FLIGHT DELAY PREDICTION MODEL USING MACHINE LEARNING

A NAALAIYA THIRAN PROJECT REPORT

Submitted by

SRIDHAR K V	(610519104095)
SRIRAM S M	(610519104098)
TAMILARASAN D	(610519104103)
VITTAL J S	(610519104116)

TEAM ID: PNT2022TMID29843

FACULTY MENTOR
MR SANKAR H

INDUSTRY MENTOR
MS LALITHA GAYATHRI

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DHIRAJLAL GANDHI COLLEGE OF TECHNOLOGY

SALEM

TABLE OF CONTENT

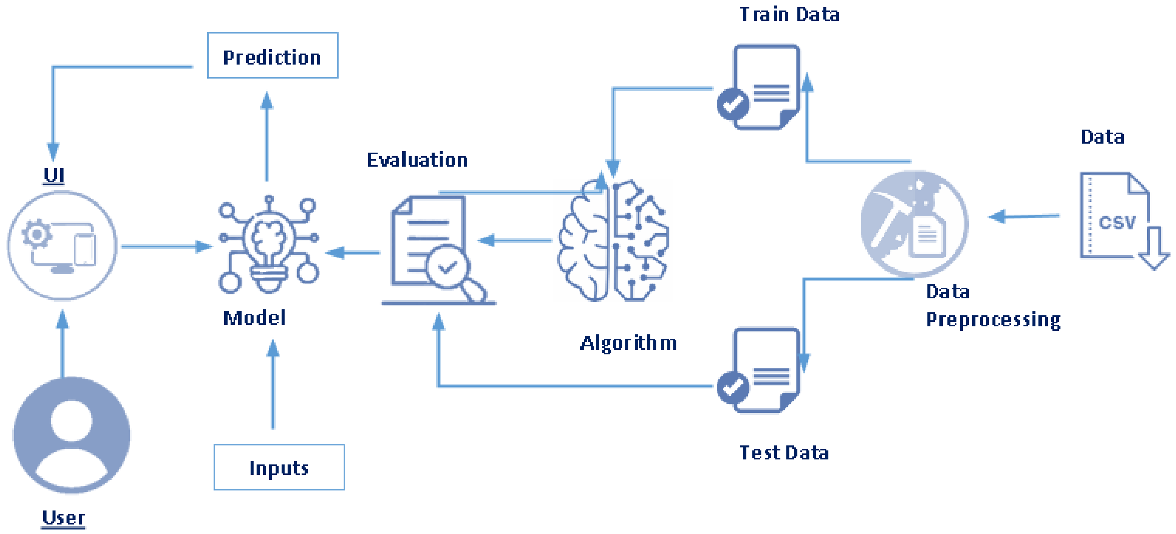
CHAPTER NO	TITLE	PAGE NO
1	**INTRODUCTION**	**1**
	1.1 Project Overview	1
	1.2 Purpose	2
2	**LITERATURE SURVEY**	**3**
	2.1 Existing problem	3
	2.2 Problem statement	5
3	**IDEATION & PROPOSED SOLUTION**	**7**
	3.1 Empathy Map Canvas	7
	3.2 Ideation and Brain storming	7
	3.3 Proposed Solution	10
	3.4 Problem Solution Fit	11
4	**REQUIREMENT ANALYSIS**	**12**
	4.1 Functional Requirement	12
	4.2 Non-Functional Requirement	13
5	**PROJECT DESIGN**	**14**
	5.1 Data Flow Diagram	14
	5.2 User Stories	14
	5.3 Solution Architecture	15
	5.4 Technology Stack	16

6	**PROJECT PLANNING & SCHEDULING**	**18**
	6.1 Sprint Planning & Estimation	18
	6.2 Sprint Delivery Schedule	19
7	**CODING AND SOLUTIONS**	**20**
8	**IBM CLOUD SERVICES**	**31**
9	**OUTPUTS AND SCREENSHOTS**	**35**
10	**ADVANTAGES & DISADVANTAGES**	**38**
11	**CONCLUSION**	**38**
12	**FUTURE SCOPE**	**39**
13	**APPENDIX**	**40**

1.INTRODUCTION

1.1Project Overview:

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

Over the last twenty years, air travel has been increasingly preferred among travellers, 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. According to the Bureau of Statistics HOS), hoor 20% of all flights are delayed by 15 minutes or more. Flight delays causes a negative impact, mainly economical for airport arities, commuters and airline industries as well.

