KNOWLEDGE INSTITUTE OF TECHNOLOGY

TEAM ID: PNT2022TMID30209

TEAM MEMBERS:

REVATHI R (611219104076)

SARADA S P (611219104086)

SIVATHARANI V (611219104099)

SPARSHA S (611219104104)

TECHNOLOGY TRACK:

APPLIED DATA SCIENCE

USE CASE:

DEVELOPING A FLIGHT DELAY USING MACHINE LEARNING

TABLE OF CONTENTS

CHAPTER - 1

INTRODUCTION

1.1 PROJECT OVERVIEW

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.

1.2 PURPOSE

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. Our research concentrated mainly on predicting flight delays for a particular airport over a specific period of time. we used a regression model to examine the significance of each feature, Here we use Logistic Regression and some algorithms like Support Vector Machine, Decision Tree, Random Forest. Further, we applied an approach called One-Hot-Encoder to create a variant of the model for evaluating potential prediction performance.

CHAPTER - 2

LITERATURE SURVEY

2.1 EXISTING SYSTEM

To predict flight delays to train models, we have collected data accumulated by the Bureau of Transportation, U.S. Statistics of all the domestic flights taken in 2018 was used. The US Bureau of Transport Statistics provides statistics of arrival and departure that includes actual departure time, scheduled departure time, scheduled elapsed time, wheels-off time, departure delay and taxi-out time per airport. Cancellation and Rerouting by the airport and the airline with the date and time and flight labelling along with airline airborne time are also provided.

2.2 REFERENCE

2.2.1 Flight Delay Prediction based on AviationBig Data and Machine Learning(Author: Rahul Garg et.al.,2022)

The dataset includes the scheduled and actual departure and arrival times for non-stop flights recorded by different airways. Information on delayed and canceled flights, actual travel time and non-stop distances is also available in the dataset. Airline origin and destination are also included. With this dataset, a predictive model was used to solve the flight delay cases. The flight dataset includesdata for 161 airports. Flightsarriving after the scheduled arrivaltime are considered delayed. In addition, the flight under attack is deleted from the dataset. , time of departure, time of boarding are applied. This study aims to predict flight delays for airlines. Three methods are used to predict flight delay, that is, Random Forest, Support Vector Machine, K-nearest neighbor.

2.2.2 Flight Delay Prediction (Author: BhuvanBhatia, 2019)

The paper titled “Flight Delay Prediction” by Bhuvan Bhatia concentrated mainly on predicting flight delays for a particular airport over a specific period of time. First, they used a regression model to examine the significance of each feature and then, a feature selection approach to examine the impact of feature combination. These two techniques determined the features to retain in the model. Instead of using the whole set, we sampled 5,000 records at a time to run through differentmachine learning models. The machine learning models implemented here were Random Forest classifier and Support Vector Machine (SVM) classifier. Further, we applied an approach called One-Hot-Encoder to create a variant of the model for evaluating potential prediction.

2.2.3 Assessing Strategic Flight Schedules at an Airport using Machine Learning based Flight Delay And Cancellation Prediction (Author: Miguel Lambelho et.al., 2020

To migrate air traffic demand-capacity imbalances, demonstrate an approach for strategic flight schedulesin the period 2013-2018. Machine learning approach to predict whether strategic, scheduled arrival/departure flights are delayedor canceled. These predictions are based on strategic flight schedules from LHR and assume a 6-month prediction horizon, i.e., we predict whether flightsare delayed or canceled 6 months prior to the day of the flight execution. LightGBM is a tree-based machine learning algorithmthat uses Gradient-based One-Side Sampling, which excludes data instances with small gradients, and Exclusive Feature Building, which bundlesmutually exclusive variables, thus, reducing the number of features. We are considering extending the set of features for the prediction algorithms to improve

the accuracy of the predictions and will evaluatethe impact of considering flight delay and cancellation predictions in the flight scheduling optimization models, at the strategic phase.

2.2.4 Flight delay prediction based on deep learning and Levenberg-Marquart algorithm(Author: M F Yazdi et.al., 2020)

The Levenberg-Marquart algorithm is applied to find weight and bias proper values, and finally the output has been optimized to produce high accurate results. To investigate the three models, we apply the proposed model on the U.S flight dataset that is an imbalanced dataset. Algorithm used in this is min max normalization and denoising autoencoder training. In order to evaluatethe model, the number of denoising autoencoders and neurons must be determined based on the values for precision, accuracy and time consuming. At the end, to evaluate the validity of the proposed model and the results from training, we evaluate the standard deviation of all the parameters after the 30 times repetition. Comparing the three models for two of imbalanced and balanced datasets shows that accuracy of SDA-LM model with imbalanced dataset respectively is greater by 8.2 and 11.3% Than SAE-LM and SDA models. On the other hand, these values for balanced datasets are respectively 10.4 and 7.3%. At the next stage, the model has been evaluated and computed for subjects of discarding with a standard deviation for all evaluation parameters during 30 times of model run. Finally, we compared the accuracy of the proposed Model against SAE-LM,SDA and RNN models .

2.2.5 Predicting flightdelays using data from US Domestic flights(2019)

The objective of the project is to "Design a Model that predicts flight delays before they are announced on the departure boards". The dataset comes from Kaggle, and it consists of a multi- year data ranging from 2009 to 2019 separated in 10 different files. The data preprocessing and

cleaningwas done in two separateparts, documented in two notebooks to make it easier to follow up due to their length. The first section is a standard cleaning involving minimal feature engineering, and the second is driven after the 20 most common arrival destinations were defined based on the number of flights and is the one that contains the most feature engineering done.The same way the data cleaning and preprocessing was done in two separate notebooks, the EDA was done in two as well, however the difference here is that the visualizations done on each of the EDAs were done with different libraries. The first was done using matplotlib and Seaborn, and the second with plotly. Six type of ML Algorithms were tested, they were

a. Bagged Trees
b. Random Forest
c. AdaBoost
d. Gradient BoostedTrees
e. XGBoost
f. Deep Neural Network (MLP)

It is quite hard to create a ML model for flight delay prediction before you even know that the flight is delayed on the departure board. Neural Networks responded a lot better under these conditions with an average difference in accuracy, precision and recall of over 15%. Maybe an even more thorough feature analysis could raise these metrics to close to 90%.

