PROJECT REPORT

Submitted by

TEAM ID	***PNT2022TMID08017***
SATHISH G	**(AC19UCS100)**
SHEIK HASSAINSADIQ S	**(AC19UCS105)**
UDAYABALAJI S	**(AC19UCS130)**
VARUN R	**(AC19UCS133)**

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

ADHIYAMAAN COLLEGE OF ENGINEERING

DR. M.G.R NAGAR, HOSUR-635130

ANNA UNIVERSITY: CHENNAI 600 025

November 2022

Developing a Flight Delay Prediction Model using Machine Learning

1.INTRODUCTION :

1.1 Project Overview :

As in this modern age of technology travelling through air and air means transportation has been a crucial part in todays technology . Undeniably the airline's and airport has become more important means of transportation than the other means of transportation.In the year 2021 , out of the 3 million air carriers that were in service , 81.17% of those carriers reported to be on time , and nearly 16.78% of the flights where reported to be delayed in arrival . 1.79% of those carriers where cancelled due to unforeseen circumstances and around 10 thousand of those flights were diverted due to emergency situations and weather conditions. Domestic flight delays cause about a $32.9 billion loss to the economy, and about half that cost is borne by airline passengers which is almost $16.7 billion. This number was calculated based on lost passenger time due to flight delays, cancellations and missed connections, as well as expenses for food and accommodations as a result of being away from home. The airlines with high rates of delay also have higher operating costs overall due to the cost of airlines included increased expenses for crew, fuel and maintenance, among others.

1.2 Purpose:

In this project, we try to resolve this problem by building a flight delay prediction model which uses machine learning techniques like decision tree classifier model and logical regression model to make a prediction using the dataset on flight delay. In this model a flight is considered delayed if its arrival and departure time is longer than 15 minutes .By predicting the delay before hand we can avoid situation where the passengers save their time and the airline companies can save money that could have been caused by the delay.

2. LITERATURE SURVEY

2.1 Existing Problem:

An accurate estimation of flight delay is critical for airlines because the results can be applied to increase customer satisfaction and incomes of airline agencies. There have been many researches on modeling and predicting flight delays, where most of them have been trying to predict the delay through extracting important characteristics and most related features. However, most of the proposed methods are not accurate enough because of massive volume data, dependencies and extreme number of parameters.

2.2 Reference:

we referred six professionally published papers by student's from all around the world , on flight delay prediction and how to predict the delay using machine learning techniques. These journals are studied and classified into simpler field's like proposed work , tools / algorithm used and a short advantage and disadvantage they hold for easy understanding. A link to each of these paper's are given the end of this document .

i) Flight delay prediction using supervised machine learning

The flight delay has been predicted by data collection mainly relies on the airport destination and their connecting routes using supervised machine learning algorithm for classification of flight delays.They used tools like Decision tree , SPARK software , Statistical modules , it has advantages and disadvantages of The supervised machine learning algorithm for classification produces higher accuracy/Requires more processing time And low accuracy rate.

ii) Flight delay prediction based on deep learning and Levenberg-Marquardt algorithm

Optimized forecasting model based on deep learning which engages LM algorithm.to study and validate the positive effect of denouncing auto-encoder and LM algorithm , Tools and algorithm used in this model are Levenberg - Marquardt algorithm , Roof mean square error , Linear regression , It has advantages and disadvantages of Accuracy of SDA-LM model with imbalanced dataset respectively is greater Than SAE-LM model/Dataset should be balanced instead of under-sampling and up-sampling

iii) Comparative Study of Flight Delay Prediction using Back Propagation and Radial Basis Function

predict the delays of flights prediction using Back-propagation network and Radial Basis and the one with most accuracy will be considered To build the model and the model , IT uses tools and algorithms like Radial Basis Function , Back propagation algorithm and it has advantages and disadvantages of RBFs can be trained much faster than the perception , The smallest training error was achieved with RBFN , / The dataset used was not large enough

iv) Flight Delay Prediction Using a Hybrid Deep Learning Method

The goal of this study is to assess the viability and efficacy of the HDL model in comparison to a feed-forward ANN and gradient boosted tree (XGBoost). and uses tools and algorithms like XGBoost , Hybrid deep learning.has a advantage and disadvantage of The model will likely outperform the XGBoost model , If weather data included / The HDL model did not result in the highest accuracy compared to ANN model.

v) Flight Delay Classification Prediction Based on Stacking Algorithm

Stacking algorithm is a combination of different algorithms with different performances that verify how strong or weak learners affect the Stacking performance.and uses tools and algorithms like SMOTE algorithm , k - fold , Features Selection , Boruta algorithm has advantages and disadvantages like Overall accuracy remained same when the algorithm stacking was implemented/ It does not add exact weather related features in the prediction model.

vi) Flight Departure Time Prediction Based on Deep Learning

Works on GRU neural network prediction model under the influence of multiple factors.and uses tools and algorithms like A gated recurrent unit (GRU) model , Deep LSTM neural network has advantages and disadvantages of static research data are easy to obtain and the modeling is simple./ If the historical data cannot be updated, it is difficult to present the flight operation law.

2.3 Problem statement Definition :

The main objective of the problem is to develop a machine learning model that is capable of handling huge volume of data and process that data to and give a prediction , if the flight is delayed or not. The prediction is positive(delayed) if the time difference between the departed time and arrival time is greater than 15 minutes , which is the standard the standard waiting time which lead's little to no financial both the passanger's and airline's . Building an model that is capable of considering all viable factors that can influence in a delaying an aircraft and help foresee these delay's and give people awareness on this delay is crucial.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map :

Before coming up with different possible solution it is important to know customers point-of-view and understand the felling's and problems they face .Empathy map is a collaborative tool teams can use to gain a deeper insight into the problems and it's effects that's been cause to the customers .
It is important know these detail's as it help's us to come up with more innovative solution that can address all the aspects of the problem. Here in this empathy map we as a team have attempted to come up all the possible concer's a customer could have regarding out problem statement.

In this empathy map there are five field's where each of them represent how the customer is and how is their surrounding when the problem from the problem statement occurs . To this question the customer may be a casual traveler , A business person or even a member of a flight crew . The surrounding where the passenger's are placed is an airport . Aeroplane's are known to be late so many airports hold waiting lounge's to provide a space for the waiting travelers .
The field's used in this empathy map are *What do they see ? , What do they hear ?, what do they think and feel ? , What do they say and do ?* and *what are the pain and gain* they obtain from this experience. In the above statements " *they* " is used to represent the customer's or the passengers who are stuck in the airport due to the flight delay . Here we try to think of all the possible reaction and emotion the travelers might go through during the waiting time. Even when we casually think about flight delay we only think about the negative aspects that affects the travelers so, we also tried to shine some light on the positive aspects that might possibly occur due to the flight delay which might result in favor of the customer or the traveler.

3.2 Ideation & Brainstorming:

Ideation essentially refers to the whole creative process of coming up with and communicating new ideas. Ideation is innovative thinking, typically aimed at solving a problem or providing a more efficient means of doing or accomplishing something. It encompasses thinking up new ideas, developing existing ideas, and figuring out means or methods for putting new ideas into practice.

Brainstorming is a process where we as a team come up with all the possible way's we could counter the problem by stating our idea and opinion . Here , the brainstorming phase contains three stage of proposing solution for the problem.

First we define the problem statement. Next every teammate would give their individually idea on how we can over come the problem . These idea's should be noted as it is used in the next stage. Now The group would as whole would give idea's on how to overcome the problem.At last the group ideas are prioritize on the basis of its importance with respect to feasibility of the solution. Finally the brainstorming model is produced as

3.3 Proposed Solution:

The proposed solution for this problem statement was building a machine learning model which worked specifically with two algorithm's namely Decision Tree Classifier and Logistical Regression while would be implemented within a simple neural network. These machine learning techniques where capable enough to handle huge dataset's and had less processing time.The proposed solution should also address some other factors like the scalability , novelty , social impact . All these feature's are addressed in the below table.

