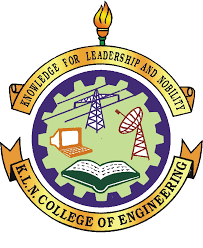
**IBM NALAIYA THIRAN**

**PROJECT REPORT**

**K.L.N COLLEGE OF ENGINEERING, POTTAPALAYAM**

(An Autonomous institution, affiliated to Anna university, Chennai)



**Problem statement :** Flight Delay Predication

**Team ID :** PNT2022TMID11613

**Team Leader :** P.Subha Sree

**Team members :** 1. A.Sanjana Devi

2. S.Susmitha

3. A.K.Sona

**Faculty mentor :** G.H.Ram Ganesh

**Evaluator :** Dr.J.S. Kanchana

**Industry mentor :** Prof Swetha

**INDEX**

|  |  |  |
| --- | --- | --- |
| **SNo** | **Topic** | **Page number** |
| **1** | **Introduction** | 2 |
| 1.1 | Project overview | 2 |
| 1.2 | Purpose | 2 |
| **2** | **Literature survey** | 3 |
| 2.1 | Existing problem | 3 |
| 2.2 | References | 3 |
| 2.3 | Problem statement definition | 6 |
| **3** | **Ideation and proposed solution** | 7 |
| 3.1 | Empathy map canvas | 7 |
| 3.2 | Ideation and Brainstorming | 7 |
| 3.3 | Proposed solution | 7 |
| 3.4 | Problem solution fit | 9 |
| **4** | **Requirement analysis** | 9 |
| 4.1 | Functional requirements | 9 |
| 4.2 | Non-functional requirements | 10 |
| **5** | **Project design** | 10 |
| 5.1 | Data flow diagrams | 10 |
| 5.2 | Solution and technical architecture | 11 |
| 5.3 | User stories | 11 |
| **6** | **Project planning and scheduling** | 12 |
| 6.1 | Sprint planning and estimation | 12 |
| 6.2 | Sprint delivery schedule | 12 |
| 6.3 | Reports from JIRA | 12 |
| **7** | **Coding and solutioning** | 13 |
| 7.1 | Sprint-1 | 13 |
| 7.2 | Sprint-2 | 16 |
| 7.3 | Sprint-3 | 16 |
| 7.4 | Sprint-4 | 22 |
| **8** | **Testing** | 25 |
| 8.1 | Test case | 25 |
| 8.2 | User acceptance testing | 25 |
| **9** | **Result** | 26 |
| 9.1 | Performance metrics | 26 |
| **10** | **Advantages and Disadvantages** | 26 |
| **11** | **Conclusion** | 27 |
| **12** | **Future scope** | 27 |
| **13** | **Appendix** | 27 |

**1.Introduction:**

over the last twenty years, air travel has been increasingly preferred among travelers, mainly because of its speed and in some cases comfort. This has led to phenomenal growth in air traffic and on the ground. An increase in air traffic growth has also resulted in massive levels of aircraft delays on the ground and in the air. These delays are responsible for large economic and environmental losses. The main objective of the model is to predict flight delays accurately in order to optimize flight operations and minimize delays.

Using a machine learning model, we can predict flight arrival delays. The input to our algorithm is rows of feature vector like departure date, departure delay, distance between the two airports, scheduled arrival time etc. We then use decision tree classifier to predict if the flight arrival will be delayed or not. A flight is considered to be delayed when difference between scheduled and actual arrival times is greater than 15 minutes. Furthermore, we compare decision tree classifier with logistic regression and a simple neural network for various figures of merit.

* 1. **Project overview:**

Artificial Intelligence is a data hunger technology, it depends heavily on data, without data, it is impossible for a machine to learn. It is the most crucial aspect that makes algorithm training possible. In Convolutional Neural Networks, as it deals with images, we need training and testing data set. It is the actual data set used to train the model for performing various actions. In this activity let's focus on gathering the dataset.

Then, as we will be obtaining numerical data as the output, we employ the regression approach. We also use the Random Forest Regression method.

We used the Random Forest Regression method because

* It provides an effective way of handling missing data.
* It can produce a reasonable prediction without hyper-parameter tuning.
* It solves the issue of overfitting in decision trees.
* In every random forest tree, a subset of features is selected randomly at the node’s splitting point.

We then uploaded our machine learning model to the IBM cloud, constructed a scoring API with Flask, and built the front end using HTML, CSS, and Bootstrap..

* 1. **Purpose**

In this milestone, we will be preprocessing the dataset that is collected. Preprocessing includes:

1. Processing the dataset.
2. Handling the null values.
3. Handling the categorical values if any.
4. Normalize the data if required.
5. Identify the dependent and independent variables.
6. Split the dataset into train and test sets.

**2.Literature survey**

In a literary survey, we analyse critically, and concisely earlier research and literature related to a particular research problem and utilize them for their own research purposes. It helps us in understanding the significance of new research and its connections to earlier work.

**2.1 Existing problem**

There are various websites that provide these services but the method to do so is not the best, different models and systems may contribute to predicting the flight delay.

**2.2 References**

**In the year of 2021, Rajkumar Goel Institute of Technology**

After applying both the models for predicting whether a flight should be delayed, as well as how much one would expect a flight should be delayed, they found the following factors to be important: week, month, airline carrier reference, planned elapsed time (in airtime), distance between two departure and destinations, flight planned departure time, departure airport code, and taxi-in and taxi-out4 time. By applying there model, on the data collected, one could be able to predict whether a flight might be delayed, and more importantly, how long delayed time she/he would expect. However, there is some limitation in our model, first, our model only included one-year data due to our computation capability, as more years of data included, the prediction could be easier. In addition, some other related information such as airplane type, e.g., detailed weather data specific to airport were not included. Therefore, researchers could try to collect more related data and deploy better computational powers to build a better model. This paper presented a methodology for predicting aggregate flight departure delays in airports by exploring supervised learning methods. This way, they may be able to predict the delays of a new flight, without needing several months of data to build a prediction model. Another step forward would be to generalize the model to flights of the entire world, or at least to exploit more data sources, to build more complete predictions. Finally, 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