2.3 PROBLEM STATEMENT DEFINITION

Problem Statement (PS)	**I am (Customer)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	Businessman	To reach the destination on time	Not able to reach	The flight is delayed	Stressed
PS-2	Tourist	To check the status	No clear information	Delay is not predicted correctly	Frustrated
PS-3	Traveller	To check list of flights with their status	No clear information	Status is unavailable	Tensed
PS-4	Passenger	Find alternate resources	Not all information is available	Limited to just one airport/ airline	Dissatisfied

CHAPTER - 3

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

Fig.1

3.2 IDEATION & BRAINSTORMING

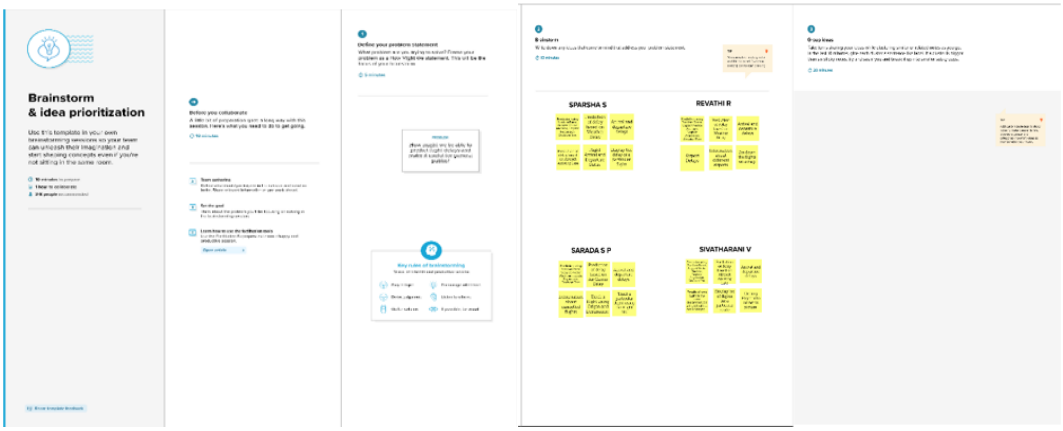


Fig.2

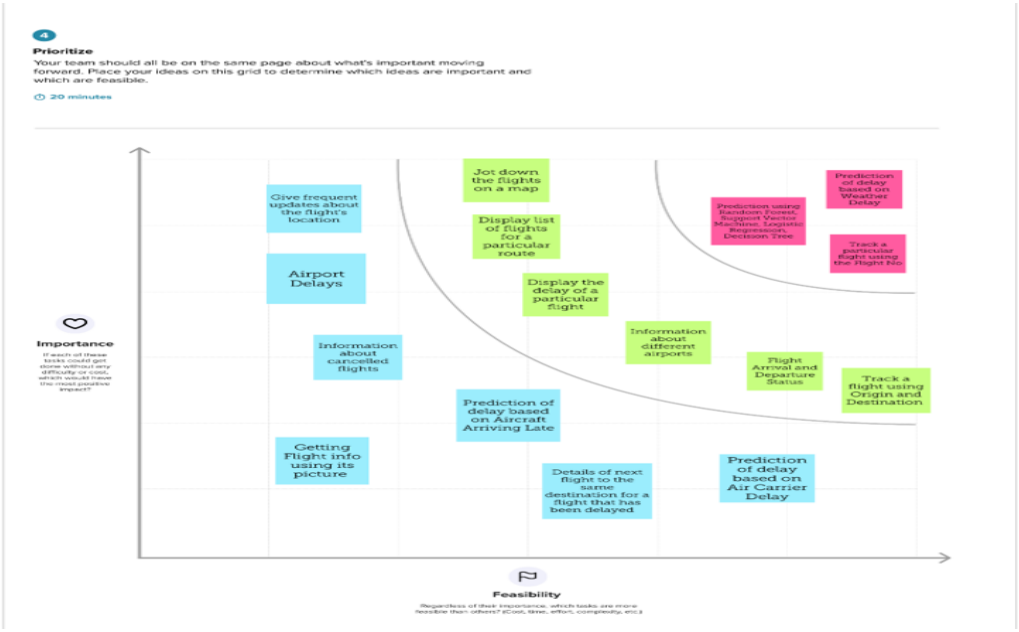


Fig.3

3.3 PROPOSED SOLUTION

S. No.	Parameter	Description
1	Problem Statement (Problem to be solved)	To predict flight delays usingML algorithm.
2	Idea / Solution description	● Predicting flight delays usingalgorithms such as Random Forest,Logistic Regression, Decision Tree and SupportVectorMachine. ● A user willbe notified of the bookedflight's location frequently. ● Givingan accurate delayprediction will help in bettercustomer service.

		● Cancellations willalso be notified. ● Multiple metrics like arrival/ departure delays, delays based on geographic areas are considered, making this solution more precise.
3	Novelty / Uniqueness	● Frequent updates on the flight's location and accurate ● Prediction of the delays. ● Gives the status of different airports too.
4	Social Impact / Customer Satisfaction	● Proper planning of trips. ● Reduction of mental pressure and stress. ● Prior information helps in avoiding loggerheads with otherpeople. ● Reduction of business losses.
5	Business Model (Revenue Model)	● This model can be used by all the people who travel viaflights. The appcan be accessed through any device. ● The existing solutions do notgive frequent updates to the customer directly. ● The ML algorithms to be used have accuracy between 87% -91%.

6	Scalability of the Solution	• The scalability of the solution is expanded for travellers allover the world,irrespective of their purpose for travelling. • This app can help customers to get updates of the flightof anypartof the flight. • This is also beneficial for all theairline authorities by reducingcomplaints and increasing customer satisfaction.