Therefore, this study develops a novel spatial analysis approach to explore the delay and canal factors which is able to take dependence and the possible problem involved including error correlation and variable lag effect of canal factors on delay into account.

The main objective of the model is to predict flight delays accurately in order to optimize flight operations and minimize delays.

2.LITERATURE SURVEY

2.1 Existing problem

Reference paper : 1

Topic : Airline Flight Delay Prediction Using Machine Learning Models, 2021 5th International Conference on E-Business and Internet, Singapore, Singapore, October 2021.

Description :

Flight delays are gradually increasing and bring more financial difficulties and customer dissatisfaction to airline companies. To resolve this situation, supervised machine learning models were implemented to predict flight delays. The data set that records information of flights departing from JFK airport during one year was used for the prediction. Seven algorithms (Logistic Regression, K-Nearest Neighbour, Gaussian Naïve Bayes, Decision Tree, Support Vector Machine, and Gradient Boosted Tree) were trained and tested to complete the binary classification of flight delays. The evaluation of algorithms was fulfilled by comparing the values of four measures: accuracy, precision, recall, and f1-score. Measures were weighted to adjust the imbalance of the selected data set. The comparative analysis showed that the Decision Tree algorithm has the best performance with an accuracy of 0.9777. Tree-based ensemble classifiers generally have better performance over other base classifiers.

Reference link: https://dl.acm.org/doi/fullHtml/10.1145/3497701.3497725

Reference paper : 2

Topic :

Flight delay prediction based on deep learning and Levenberg-Marquart algorithm, 26 November 2020.

Description :

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. However, most of the proposed methods are not accurate enough because of massive volume data, dependencies and extreme number of parameters. This paper proposes a model for predicting flight delay based on Deep Learning (DL). DL is one of the newest methods employed in solving problems with high level of complexity and massive amount of data. Moreover, DL is capable to automatically extract the important features from data. Furthermore, due to the fact that most of flight delay data are noisy, a technique based on stack denoising auto-encoder is designed and added to the proposed model.

Reference Link : https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00380-z

Reference paper : 3

Topic : Machine Learning Model - based Prediction of Flight Delay

Description :

Prior prediction of flight arrival delays is necessary for both travelers and airlines because delays in flights not only trigger huge economic loss but also airlines end up losing their reputation that was built for several years and passengers lose their valuable time. Our paper aims at predicting the arrival delay of a scheduled individual flight at the destination airport by utilizing available data. The predictive model presented in this work is to foresee airline arrival delays by employing supervised machine learning algorithms. US domestic flight data along with the weather data from July 2019 to December 2019 were acquired and are used while training the predictive model. XGBoost and linear regression algorithms were applied to develop the predictive model that aims at predicting flight delays.. Flight data along with the weather data was given to the model. Using this data, binary classification was carried out by the XGBoost trained model to predict whether there would be any arrival delay or not, and then linear regression model predicts the delay time of the flight. Reference Link : https://ieeexplore.ieee.org/document/9243339

PROPOSED METHODOLOGY :

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.

2.2 Problem statement

Over the last twenty years, air travel has been increasingly preferred among travellers, 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. According to the Bureau of Statistics HOS), hoor 20% of all flights are delayed by 15 minutes or more.

Flight delays causes a negative impact, mainly economical for airport arities, commuters and airline industries as well. Therefore, this study develops a novel spatial analysis approach to explore the delay and canal factors which is able to take dependence and the possible problem involved including error correlation and variable lag effect of canal factors on delay into account.

