DEVELOPING A FLIGHT DELAY PREDICTION USING MACHINE LEARNING

TEAM ID: PNT2022TMID43039

IBM NALAIYA THIRAN

Submitted by

JAYADHARANI B (713119205002)

JEEVALAKSHMAN B (713119205003)

SANTHAVALIYAN M (713119205007)

SARUTHI M (713119205008)

In partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

SRI SAI RANGANATHAN ENGINEERING COLLEGE

COIMBATORE-641 109

ANNA UNIVERSITY: CHENNAI 600 025

TABLE OF CONTENT

ABSTRACT

Flight delays hurt airlines, airports, and passengers. Their prediction is crucial during the decision-making process for all players of commercial aviation. Moreover, the development of accurate prediction models for flight delays became cumbersome due to the complexity of air transportation system, the number of methods for prediction, and the deluge of flight data. In this context, this paper presents a thorough literature review of approaches used to build flight delay prediction models from the Data Science perspective. We propose a taxonomy and summarize the initiatives used to address the flight delay prediction problem, according to scope, data, and computational methods, giving particular attention to an increased usage of machine learning methods. Besides, we also present a timeline of significant works that depicts relationships between flight delay prediction problems and research trends to address them.

Keywords: Airlines, Flight delay, Prediction

1.INTRODUCTION

1.1 Project overview

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.

1.2 Purpose

To solve this issue, at currently predicting the flight delay allows passengers to be well prepared for the deterrent caused to their journey and enables airlines to respond to the potential causes of the flight delays in advance to diminish the negative impact.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM:

The existing predicts only the historical data of a single airport, without considering the interaction between airports in the airport cluster, as well as the historical data of multi-airport terminal airspace, which is a future research direction

2.2 REFERENCES:

1. **TITLE:** Study of flight departure delay and casual factor using spatial analysis**.**

   AUTHOR: shaowu cheng, yapingzh ang, siqi Hao,

   ABSTRACT:

   Analysis of flight delay and causal factors is crucial in maintaining airspace efficiency and safety. However, delay samples are not independent since they always show a certain aggregation pattern. Therefore, this study develops a novel spatial analysis approach to explore the delay and causal factors which is able to take dependence and the possible problem involved including error correlation and variable lag effect of causal factors on delay into account. The study first explores the delay aggregation pattern by measuring and quantifying the spatial dependence of delay. The spatial error model (SEM) and spatial lag model (SLM) are then established to solve the error correlation and the variable lag effect, respectively. Results show that the SEM and SLM achieve better fit than ordinary least square (OLS) regression, which indicates the effectiveness of considering dependence by employing spatial analysis. Moreover, the outcomes suggest that, aside from the well-known weather and flow control factors, delay-reduction strategies also need to pay more attention to reducing the impact of delay at the previous airport

2. **TITLE:** flight delay forecasting and analysis of direct and indirect factors

AUTHOR: Fujun wang, jun Bi, Dongfan Xie, Xiaomei zhao

ABSTRACT:

The accurate prediction of flight delays is of great significance to airports, airlines and passengers. This paper presents a causal flight delay prediction model developed for a single airport. A long short-term memory network of delay prediction with an attention mechanism (LSTM-AM) is established to predict flight delays and analyse their primary causes. In this model, the direct and indirect factors related to delays are comprehensively considered. LSTM-AM can focus on input data combined with the attention vector to capture the critical time points, which can make the prediction more accurate. The model's performance is verified by actual operational data of Beijing International Airport, one of the busiest airports in the world. Experimental results show that LSTM-AM has better prediction accuracy than baseline algorithms such as some machine learning methods and deep learning methods. The mean absolute error of LSTM-AM is about 8.15 min on the test dataset. The study found that using the predicted results of this paper to release delayed information in advance can effectively alleviate the nervousness of passengers. The critical time point captured by LSTM-AM combined with runway and apron flow control can reduce or eliminate delays of one flight.

2.3 PROBLEM STATEMENT DEFINITION

- Due to poor weather condition, some of technical problems occurred in aircraft leads the flight delay.

- so, the travelers hates flying. Due to this problem the air travelers count will decreased day by day.

- We need to fix the problem to improve airline operations and passenger satisfaction, which will result in a positive impact on the economy.

3. IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS:

THINK AND FEEL:

- Passengers can’t want to plan for next trip due to flight delays.
- Passengers feel like " Why did these kinds of things happen for us?"
- They think "when will this problem and take off fight?"

WHAT DO THEY HEAR:

- Passengers hear some roomers which may be positive or negative around them.
- Airport authorities and airline hear like "Passengers are very anger and vague"
- Use different algorithms to predict delay of flight.

WHAT DO THEY SEE:

- Passengers feel very confused surrounding about delaying fight.
- Increasing operational profitability through better arrival flight prediction.

WHAT THEY SAY AD DO:

- Assign a proper schedule for arrival and departing of flight.

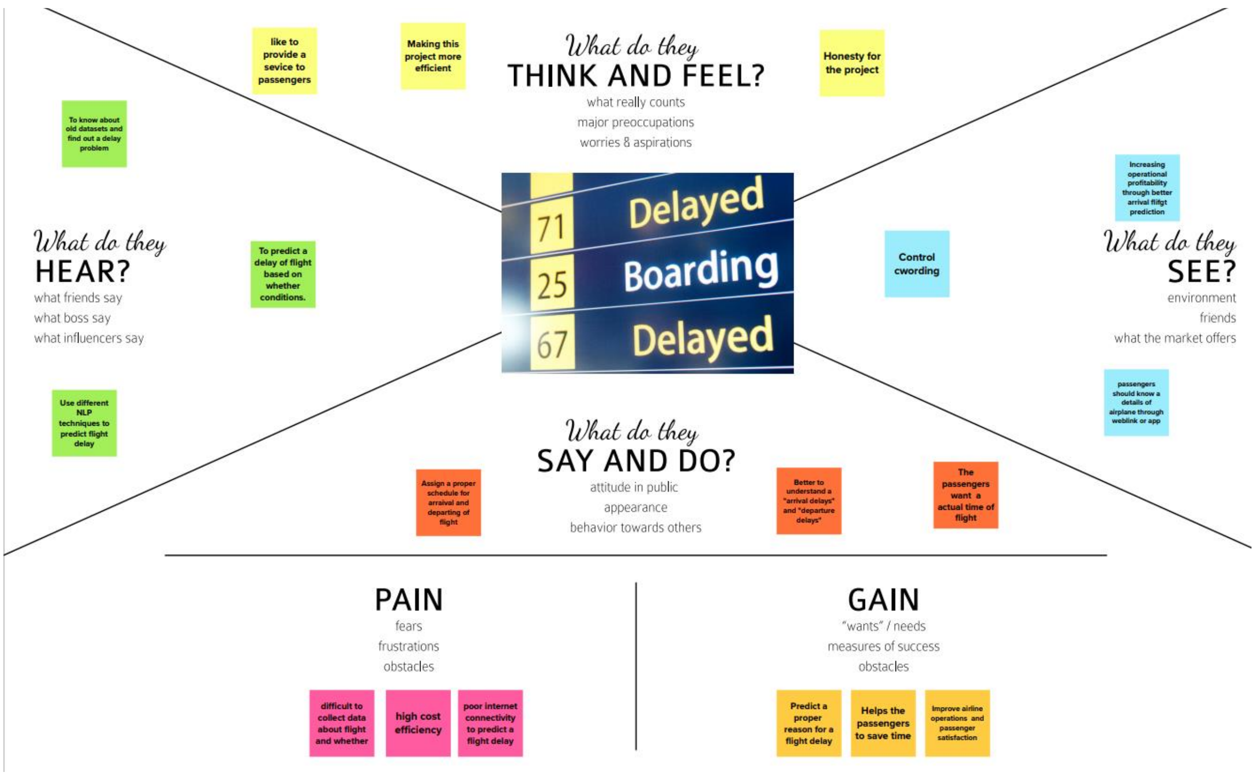
- ➢ Airport authorities try to find out the solution for that problem.

PAIN:

- ➢ Passengers worry about their planned works & schedules.
- ➢ Passengers feels frustrated about the long waiting in that environment.

GAIN:

- ➢ If delay is known earlier, passengers can reschedule their works and find alternate ways to reach their destination.
- ➢ They can reduce their financial loses and save their company's reputation on society.