3.4 PROBLEM SOLUTION FIT

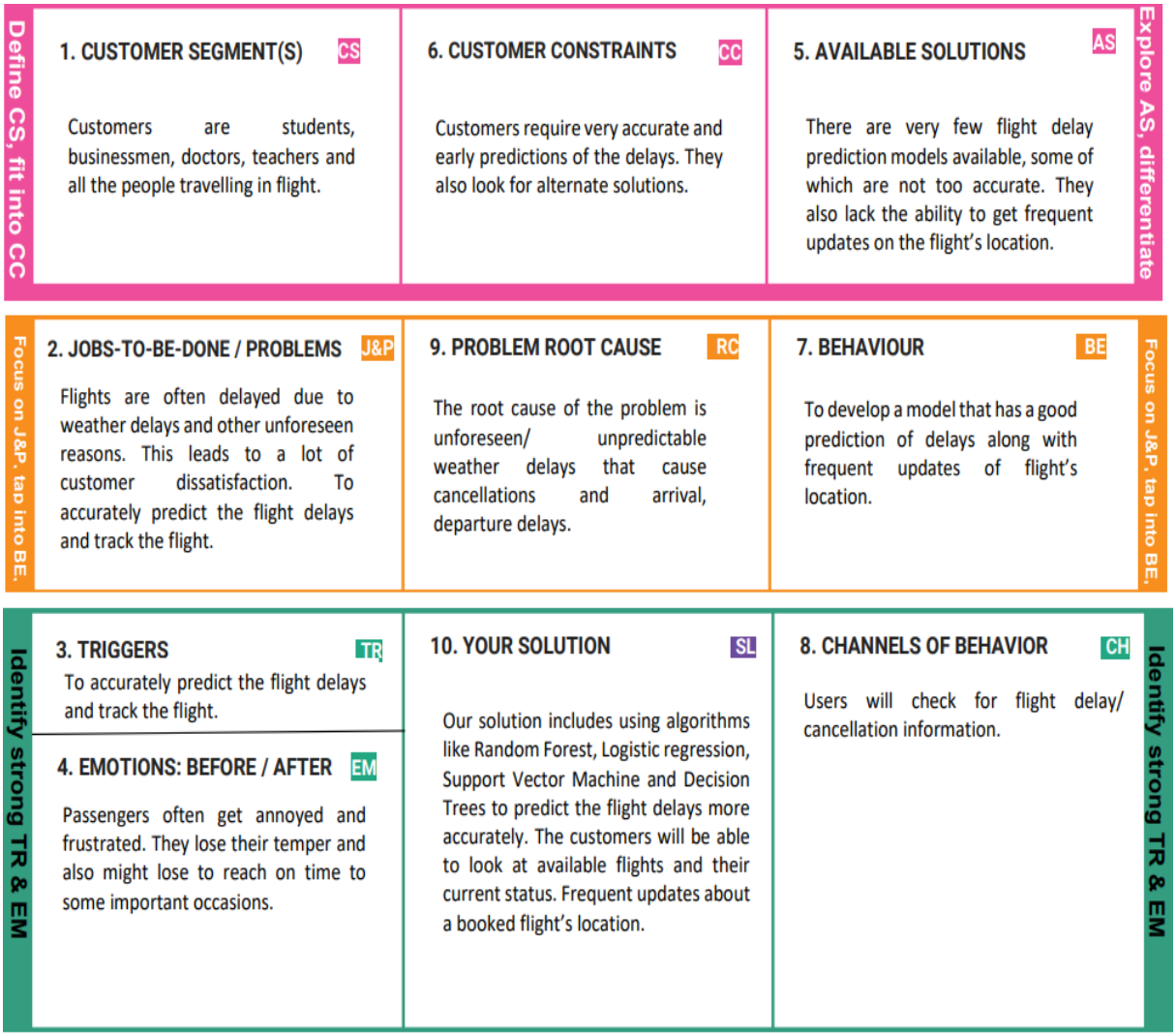


Fig.4

CHAPTER - 4

REQUIREMENTS 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 Registration through Phone number
FR-2	User Confirmation	Confirmation via mail Confirmation via OTP
FR-3	User Login	Login using Credentials
FR-4	Search	Get Flight Details
FR-5	GPS	Track flight
FR-6	Analysis	Fetch Dataset
FR-7	Prediction	Train and TestPredictive Models

4.2 NON-FUNCTIONAL REQUIREMENT

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Can be usedto get detailsof multiple airlines
NFR-2	Security	Two stepverification
NFR-3	Reliability	The flightdelays will be accurately predicted
NFR-4	Performance	The web app will take less time to load anddisplay the content

NFR-5	Availability	Can be accessed at anytime from anywhere Available for all irrespective of their purposes
NFR-6	Scalability	The solution can help in reducing the customerdissatisfaction and helpin better handling of delays/ cancellations