3.4 Problem Solution fit :

Problem solution fit helps us to see the potential of our solution and how adaptable it is to the problem and how it help's in addressing different problem's faced by the customer at each stage of our proposed solution. The problem fit solution has Ten aspects to it and these aspects are classified into Three major division . First three helps us to know who our customer's are , Second three aspect's help's us to know about the problem and its related behavior . Finally the Third stage point's out triggers , emotions and channels that the customer would go through . The solution is represented at the bottom middle aspect of the Problem solution fit which is given as,

4. REQUIREMENT ANALYSIS

4.1 Functional requirement:

The functional requirement's of a solution contain's all the technical functionalities of the solution like the data processing, programming approach and calculation that define what a system is supposed to accomplish. The Functional requirement's should never be skipped and else the solution might not work as intended. Functional Requirements for this problem statement is mentioned as below,

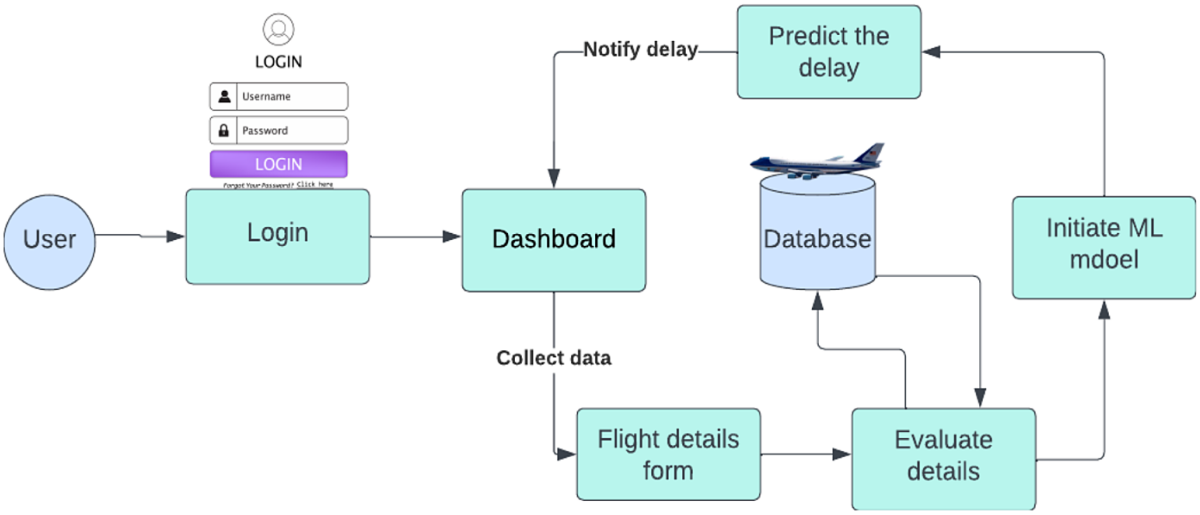
4.2 Non-Functional requirements :

Non-Functional Requirements are the not so important aspects of the solution that doesn't involve any technical aspect's to it . Usually the requirements tell's us about the system and performance constraint's . If failed to meet these requirements , the user experience with our solution will be bad which in turn make's our solution unreliable , so it is essential to construct a solution which can address all the Non - Functional Requirement's of the problem and the system in which the solution will be added to. The Non - Functional Requirements for this problem statement's solution is given as below,

5. PROJECT DESIGN

5.1 Data Flow Diagrams :

The data flow diagram helps us to easily understand how the data is being exchanged within the solution . Here is an short version of how the data travel's in this solution , first the user create's a login where he either create's a new account with their e-mail and new password ,or enter's the site with the already existing credentials. From there the user is redirected to the dashboard where is they are asked to enter the details related to the flight like flight number, arrival time , departure time etc. After getting these information the data is now sent to the machine learning model which compare the data with the prediction which is derived from the dataset . Finally the prediction is displayed in form of a notification on the user's screen.



5.2 Solution & Technical Architecture :

Our solution for this problem statement is to create a simple neural network and implement either one of machine learning algorithm is; Decision Tree Classifier or Logistic regression. The technical architecture explain's in detail how the data flow within the system while also mentioning all the technical components the solution will contain .

The technical components can be classified into three divisions naming : Cloud , User, Data processing . Each component has it's own functionality and perform certain task . In the User side , the webpage is used to fetch data from the user . With the help of the internet the data from the user is sent to the cloud. The Cloud contains the machine learning model which test and train the data from the dataset from where the model get's its prediction . Data processing contains the dataset and initializes and split's the data into test and train data. The cloud compare the user's value with the prediction value and send the output to the user through internet. The below diagram represent's all the technical components along with how the data is communicated between them,

5.3 User Stories :

A user story is an informal, general explanation of a software feature written from the perspective of the end user. The user story for this solution is given as below,

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile 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 Gmail	I can login at any time	Medium	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering email & password	I can Fill the flight details	High	Sprint-1
	Dashboard	USN-5	As a user,i can fill the flight details for which i want to get the prediction	I can get the delay prediction	High	Sprint-2
Customer (Web user)	Login	USN-6	As a user, I can login to my account and enter the webpage through Email & password.	I can access the webpage	HIgh	Sprint-1
Administrator	Access	USN-7	As a administrator, I can access and change the database and model	I can update the database and model	High	Sprint-2
	Modify	USN-8	As a administrator, I can edit the webpage based on the feedbacks	I can customize the web page	Medium	Sprint-2

6. PROJECT PLANNING & SCHEDULING :

In this stage we begin the planning and flow of work regarding developing the solution . A well planned out project would always help in prioritizing the most important requirements and schedule time according to the task in hand.

6.1 Sprint Planning & Estimation:

The sprint planning and estimation is developed under JIRA software development model . JIRA help's us in tracking the work flow of the project by creating user stories (epic) explaining each use case of the solution , keep track of the work done in the form of sprint's and also track all thed errors or chage's that the model goes through the development. As said before the development phase of the software solution in divided into sprints . Each sprint contains some specific task or use case, which has to completed within the time span that was planned before hand. Creating sprint help's us in dividing the work amongst over self and hold the person or persons accountable when the specific task is not done. Each user story or use case is given story point with respect to its relevance in the overall solution.

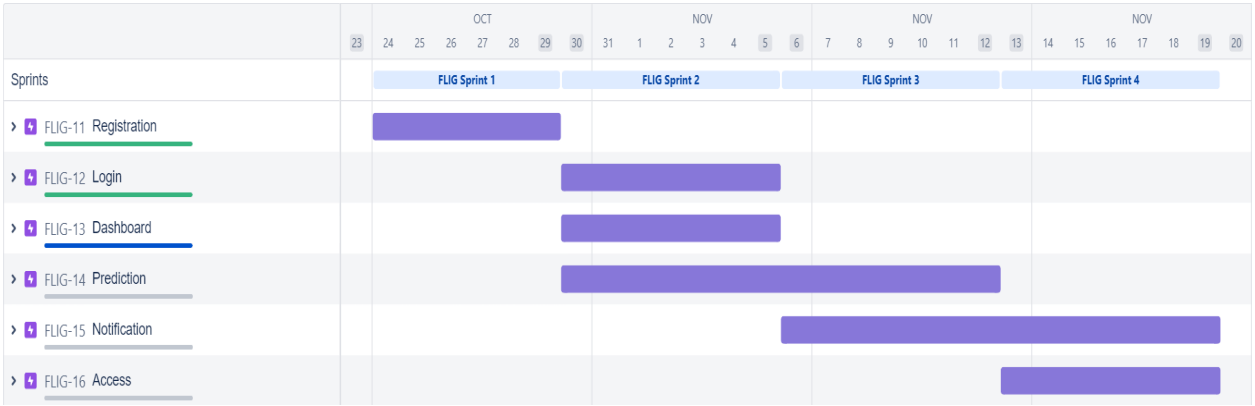
6.2 Sprint Delivery Schedule :