**In the year of 2020, Nanjing University of Aeronautics and Astronautics**

This paper presented a methodology for predicting aggregate flight departure delays in airports by exploring supervised learning methods. The proposed new model was enabled by four types of airport-related aggregate characteristics, including time characteristics, flight plan characteristics, delay characteristics and local weather characteristics. The results obtained show that for a 1-h forecast horizon, LightGBM model provides the best result, giving 0.8653 accuracy with 6.58 min mean absolute error, which is 1.83 min less than previous research. Analysis also found that accumulated number of departure demand in the prediction period is the dominating factor in the LightGBM model. The number of planned departures in the prediction period and the expected delay time of departures before the prediction period are two other obvious factors, while the expected delay time of arrivals before the prediction period, hour of the day, and the number of planned departures before the prediction period are three following characteristics. Of special note is that the model performances with local weather characteristics are not as good as those without meteorological data. Two potential reason are that the cancelled and returned flight records caused by local weather characteristics can hardly be translated into specific delay time in departing airports, and the local flight delays are often caused by weather conditions en-route or in the airports of the previous flights, not just in local airports. The prediction model presented in this paper yields a better understanding of delays interactions between time, flight plan and previous delay. Since we predict the flight departure delays from the airport aspect, the model could be used for reminding airport managers, air traffic controllers and passengers to deal with the impending congestion in airports. Future works include exploring some other explanatory characteristics such as national weather, city-pair, and network states, etc., and extending forecast horizon with more accuracy results.

**In the year of 2019, H. Khaksar and A. Sheikholeslami,Sharif University of Technology**

FDP methods, namely decision tree, cluster, Bayesian, random forest, and hybrid classication, were proposed in this research. These approaches were examined on the basis of real datasets on US and Iranian ight networks. The results indicated that the hybrid approach exhibited a performance superior to those of the other methods and was therefore adopted as the FDP model. Parameters such as eet age and aircraft type exert strong eects on ight delays in the Iranian network, whereas weather conditions strongly in uence ight delays in the US network. The accuracy levels of the hybrid approach were 71.39% and 76.44% in predicting delay occurrence and 70.16% and 75.93% in predicting delay magnitude in the US and Iranian networks, respectively. These results may be of interest to airlines that want to implement measures for preventing delay propagation, especially those based in developing countries, such as Iran. For the future studies, researchers can implement other exciting data mining methods and compare the results. The proposed combined method of delay anticipation and its results can also be further explored in other studies. For example, combing the hybrid method with robust ight scheduling shows potential as an interesting research direction.

**In the year of 2017 , Anish M. Kalliguddi\* , Aera K. Leboulluec,College of Engineering, University of Texas, United States**

This study is devoted to develop a predictive model to forecast flight delays. Data spanning for over 1million observations including US domestic flights variables was used. Models based on multiple linear regression, decision trees and random forest algorithms are created and tested in R-studio software concluding that Random forest model outperforms other two models based on the evaluation criteria. In addition, the study also sheds light on the significant factors responsible for departure delay. The splitting variables or the significant variable are found to be late aircraft delay, Carrier delay, weather delay and NAS delay which have the most effect on on-time flight departure. The predictive model was developed for a period of 365 days for all US domestic airports. It is seen that the longer forecast horizon helps in better prediction accuracy with minimum prediction error for random forest. These models can be used to improved traffic management decision in comparison with the current applications of Enhanced Traffic Management Systems (ETMS). Although the model gives very good prediction accuracy, more variables can be considered to develop a predictive model. For example, Weather data can be extracted and used to better develop a predictive model for flight delay. The future scope of this study involves various approaches that can be used to analyze the data. Principal component analysis or transformation can be done to uncover hidden relations between variables. In addition, since the data is not exactly linear, artificial neural networks or Support vector machines can be used to analyze the effect of various variables on flight delay.