CHAPTER - 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

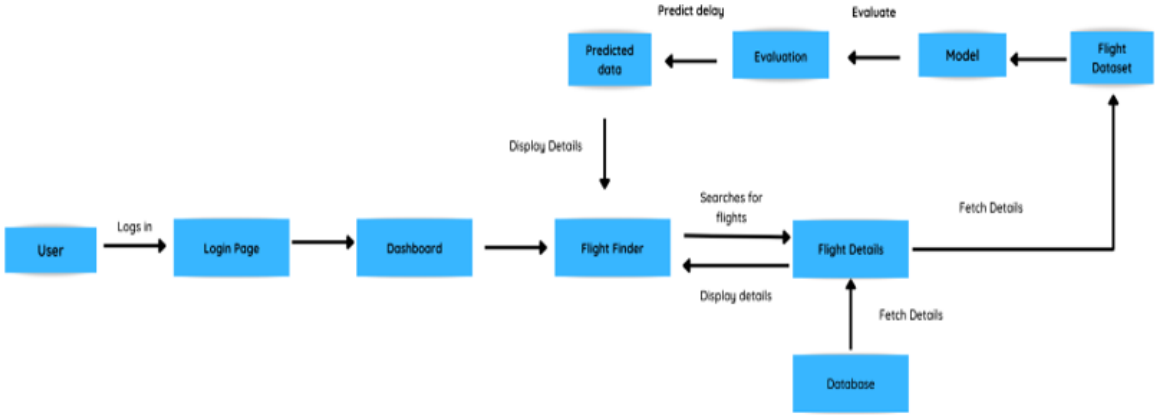


Fig.5

5.2 SOLUTION & TECHNICAL ARCHITECTURE

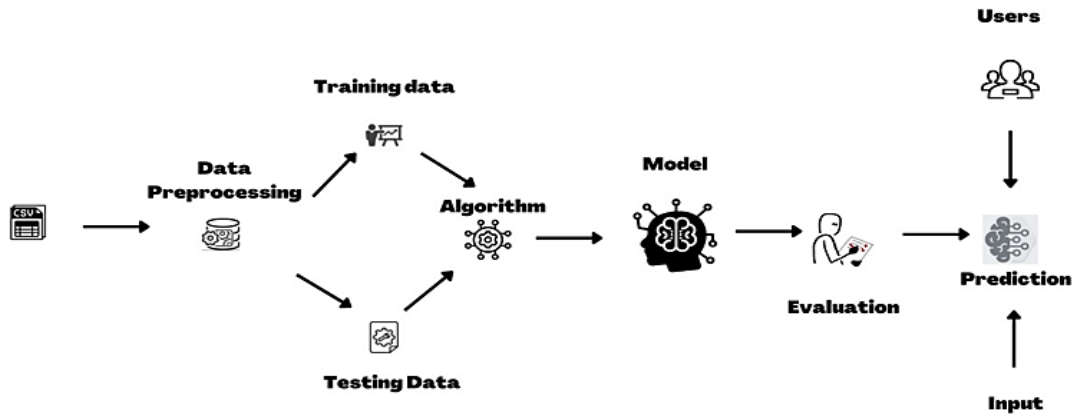


Fig.6 Solution architecture

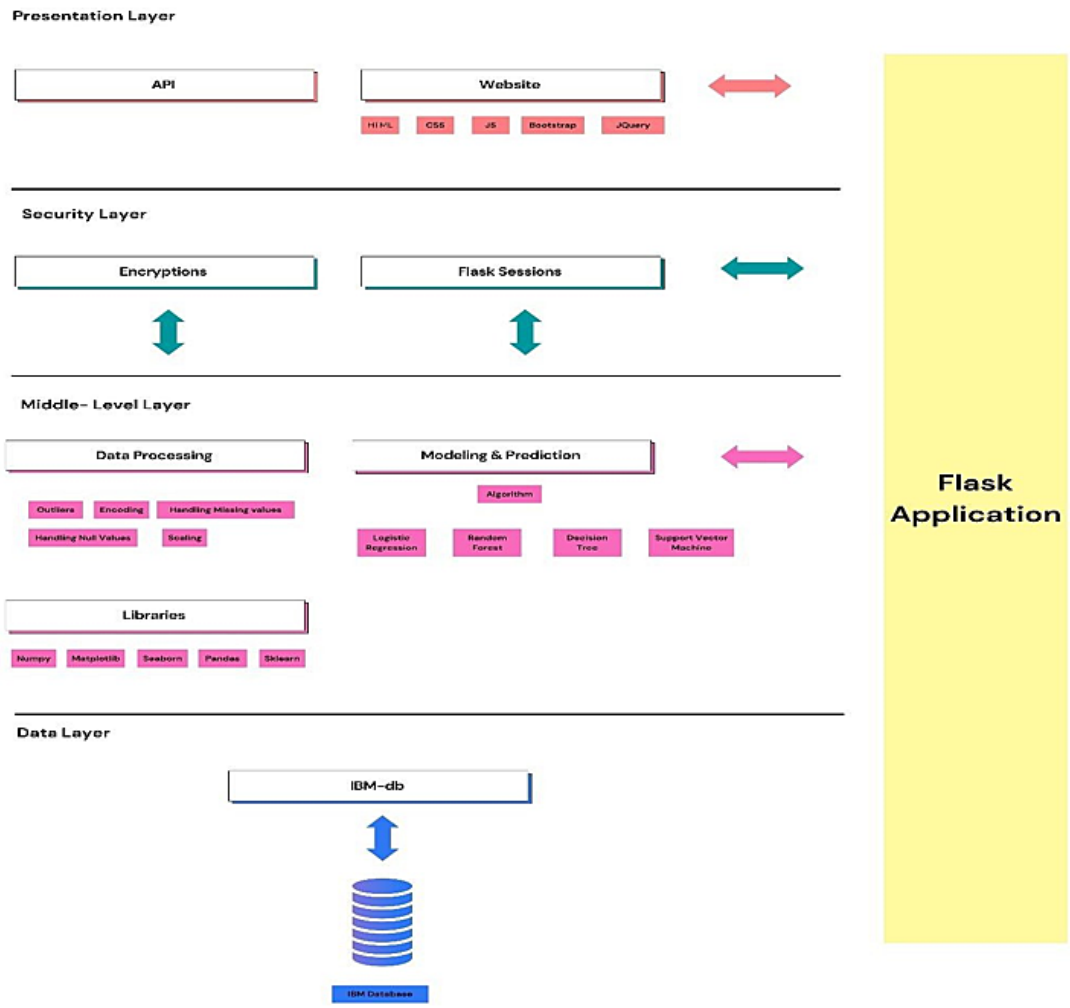


Fig.7 Technical architecture

5.3 USER STORIES

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password	High

Sprint-1		USN-2	As a user, I will receive confirmation email once I haveregistered for the application	High
Sprint-1		USN-3	As a user,I can register for the application through Google	Medium
Sprint-1	Login	USN-4	As a user, I can log into the application by entering email & password	High
Sprint-2	Predict	USN-5	As an admin, I will predict the delays using the dataset	High
Sprint-3	Result page	USN-6	Link for nearest top Restaurants, Hotels and inns, Mode of transportation	High
Sprint-4	Cloud deployment	USN-7	In cloud we deployed the model	Medium
Sprint-4		USN-8	Created Watson studio service, Model creation and project creation	High

CHAPTER - 6

PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Sprint - 1

We created Home page, Registration page, Login page and Profile page. In Registration page the user will Register by entering email, password and confirm password. The user can also Register through Google. OTP verification mail will sent to the user for confirmation. They can login and can view their profile page.

Sprint - 2

We created a search bar to find airlines and flights available at some airports. We analysed the dataset and predicted the flight delays.

Sprint - 3

We predicted the flight delays and integrated the flask.

Sprint - 4

We have added extra features in the result page where user can look for Restaurants, Hotels and Inns and Modes of transportation.

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application byentering	3	High	Sarada S P

			my email, password			
Sprint-1		USN-2	As a user, I will receive confirmation email once I haveregistered for the application	1	High	Sivatharani V
Sprint-1		USN-3	As a user,I can register for the applicationthrough Google	2	Medium	Sparsha S
Sprint-1	Login	USN-4	As a user, I can log intothe application by entering email & password	1	High	Revathi R
Sprint-2	Predict	USN-5	As an admin, I will predictthe delays	8	High	Sparsha S

Sprint-3	Result page	USN-6	Link for nearest top Restaurants, Hotels and inns, Mode of transportation	3	High	Sarada S P
Sprint-4	Cloud deployment	USN-7	In cloud we deployed the model	5	Medium	Sivatharani V
Sprint-4		USN-8	Created Watson studio service, Model creation and project creation	3	High	Revathi R

6.2 SPRINT DELIVERY SCHEDULE

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	8	4 Days	22 Oct 2022	25 Oct 2022	8	26 Oct 2022
Sprint-2	15	9 Days	26 Oct 2022	03 Nov 2022	15	04 Nov 2022

Sprint-3	19	9 Days	02 Nov 2022	10 Nov 2022	19	12 Nov 2022
Sprint-4	14	9 Days	04 Nov 2022	12 Nov 2022	14	15 Nov 2022

Velocity:

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Average velocity in sprint - 1