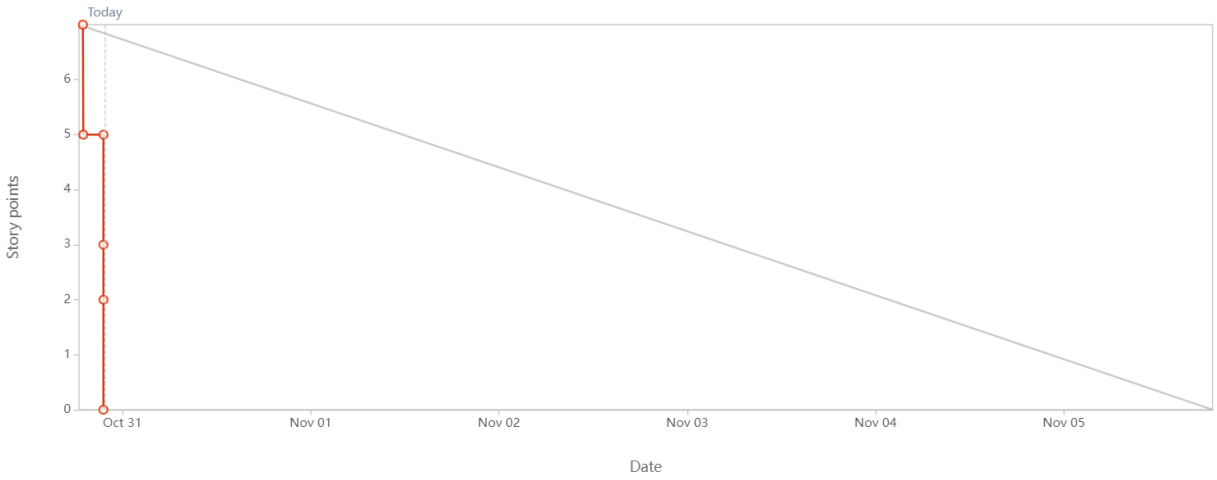
The sprint delivery is divided into four parts or four sprint's each sprint consist of two or three User Story / Task . Each task has its own story point which indicate's it's relevance to the solution and they are given the priority of the task.

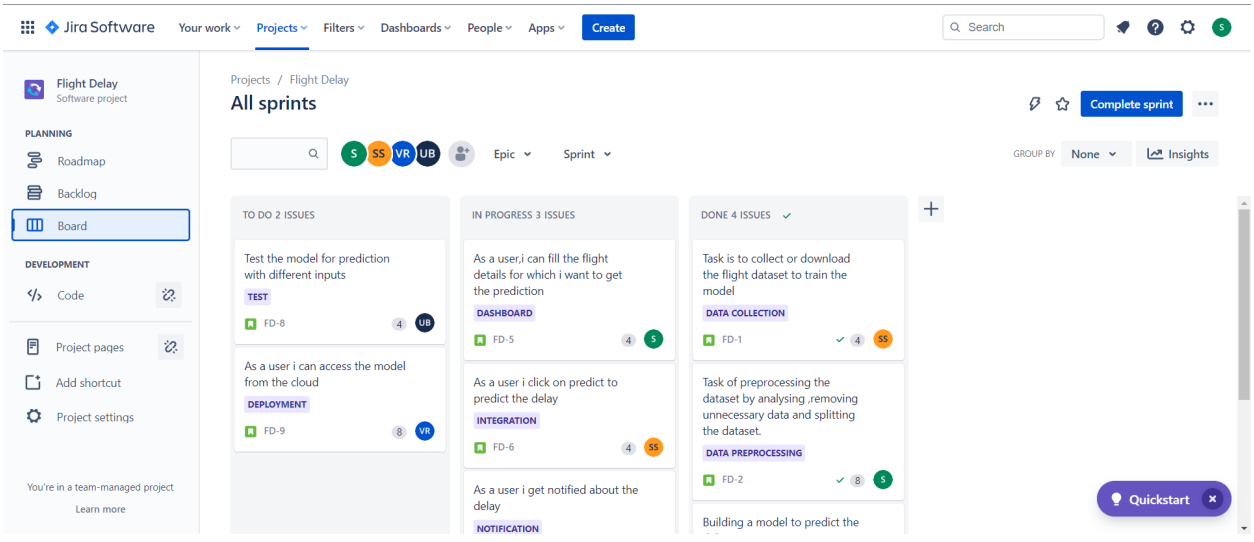
Sprint-3	Integration	USN-6	As a user i click on predict to predict the delay	4	High	Sathish, Sheik Hassain Sadiq
Sprint-3	Notification	USN-7	As a user i get notified about the delay	4	High	Sathish, Sheik Hassain Sadiq
Sprint-4	Test	USN-8	Test the model for prediction with different inputs	4	High	Varun, Udaya Balaji
Sprint-4	Deployment	USN-9	As a user i can access the model from the cloud	8	High	Varun, Udaya Balaji

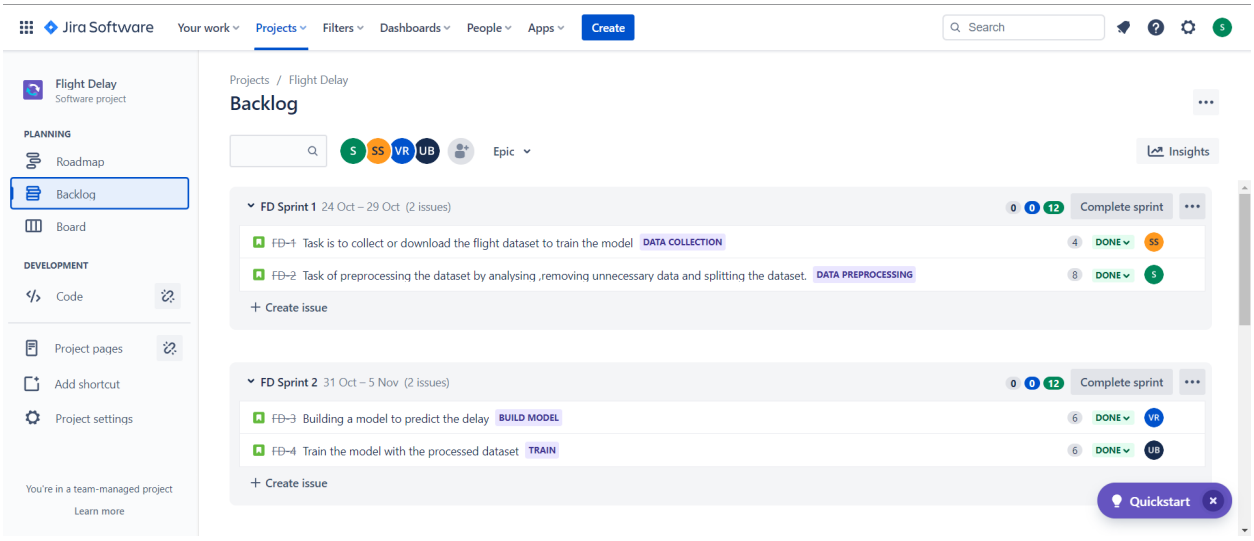
6.3 Reports from JIRA :

The JIRA software generate's reports based up on our development . These reports are generated on the basis of sprint completion and changes in sprint schedule.




Today
Story points
6
5
4
3
2
1
0
Oct 31
Nov 01
Nov 02
Nov 03
Nov 04
Nov 05
Date


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Flight Delay
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Projects / Flight Delay
All sprints
Complete sprint
Epic
Sprint
GROUP BY
None
Insights
TO DO 2 ISSUES
Test the model for prediction with different inputs
TEST
FD-8
As a user i can access the model from the cloud
DEPLOYMENT
FD-9
IN PROGRESS 3 ISSUES
As a user,i can fill the flight details for which i want to get the prediction
DASHBOARD
FD-5
As a user i click on predict to predict the delay
INTEGRATION
FD-6
As a user i get notified about the delay
NOTIFICATION
DONE 4 ISSUES
Task is to collect or download the flight dataset to train the model
DATA COLLECTION
FD-1
Task of preprocessing the dataset by analysing ,removing unnecessary data and splitting the dataset.
DATA PREPROCESSING
FD-2
Building a model to predict the
Quickstart


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Flight Delay
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Projects / Flight Delay
Backlog
Epic
Insights
FD Sprint 1 24 Oct – 29 Oct (2 issues)
Complete sprint
FD-1 Task is to collect or download the flight dataset to train the model DATA COLLECTION
FD-2 Task of preprocessing the dataset by analysing ,removing unnecessary data and splitting the dataset. DATA PREPROCESSING
Create issue
FD Sprint 2 31 Oct – 5 Nov (2 issues)
Complete sprint
FD-3 Building a model to predict the delay BUILD MODEL
FD-4 Train the model with the processed dataset TRAIN
Create issue
Quickstart