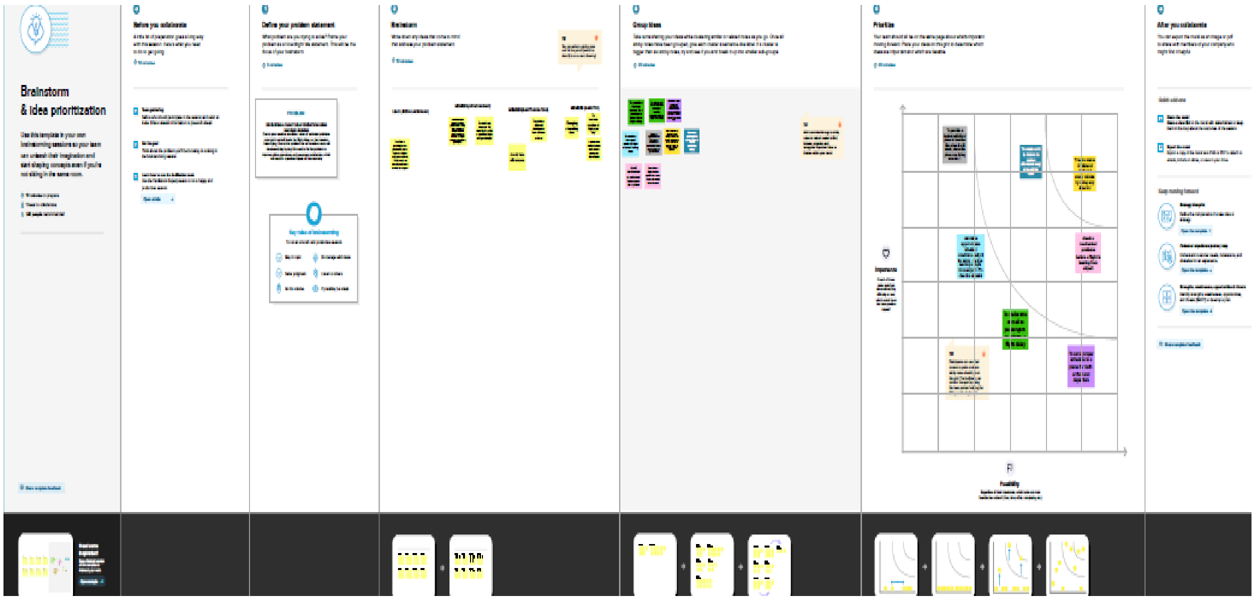
3.2. IDEATION AND BRAINSTROMING:

PROCESS:

- In our brainstorming we will collaborate our team members and enter our problem statement.
- Each member of our team gives nine ideas to solving the problems.
- Separate the similar ideas and graph it properly.

PURPOSE:

- To find innovative solution to problem
- To leverage creativity and motivate to higher plateau of thinking
- Create the opportunity for expression of uncultivated ideas
- To draw from the diversity of new skills.



3.3. PROPOSED SOLUTION:

By building prediction model that give prediction of flight delay using machine learning algorithms (Random Forest classifier, decision tree classifier, Linear regression, Logistic regression) which gives the best accuracy and less error. The prediction provides the indication of flight delay by identifying and solving issues or take other precaution steps to avoid economic losses, tension etc.

S. No.	Parameter	Description
1	Problem Statement (Problem to besolved)	**Problem:** Due to poor weather condition, some of technical problems occurred in aircraft leads the fight delay. so, the travelers hates flying. Due to this problem the air travelers count will decreased day by day. We need to fix the problem to improve airline operations and passenger satisfaction, which will result in a positive impact on the economy **Solution:** ➢ **By controlling a mechanical issue occurred in flight and find a daily whether condition. Fast connecting of passengers and bags.**
2	Idea / Solution description	**Idea:** ➢ Collect the Passengers flight on-time performance data, pre-process the collected data, and apply some learning algorithms with data science to predict a delay of flight.
3	Novelty / Uniqueness	**Uniqueness**: ➢ To collect a data of flight and whether conditions to train our model to predicta outcome(delays**)**
4	Social Impact / Customer Satisfaction	**Customer Satisfaction** Customer should able to go at correct destination at his targeted time.
5	Business Model (Revenue Model)	➢ Application ➢ Website
6	Scalability of the Solution	By using this type of application or a website we should know about a flight delays. Add extra features to our traveler's home page to know a details about our flight and where the flight is being fly and when we reach a destination.

3.3PROBLEM SOLUTION FIT:

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e., working parents of 0-5 y.o. kids

*Domestic and international passengers or a traveler
*Airport maintaining companies

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e., spending power, budget, no cash, network connection, available devices.

*Less influence
*Data collection
*some solutions need high budget
*Out of capacity for some of passengers
*The known solutions will quit complex

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem?

or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e., pen and paper is an alternative to digital notetaking

*Predefined plans if any issue will occur
*Change their schedule
*backup officers, flight, infrastructure of airport
*Be ready with precautions

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

* change their schedules
*Unable to attend important meetings it will spoiling the image of airlines where they will be cancelled or delayed
*Unnecessary tensions, crowding in airport

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

*Air traffic
*Whether condition
*Technical fault
*Medical emergency
*previous flight delay

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?

i.e., directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

*Deep thinking about why the problem will be occur and find a correct reason
*Need to think about what we will do next to avoid this problem (find a solution)

Focus on J&P, tap into BE, understand RC

IDENTIFY STRONG TR & EM

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

*Economic losses
*Unable to attend the important meeting at correct time
*Frustration created by delay of flights

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

* At initial stage they are well motivated, energetic and pleasant mind to reach a destination and do their planned jobs
* After customers facing a problem, they are frustrated

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

*Build a model using various machine learning algorithms to predict a delay of flight which gives the best accuracy. This prediction helps us to know the flight delay earlier.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

*Through online the customers try to contact via helpline, customer care number to inform airline officer

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

*Avoid violence and try to know a details of flight delay with his neighbors and find a proper solution

IDENTIFY STRONG TR & EM

4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS:

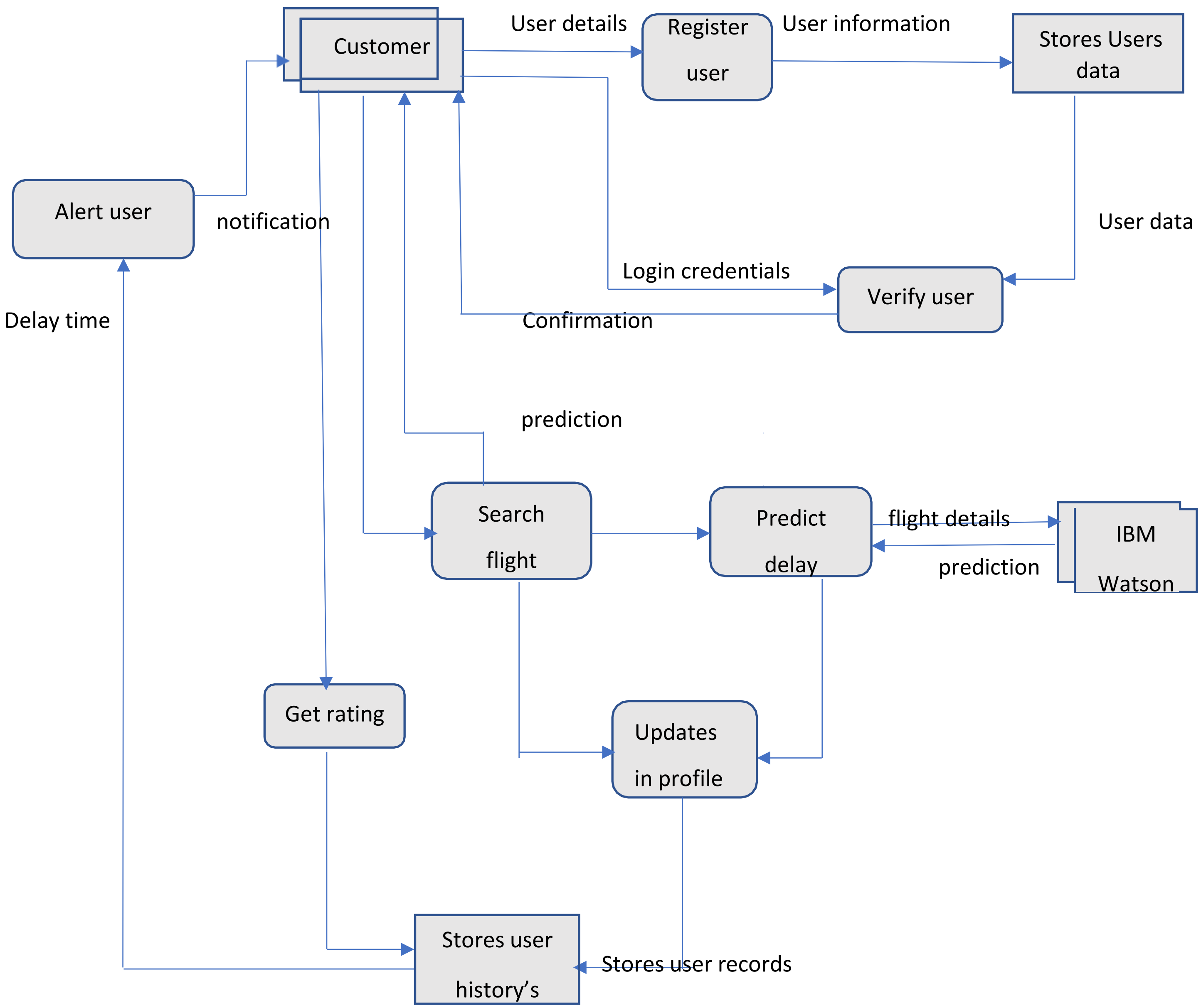
FR NO.	Functional requirements	Sub requirement(story/sub-task)
FR-1	User Registration	Registration through a form or Gmail or username password
FR-2	User Confirmation	Confirmation via mobile OTP or Gmail
FR-3	Log In	Login with given user credentials like username password
FR-4	Support	Asking queries or contact with customer support team
FR-5	Prediction of delays	Requesting to enter a detail of flight (predict delays in backend pre-programmed model and shows results)
FR-6	Notify to User	Notify users to know about flight arrival/ departure before 1 hour
FR-7	Get Feedback	Get feedback and ratings from user
FR-8	Log out	Logout from the application