AV = 8/4 = 2

Average velocity in sprint - 2

AV = 15/9 = 1.6

Average velocity in sprint - 3

AV = 19/9 = 2.1

Average velocity in sprint - 4

AV = 14/9 = 1.5

6.3 Reports for JIRA

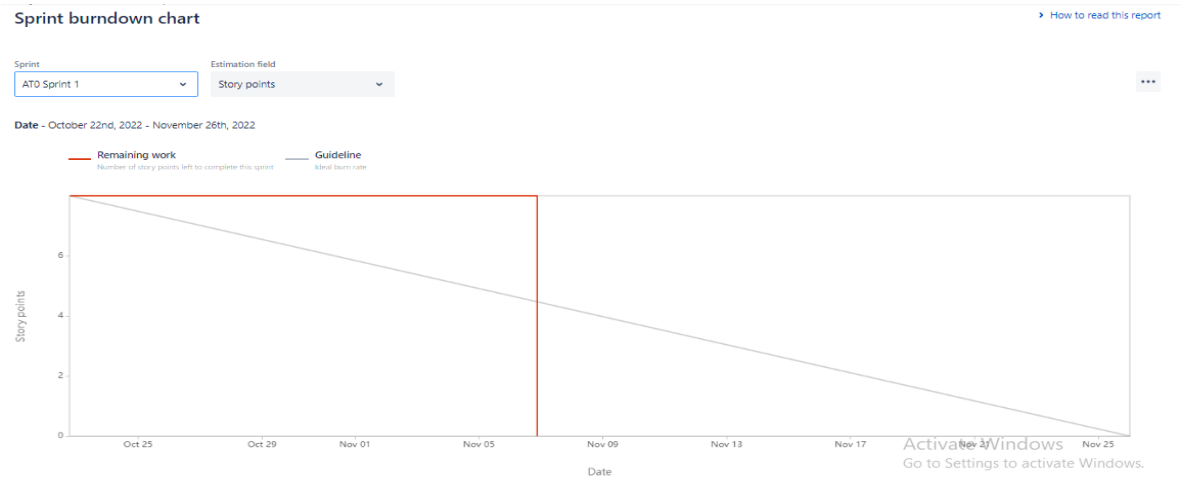


Fig.8 Burndown chart Sprint 1

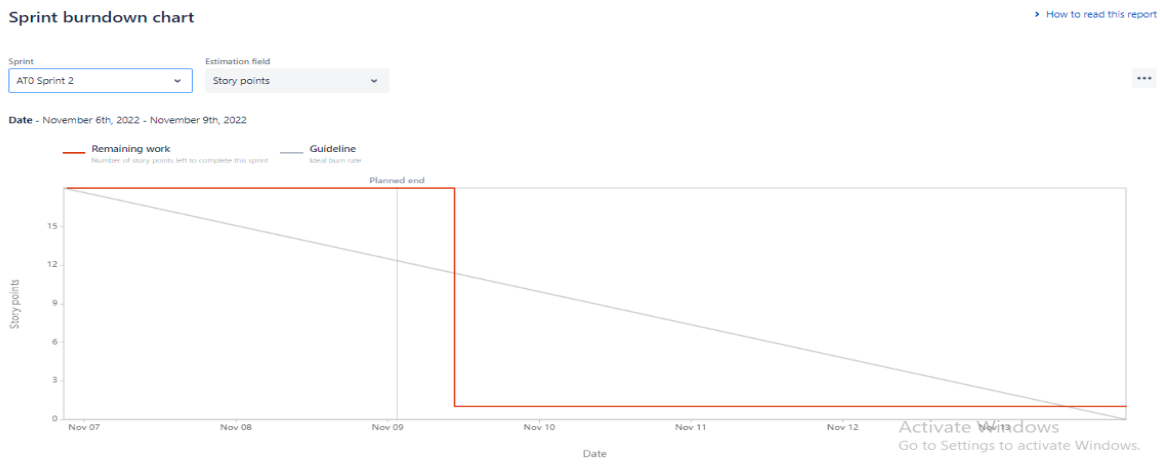


Fig.9 Burndown chart Sprint 2

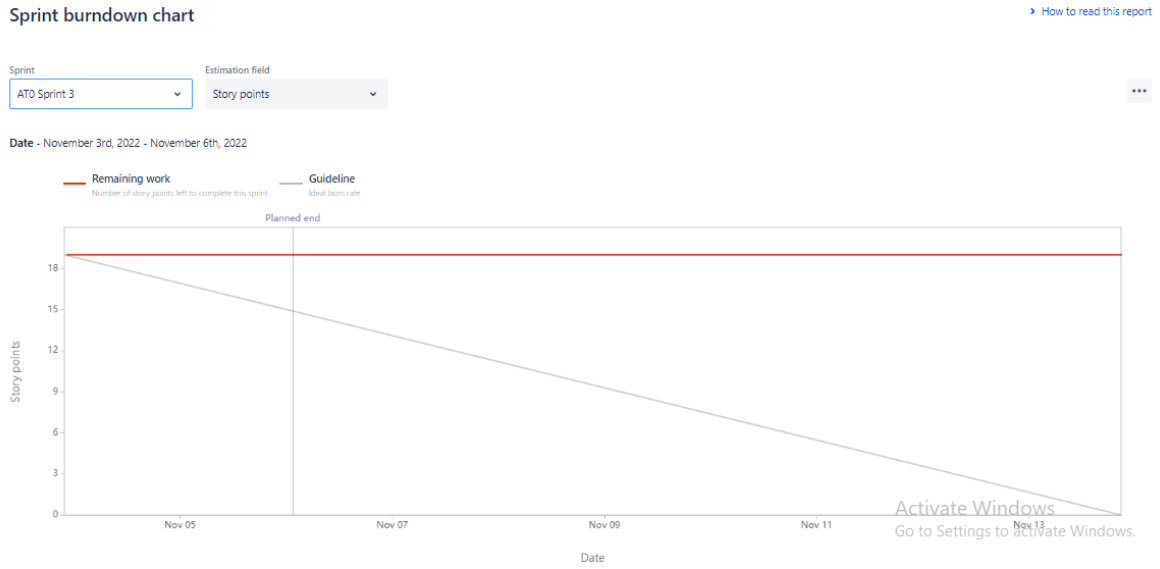


Fig.10 Burndown chart Sprint 3

Fig.11 Burndown chart Sprint 4

Fig.12 Road Map

Fig.13 Velocity chart

CHAPTER - 7

CODING AND SOLUTIONING

7.1 FEATURE 1

User can give their credentials for Registration purpose. OTP verification will send to the registered mail id. After the verification the account will be created. Mail id and password must be given for login, Google sign-in is also used for sign-in. If the user forget the password they can click the forget password option and it will be redirected to the recovery mail page.

7.2 FEATURE 2

We predicted whether the flight is delayed or not using the dataset. If the user gives Flight number, Month, Day of week, Origin, Destination, scheduled departure time, scheduled arrival time and Actual departure time, then it will predict the result and displays it. We have added extra feature where the user can view the nearby top Restaurants, Hotels and inns, Mode of transportation by selecting the destination airports.

CHAPTER - 8

TESTING

8.1 TEST CASES

Test case ID	Component	Test Scenario	Steps To Execute	Expected Result
1	Home Page		1. Click on sign up for login. 2. Email text box. 3. Password text box. 4. Login button. 5. Login with Google. 6. I am new here. 7. Forgot Password.	Application should show below UI elements: 1. Email text box 2. Password text box 3. Login button 4. Login with Google 5. I'm new here 6. Forgot Password
2	Home Page	Verify user is able to log into application with Valid credentials	1. Click on sign up for login. 2. Enter valid username/email in email text box. 3. Enter valid password in. password text box.	User should navigate to user account homepage

			4. Click on login button.	
3	Login Page	Verify user is able to log into application with invalid credentials	1. Click on sign up for login. 2. Enter Invalid username/email in email text box. 3. Enter valid password in the password text box. 4. Click on login button.	Application should show 'Incorrect email or password ' validation message.

4	Predication Page	To predict the flight is delayed or not	1. Click on prediction button. 2. Enter flight number. 3. Enter month. 4. Enter day of the month. 5. Enter day of the week. 6. Enter origin. 7. Enter destination. 8. Enter scheduled departure time. 9. Enter scheduled arrival time. 10. Enter actual departure time. 11. Click submit button.	Application should show the flight is delayed or not
5	Result page	It shows flight is delayed or not and shows the hotels, restaurants, and transportation facilities near	1. Enter all details in prediction page. 2. Click submit. 3. Show all hotels, restaurants, and transportation near to airports.	Application should show if the flight is delayed or not.

		the airports.		

8.2 USER ACCEPTANCE TESTING

8.2.1 Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issuesof the Flight Delay prediction model project at the time of the release to User Acceptance Testing (UAT).

