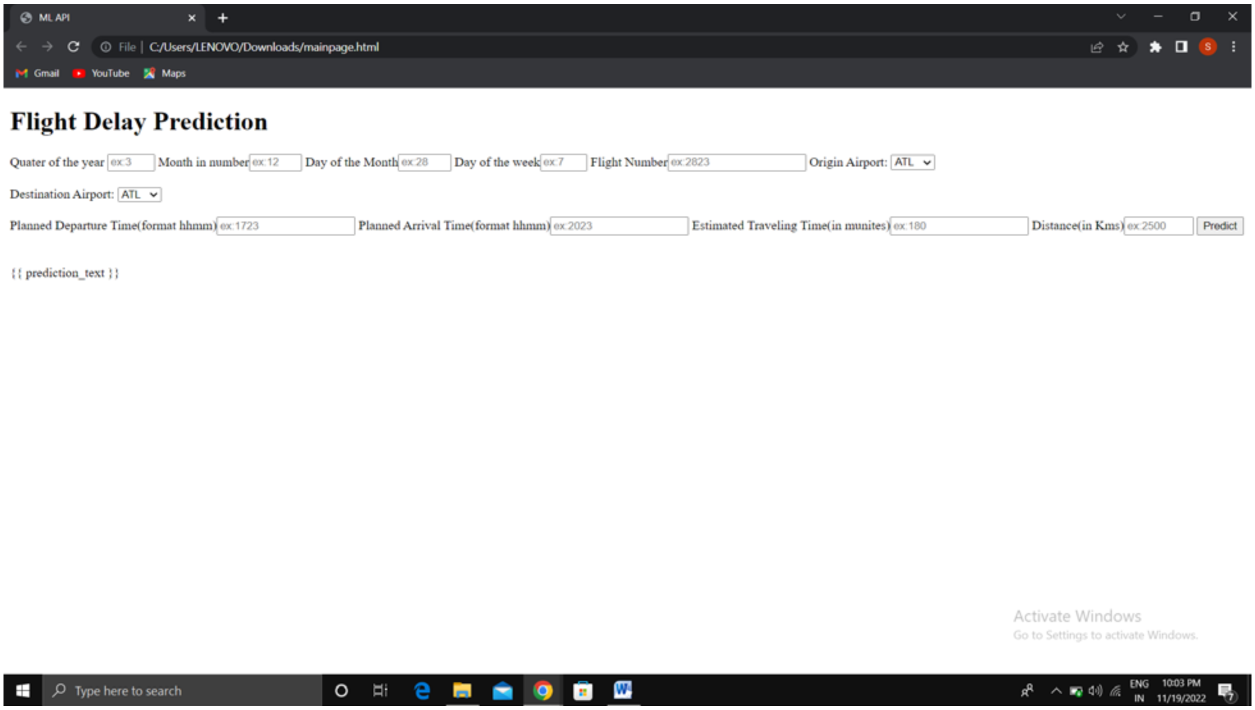
13.1.1. Train the ML Model

```
# # SPRINT-2

# **TRAIN-TEST-SPLIT**

# In[45]:

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)

# In[46]:

x_test.shape

# In[47]:

x_train.shape

# In[48]:

y_test.shape

# In[49]:

y_train.shape

# **Scaling**

# In[50]:

sc = StandardScaler()

# In[51]:

x_train=sc.fit_transform(x_train)
```

```
# In[52]:

x_test=sc.fit_transform(x_test)

# **Model Building**

# In[53]:

classifier = DecisionTreeClassifier(random_state=0)

# In[54]:

classifier.fit(x_train,y_train)

# In[55]:

predicted = classifier.predict(x_test)

# In[56]:

predicted

# In[57]:

y_test

# **MODEL EVALUATION**

# In[58]:

acc=accuracy_score(predicted,y_test)

# In[59]:

acc

# In[ ]:


# In[60]:
```

```
data[data['ARR_DEL15']>0].iloc[33].values

# In[61]:

sample=[[1.187e+03, 1.000e+00, 1.500e+01, 5.000e+00, 1.900e+01, 1.000e+00,
0.000e+00, 0.000e+00, 0.000e+00, 1.000e+00, 0.000e+00,
       0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 1.000e+00]]

# In[62]:

classifier.predict(sample)

# **SAVING THE MODEL**

# In[63]:

pickle.dump(classifier,open('flight_new.pk1','wb'))

# In[64]:

from sklearn.metrics import confusion_matrix
confusion_matrix(predicted, y_test)

# In[66]:

from sklearn.metrics import classification_report
print(classification_report(predicted, y_test, labels=[1, 2]))

# In[ ]:
```