7. CODING & SOLUTIONING

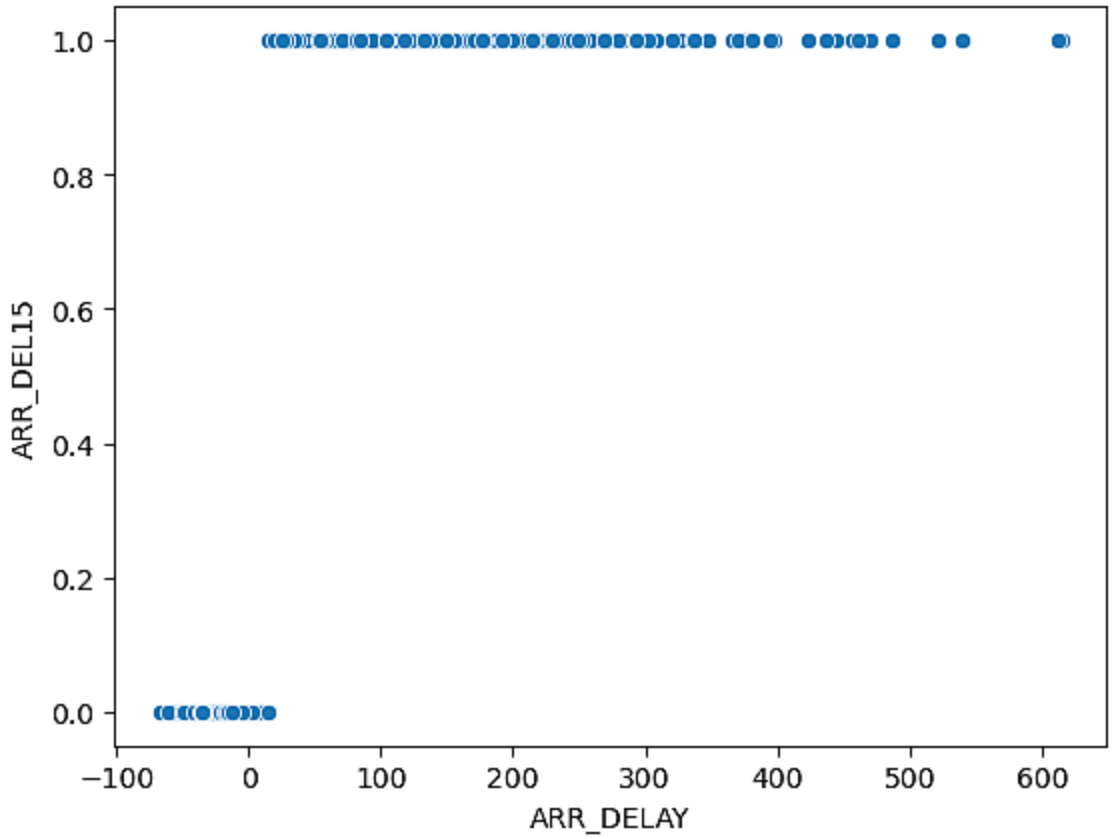
In this project we have created a machine learning model using python code in Jupyter Notebook using python code and then saved the model using the pickle library , Then the model is integrated with the html page with the help of flask application ,Finaly the model is deployed in the IBM cloud.

7.1 Data Preprocessing :

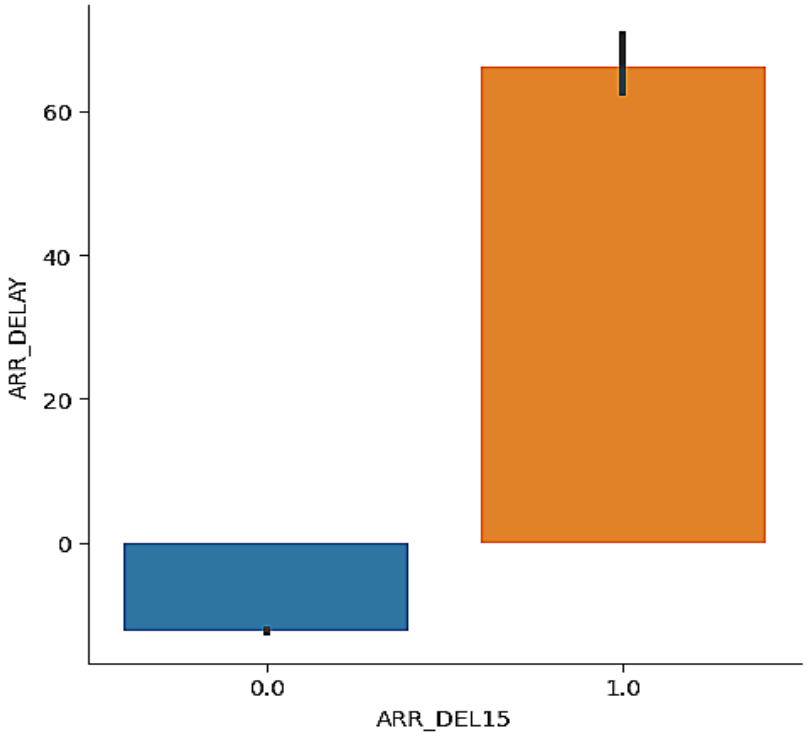
The flight delay dataset is collected and loaded in the notebook and the various preprocessing techniques are performed in the dataset like Analyzing data , Data visualization , Handle missing values , Dropping un necessary columns ,Label encoding &One hot encoding , Splitting into dependent and independent and into Train and test data .

```
# importing the required libraries
import sys
import numpy as np
import pandas as pd
import seaborn as sns
import pickle
%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.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import sklearn.metrics as metrics
```

```
import warnings
warnings.filterwarnings('ignore')
# importing the dataset
dataset = pd.read_csv("flightdata.csv")
# Analysing the data
dataset.info()
dataset.describe()
# Handling missing values
dataset.isnull().sum()
dataset['DEST'].unique()
# Data visualization
sns.scatterplot(dataset['ARR_DELAY'],dataset['ARR_DEL15'])
```



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




```
sns.heatmap(dataset.corr())
# Dropping unnecessary columns
dataset = dataset.drop('Unnamed: 25',axis=1)
dataset.isnull().sum()
dataset=dataset[["FL_NUM","MONTH","DAY_OF_MONTH","DAY_OF_WEEK
","ORIGIN","DEST","CRS_ARR_TIME","DEP_DEL15","ARR_DEL15"]]
dataset.isnull().sum()
# replace missing values with 0 and 1
dataset = dataset.fillna({'ARR_DEL15':1})
dataset = dataset.fillna({'DEP_DEL15':0})
dataset.iloc[177:185]
#Label encoding & One hot encoding

le = LabelEncoder()
```

```
dataset['DEST'] = le.fit_transform(dataset['DEST'])
dataset['ORIGIN'] = le.fit_transform(dataset['ORIGIN'])
oh = OneHotEncoder()
z = oh.fit_transform(dataset[['ORIGIN','DEST']]).toarray()
t = oh.fit_transform(dataset[['DEST','CRS_ARR_TIME']]).toarray()
#Splitting into dependent and independent variables
dataset = pd.get_dummies(dataset , columns=['ORIGIN','DEST'])
dataset.head()
x=dataset.drop(columns=['ARR_DEL15']).values
y=dataset['ARR_DEL15'].values
#Splitting into train and test data
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=0)
x_test.shape
x_train.shape
y_test.shape
y_train.shape
```

7.2 Model Building :

A model with a decission tree classifier is built and trained with the train data and tested with the test data from the processed dataset and accuracy score Is evaluated for the model ,The model is saved using the pickle library in python

```
#Train and test the model Using Decision tree classifier
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
```