**In the year of 2016, Conference Organized by Missouri University of Science and Technology**

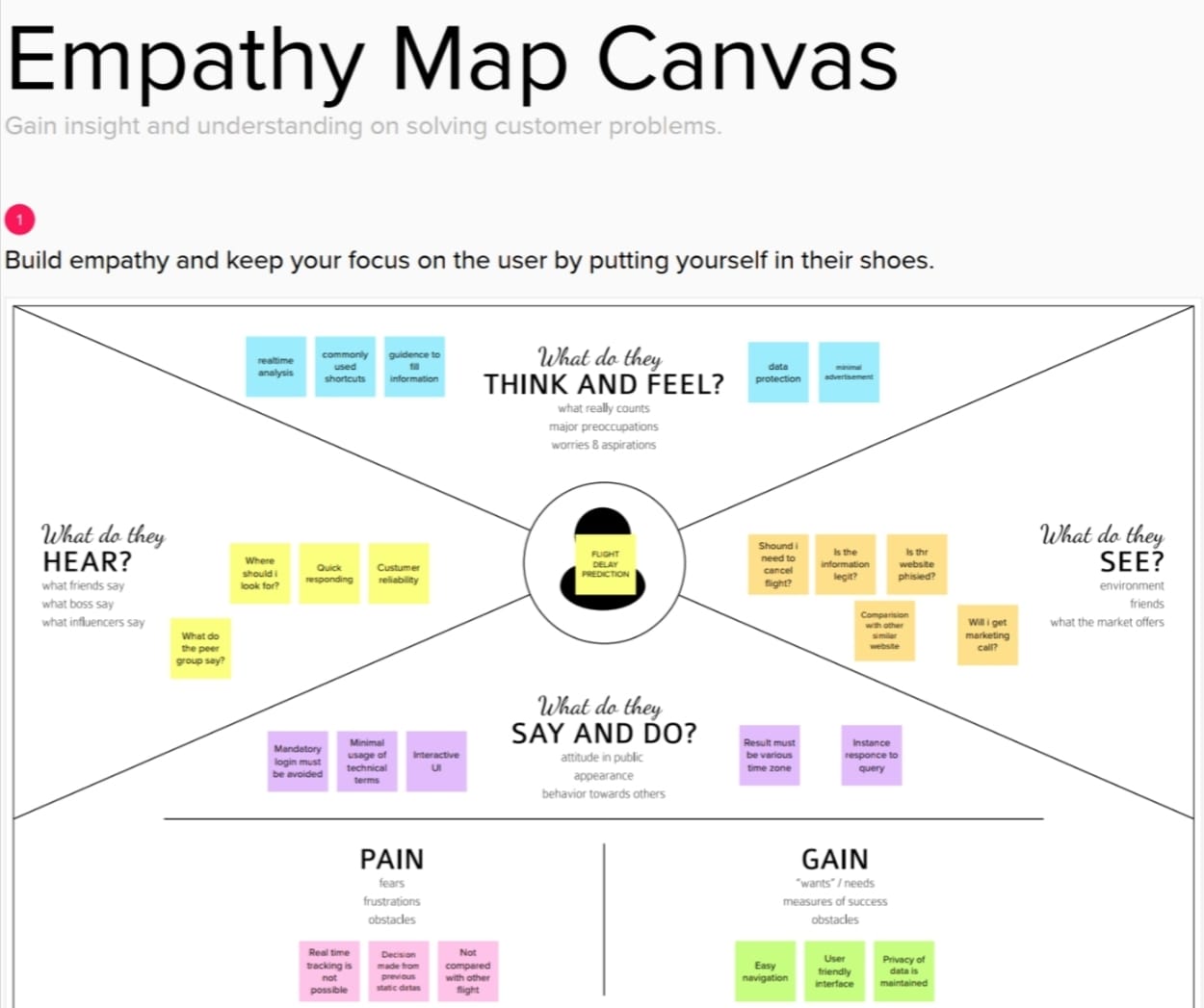
A new ANN structure (DMP-ANN) is introduced which is suitable for prediction of defects such as delays in operations. This structure is appropriate for problems with nominal variables, where traditional ANN models have difficulties. For example, the types of cargo or ID number of origin of departure are variables that cannot be directly used in a traditional ANN. The input layer in proposed DMP-ANN consists of several sublayers in which one or more neurons are active (output=1) and others where they are inactive (output=0). Hence, the learning process involves updating the weights of active neurons. The introduced ANN model is applied to a system of airport traffic control where the arriving flights are prioritized for landing based on the expected possible delays. The results suggest that the proposed method can be effective for specific problems that include many nominal variables, such as the transportation problem. One of the limitations of this study that needs to be addressed in our future work is the complexity of the proposed method (as the number of variables increases the number of connections also significantly increase). Furthermore, we will consider the integration of the proposed method with fuzzy logic to expand the real-world applications of the proposed method.