8.2.2 Defect Analysis

This report showsthe 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	11	4	3	7	25
Duplicate	2	1	1	2	6
External	3	3	0	2	8
Fixed	10	5	3	18	38
Not Reproduced	1	1	0	0	2
Skipped	0	0	1	1	2

Won't Fix	0	5	3	0	8
Totals	27	19	11	30	87

8.2.3 Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Login	4	0	0	4
Sign-up	7	0	0	7
Security	2	0	0	2
Prediction	9	0	0	9
Final Report Output	2	0	0	2

CHAPTER - 9

RESULTS

9.1 PERFORMANCE METRICS

S.No.	Parameter	Values
1.	Metrics	**Classification Model:** Confusion Matrix – [1840,0,0,407] Accuracy Score- 100% Classification Report – 100%
2.	Tune the Model	Hyperparameter Tuning – 100% Validation Method – RandomizedSearchCV

SCREENSHOTS

METRICS

```
from sklearn.metrics import confusion_matrix
```

```
from sklearn import tree
fig = plt.figure(figsize =(80,20))
_ = tree.plot_tree(dtc,
                   feature_names=data.columns,
                   class_names=['No Delay', "Delay"],
                   filled=True)
```

Fig.14

```
In [46]: metrics.confusion_matrix(y_test, dt)
Out[46]: array([[1840,    0],
                [   0,  407]])

In [49]: metrics.classification_report(dt, y_test)
Out[49]: '              precision    recall  f1-score   support\n\n           0       1.00      1.00      1.00      1840\n           1       1.00
          1.00      1.00       407\n\n    accuracy                           1.00      2247\n   macro avg       1.00      1.00      1.00      2247\nw
         eighted avg       1.00      1.00      1.00      2247\n'
```

Deployment

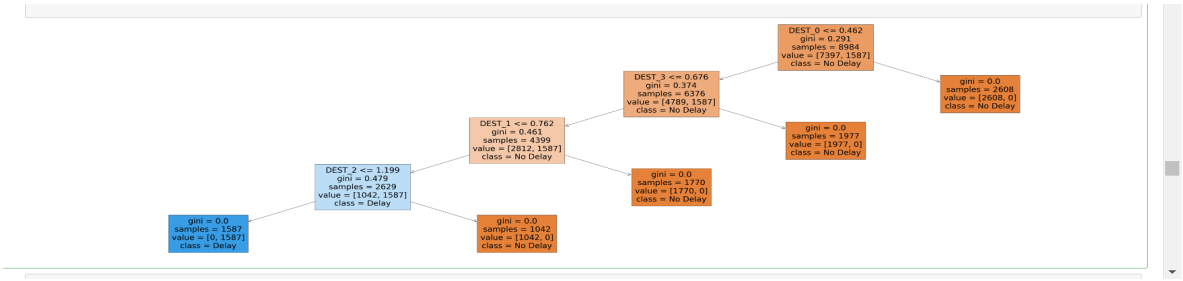
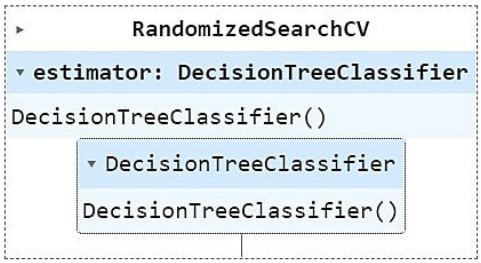


Fig.15

```
import matplotlib.pyplot as plt
import numpy
from sklearn import metrics

confusion_matrix = metrics.confusion_matrix(y_test, dt)

cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix = confusion_matrix, display_labels = [False, True])

cm_display.plot()
plt.show()
```

Fig.16

TUNING THE MODEL




```
DT_grid.best_estimator_
```

```
DecisionTreeClassifier(criterion='entropy', max_depth=15, max_features='log2',
                       min_samples_split=6)
```

Fig.17

```
from sklearn.model_selection import RandomizedSearchCV
```

```
parameters = {'max_depth' : (3,5,7,9,10,15,20,25)
              , 'criterion' : ('gini', 'entropy')
              , 'max_features' : ('auto', 'sqrt', 'log2')
              , 'min_samples_split' : (2,4,6)
             }
```

```
DT_grid  = RandomizedSearchCV(DecisionTreeClassifier(), param_distributions = parameters, cv = 5, verbose = True)
```

```
DT_grid.fit(x_train,y_train)
```

```
DT_grid.fit(x_train,y_train)
```

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
```

Fig.18

CHAPTER - 10

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- Flight delay is inevitable and it plays an important role in both profits and loss of the airlines. 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.
- A Flight Data Monitoring Program assists an operator to identify, quantify, assess and address operational risks.
- Simple delay models also can identify a small number of critical paths to be simulated in more detail and allow CAD tools to perform basic optimization and sizing of many circuits.

DISADVANTAGES

- The reasons for these delays vary a lot going from air congestion to weather conditions, mechanical problems, difficulties while boarding passengers, and simply the airlines inability to handle the demand given its capacity.
- It's difficult to predict all the delays.
- Sometimes it is difficult to know the weather condition.

CHAPTER - 11

CONCLUSION

Over the last twenty years, air travel has been increasingly preferred among travelers, mainly because of its speed and in some cases comfort. So here we try to solve those problems, We used flight dataset to predict whether the flight is delayed or not. The result shows that the highest values of accuracy 100% in decision tree. We have added extra feature restaurants, Hotels and Inns and Modes of transportation.

CHAPTER - 12

FUTURE SCOPE

In future we will track the flight's location and it will be displayed on the Map. Notification will be sent to the user for continous updation about flight's status. The flight details will be diplayed. If the user gives the origin and destination the available flights will be displayed. Although weather conditions are the major reasons for flight delay, other unprecedented events such as major calamities , natural or man-made can cause major delay in flight.

CHAPTER - 13

APPENDIX

13.1 SOURCE CODE

App.py

```
from errno import ENAMETOOLONG
from turtle import st
from flask import Flask, render_template, request, redirect, url_for,
session
import numpy as np
import pandas as pd
import pickle
import os
import requests
from markupsafe import escape
from flask_mail import Mail, Message
from random import randint
import ibm_db
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=764264db-
9824-4b7c-82df-
40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;POR
T=32536;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRoot
CA.crt;UID=kzy64909;PWD=ex8AdzTg57abUK25",",")
GOOGLE_CLIENT_ID = "340644155083-
hm83b3k5d7mbb0ps5u33ck7qkbder4uf.apps.googleusercontent.com"
GOOGLE_CLIENT_SECRET = "GOCSPX-
JDnPmqWt00uo5xaBgkRukzrhPqAl"
REDIRECT_URI = '/gentry/auth'
import json
```

```
# NOTE: you must manually set API_KEY below using information
retrieved from your IBM Cloud account.
API_KEY =
"mYBrvKJylOO4wCWoS_TesMMELMxEBSW9rQ1NzP0Wn-se"
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.secret_key = 'flightdelayflyhigh2022'
app.config["MAIL_SERVER"] = 'smtp.gmail.com'
app.config["MAIL_PORT"] = 465
app.config["MAIL_USERNAME"] = '2k19cse104@kiot.ac.in'
app.config['MAIL_PASSWORD'] = 'kiotcse@19'
app.config['MAIL_USE_TLS'] = False
app.config['MAIL_USE_SSL'] = True
mail = Mail(app)
@app.route('/')
def home():
  return render_template('home_page.html')
@app.route('/signup')
def signup():
  return render_template('Sign_Up.html')
@app.route('/prediction',methods=["POST","GET"])
def prediction():
```