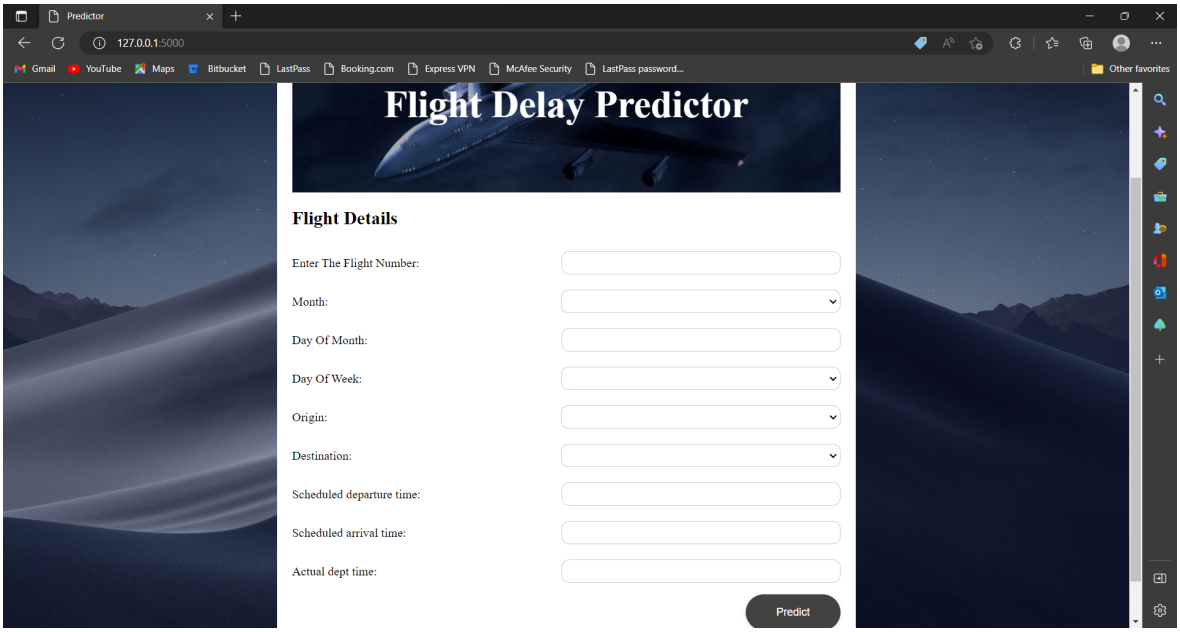
```
x_test = sc.transform(x_test)

classifier = DecisionTreeClassifier(random_state = 0)
classifier.fit(x_train,y_train)

decisiontree = classifier.predict(x_test)
decisiontree

#Predict The Result
decisiontree = classifier.predict(x_test)

#Score Of The Model
desacc = accuracy_score(y_test,decisiontree)
desacc
```

```
0.8722741433021807
```

```
#Saving Our Model
pickle.dump(classifier,open('flight.pkl','wb'))
```

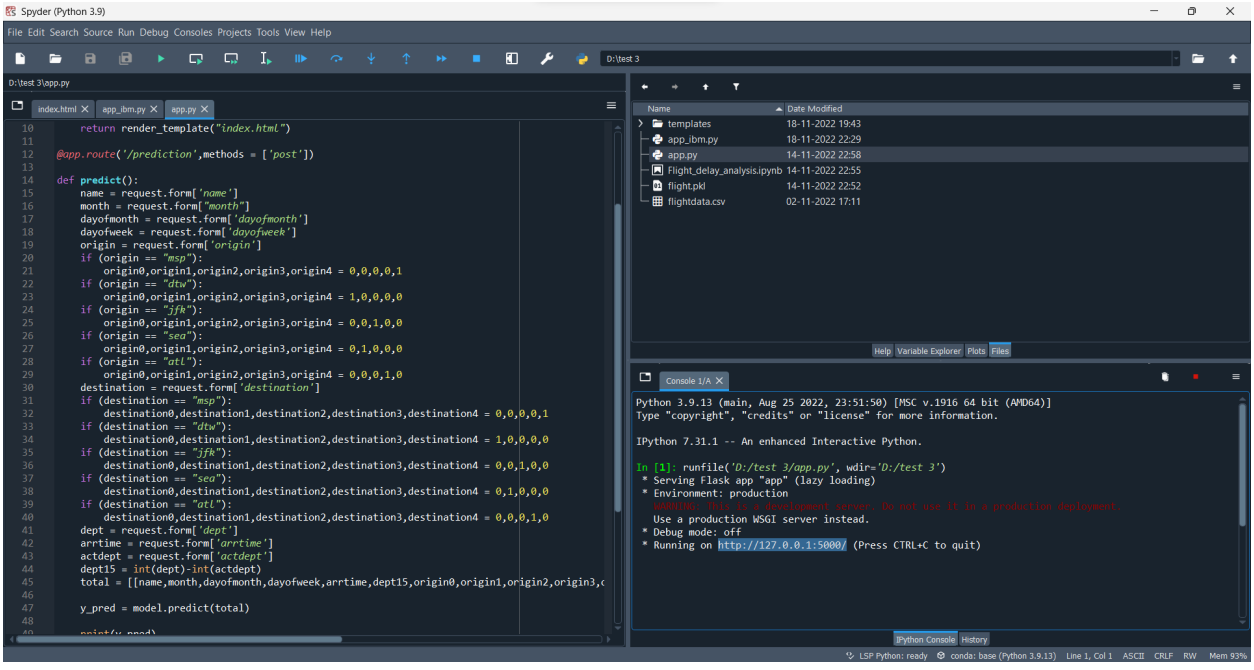
7.3 Web Page Building :

The web page which helps the user to input their details for the prediction



7.4 Flask Integration:

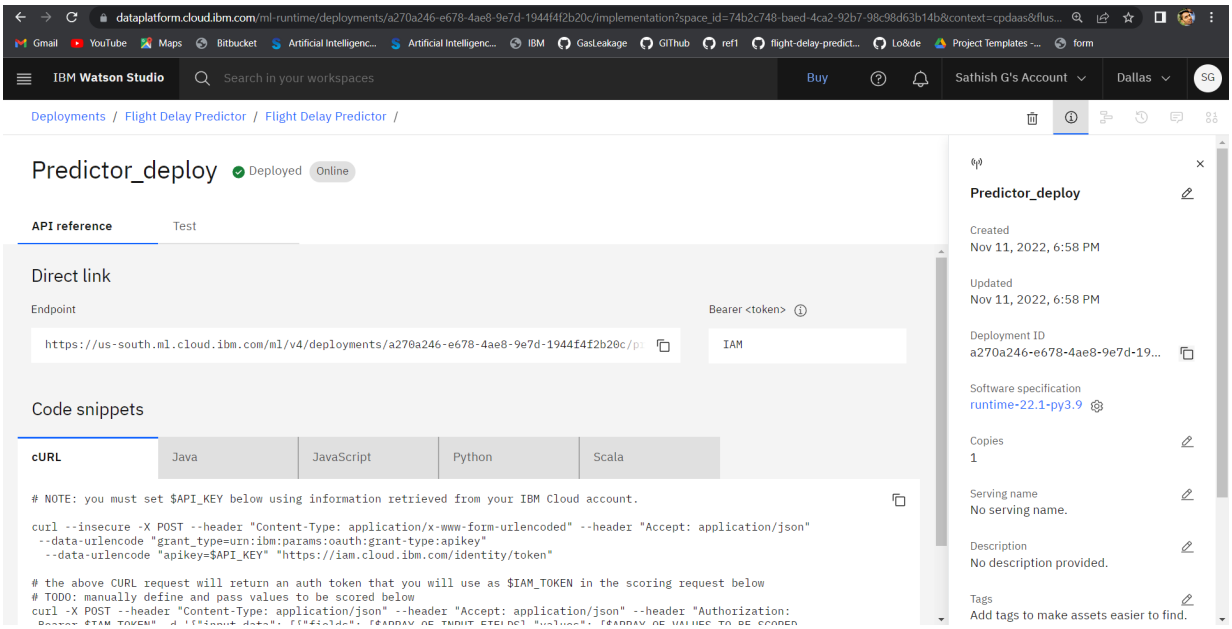
The web page is integrated with the model using the flask application



7.5 Cloud Deployment :

In cloud deployment model after integrating with the flask is the deployed in the IBM cloud so that it can be hosted online

```
########################################################################################
#############
Synchronous deployment creation for uid: 'd9a1f746-41b8-4936-8378-
25bf7013f32f' started
########################################################################################
#############
initializing
Note: online_url is deprecated and will be removed in a future release.
Use serving_urls instead.
ready
------------------------------------------------------------------------------------
----------------------
Successfully finished deployment creation, deployment_uid='a270a246-e678-
4ae8-9e7d-1944f4f2b20c'
------------------------------------------------------------------------------------
----------------------
```



8. TESTING

The testing phase is crucial for all project development . Here we test all model and their functionality. Testing phase can weed out any error's or bugs that has been in the program .