**2.3 Problem Statement Definition**

****

**3. IDEATION & PROPOSED SOLUTION**

**3.1 Empathy Map Canvas**

****

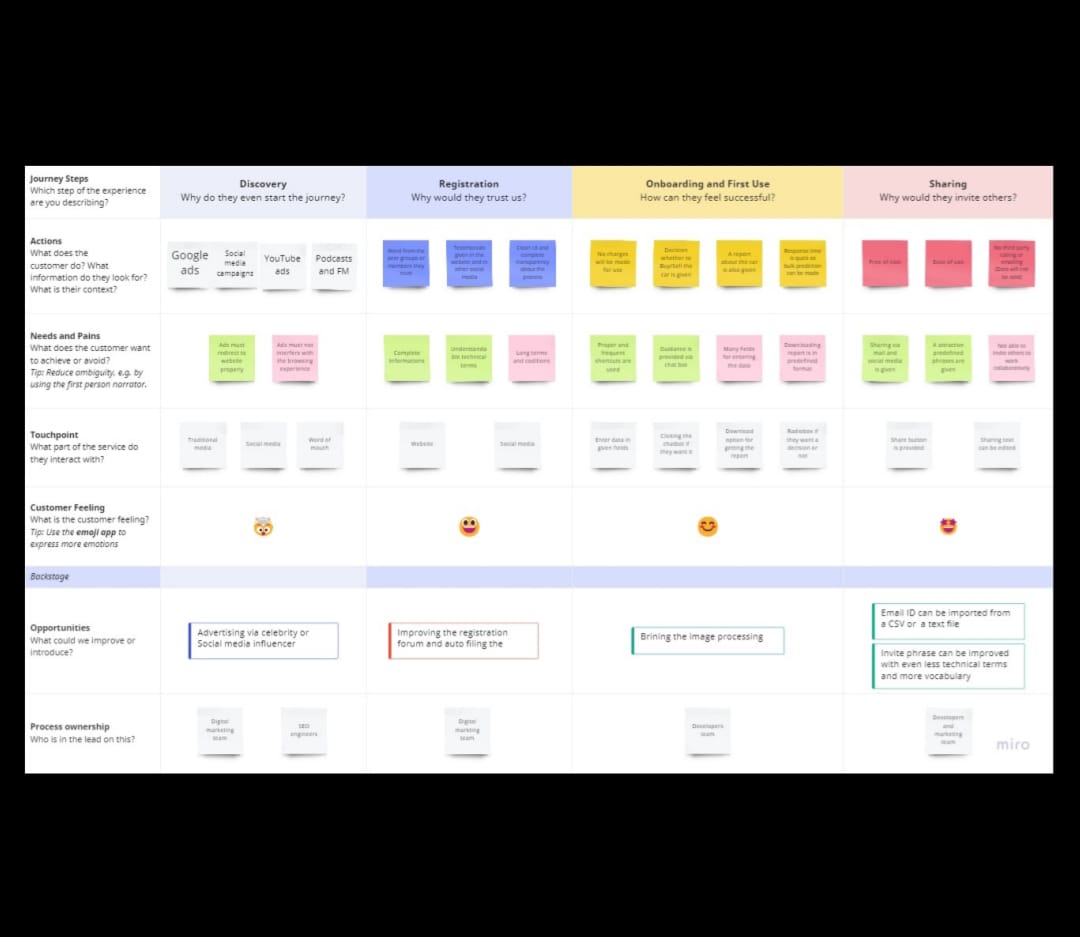
**3.2 Ideation & Brainstorming**



**3.3 Proposed Solution**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Description** |
|  | Problem Statement (Problem to be solved) | As there are a lot of factors via which the flight are getting delayed, these delay can cost the customer's time and money a lot. The customer need to find a place to stay, as this was unplanned the rates of booking for the hotels may cost more, or he may also want to find other models or transportation ,but compared to flights no other transport's are faster. Imagine the same scenario for 100 to 300 people, that will be a nightmare , therefore we are developing a model that can predict the arrival of the flights. |
|  | Idea / Solution description | Using machine learning model ,we can predict flight arrival delays .The input to our algorithm is rows of features vector like departure date, departure delay, distance between the two airports , scheduled arrival time etc..We then use decision tree classifier to predict if the flight arrival will be delayed or not. A flight is considered to be delayed when difference between scheduled and actual arrival times is greater than 15 minutes . Furthermore , we compare decision tree classifier with logistic regression and a simple neural network for various figures of merit . |
|  | Novelty / Uniqueness | We will be implementing an classification algorithm , whether the user can wait for the flight or cancel it go for the next one. The inputs will the , ETA of the user, budget he can allocate and some other factors on demand . |
|  | Social Impact / Customer Satisfaction | Time management  Trip management can be easy  Cost effective  Social impact |
|  | Business Model (Revenue Model) | Generate revenue via good ads able to get a commission if we refer the details of the customer to other company on demand of the customer |
|  | Scalability of the Solution | Can even add the flight booking system also or refer to the some authorised agents . Using the IBM cloud the systems can withstand the user loads, so that addition of any new features will not be that difficult . |

**3.4 Problem Solution fit**

****

**4. REQUIREMENT ANALYSIS**

**4.1 Functional requirement**

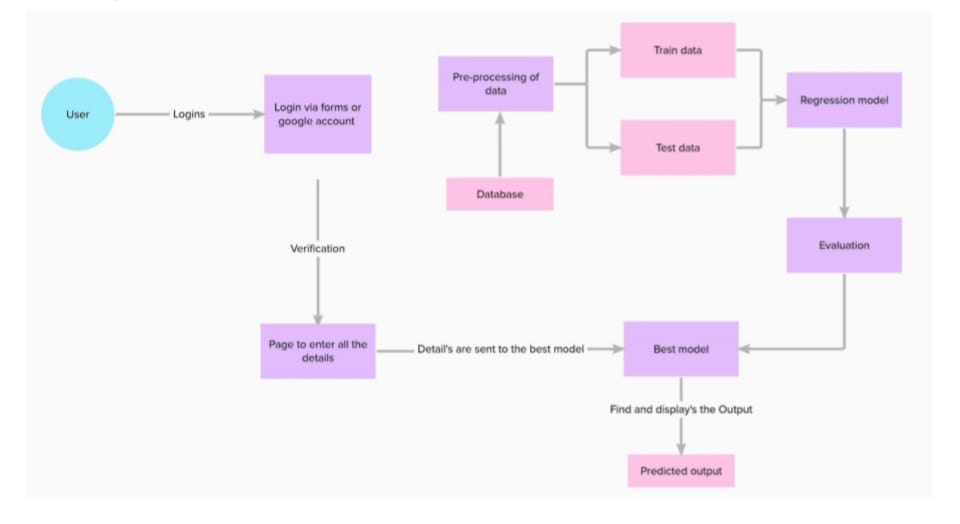
|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | Registration through Form  Registration through Gmail |
| FR-2 | User Confirmation | Confirmation via Email |
| FR-3 | User flight details | * flight name * departure time * departure delay * schedule arrival time * actual ETA * distance |
| FR-4 | Classification to predict flight delay whether to take off or not? | Using the multiple existing features, loading it into an algorithm to find whether the flight delay is there are not. |

**4.2 Non-Functional requirements**

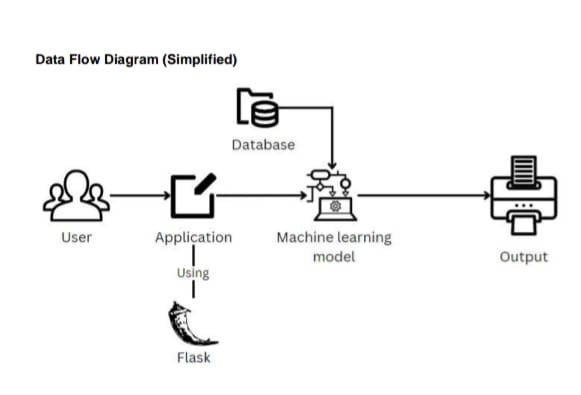
|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | User interface (UI) is very much user-friendly and easy to use for a seamless flow of the process |
| NFR-2 | **Security** | Must guard against SQL injection and other types of attacks that could lead to data theft. |
| NFR-3 | **Reliability** | A trustworthy source where user information is encrypted and protected from attackers |
| NFR-4 | **Performance** | The user interface needs to be fast and able to handle a significant quantity of network traffic. |
| NFR-5 | **Availability** | The website must not crash because of network load and must always be accessible to users. |
| NFR-6 | **Scalability** | The website must be able to withstand fluctuations in network traffic and resource usage and be error-free. |

**5. PROJECT DESIGN**

**5.1 Data Flow Diagrams**



**5.2 Solution & Technical Architecture**

****

**5.3 user stories**

| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint-1 | Home page | USN-1 | As a user, I can view my home page of the application. | 20 | Low | Subha sree,Susmitha |
| Sprint-2 | Flight Delay Display | USN-2 | As a user, I will be able to enter the data in the application | 20 | Medium | Sanjana Devi, Sona |
| Sprint-3 | Data entry | USN-3 | As a user, there will be fields in which I need to give my data | 20 | High | Subha sree, Sanjana Devi |
| Sprint-4 | Flight Delay Prediction | USN-4 | As a user, I will expect my predicted value to be displayed | 20 | High | Sona,Susmitha |