4.2 NON-FUNCTIONAL REQUIREMENTS:

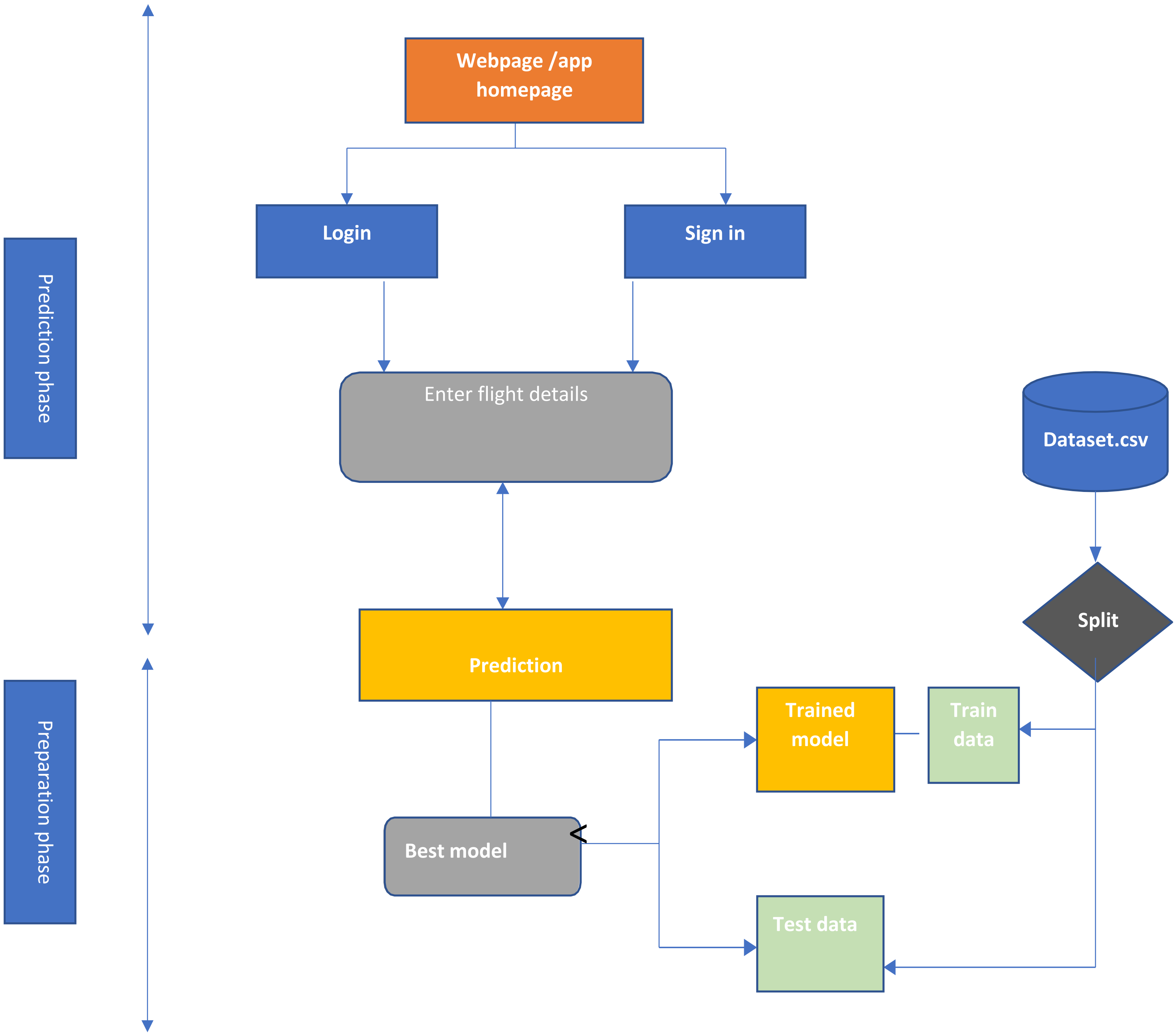
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	How effectively and easy users can learn and use a system. (To guide a new user easily understands a home page)
NFR-2	Security	system and its data were protected against malware attacks or unauthorized access (using two factor authentication during registration its leads to confirmation from user to get a next page)
NFR-3	Reliability	The system would run without failures for a given period of time under predefined conditions. (There is a less chance to occur any failure, then it has 80% of restoring capability even if the system fails.)
NFR-4	Performance	Deals with measure of the systems response time under different load conditions (The application takes minimum 2 to 3 seconds to load a page and a predicted result will be displayed within 5 seconds with WIFI OR LTE connection)
NFR-5	Availability	If any of issues occurred in a model the user will receive an alert there is a problem in prediction and the system would get back within 5 to 10 minutes. The system will be available during other times.
NFR-6	Scalability	The system is scalable, more than 1,000,000 of customers can use the application at same time to know about a flight delays.