8.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	Executed By
Dashboard	UI	Home Page	Testing the prediction button with completely filled form	Web browser	1.Enter URL and click go 2.Fill the details form completely 3.Click on predict button	http://127.0.0.1:5000/	Notification of prediction should appear	Working as expected	Pass	A prediction of delay or on time is appeared	Udaya balaji
Dashboard	UI	Home Page	Testing the prediction button without filling the form	Web browser	1.Enter URL and click go 2.Fill the details form in complete 3.Click on predict button	http://127.0.0.1:5000/	Notify to fill the form completely	Working as expected	Fail	A error message of "this field is required" is apeared	Sheik Hasain sadiq
Prediction model	Functional	Model	Test the model for delay prediction	Jupyter Notebook	1.Importing the model 2.Fill the details that shows flight is delayed 3.Execute prediction	1.Flight no:1699 2.Month : 1 3.Day of month : 5 4.Day of week :3 5.Origin : 00001 6.Destination:01000 7.Dept time:1950 8.Arr time:2020	Predicted as delay	Working as expected	Pass	Notified as flight will be delayed	Sathish
Prediction model	Functional	Model	Test the model with details of flight on time	Jupyter Notebook	1.Importing the model 2.Fill the details that shows flight is delayed 3.Execute prediction	1.Flight no:1480 2.Month : 5 3.Day of month : 3 4.Day of week :6 5.Origin : 00010 6.Destination:00010 7.Dept time:1700 8.Arr time:1950	Predicted as on time	Working as expected	Pass	Notified as flight will be on time	Varun
Deployment	Functional	Cloud	Testing the web application integration with cloud	Web browser	1.Enter URL and click go 2.Fill the details form in complete 3.Click on predict button	1.Flight no:1600 2.Month : 7 3.Day of month : 2 4.Day of week :5 5.Origin : 10000 6.Destination:01000 7.Dept time:0400 8.Arr time:0600	Notification of flight delay prediction	Working as expected	Pass	Model in cloud is accesed and verified	Sathish

8.2 User Acceptance Testing

Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	3	2	2	3	10
Duplicate	1	0	0	0	1
External	1	0	3	0	4
Fixed	5	2	3	3	13
Not Reproduced	0	0	2	0	2
Skipped	0	0	1	1	2
Won't Fix	0	0	1	1	2
Totals	10	4	12	8	34

Test Case Analysis

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

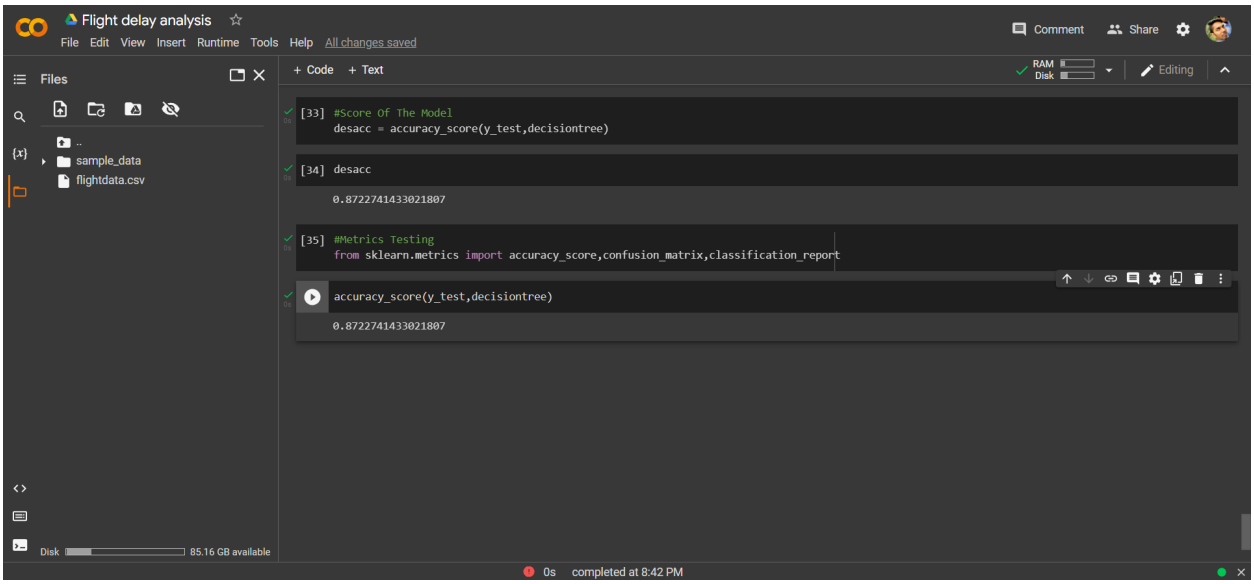
Section	Total Cases	Not Tested	Fail	Pass
Homepage	20	0	0	20
Model	14	0	0	14

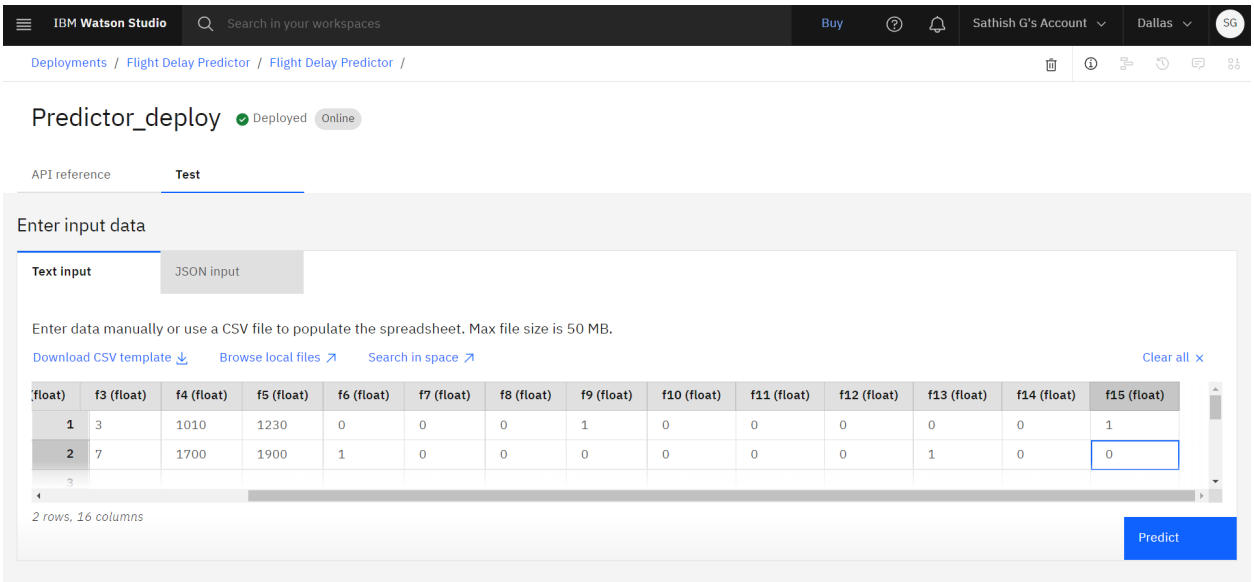
---

9. RESULTS

This below Information contains the combined result of this project. The algorithm Decision tree classifier which we used has a 87% accuracy for the prediction.

9.1 Performance Metrics





10. ADVANTAGES & DISADVANTAGES

The advantages and disadvantages of this project is listed below ,

ADVANTAGES :

- The project is developed with the goal to develop a flight delay prediction model which operates on machine learning algorithm Decision tree classifier.
- This algorithm is capable of handling the huge volume of data that is required for the prediction .
- Decision tree classifier can also handle both numeric and categorical data and show Versatility, Non-Linearity in prediction
- This model can help reduce the time required and also has possibilities to develop in future if the data set is changed.