**6. PROJECT PLANNING & SCHEDULING**

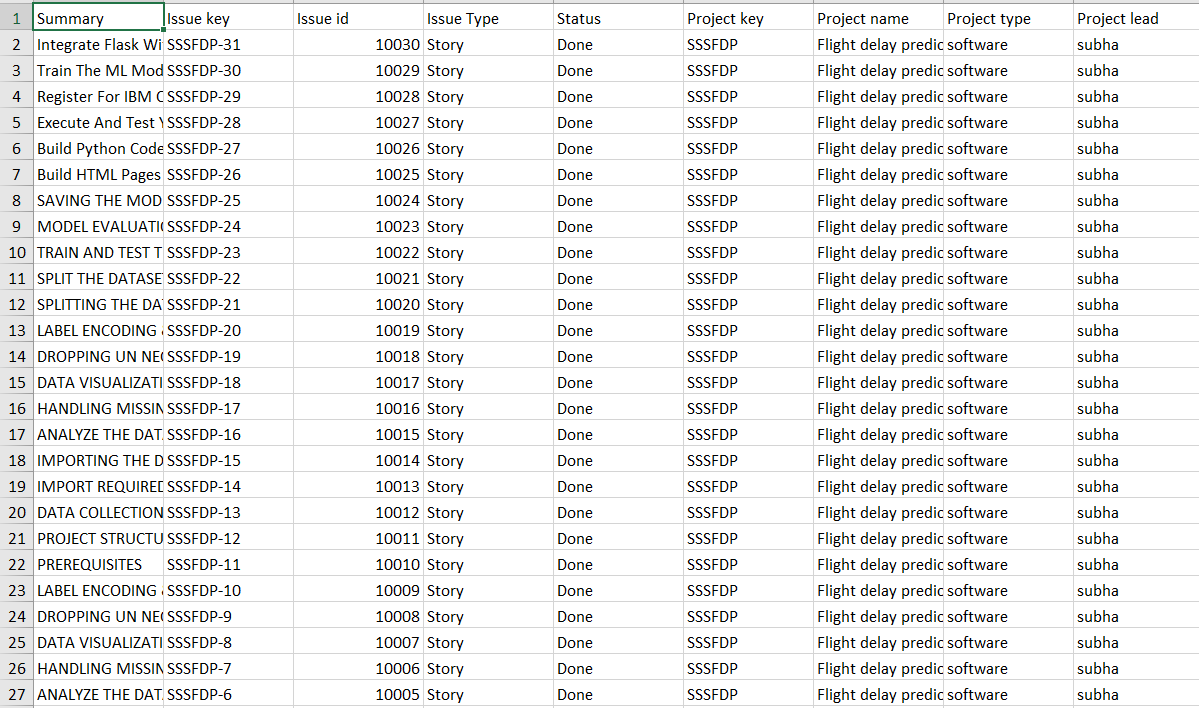
**6.1 Sprint Planning & Estimation**

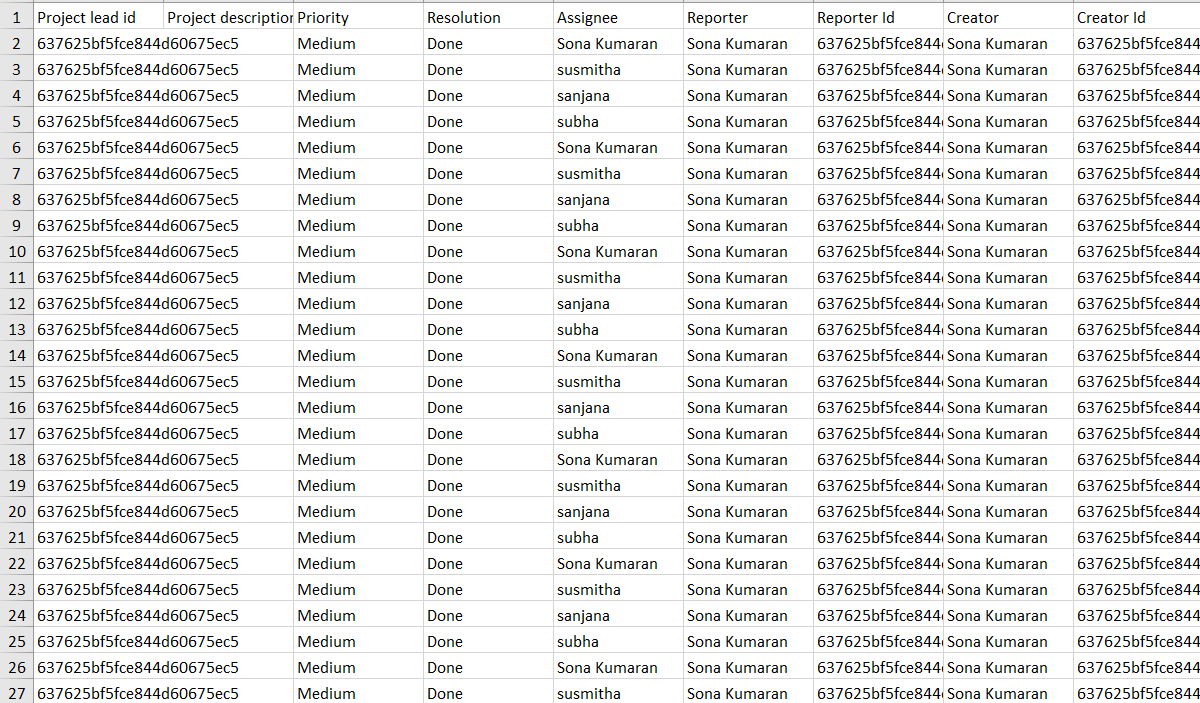
| **Sprint** | **Total Story Points** | **Duration** | **Sprint Start Date** | **Sprint End Date (Planned)** | **Story Points Completed (as on Planned End Date)** | **Sprint Release Date (Actual)** |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint-1 | 20 | 6 Days | 26 Oct 2022 | 30 Oct 2022 | 20 | 31 Oct 2022 |
| Sprint-2 | 20 | 6 Days | 1 Nov 2022 | 06 Nov 2022 | 20 | 09 Nov 2022 |
| Sprint-3 | 20 | 6 Days | 09 Nov 2022 | 14 Nov 2022 | 20 | 14 Nov 2022 |
| Sprint-4 | 20 | 6 Days | 16 Nov 2022 | 19 Nov 2022 | 20 | 19 Nov 2022 |

**6.2 Sprint Delivery Schedule**

Pre-requites

* Imported the necessary libraries
* Reading the data
* Cleaning the data
* Exporting the pre-processed data
* Creating the .npy files
* Splitting the train and testing data
* Save the ML model
* Create HTML pages
* Integrate with flask
* Deploy the ML model
* Develop the scoring endpoint
  1. **Reports from JIRA**