13.1.2. Main page – HTML Code



13.1.3 Prediction Page - HTML Code

```html
<!doctype html>
<html>
  <body background="C:\Users\Public\project\templates\flight_2.jpg">
        <center>
    <h1><strong>Thanks for asking</strong></h1>
    <h2>{{data}}</h2>
    <a href='/'>Go back to home page</a>
        </center>
  </body>
</html>
```

13.1.3. Flask Application

```python
import numpy as np
import os
from flask import Flask, request, jsonify, render_template,json
import pickle
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM
Cloud account.
API_KEY = "okbr7ARnOQjyplTOyvNFC2QVkCF6q7afpci065Hucby8"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
app = Flask(__name__)
model = pickle.load(open('rfmodel.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('mainpage.html')

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

    sm=[6,7,8]
    wt=[9,10,11]
    sp=[12,1,2,3]
    fl=[4,5]
    farr= [int(x) for x in request.form.values()]
    if farr[1] in sm:
        farr.append(0)
    elif farr[1] in wt:
```

```
        farr.append(1)
    elif farr[1] in sp:
        farr.append(2)
    else:
        farr.append(3)
    final_features=[int(x) for x in farr]
    print(final_features)
    payload_scoring = {"input_data": [{"fields":
[['QUARTER','MONTH','DAY_OF_MONTH','DAY_OF_WEEK','FL_NUM','ORIGIN','DEST',
'CRS_DEP_TIME.1','CRS_ARR_TIME.1','CRS_ELAPSED_TIME','DISTANCE','SEASON']],
"values": [final_features]}]}

    response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/b54f9857-1352-432a-8ab1-
144ebda20501/predictions?version=2022-11-08',
json=payload_scoring,headers={'Authorization': 'Bearer ' + mltoken})
    print("Scoring response")
    pred=response_scoring.json()
    print(pred)
    prediction=pred['predictions'][0]['values'][0][0]
    prediction = model.predict([final_features])
    print(prediction)

    output =prediction

    if output==0:
        return render_template('mainpage.html', prediction_text='No delay will happen
{}'.format(output))
    elif output==1:
        return render_template('mainpage.html', prediction_text='There is a chance to
departure delay will happen {}'.format(output))
    elif output==2:
        return render_template('mainpage.html', prediction_text='here is a chance to both
departure and arrival delay will happen {}'.format(output))
    elif output==3:
        return render_template('mainpage.html', prediction_text='here is a chance to flight  will
diverted {}'.format(output))
    elif output==4:
        return render_template('mainpage.html', prediction_text='here is a chance to cancel
 the flight {}'.format(output))
    else:
        return render_template('mainpage.html', prediction_text='output {}'.format(output))
'''@app.route('/predict_api',methods=['POST'])
def predict_api():

    For direct API calls trought request

    data = request.get_json(force=True)
    prediction = model.predict([np.array(list(data.values()))])
```

```
        output = prediction[0]
        return jsonify(output)'''

    if __name__ == "__main__":
        os.environ.setdefault('FLASK_ENV', 'development')
        app.run(debug=False)
```

13.2. GITHUB & PROJECT DEMO LINK

Github link

https://github.com/IBM-EPBL/IBM-Project-22576-1659854171

Project Demo link

https://youtu.be/E2x2s-kboJU