```
    if request.method == 'POST':
      global name
      global month
      global day_of_month
      global day_of_week
      global origin
      global ans
      name = request.form['name']
      month = request.form['month']
      day_of_month = request.form['day_of_month']
      day_of_week = request.form['day_of_week']
      origin = request.form['origin']
      ans = 'No delay'
      print(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  == 'ALT'):
        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 == 'ALT'):
      destination1, destination2, destination3, destination4, destination5 =
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, day_of_month, day_of_week,dept15, arrtime,
origin1,origin2,origin3,origin4,origin5,destination1,destination2,destina
tion3,destination4,destination5]]
    print(total)
    payload_scoring = {"input_data": [{"field": ['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/739a6f52-043e-49ec-b5ee-
d9b2d5a4e6b6/predictions?version=2022-11-19', json=payload_scoring,
    headers={'Authorization': 'Bearer ' + mltoken})
    y_pred = response_scoring.json()
    print(y_pred)
```

```
    pred_result = y_pred['predictions'][0]['values'][0][0]
    print(pred_result)
    if(pred_result == 0):
      ans = 'Yippeeee! The flight will be on time.'
    else:
      ans = 'Sorry! Your flight will be delayed.'
    print(ans)
    return render_template('result.html', prediction = ans)
  return render_template('Prediction.html')
@app.route('/gentry')
def gentry():
    return
redirect(f"https://accounts.google.com/o/oauth2/v2/auth?scope=https://
www.googleapis.com/auth/userinfo.profile&access_type=offline&inclu
de_granted_scopes=true&response_type=code&redirect_uri=http://127.
0.0.1:5000/gentry/auth&client_id={GOOGLE_CLIENT_ID}")
@app.route('/gentry/auth')
def gentry_auth():
    r = requests.post("https://oauth2.googleapis.com/token",
    data={
        "client_id": GOOGLE_CLIENT_ID,
        "client_secret": GOOGLE_CLIENT_SECRET,
        "code": request.args.get("code"),
        "grant_type": "authorization_code",
        "redirect_uri": "http://127.0.0.1:5000/gentry/auth"
    })
r=requests.get(f'https://www.googleapis.com/oauth2/v2/userinfo?access
_token={r.json()["access_token"]}').json()
```

```
print("============================================
======")
print(r)
print("============================================
======")
 return redirect(url_for('home'))
@app.route('/recoverymail')
def recoverymail():
  return render_template('recoverymail.html')
@app.route('/sendpassword', methods = ['POST'])
def sendpassword():
  if request.method == 'POST':
    email = request.form.get('email')
    sql = "SELECT * FROM Profiles WHERE Email =?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,email)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    print("-------------------------------")
    print(account)
    if not account:
      return render_template('recoverymail.html', msg="You are not
signed up")
    else:
      password = account['PASSWORD']
      msg = Message(subject='OTP',
sender='2k19cse104@kiot.ac.in',recipients=[email])
      msg.body = "Your Password is: " + password
```

```
        mail.send(msg)
        return render_template('Sign_Up.html')
@app.route('/register',methods = ['POST', 'GET'])
def register():
  if request.method == 'POST':
    global Name
    global Email
    global Password
    global ConfirmPassword
    Name = request.form['Name']
    Email = request.form['Email']
    Password = request.form['Password']
    ConfirmPassword = request.form['ConfirmPassword']
    sql = "SELECT * FROM Profiles WHERE name =?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,Name)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    if account:
      return render_template('Sign_Up.html', msg="You are already a
member, please login using your details")
    else:
      global otp
      otp = randint(000000, 999999)
      email = Email
      msg = Message(subject='OTP', sender='2k19cse104@kiot.ac.in',
                    recipients=[email])
      msg.body = "You have succesfully registered!\nThe OTP for
```

```
verification is\n\t" + \
            str(otp)
    mail.send(msg)
    return render_template('otp_verification.html', resendmsg="OTP has
been resent")
  email = request.form['Email']
  msg = Message(subject='OTP', sender='2k19cse104@kiot.ac.in',
              recipients=[email])
  msg.body = "You have succesfully registered !\nThe OTP for
verification is\n\t" + \
    str(otp)
  mail.send(msg)
  return render_template('otp_verification.html')
@app.route('/validate', methods=['POST'])
def validate():
   global otp
   user_otp = request.form['otp']
   if otp == int(user_otp):
     insert_sql = "INSERT INTO Profiles VALUES (?,?,?,?)"
     prep_stmt = ibm_db.prepare(conn, insert_sql)
     ibm_db.bind_param(prep_stmt, 1, Name)
     ibm_db.bind_param(prep_stmt, 2, Email)
     ibm_db.bind_param(prep_stmt, 3, Password)
     ibm_db.bind_param(prep_stmt, 4, ConfirmPassword)
     ibm_db.execute(prep_stmt)
     return render_template('Sign_Up.html')
   else:
      return render_template('otp_verification.html', msg="OTP is
```

```
invalid. Please enter a valid OTP")
@app.route("/login", methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        Email = request.form.get('Email')
        Password = request.form.get('Password')
        sql = "SELECT * FROM user WHERE Email =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, Email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        if account:
            if (Password == str(account['PASS']).strip()):
                return render_template('home_page.html')
        else:
         return render_template('Sign_Up.html', msg="Password is
invalid")
        else:
            return render_template('Sign_Up.html', msg="Email is invalid")
    else:
        return render_template('Sign_Up.html')
if(_name_ ==  '_main_'):
  app.run(host = '0.0.0.0', port =5000)
```

Prediction

```
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.metrics import accuracy_score
import sklearn.metrics as metrics
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3
def _iter_(self): return 0
# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage.
It includes your credentials.
# You might want to remove those credentials before you share the
notebook.
cos_client = ibm_boto3.client(service_name='s3',

ibm_api_key_id='3k1NmXn2fW7Wruf4t8qQ85rXpg4Aq4zv5rNafwhR
PyIt',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-
storage.appdomain.cloud')
```

```
bucket = 'flightdelayprediction-donotdelete-pr-yswofidmzwspa4'
object_key = 'IBM Dataset.csv'
body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing _iter_ method, so pandas accepts body as file-like object
if not hasattr(body, "_iter"): body.iter_ = types.MethodType( _iter_,
body )
data = pd.read_csv(body)
data.head()
```

Data Analysis:

```
data.info()
data.shape
```

Handling Missing Values:

```
data.isnull().sum()
data.dropna()
data['ARR_TIME'] =
np.where(data['ARR_TIME'].isnull(),data['CRS_ARR_TIME'],
data['ARR_TIME'])
data['DEP_DELAY'] = np.where(data['DEP_DELAY'].isnull(),0,
data['DEP_DELAY'])
```

Data Visualisation:

```
sns.heatmap(data.corr())
sns.pairplot(data)
```

Encoding:

```
le = LabelEncoder()
data['DEST'] = le.fit_transform(data['DEST'])
data['ORIGIN'] = le.fit_transform(data['ORIGIN'])
data.head(5)
```

Splitting the dataset into X and Y

```
data = pd.get_dummies(data, columns = ['ORIGIN', 'DEST'])
data.head()
x = data.iloc[:, 0:16].values
y = data.iloc[:, 16:17].values
ohe = OneHotEncoder()
z = ohe.fit_transform(x[:,4:5]).toarray()
t = ohe.fit_transform(x[:,5:6]).toarray()
x.shape
y.shape
```

Splitting into Train and Test

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2,
random_state = 0)
x_train.shape
y_train.shape
x_test.shape
y_test.shape
from sklearn.preprocessing import StandardScaler
ss = StandardScaler()
x_train = ss.fit_transform(x_train)
x_test = ss.transform(x_test)
```

Decision Tree

```
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier(random_state = 0)
dtc.fit(x_train, y_train)
dt = dtc.predict(x_test)
dt
from sklearn.metrics import accuracy_score
acc = accuracy_score(y_test, dt)
```

```
acc
def guid_from_space_name(client, space_name):
    space = client.spaces.get_details()
    return(next(item for item in space['resources'] if
item['entity']["name"] == space_name)['metadata']['id'])
space_uid = guid_from_space_name(client, 'Model Deployments')
print("Space UID = " + space_uid)
client.set.default_space(space_uid)
model_name = "Decision Tree"
deployment_name= "Model Deployments"
model = dtc
client.software_specifications.list()
software_spec_uid =
client.software_specifications.get_uid_by_name("runtime-22.1-py3.9")
software_spec_uid
model_props ={
    client.repository.ModelMetaNames.NAME : model_name,
    client.repository.ModelMetaNames.TYPE : "scikit-learn_1.0",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID :
software_spec_uid
}
model_details = client.repository.store_model(
    model = model,
    meta_props =model_props,
    training_data = x_train,
    training_target =  y_train
)
model_details
```

RESULT PAGE

```
<!DOCTYPE html>
<html>
 <head>
  <title>Prediction</title>
  <style>
  body{
   font-family: Arial, Helvetica, sans-serif;
   background-repeat: no-repeat;
   background-size: 100% 100%;
   background-attachment: fixed;
  }
  p{
   display: block;
   font-size: 60px;
  }
  .card-front{
   position: absolute;
   width:100%;
   height: 100%;
   padding: 50px;
   box-sizing: border-box;
   border-radius: 10px 10px 10px 10px;
  }
.row {
 display: flex;
 flex-wrap: wrap;
 width: 100%;
```

```
  justify-content: center;
  padding:20px;
 }
 .column {
  display: flex;
  flex-basis: 50%;
  flex: 1;
  text-align: center;
  font-weight: bold;
  padding-left: 0px;
 }
 .container {
  position: absolute;
  background-color: rgb(209, 217, 219);
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.5);
  border-radius: 3px;
  padding-left: 0px;
 }
 a{
  text-decoration: none;
  font-size: 20;
  float: none;
  color: black;
  padding: 3px 5px;
  display: block;
  text-align: center;
 }
 a:hover {
```

```
  background-color: #ddd;
  display: block;
}
.logo{
  display:flex;
  flex-direction:right;
  justify-content: space-between;
  padding-top: 0px;
  padding-right:30px;
  height: 70px;
  width: 70px;
}
.container p{
  font-size: 20px;
  font-weight: bold;
}
  </style>
  </head>
    <body background = “{{url_for(‘static’,filename =
‘resultcover.png’)}}”>
      <a href= “{{url_for(‘home’)}}” ><img
src=”{{url_for(‘static’,filename = ‘newlog.png’)}}” class=”logo”
align=”right” title=”FlyHigh”></a>
      <div class=”card-front”>
        <p>{{ prediction }}</p>
      <div class=”row”>
        <div class=”column”>
          <div class=”container”>
```

```
            <p>RESTAURANTS</p>
            <h5>Select your destination to know good restaurants near your
airport…</h5>
            <a href=”https://www.opentable.com/landmark/restaurants-near-
msp-airport”>MSP</a><br>
            <a href=”https://www.opentable.com/landmark/restaurants-near-
dtw-airport”>DTW</a><br>
            <a href=”https://www.opentable.com/landmark/restaurants-near-
jfk-airport”>JFK</a><br>
            <a href=”https://www.opentable.com/landmark/restaurants-near-
seattle-tacoma-international-airport”>SEA</a><br>
            <a href=”https://www.opentable.com/landmark/restaurants-near-
hartsfield-jackson-atlanta-international-airport”>ALT</a><br>
          </div>
        </div>
        <div class=”column”>
          <div class=”container”>
            <p>HOTELS AND INNS</p>
            <h5>Select your destination to know the nearest hotels near your
airport…</h5>
            <a href=”https://www.tripadvisor.com/HotelsNear-g43323-
qMSP-Minneapolis_Minnesota.html”>MSP</a><br>
            <a href=”https://www.tripadvisor.com/HotelsNear-g42139-
qDTW-Detroit_Michigan.html”>DTW</a><br>
            <a href=”https://www.tripadvisor.com/HotelsNear-g60763-
qJFK-New_York_City_New_York.html”>JFK</a><br>
            <a href=”https://www.tripadvisor.com/HotelsNear-g60878-
qSEA-Seattle_Washington.html”>SEA</a><br>
```

```
            <a href=”https://www.tripadvisor.com/HotelsNear-g60898-
qATL-Atlanta_Georgia.html”>ALT</a><br>
          </div>
        </div>
        <div class=”column”>
          <div class=”container”>
            <p>TRANSPORTATION</p>
            <h5>Select your destination to know about transport facility
near your airport…</h5>
            <a href=”https://www.mspairport.com/directions/ground-
transportation”>MSP</a><br>
            <a href=”https://www.jayride.com/airport-transfer/united-
states/detroit-airport-dtw”>DTW</a><br>
            <a href=”https://www.jfkairport.com/to-from-airport/taxi-car-
and-van-service”>JFK</a><br>
            <a href=”https://www.portseattle.org/sea-tac/ground-
transportation”>SEA</a><br>
            <a href=”https://www.atlanta-airport-cab.com/”>ALT</a><br>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

13.2 GitHub & Project Demo Link

https://www.youtube.com/embed/Q-ASAyIo3ck

14 REFERENCE

[1] Flight Delay Prediction based on AviationBig Data and Machine Learning(Author: Rahul Garg et.al.,2022)

[2] Flight Delay Prediction (Author: BhuvanBhatia, 2019)

[3] Assessing Strategic Flight Schedules at an Airport using Machine Learning based Flight Delay And Cancellation Prediction (Author: Miguel Lambelho et.al., 2020)

[4] Flight delay prediction based on deep learning and Levenberg-Marquart algorithm(Author: M F Yazdi et.al., 2020)

[5] Predicting flightdelays using data from US Domestic flights(2019)