****



**7. CODING & SOLUTIONING**

**7.1 Sprint-1**

import sys

import numpy as np #Linear Algebra

import pandas as pd

import seaborn as sns

import pickle

import matplotlib.pyplot as plt

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

dataset = pd.read\_csv("flightdata.csv")

dataset.drop('Unnamed: 25', axis = 1, inplace = True)

dataset.info()

dataset.describe()

dataset.isnull().sum()

dataset.dropna(subset=['DEP\_TIME','ARR\_DELAY'], inplace = True)

sns.heatmap(dataset.corr())

fig, ax = plt.subplots(1, 2, figsize = (10,10))

ax[0].pie(dataset['DEP\_DEL15'].value\_counts(), labels = ['On Time', 'Delayed'], autopct = '%1.2f%%',startangle = 90, explode = (0,0.1));

ax[0].title.set\_text("Ratio of Delayed Departure Flights");

ax[1].pie(dataset['ARR\_DEL15'].value\_counts(), labels = ['On Time','Delayed'], autopct = '%1.2f%%',startangle = 90, explode = (0,0.1));

ax[1].title.set\_text("Ratio of Delayed Arrival Flights");

new\_df = pd.get\_dummies(dataset, columns = ['ORIGIN','DEST'])

new\_df.head()

X = new\_df[['MONTH','DAY\_OF\_MONTH','DAY\_OF\_WEEK','ORIGIN\_ATL','ORIGIN\_DTW','ORIGIN\_JFK','ORIGIN\_MSP','ORIGIN\_SEA','DEST\_ATL','DEST\_DTW','DEST\_JFK','DEST\_MSP','DEST\_SEA','CRS\_DEP\_TIME','DEP\_TIME','DEP\_DEL15','CRS\_ARR\_TIME']]

y = new\_df['ARR\_DEL15']

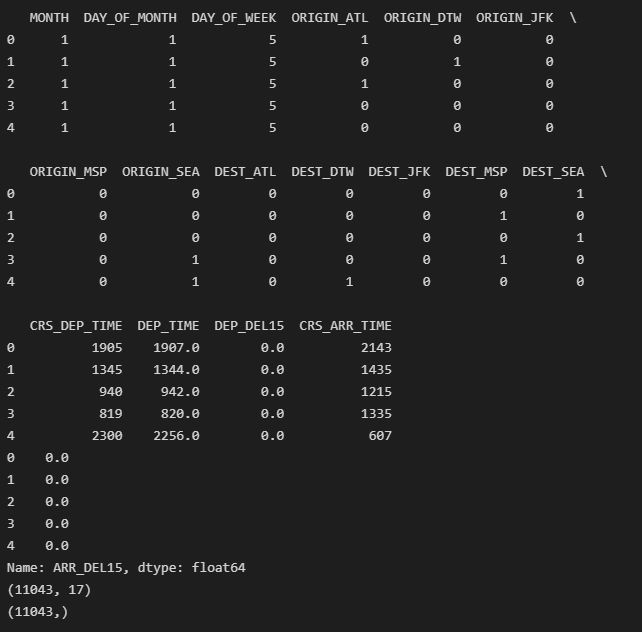
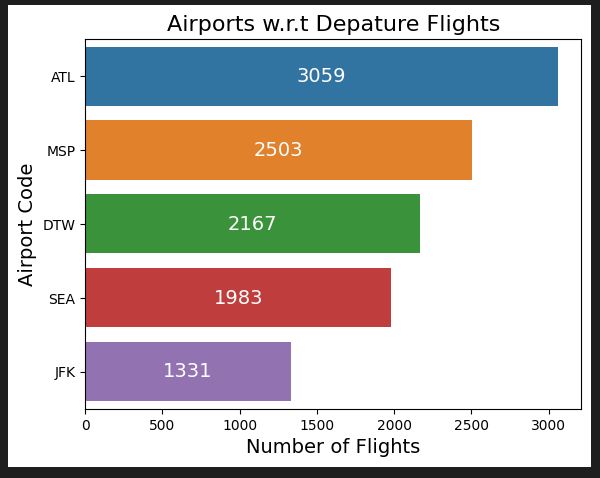
print(X.head())

print(y.head())

print(X.shape)

print(y.shape)

Output:



**7.2 Sprint-2**

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.30)

from sklearn.tree import DecisionTreeClassifier

clf = DecisionTreeClassifier(max\_depth = 4, min\_samples\_split = 4, random\_state = 25)

clf.fit(X\_train, y\_train)

pred = clf.predict(X\_test)

from sklearn.metrics import accuracy\_score

print(accuracy\_score(y\_test, pred))

import pickle

pickle.dump(clf, open('flightdelay.pkl','wb'))

**output**

****

**7.3 Sprint-3**

# importing the necessary dependencies

from flask import Flask, request,render\_template

import numpy as np

import pandas as pd

import pickle

import os

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

app = Flask(\_name\_)

@app.route('/')

def home():

return render\_template("index.html")

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

def predict() :

if request.method=="POST":

name=request.form["name"]

month=request.form["month"]

dayofmonth = request.form['dayofmonth']

dayofweek = request.form['dayofweek']

origin = request.form['origin']

destination=request.form['destination']

origin1,origin2,origin3,origin4,origin5=0,0,0,0,0

if(origin=="msp"):

origin1,origin2,origin3,origin4,origin5=0,0,0,1,0

if(origin=="dtw"):

origin1,origin2,origin3,origin4,origin5=0,1,0,0,0

if(origin=="jfk"):

origin1,origin2,origin3,origin4,origin5=0,0,1,0,0

if(origin=="sea"):

origin1,origin2,origin3,origin4,origin5=0,0,0,0,1

if(origin=="alt"):

origin1,origin2,origin3,origin4,origin5=1,0,0,0,0

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 == "alt"):

destination1,destination2,destination3,destination4,destination5 = 0,0,0,1,0

dept = request.form['dept']

arrtime = request.form['arrtime']

actdept = request.form['actdept']

depti5=int(dept)-int(actdept)

total = [[month,dayofmonth,dayofweek,origin1,origin2,origin3,origin4,origin5,destination1,destination2,destination3,destination4,destination5,dept,actdept,depti5,arrtime]]

y\_pred = model.predict(total)

print(y\_pred)

if(y\_pred==[0.]):

return "The Flight will be on time"

else:

return "THE FLIGHT WILL BE DELAYED"

app.run(debug=True)