5. PROJECT DESIGN

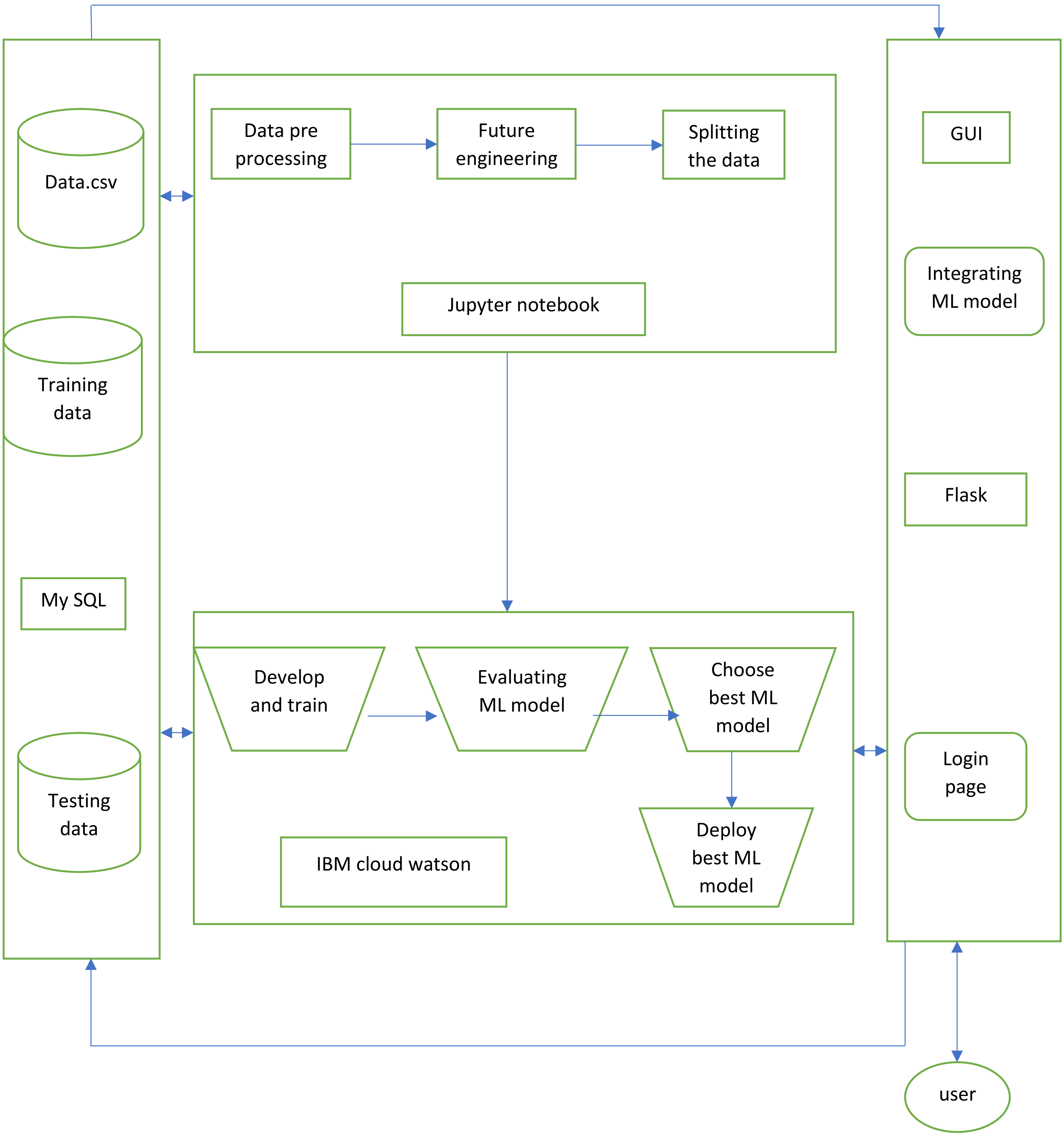
5.1 DATA FLOW DIAGRAM



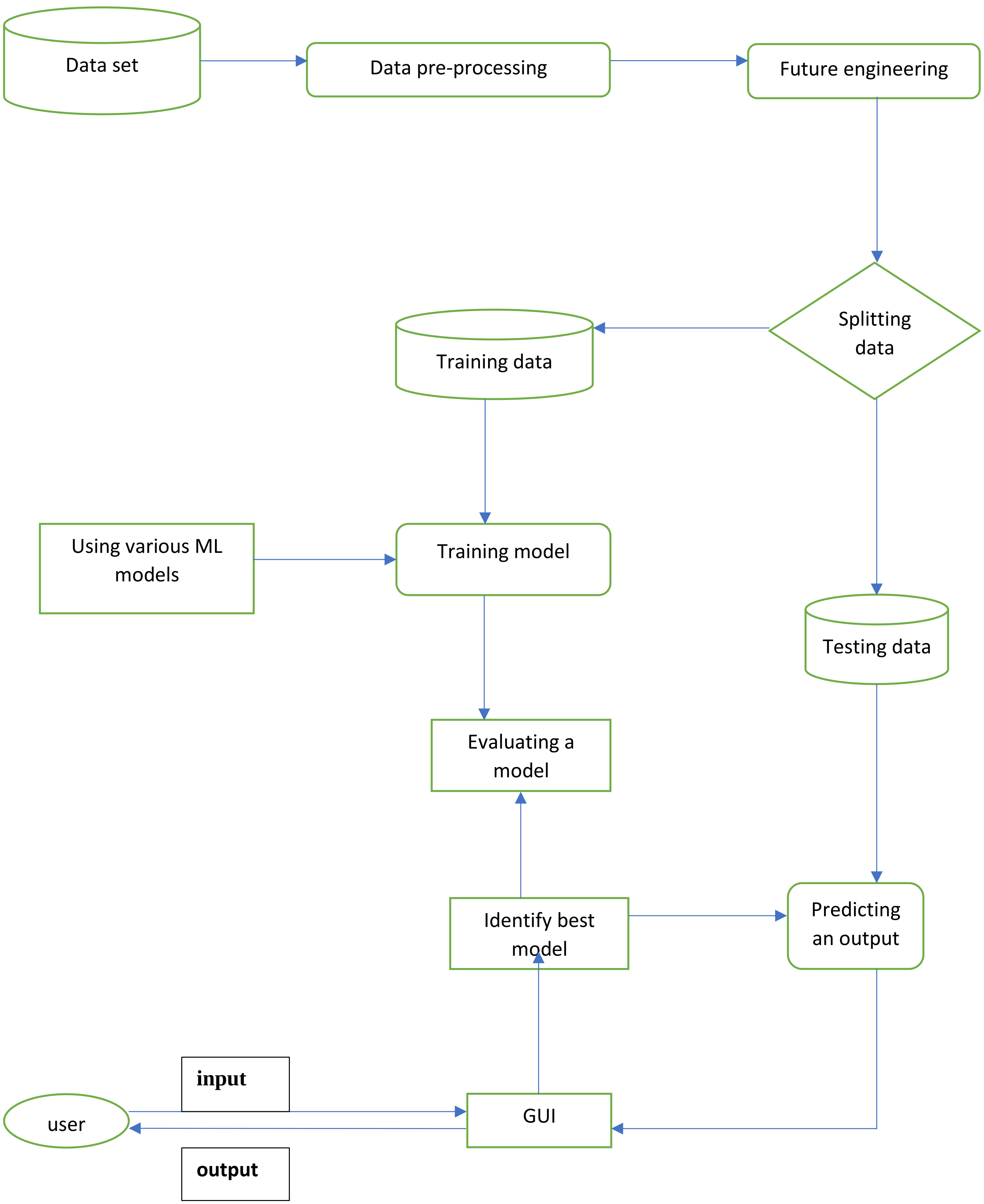
5.2 PROJECT FLOW DIAGRAM



5.3 TECHNICAL ARCHITECTURE



5.4 SOLUTION ARCHITECTURE:



5.5.USER STORIES

User type	Functional requirement	User story number	User story/ task	Acceptance criteria	priority	Release
Customer	User signup	USN-1	As a user, I can register for the application by email, passwordand other details	I can access my account/dashboard	high	Sprint-1
Customer	Signup through google	USN-2	As a user, I can use signup through google and exempt from typing other information	I can access my account/dashboard	low	Sprint-3
Customer	User confirmation	USN-3	As a user, I will receive a conformation email once I have registered for the application	I can receive a confirmation email& click confirm	medium	Sprint-2
Customer	User login	USN-4	As user, I can log in to the application using my usernameand password	I can log in successfully and seemy dashboard	high	Sprint-1
Customer	User login through Gmail	USN-5	As a user, I can log in to this application without have to remember the username and Password	I can log in successfully and seemy dashboard	low	Sprint-3
Customer	Search flight	USN-6	As a user, I can search for theflight whose delay I am interested of even if I don't know the flight complete details.	I can get a list of possible flight out of which one is my intended flight	high	Sprint-1
Customer	Predict delay time	USN-7	As a user, I can click 'Predict delay' to get the predicted delay time after entering flight detail	An estimated time of delay is displayed	high	Sprint-2
Customer	Predict delay accuracy	USN-8	As a user, I can see how reliableand accurate the prediction is.	The accuracy of the predicted delay is displayed	medium	Sprint-2
Customer	Notify the user	USN-9	As a user, I want to get reminded of my flight's delayedarrival/departure so that I can get prepared	A notification emailis received in my inbox	low	Sprint-3
Customer	Get feedback	USN-10	As a co-user, I want to help other users about the application through my ratings	I am able to rate the application out of 5 and submit a descriptive	low	Sprint-3
Customer	User logout	USN-11	As a user, I do not want to say logged in for long.		medium	Sprint-2

6.PROJECT PLANNING &SCHEDULING

6.1 Sprint planning & estimation:

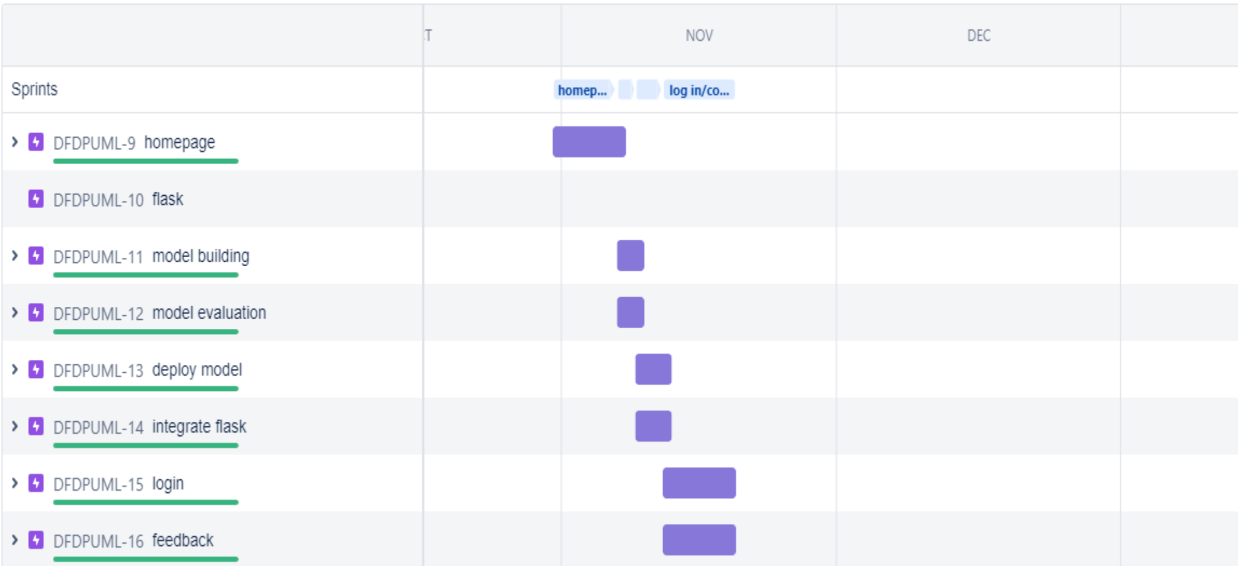
Sprint	Requirements	Estimation	Team Members
Sprint-1	➢ Home page ➢ Import flask	medium	Jeevalakshman .B Jayadharanai. B
Sprint-2	➢ Model building ➢ model evaluation	high	Saruthi . M Jeevalakshman.B
Sprint-3	➢ deploy the model ➢ integrate trained model with flask	high	Jayadharani .B Santhavaliyan .M
Sprint-4	➢ login /sign in	medium	Santhavaliyan.M Jeevalakshman.B