DISADVANTAGES :

- The main disadvantage for this model is it does not include real time data meaning the data used for prediction does not include the currently operating data.
- The model also failed to include weather related data , so this model cannot predict the delay caused by weather's

11. CONCLUSION

In this project we use a simple webpage to get the details and implement the flight dataset to predict the flight delay. Based on the data derived from the dataset we produce the predicted result. We also conclude that the decision tree classifier can produce good performance compared to the logistic regression model which

had less accuracy rate compared to the decision tree classifier.

12. FUTURE SCOPE

This project can further be developed by implementing real-time flight information and using more powerful systems we can also use weather related data . We can also implement new technologies which might make the data processing and prediction easier and implement real time notification for the user.

13. APPENDIX

App.py

```
from flask import Flask,request,render_template
import pickle

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

app = Flask(__name__)

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

@app.route('/prediction',methods = ['post'])

def predict():
    name = request.form['name']
    month = request.form["month"]
```

```
    dayofmonth = request.form['dayofmonth']
    dayofweek = request.form['dayofweek']
    origin = request.form['origin']
    if (origin == "msp"):
        origin0,origin1,origin2,origin3,origin4 = 0,0,0,0,1
    if (origin == "dtw"):
        origin0,origin1,origin2,origin3,origin4 = 1,0,0,0,0
    if (origin == "jfk"):
        origin0,origin1,origin2,origin3,origin4 = 0,0,1,0,0
    if (origin == "sea"):
        origin0,origin1,origin2,origin3,origin4 = 0,1,0,0,0
    if (origin == "atl"):
        origin0,origin1,origin2,origin3,origin4 = 0,0,0,1,0
    destination = request.form['destination']
    if (destination == "msp"):
        destination0,destination1,destination2,destination3,destination4 = 0,0,0,0,1
    if (destination == "dtw"):
        destination0,destination1,destination2,destination3,destination4 = 1,0,0,0,0
    if (destination == "jfk"):
        destination0,destination1,destination2,destination3,destination4 = 0,0,1,0,0
    if (destination == "sea"):
        destination0,destination1,destination2,destination3,destination4 = 0,1,0,0,0
    if (destination == "atl"):
        destination0,destination1,destination2,destination3,destination4 = 0,0,0,1,0
    dept = request.form['dept']
    arrtime = request.form['arrtime']
    actdept = request.form['actdept']
    dept15 = int(dept)-int(actdept)
                                                                total                    =
[[name,month,dayofmonth,dayofweek,arrtime,dept15,origin0,origin1,origin2,origi
n3,origin4,destination0,destination1,destination2,destination3,destination4,]]
```

```
    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",showcase = ans)

if __name__ == '__main__' :


    app.run(debug= False)
```

Index.html

```
<!DOCTYPE html>
<html>
<head>
<title>Predictor</title>
</head>
<style>
html, body {
      min-height: 100%;
      }
      body{
      background-image: url("https://wallpaperaccess.com/full/752854.jpg");
      height: 100%;
  background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
```

```
}
h1 {
position: absolute;
margin: 0;
line-height: 55px;
font-size: 50px;
color: #fff;
z-index: 2;
}
b{
margin-top:10px;
font-size: 30px;
color: #424242;
}
.testbox {
display: flex;
justify-content: center;
align-items: center;
height: inherit;
padding-left: 350px;
padding-right: 350px;
padding-top: 30px;
    padding-bottom:30px;
}
form {
width: 100%;
padding: 20px;
border-radius: 6px;
background: #fff;
}
.banner {
position: relative;
```

```
    height: 230px;
    background-image: url("https://pics.imcdb.org/0is16/flight3.1513.jpg");
        background-position: center;
        background-repeat: no-repeat;
        background-size: cover;
    display: flex;
    justify-content: center;
    align-items: center;
    text-align: center;
    }
    .banner::after {
    content: "";
    background-color: rgba(0, 0, 0, 0.2);
    position: absolute;
    width: 100%;
    height: 100%;
    }


.form-group {
display: flex; }
.form-group p {
  width: 50%; }
  .form-group input {
    margin-top: 10px;
    margin-bottom: 10px;
    border: 1px solid #ccc;
    border-radius: 10px;
    width: 50%;
    padding: 5px;
    }
    .form-group input:hover {
```

```
    border-color: #424242;
    }
    .form-group input:focus,
    .form-group select:focus {
      outline: none;
      Border-color:#424242;
    }
    .form-group select {
    margin-top: 10px;
    margin-bottom: 10px;
    border: 1px solid #ccc;
    border-radius: 10px;
    width: 52%;
    padding: 5px;
    }
    .form-group select:hover {
    border-color:#424242;
    }
    .form-group button {
width:  auto;
min-width:  100px;
border-radius:  24px;
text-align:  center;
padding:  15px 40px;
margin-top:  5px;
background-color: #424242;
color:  #fff;
font-size:  14px;
margin-left:  auto;
font-weight:  500;
box-shadow:  0px 2px 6px -1px rgba(0,0,0,.13);
border:  none;
```

```
  transition:  all .3s ease;
  outline: 0;
  }
  .form-group button:hover {
    transform:  translateY(-3px);
    box-shadow:  0 2px 6px -1px rgba(#424242, .65);
  }
  .form-group button:active {
    transform:  scale(.99);
  }
  </style>
<body>
<div class="testbox">
<form action="/prediction" method="post">
<div class="banner">
<h1>Flight Delay Predictor</h1>
</div>
<div class="item">
<h2>Flight Details</h2>
</div>
<div class="form-group">
<p>Enter The Flight Number:</p>
<input type="number" name ="name" required/></div>
<div class="form-group">
<p><label for = "month">Month:</label></p>
<select name="month" required>
<option value="none" selected disabled hidden></option>
<option value ="1">Jan</option>
<option value ="2">Feb</option>
<option value ="3">Mar</option>
<option value ="4">Apr</option>
<option value ="5">May</option>
```

```
<option value ="6">Jun</option>
<option value ="7">Jul</option>
<option value ="8">Aug</option>
<option value ="9">Sep</option>
<option value ="10">Oct</option>
<option value ="11">Nov</option>
<option value ="12">Dec</option>
</select></div>
<div class="form-group">
<p>Day Of Month:</p>
<input type="number" name ="dayofmonth" required/></div>
<div class="form-group">
<p><label for = "dayofweek">Day Of Week:</label></p>
<select name="dayofweek" required>
<option value="none" selected disabled hidden></option>
<option value ="1">Sun</option>
<option value ="2">Mon</option>
<option value ="3">Tue</option>
<option value ="4">Wed</option>
<option value ="5">Thu</option>
<option value ="6">Fri</option>
<option value ="7">Sat</option>
</select></div>
<div class="form-group">
<p><label for = "Origin">Origin:</label></p>
<select name="origin" required>
<option value="none" selected disabled hidden></option>
<option value ="msp">msp</option>
<option value ="dtw">dtw</option>
<option value ="jfk">jfk</option>
<option value ="sea">sea</option>
<option value ="atl">atl</option>
```

```
</select></div>
<div class="form-group">
<p>
<label for = "destination">Destination:</label></p>
<select name="destination" required>
<option value="none" selected disabled hidden></option>
<option value ="msp">msp</option>
<option value ="dtw">dtw</option>
<option value ="jfk">jfk</option>
<option value ="sea">sea</option>
<option value ="atl">atl</option>
</select>
</div>
<div class="form-group">
<p>Scheduled departure time:</p>
<input type="number" name ="dept" required/></div>
<div class="form-group">
<p>Scheduled arrival time:</p>
<input type="number" name ="arrtime" required/></div>
<div class="form-group">
<p>Actual dept time:</p>
<input type="number" name ="actdept" required/></div>
<div class="form-group">
<b>{{showcase}}</b>
<button type="submit" value="submit">Predict</button></div>
</form>

</body>
</html>
```