**html code:**

<!DOCTYPE html>

<html>

<style >

input[type=text], select {

width: 50%;

padding: 12px 20px;

margin: 8px 0;

display: inline-block;

border: 1px solid #ccc;

border-radius: 4px;

box-sizing: border-box;

}

input[type=submit] {

width: 50%;

background-color: #4CAF50;

color: white;

padding: 14px 20px;

margin: 8px 0;

border: none;

border-radius: 4px;

cursor: pointer;

}

input[type=submit]:hover {

background-color: #45a049;

}

div {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

}

</style>

<head>

<title>Predication of Flight Delay</title>

</head>

<body style="background-image:url('ibm.jpg');background-repeat:no-repeat;background-size:100% 100%">

<h1><b><center> Predication of Flight Delay</center></b></h1>

<form action ="http://localhost:5000/predicition" method="POST">

<label for="fname"><b>Enter The Flight Number:</b></label>

<input type="text" id="name" name="name"><br><br>

<label for="lname"><b>Month:</b></label>

<input type="text" id="month" name="month"><br><br>

<label for="fname"><b>Day Of Month:</b></label>

<input type="text" id="dayofmonth" name="dayofmonth"><br><br>

<label for="lname"><b>Day Of Week</b></label>

<input type="text" id="dayofweek" name="dayofweek"><br><br>

<label for="fname"><b>Origin:</b></label>

<input type="text" id="origin" name="origin"><br><br>

<label for="lname"><b>Destination:</b></label>

<input type="text" id="destination" name="destination"><br><br>

<label for="fname"><b>Scheduled Departure Time:</b></label>

<input type="text" id="dept" name="dept"><br><br>

<label for="lname"><b>Scheduled Arrival Time:</b></label>

<input type="text" id="arrtime" name="arrtime"><br><br>

<label for="fname"><b>Actual Departure Time:</b></label>

<input type="text" id="actdept" name="actdept"><br><br><br>

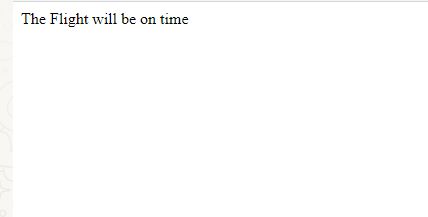
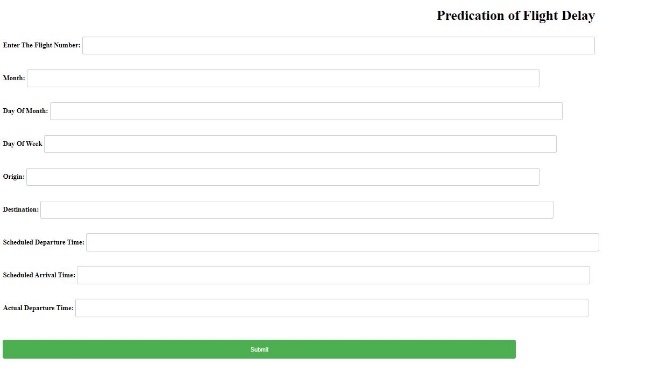
<input type="submit" value="Submit">

</form>

</body>

</html>

**Output**

****

**7.4 Sprint-4**

# importing the necessary dependencies

from flask import Flask, request,render\_template

import numpy as np

import pandas as pd

import pickle

import requests

import json

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

app = Flask(\_name\_)

@app.route('/')

def home():

return render\_template("index.html")

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

def predict() :

if request.method=="POST":

name=request.form["name"]

month=request.form["month"]

dayofmonth = request.form['dayofmonth']

dayofweek = request.form['dayofweek']

origin = request.form['origin']

destination=request.form['destination']

origin1,origin2,origin3,origin4,origin5=0,0,0,0,0

if(origin=="msp"):

origin1,origin2,origin3,origin4,origin5=0,0,0,1,0

if(origin=="dtw"):

origin1,origin2,origin3,origin4,origin5=0,1,0,0,0

if(origin=="jfk"):

origin1,origin2,origin3,origin4,origin5=0,0,1,0,0

if(origin=="sea"):

origin1,origin2,origin3,origin4,origin5=0,0,0,0,1

if(origin=="alt"):

origin1,origin2,origin3,origin4,origin5=1,0,0,0,0

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 == "alt"):

destination1,destination2,destination3,destination4,destination5 = 0,0,0,1,0

dept = request.form['dept']

arrtime = request.form['arrtime']

actdept = request.form['actdept']

depti5=int(dept)-int(actdept)

total = [[month,dayofmonth,dayofweek,origin1,origin2,origin3,origin4,origin5,destination1,destination2,destination3,destination4,destination5,dept,actdept,depti5,arrtime]]

# y\_pred = model.predict(total)

# NOTE: you must manually set API\_KEY below using information retrieved from your IBM Cloud account.

API\_KEY = "IU\_76Xzg4uhR9DbnmjkoLeMoa5ePG3QAmmPfcUBPjziH"

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}

# NOTE: manually define and pass the array(s) of values to be scored in the next line

payload\_scoring = {"input\_data": [{"field": [["f0","f1","f2","f3","f4","f5","f6","f7","f8","f9","f10","f11","f12","f13","f14","f15","f16"]], "values": total}]}

response\_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/9b151467-653e-419b-bb25-1b3a6c11228d/predictions?version=2022-11-22', json=payload\_scoring,

headers={'Authorization': 'Bearer ' + mltoken})

print("Scoring response")

print(response\_scoring.json())

y\_pred=response\_scoring.json()

print(y\_pred['predictions'][0]['values'][0][1][1])

v=y\_pred['predictions'][0]['values'][0][0]

print(type(v))

if(v==0.):

return "The Flight will be on time"

else:

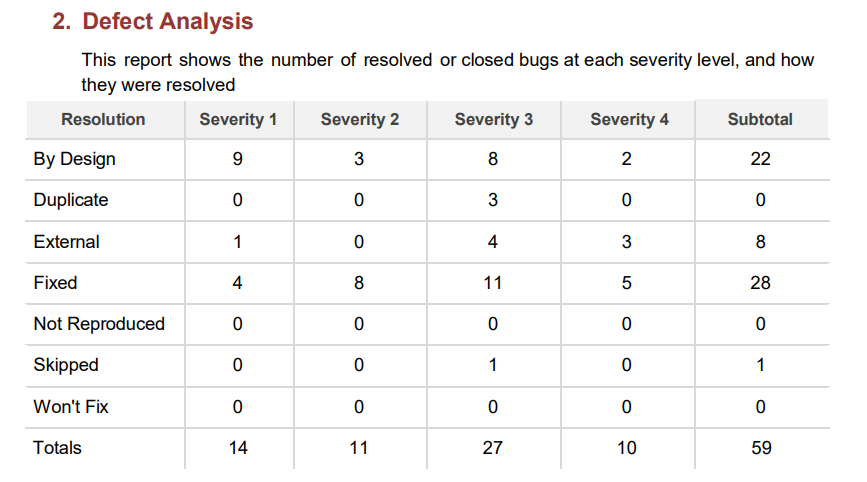
return "THE FLIGHT WILL BE DELAYED"

app.run(debug=True)

**8. TESTING**

**8.1 Test Cases**

* Page loading and proper integration with CSS and Bootstrap
* Able to enter all the data and send to backend
* Ability to capture the entered data and send it to ML model in the cloud
* To ensure the end point scoring is properly integrated
* Prediction is proper?

**8.2 User Acceptance Testing**

**Table

Description automatically generated**

**Table

Description automatically generated**

**9. RESULTS**

**9.1 Performance Metrics**

|  |
| --- |
| **Classification Model:** Confusion Matrix –  [[2804 109] [ 169 231]]  Accuray Score-  0.9160881376396016  Classification Report – |
| Hyperparameter Tuning –  max\_depth = 4, min\_samples\_split = 4, random\_state = 25  Validation Method -  used the KFold method  n\_split = 3  shuffle = Ture  random\_state = 34 |

**10. ADVANTAGES & DISADVANTAGES**

**Advantages:**

* An effective and efficient model
* Has a good accuracy
* Simple and easy UI
* Data privacy is maintained

**Disadvantages:**

* Not many options

**11. CONCLUSION**

The studies mentioned above offer some suggestions for contemporary approaches to tracking expenses. These kinds of studies frequently demonstrate how ideas change over time. Evolution is not a prerequisite; rather, it's a shift in thinking and time, during which we estimate, evaluate, and assess things in accordance with changing needs. Although there are some android apps, the technology used in these projects was similar to that used in earlier times. Nevertheless, these apps have their own set of problems. Additionally, I believe that things need to be much simpler to use on a desktop device. Because Android apps sometimes produce accurate results.

**12. FUTURE SCOPE**

We would like to add the features like

* Providing chatbot function
* Sending the details to third parties if user wants to

**13. APPENDIX**

Source code:

**app.py**

# importing the necessary dependencies

from flask import Flask, request,render\_template

import numpy as np

import pandas as pd

import pickle

import requests

import json

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

app = Flask(\_name\_)

@app.route('/')

def home():

return render\_template("index.html")

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

def predict() :

if request.method=="POST":

name=request.form["name"]

month=request.form["month"]

dayofmonth = request.form['dayofmonth']

dayofweek = request.form['dayofweek']

origin = request.form['origin']

destination=request.form['destination']

origin1,origin2,origin3,origin4,origin5=0,0,0,0,0

if(origin=="msp"):

origin1,origin2,origin3,origin4,origin5=0,0,0,1,0

if(origin=="dtw"):

origin1,origin2,origin3,origin4,origin5=0,1,0,0,0

if(origin=="jfk"):

origin1,origin2,origin3,origin4,origin5=0,0,1,0,0

if(origin=="sea"):

origin1,origin2,origin3,origin4,origin5=0,0,0,0,1

if(origin=="alt"):

origin1,origin2,origin3,origin4,origin5=1,0,0,0,0

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 == "alt"):

destination1,destination2,destination3,destination4,destination5 = 0,0,0,1,0

dept = request.form['dept']

arrtime = request.form['arrtime']

actdept = request.form['actdept']

depti5=int(dept)-int(actdept)

total = [[month,dayofmonth,dayofweek,origin1,origin2,origin3,origin4,origin5,destination1,destination2,destination3,destination4,destination5,dept,actdept,depti5,arrtime]]

# y\_pred = model.predict(total)

# NOTE: you must manually set API\_KEY below using information retrieved from your IBM Cloud account.

API\_KEY = "IU\_76Xzg4uhR9DbnmjkoLeMoa5ePG3QAmmPfcUBPjziH"

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}

# NOTE: manually define and pass the array(s) of values to be scored in the next line

payload\_scoring = {"input\_data": [{"field": [["f0","f1","f2","f3","f4","f5","f6","f7","f8","f9","f10","f11","f12","f13","f14","f15","f16"]], "values": total}]}

response\_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/9b151467-653e-419b-bb25-1b3a6c11228d/predictions?version=2022-11-22', json=payload\_scoring,

headers={'Authorization': 'Bearer ' + mltoken})

print("Scoring response")

print(response\_scoring.json())

y\_pred=response\_scoring.json()

print(y\_pred['predictions'][0]['values'][0][1][1])

v=y\_pred['predictions'][0]['values'][0][0]

print(type(v))

if(v==0.):

return "The Flight will be on time"

else:

return "THE FLIGHT WILL BE DELAYED"

app.run(debug=True)

**Github link**

[https://github.com/IBM-EPBL/IBM-Project-32067-1660207960](https://github.com/IBM-EPBL/IBM-Project-32067-1660207960%20%20%20%20%20%20%20%20%20%204)

**Demo video link**

<https://drive.google.com/file/d/1VcrgameWm9fNQnD7BZX1EjQoM0B6TG9b/view?usp=sharing>