6.2 sprint deliverable schedule:

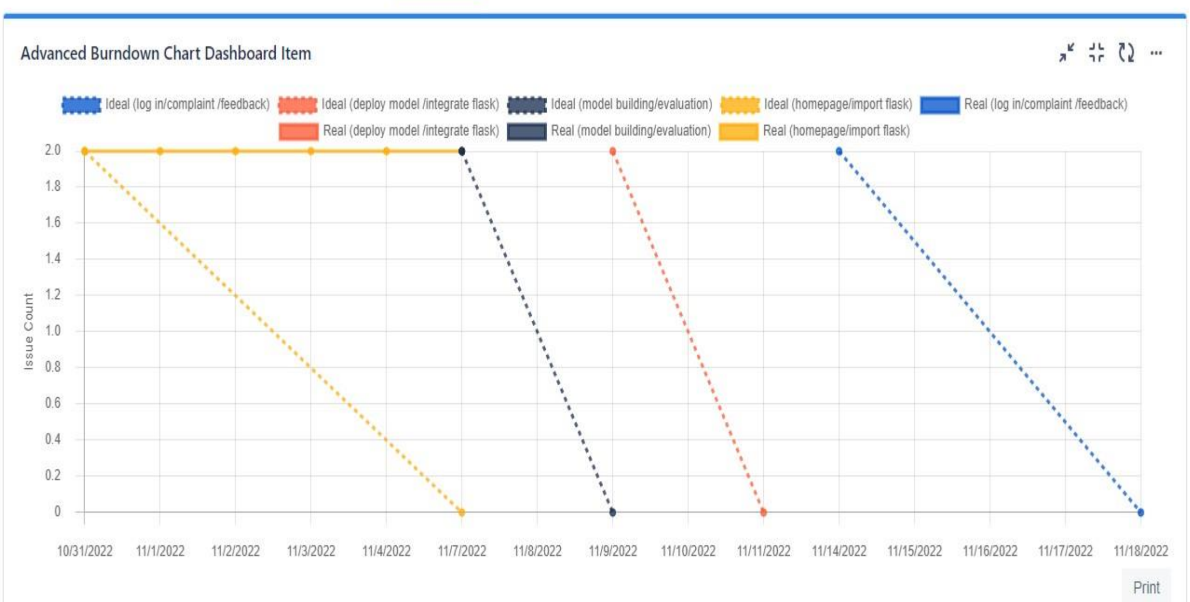
Sprint	Sprint start date	Sprint end date	Story point	Sprint release date
Sprint-1	24 Oct 2022	29 Oct 2022	20	06 Nov 2022
Sprint-2	31 Oct 2022	05 Nov 2022	20	09 Nov 2022
Sprint-3	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	12 Nov 2022	12 Nov 2022	20	19 Nov 2022

6.3. REPORTS FROM JIRA:

Road map:



Burndown graph:



7.CODING AND SOLUTIONING:

7.1. Feature 1

```
<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

 h2 {text-align: center;}

* {

 box-sizing: border-box;

}

body{

 font-family: Arial, Helvetica, sans-serif;

 background-image:
url(https://www.washingtonpost.com/wpapps/imrs.php?src=https://arc-anglerfish-
washpost-
prodwashpost.s3.amazonaws.com/public/PBKJ5C6KJJC75BO46RZEWUGL6A.jp
g&w

=860);

 background-size: cover;

 background-attachment: fixed;
```

```
}
input[type=text], select, textarea {
 width: 100%;
 padding: 12px;
 border: 1px solid #ccc;
 border-radius: 4px;
 resize: vertical;
}
label {
 padding: 12px 12px 12px 0;
 display: inline-block;
}
input[type=submit] {
 background-color: #04AA6D;
 color:white;
 padding: 12px 20px;
 border: none;
 border-radius: 4px;
 cursor: pointer;
 float: right;
```

```
}
input[type=submit]:hover {
 background-color: #45a049;
}
.container {
 border-radius: 5px;
 background-color:transparent;
 padding: 20px;
}
.col-25 {
 float: left;
 width: 25%;
 margin-top: 6px;
}
.col-75 {
 float: left;
 width: 75%;
 margin-top: 6px;
}
/* Clear floats after the columns */
```

```
.row:after {
 content: "";
 display: table;
 clear: both;
}
</style>
</head>
<body>
<h2 style="color: #ebf7f3">Prediction of flight delay</h2>
<div class="container">
 <form action="/action_page.php">
 <div class="row">
 <div class="col-25">
 <label style="color: #ebf7f3">Enter flight Number</label>
 </div>
 <div class="col-75">
 <input type="text" id="fname" numbers="flight number" placeholder="flight
number..">
 </div>
 </div>
```

```
<div class="row">

<div class="col-25">

<label style="color: #ebf7f3">Month</label>

</div>

<div class="col-75">

<input type="text" id="" name="month" placeholder="month..">

</div>

</div>

<div class="row">

<div class="col-25">

<label style="color: #ebf7f3">Day of month</label>

</div>

<div class="col-75">

<input type="text" id="" name="month" placeholder="day of month..">

</div>

</div>

<div class="row">

<div class="col-25">

<label style="color: #ebf7f3">Day of week</label>

</div>
```

```
 <div class="col-75">
 <input type="calender" id="fname" numbers="Day of week"
placeholder="Day of week..">
 </div>
 </div>
<div class="row">
 <div class="col-25">
 <label style="color: #ebf7f3">Origin</label>
 </div>
 <div class="col-75">
 </div>
<div class="col-25">
 <select id="country" name="origin">
 <option value="region">msp</option>
 <option value="region">sea</option>
 <option value="region">dtw</option>
 <option value="region">jfk</option>
 <option value="region">alt</option>
 </select>
 </div>
```

```
</div>
<div class="row">
<div class="col-25">
<label style="color: #ebf7f3">Destination</label>
</div>
<div class="col-75">
</div>
<div class="col-25">
<select id="region" name="origin">
<option value="region">msp</option>
<option value="region">sea</option>
<option value="region">dtw</option>
<option value="region">jfk</option>
<option value="region">alt</option>
</select>
</div>
</div>
<div class="row">
<div class="col-25">
<label style="color: #ebf7f3">Scheduled Departure Time</label>
```

```
 </div>
 <div class="col-75">
 <input type="time" id="fname" numbers="predict" placeholder="scheduled
Depature Time..">
 </div>
 </div>
 <div class="row">
 <div class="col-25">
 <label style="color: #ebf7f3"> Scheduled Arrival Time</label>
 </div>
 <div class="col-75">
 <input type="time" id="fname" numbers="predict" placeholder="Arrival
Depature Time..">
 </div>
 </div>
<div class="row">
 <div class="col-25">
 <label style="color: #ebf7f3">Actual Departure Time</label>
 </div>
 <div class="col-75">
```

```
 <input type="time" id="fname" numbers="predict" placeholder="..">
 </div>
 </div>
<div class="row">
 <input type="submit" value="Submit"></div>
 </form>
</div>
</body>
</html>
```

LOGIN PAGE

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>LOGIN PASSWORD VALIDATION | PRAROZ TUTORIAL</title>
 <link rel="stylesheet" href="style.css">
 <script src="valid.js"></script>
</head>
```

```html
<body>

 <div class="form">

 <h1>LOGIN HERE</h1>

 <p>Username :</p>

 <input type="text" name="" placeholder="Name Here">

 <p>Password :</p>

 <input type="password" name="" placeholder="Password Here" id="pass">

 <input type="checkbox" onclick="myfunction()">

 <input type="submit" name="" value="LOGIN" onclick="validate()">

 </div>

 <div>

 <p id="length"></p>

 </div>

 <input type="submit" name="" value="LOGIN" onclick="lpage()">


</body>

</html>
```

7.2. Feature 2

FLASK

APP.PY

```
from flask import Flask ,request,render_template

import numpy as np

import pandas as pd

import pickle

import os

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

app=Flask(__name__)