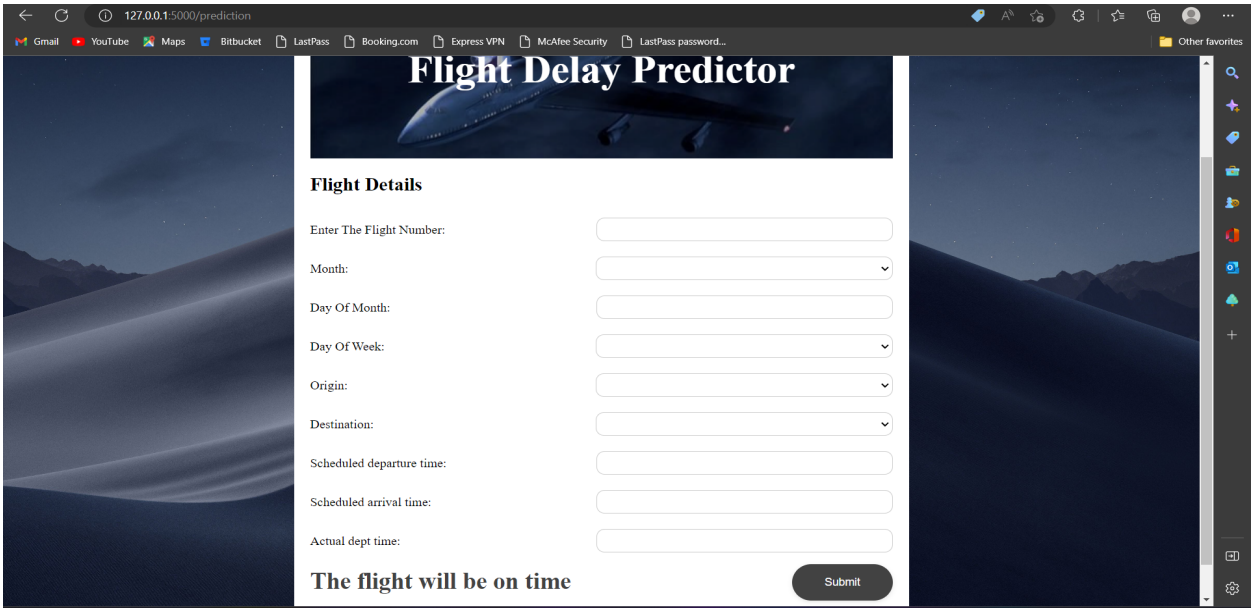
app_ibm.py

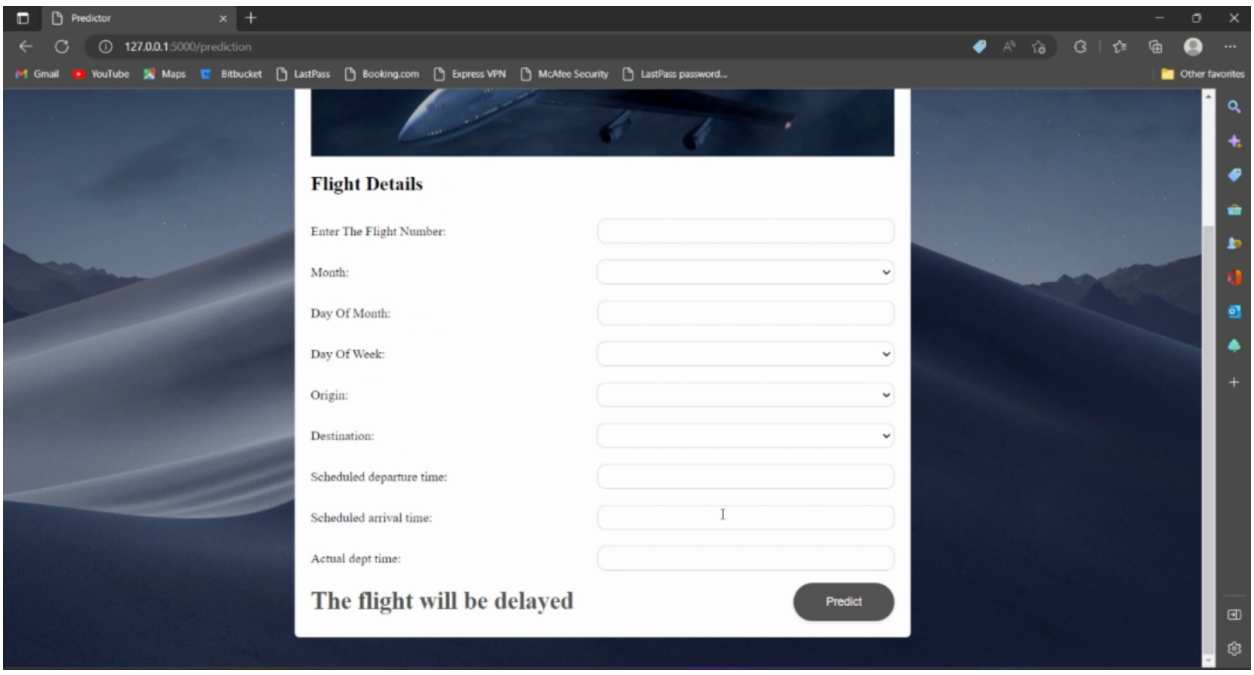
```
from flask import Flask,request,render_template
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "9rgwTXNNVEK-K7ZBxOJ8ypezVmm8-qZLspTCTlpEoejm"
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__)

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

@app.route('/prediction',methods = ['post'])

def predict():
    name = request.form['name']
    month = request.form['month']
    dayofmonth = request.form['dayofmonth']
    dayofweek = request.form['dayofweek']
    origin = request.form['origin']
    if (origin == "msp"):
        origin0,origin1,origin2,origin3,origin4 = 0,0,0,0,1
    if (origin == "dtw"):
        origin0,origin1,origin2,origin3,origin4 = 1,0,0,0,0
    if (origin == "jfk"):
```

```
    origin0,origin1,origin2,origin3,origin4 = 0,0,1,0,0
  if (origin == "sea"):
    origin0,origin1,origin2,origin3,origin4 = 0,1,0,0,0
  if (origin == "atl"):
    origin0,origin1,origin2,origin3,origin4 = 0,0,0,1,0
  destination = request.form['destination']
  if (destination == "msp"):
    destination0,destination1,destination2,destination3,destination4 = 0,0,0,0,1
  if (destination == "dtw"):
    destination0,destination1,destination2,destination3,destination4 = 1,0,0,0,0
  if (destination == "jfk"):
    destination0,destination1,destination2,destination3,destination4 = 0,0,1,0,0
  if (destination == "sea"):
    destination0,destination1,destination2,destination3,destination4 = 0,1,0,0,0
  if (destination == "atl"):
    destination0,destination1,destination2,destination3,destination4 = 0,0,0,1,0
  dept = request.form['dept']
  arrtime = request.form['arrtime']
  actdept = request.form['actdept']
  dept15 = int(dept)-int(actdept)
                                                    total =
[[name,month,dayofmonth,dayofweek,arrtime,dept15,origin0,origin1,origin2,origi
n3,origin4,destination0,destination1,destination2,destination3,destination4,]]

  # NOTE: manually define and pass the array(s) of values to be scored in the next
line
                    payload_scoring = {"input_data": [{"fields":
[['f0','f1','f2','f3','f4','f5','f6','f7','f8','f9','f10','f11','f12','f13','f14','f15']], "values":
total}]}

                    response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/a270a246-e678-4ae8-9e7d-
```

```
1944f4f2b20c/predictions?version=2022-11-11', json=payload_scoring,
     headers={'Authorization': 'Bearer ' + mltoken})
    print("Scoring response")
    print(response_scoring.json())
    pred = response_scoring.json()
    y_pred = pred['predictions'][0]['values'][0][0]
    if(y_pred==0):
        ans = "The flight will be on time"
    else:
        ans = "The flight will be delayed"
    return render_template("index.html",showcase = ans)

if __name__ == '__main__' :


    app.run(debug= False)
```

OUTPUT:


127.0.0.1:5000/prediction
Gmail
YouTube
Maps
Bitbucket
LastPass
Booking.com
Express VPN
McAfee Security
LastPass password...
Other favorites
Flight Delay Predictor
Flight Details
Enter The Flight Number:
Month:
Day Of Month:
Day Of Week:
Origin:
Destination:
Scheduled departure time:
Scheduled arrival time:
Actual dept time:
The flight will be on time
Submit


Predictor
127.0.0.1:5000/prediction
Gmail
YouTube
Maps
Bitbucket
LastPass
Booking.com
Express VPN
McAfee Security
LastPass password...
Other favorites
Flight Details
Enter The Flight Number:
Month:
Day Of Month:
Day Of Week:
Origin:
Destination:
Scheduled departure time:
Scheduled arrival time:
Actual dept time:
The flight will be delayed
Predict