@app.route('homepage.html')

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['origin']

 if(origin1=="msp"):

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

```
 if(origin2 == "dtw"):
 origin1,origin2,origin3,origin4,origin5= 1,0,0,0,0
 if(origin3== "jfk"):
 origin1,origin2,origin3,origin4,origin5 = 0,0,1,0,0
 if(origin4 == "sea"):
 origin1,origin2,origin3,origin4,orgin5 = 0,1,0,0,0
 if(origin5 == "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,dayofweek,

origin1,origin2,origin3,origin4,origin5,destination1,destination2,

destination3,destination4,destination5')


y_pred = model.predict(total)

print(y_pred)

if (y_pred == [0.1]):

 ans="The Flight will be on time"

else:

 ans="The Flight will be delayed"

 def index():

 return render_template('homepage.html')
```

APP.IBM.PY

```
import time

import requests

import flask

from flask import request, render_template
```

```
from flask_cors import CORS

import requests

# NOTE: you must manually set API_KEY below using information

retrieved from your IBM Cloud account.

API_KEY = "give your api key"

token_response =

requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":

API_KEY, "grant_type":
'urn:ibm:params:oauth:granttype:4fJbepuok7oCO1UkeKU831Sq5rz5-
JP0R_hljeB2oaEL'})

mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' +

mltoken}

app = flask.Flask(__name__,

static_url_path='c:/Users/ELCOT/Desktop/templates/app.py')

CORS(app)

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

def sendHomePage():

 return render_template('index.html')

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

def predict():
```

```
 EnterflightNumber = str(request.form['Enter flight Number'])
 Month = int(request.form['month'])
 Dayofmonth= int(request.form['day of month'])
 Dayofweek= float(request.form['day of week'])
 origin=float(request.form['origin'])
 Destination=float(request.form['destination'])
 scheduleddeparturetime=time(request.form['scheduled departure
time'])
 scheduledarrivaltime=time(request.form['scheduled arraival time'])
 Actualtime=time(request.form['actual time'])
 X = [('Enter flight Number,Month,Day of month,Day of
week,origin,Destination,scheduled departure time,scheduled arrival
time,actual time')]
 payload_scoring =
"input_data";[("field")];("EnterflightNumb','Month','Dayofmonth','Dayof
week','origin','
Destination','scheduledarrivaltime,'scheduledarrivaltime','actual time"),
"values": X
 response_scoring = requests.post(' https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/72e6ce01-fcd7-4fe4-9351-
e903f4390880/predictions?version=2022-11-14', json=payload_scoring,
```

```
headers={'Authorization': 'Bearer ' + mltoken})
 print(response_scoring)
 predictions = response_scoring.json()
 predict = predictions['predictions'][0]['values'][0][0]
 print("Final prediction :",predict)


 # showing the prediction results in a UI# showing the prediction
results in a UI
 return render_template('predict.html', predict=predict)
if __name__ == '__main__' :
 app.run(debug= False)
```

8.TESTING

8.1. Test cases

Test cases	Feature type	Test scenario	Pre-requisite	Steps to execute	working	results
Login page	Functional	Enter a user credentials	HTML	Enter user name, password	Login popup should display	pass
Home page	Functional	Enter a flight detail	Integrate with flask	Enter flight details (arrival, departure time)	Home page should display to enter flight details	pass

8.2. User acceptance testing

Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [developing a flight delay prediction using machine learning] project at the time of the releaseto User Acceptance Testing (UAT).

2.Defect Analysis

This report executes our user scheduling and their approaches.

Task	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
Login	5	1	2	4	12
Home page	4	1	7	5	17
Model building	1	0	3	0	4
Execute the model	1	0	0	1	2
Flask(app.py)	1	2	2	2	7
Flask(ibm app.py)	0	0	1	0	1

Deploying themodel	0	0	1	1	2
Totals	12	4	16	13	45

1. 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	1	0	0	pass
Homepage	1	0	0	pass

9.RESULTS

9.1. Performance metrics

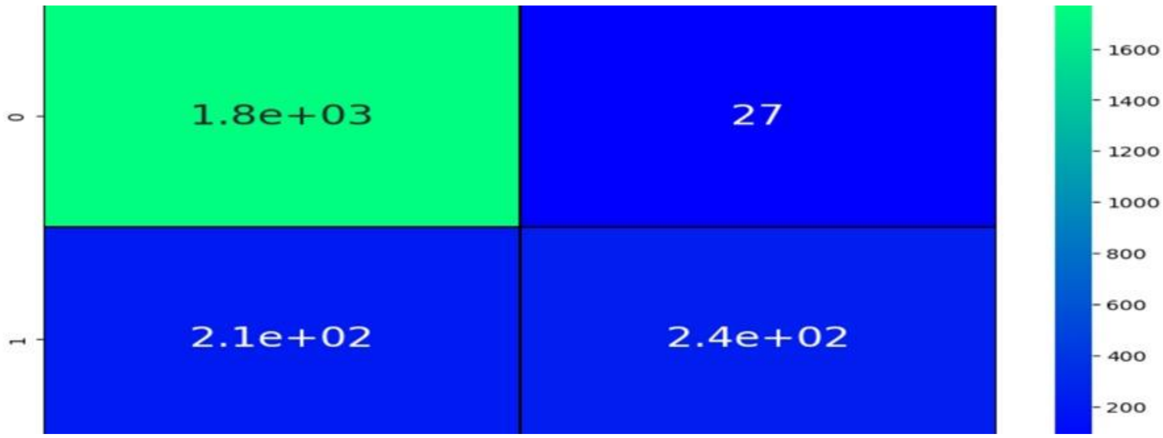
1.METRICS

REGRESSION MODEL

MAE-, MSE-, RMSE-, R2 Score

RANDOM FOREST

```
In [48]: from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
pred=rf.predict(x_test)
cm=confusion_matrix(y_test, pred)
plt.figure(figsize=(10,6))
sns.heatmap(cm, annot=True,cmap='winter',linewidths=0.3, linecolor='black',annot_kws={"size": 20})
TP=cm[0][0]
TN=cm[1][1]
FN=cm[1][0]
FP=cm[0][1]
#print(round(accuracy_score(prediction3,y_test)*100,2))
#print('Testing Accuracy for knn',(TP+TN)/(TP+TN+FN+FP))
print('Testing Sensitivity for Random Forest',(TP/(TP+FN)))
print('Testing Specificity for Random Forest',(TN/(TN+FP)))
print('Testing Precision for Random Forest',(TP/(TP+FP)))
print('Testing accuracy for Random Forest',accuracy_score(y_test, pred))

Testing Sensitivity for Random Forest 0.8942065491183879
Testing Specificity for Random Forest 0.8969465648854962
Testing Precision for Random Forest 0.9850166481687015
Testing accuracy for Random Forest 0.8945260347129506
```



CLASSIFICATION MODEL

Confusion matrix-, accuracy score- & classification report

```
In [49]: print(classification_report(y_test,pred))

              precision    recall  f1-score   support

           0       0.89      0.99      0.94      1802
           1       0.90      0.53      0.66       445

    accuracy                           0.89      2247
   macro avg       0.90      0.76      0.80      2247
weighted avg       0.89      0.89      0.88      2247
```

2.TUNE THE MODEL:

Hyperparameters tuning-

Validation method

```
In [58]: #HYPERPARAMETER TUNING
         grid.fit(X_train, y_train)

Out[58]:
                                GridSearchCV
GridSearchCV(cv=5,
             estimator=GradientBoostingClassifier(learning_rate=0.7,
                                                  max_depth=4),
             param_grid={'max_features': array([1, 2, 3, 4, 5]),
                         'n_estimators': array([ 10,  20,  30,  40,  50,  60,  70,  80,  90, 100, 110, 120, 130,
       140, 150, 160, 170, 180, 190, 200])})
                  estimator: GradientBoostingClassifier
          GradientBoostingClassifier(learning_rate=0.7, max_depth=4)
                       GradientBoostingClassifier
          GradientBoostingClassifier(learning_rate=0.7, max_depth=4)
```

```
In [59]: print("The best parameters are %s with a score of %0.2f"
               % (grid.best_params_, grid.best_score_))

         The best parameters are {'max_features': 5, 'n_estimators': 200} with a score of 0.97
```

10.ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- Therefore, predicting flight delays can improve airline operation and passenger satisfactions, which will result in a positive.
- In this study, the main goal is to compare the performance of machine learning classification algorithms when predicting flight delays.
- Several explanatory variables were tested to discover their association with flight delay, airport operation, and flow management.
- These classifiers were compared based on their accuracy, running time, and efficiency. The author also used association techniques and evaluated them to obtain flight delays information.

DISADVANTAGES:

- Delays were occurred due to weather conditions.
- Takes more time to data preprocessing

11.CONCLUSION

In this project we use the flight data to predict the flight departure delay by giving a flight details (arrival time, departure time). Our result shows that the random forest method yields the best performance compared to decision tree SVM model. In the end, our model correctly predicts 91% of non-delayed flights.

12.FUTURE WORK:

This project is based on data analysis (data pre-processing and cleaning the data). Therefore, the feature work of the project includes incorporating a larger dataset. To add extra features like weather condition to know a delay of flight. Add a weather data in our dataset and preprocess it then train the model to predict the delay of flight.

13.APPENDIX

Source code:

```
import sys
import numpy as np
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='OA-1cINHTyxIFeREJY4T2JnWjrP1kKLNljtQ47ry0ghP',
    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 = 'developingaflightdelaypredictionu-donotdelete-pr-hez261vc4alj9r'
object_key = 'flightdata.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 )

df = pd.read_csv(body)
df.head()
```

#IMPORT 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.metrics import accuracy_score
import sklearn.metrics as metrics
```

#DOWNLOAD THE DATASET

```
df=pd.read_csv('C:\\Users\\ketziyal\\Downloads\\flightdata.csv')
```

#ANALYZE THE DATA

```
df.info()

df.describe()
```

#HANDLE MISSING VALUES

```
df.isnull().sum()

df['DEST'].unique()
```

#DATA VISUALIZATION

```
from matplotlib import pyplot as plt

plt.scatter(df['ARR_DELAY'],df['ARR_DEL15'])
plt.xlabel('ARR_DELAY')
plt.ylabel('ARR_DEL15')
plt.title('scatter plot between ARR_DELAY and ARR_DEL15')

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

sns.heatmap(df.corr())
```

```
#DROP UNNECESSARY COLUMNS

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

df=df[["FL_NUM","MONTH","DAY_OF_MONTH","DAY_OF_WEEK","ORIGIN","DEST","CRS_ARR_TIME","DEP_DEL15","ARR_DEL15"]]
df.isnull().sum()

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

import math

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

from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()

df['DEST']=le.fit_transform(df['DEST'])
df['ORIGIN']=le.fit_transform(df['ORIGIN'])
```

```
df.head()

from sklearn.preprocessing import OneHotEncoder

oh=OneHotEncoder()

z=oh.fit_transform(x[: 4:5]).toarray()
t=oh.fit_transform(x[: 5:6]).toarray()

z

t

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

# DEPENDANT AND INDEPENDENT VALUES

x=df.iloc[:,0:8].values
y=df.iloc[:,8:9].values

#SPLIT THE DATA TO TRAIN TEST SPLIT

from sklearn.model_selection import train_test_split
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

#MODEL BUILDING

#DECISION TREE

from sklearn.tree import DecisionTreeClassifier
dc=DecisionTreeClassifier()
dc.fit(x_train,y_train)
dc.score(x_test,y_test)

#RANDOM FOREST

from sklearn.ensemble import RandomForestClassifier
model= RandomForestClassifier()
rf=RandomForestClassifier(n_estimators=50,random_state=42)
rf.fit(x_train,y_train)
rf.score(x_test,y_test)

pd.DataFrame(rf.predict(x_test)).value_counts()
```

#LOGESTIC REGRESSION

```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression(solver='sag')
lr.fit(x_train,y_train)
lr.score(x_test,y_test)

lr.predict(x_test).sum()
```

#SVM

```
from sklearn.svm import SVC
svm=SVC(kernel='sigmoid')
svm.fit(x_train,y_train)
svm.score(x_test,y_test)

pd.DataFrame(svm.predict(x_test)).value_counts()

pd.DataFrame(y_test).value_counts()
```

 # K-NEAREST NEIGHBOUR CLASSIFIER

```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=5)
knn.fit(x_train,y_train)
knn.score(x_test,y_test)
```

#EVALUATING A MODEL (RANDOM FOREST)

```
from sklearn.metrics import
confusion_matrix,accuracy_score,classification_report
pred=rf.predict(x_test)
cm=confusion_matrix(y_test, pred)
plt.figure(figsize=(10,6))
sns.heatmap(cm, annot=True,cmap='winter',linewidths=0.3,
linecolor='black',annot_kws={"size": 20})
TP=cm[0][0]
TN=cm[1][1]
FN=cm[1][0]
FP=cm[0][1]
#print(round(accuracy_score(prediction3,y_test)*100,2))
#print('Testing Accuracy for knn',(TP+TN)/(TP+TN+FN+FP))
print('Testing Sensitivity for Random Forest',(TP/(TP+FN)))
print('Testing Specificity for Random Forest',(TN/(TN+FP)))
print('Testing Precision for Random Forest',(TP/(TP+FP)))
print('Testing accuracy for Random Forest',accuracy_score(y_test, pred))

print(classification_report(y_test,pred))
```

#EVALUATION OF DECISSION TREE

```
pred1=dc.predict(x_test)
cm1=confusion_matrix(y_test, pred1)
plt.figure(figsize=(10,6))
sns.heatmap(cm1, annot=True,cmap='winter',linewidths=0.3,
linecolor='black',annot_kws={"size": 20})
TP=cm1[0][0]
TN=cm1[1][1]
FN=cm1[1][0]
FP=cm1[0][1]
#print(round(accuracy_score(prediction3,y_test)*100,2))
print('Testing Accuracy for Decision Tree',(TP+TN)/(TP+TN+FN+FP))
print('Testing Sensitivity for Decision Tree',(TP/(TP+FN)))
print('Testing Specificity for Decision Tree',(TN/(TN+FP)))
print('Testing Precision for Decision Tree',(TP/(TP+FP)))
print('Testing accuracy for Decision Tree',accuracy_score(y_test, pred1))
print(classification_report(y_test,pred1))
```

#Download pickle file

```
import pickle
pickle.dump(rf,open("flight.pkl",'wb'))
```

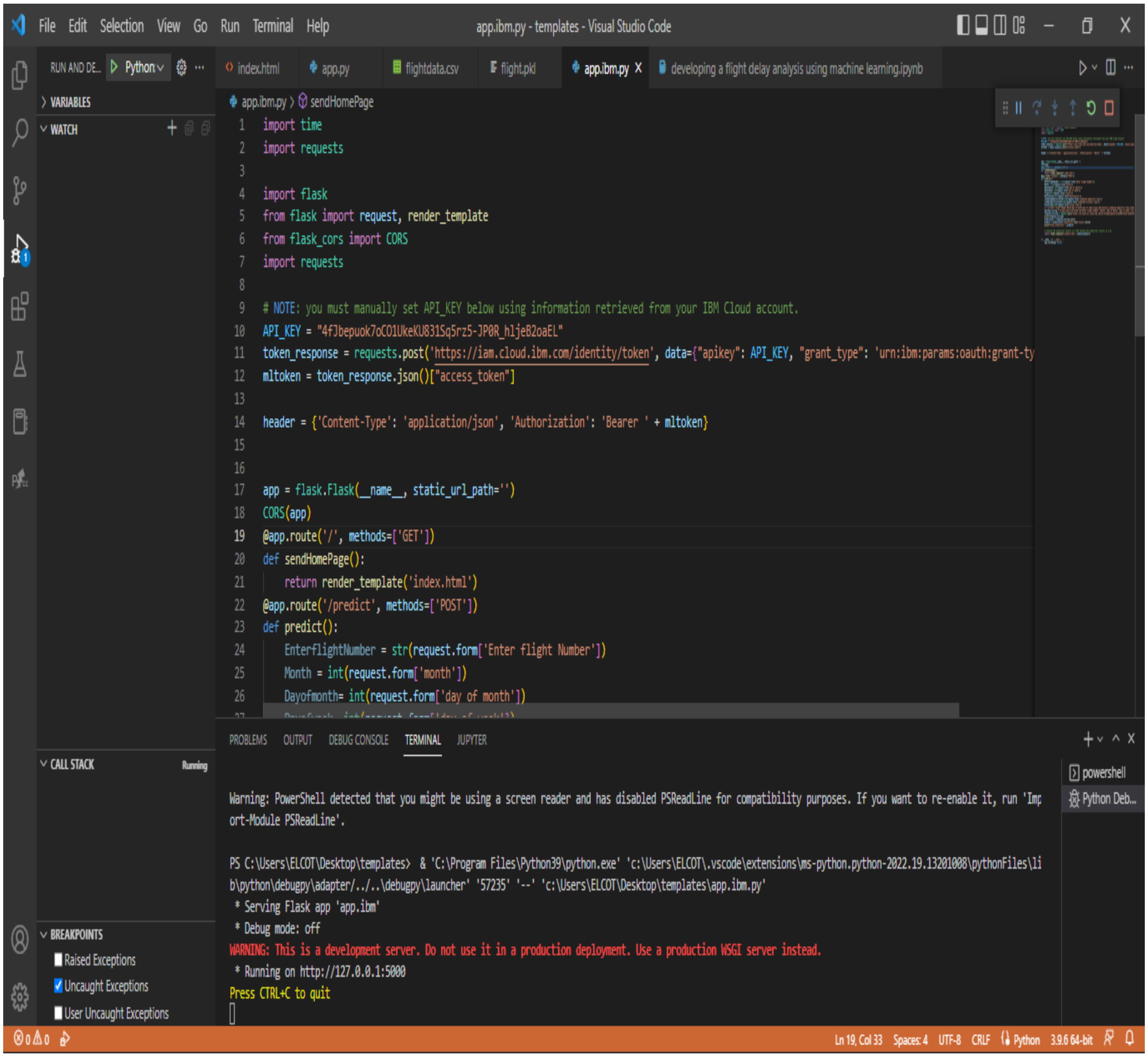
```
#DEPLOY THE MODEL ON IBM CLOUD
!pip install -U ibm-watson-machine-learning

from ibm_watson_machine_learning import APIClient
import json

#Authenticate and Set Space
wml_credentials = {
   "apikey":"4fJbepuok7oCO1UkeKU831Sq5rz5-JP0R_hljeB2oaEL",
   "url":"https://us-south.ml.cloud.ibm.com"
}

wml_client = APIClient(wml_credentials)
wml_client.spaces.list()

SPACE_ID="deaaa6e0-4843-467d-94d8-71d0272de83b"

wml_client.set.default_space(SPACE_ID)


wml_client.software_specifications.list(500)
import sys
sys.version
pip install ibm-watson-machine-learning
```

#Save and Deploy the model

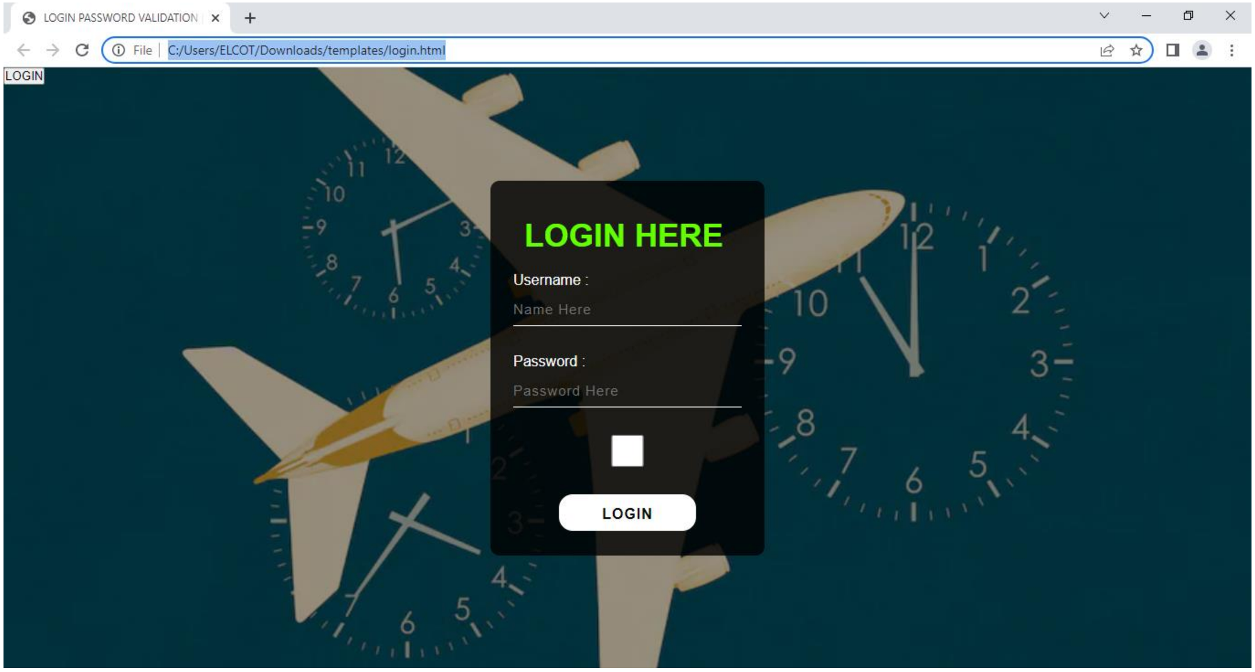
```
import sklearn
sklearn.__version__

MODEL_NAME = "RandomForestClassifier()"
DEPLOYMENT_NAME = 'flight delay'
DEMO_MODEL = model

# Set Python Version
software_spec_uid =
wml_client.software_specifications.get_id_by_name('runtime-22.1-py3.9')


# Setup model meta

model_props = {
    wml_client.repository.ModelMetaNames.NAME: MODEL_NAME,
    wml_client.repository.ModelMetaNames.TYPE: 'scikit-learn_1.0',
    wml_client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:
software_spec_uid
}

#Save model
model_details = wml_client.repository.store_model(
    model=DEMO_MODEL,
    meta_props=model_props,
```

```
    training_data=x_train,
    training_target=y_train
)

model_details
model_id = wml_client.repository.get_model_id(model_details)
model_id
# Set meta
deployment_props = {
    wml_client.deployments.ConfigurationMetaNames.NAME:
DEPLOYMENT_NAME,
    wml_client.deployments.ConfigurationMetaNames.ONLINE: {}
}
# Set meta
deployment_props = {
    wml_client.deployments.ConfigurationMetaNames.NAME:
DEPLOYMENT_NAME,
    wml_client.deployments.ConfigurationMetaNames.ONLINE: {}
}
```

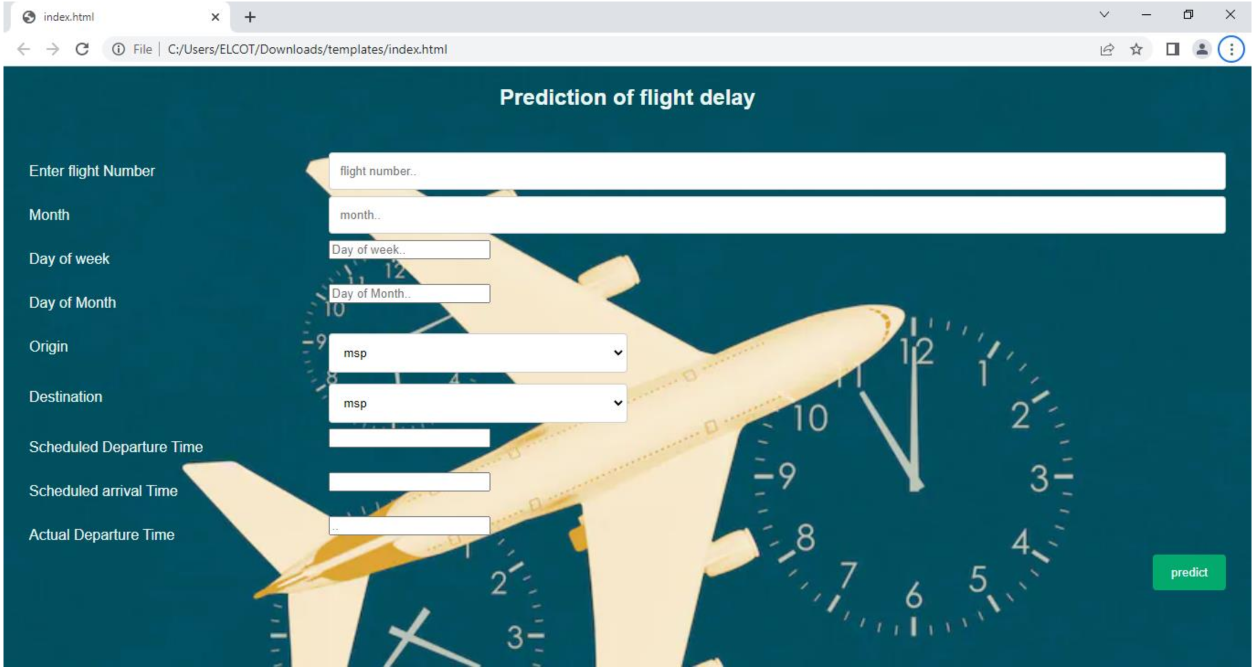
SCREENSHOTS:



LOGIN PAGE:



HOME PAGE:



GITHUB LINK: https://github.com/IBM-EPBL/IBM-Project-48849-1660813651

DEMO LINK :

https://drive.google.com/file/d/1tLZzO_4cx08AHnF2Oaee7muamMarlsBB/view?usp=sharing