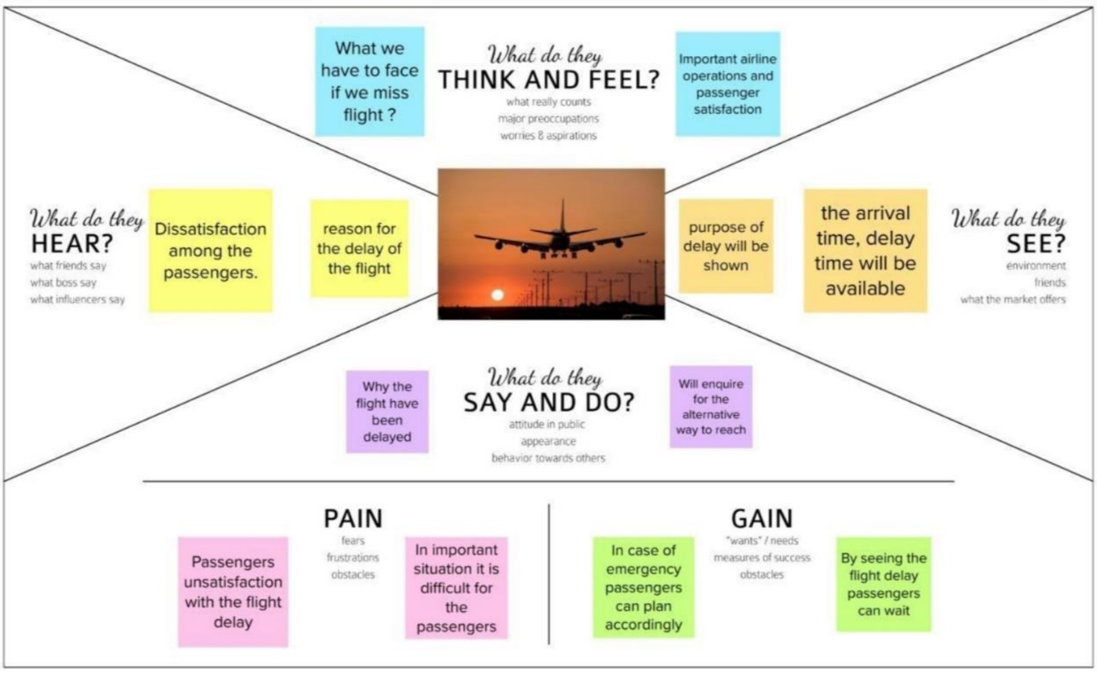
The main objective of the model is to predict flight delays accurately in order to optimize flight operations and minimize delays. Using a machine learning model, we can predict flight arrival delays.

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.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy map canvas:



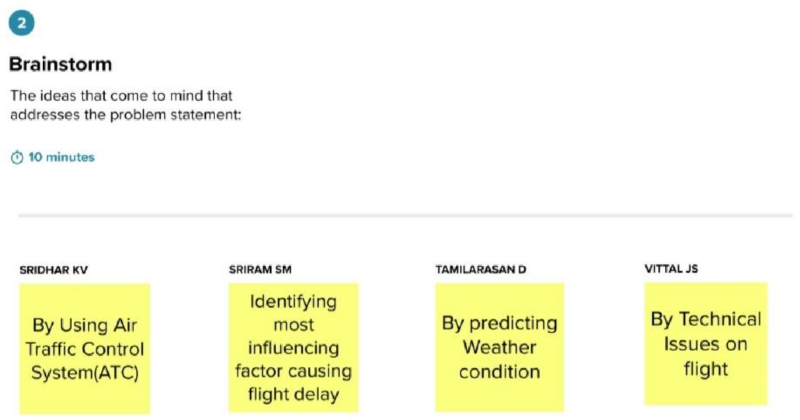
3.2 Ideation and Brain storming:

The paper uses dataset originally sourced from the Bureau of Transportation Statistics. The objective is to analyze and predict flight departure delays for a sample of flights in the USA, the main goals being:

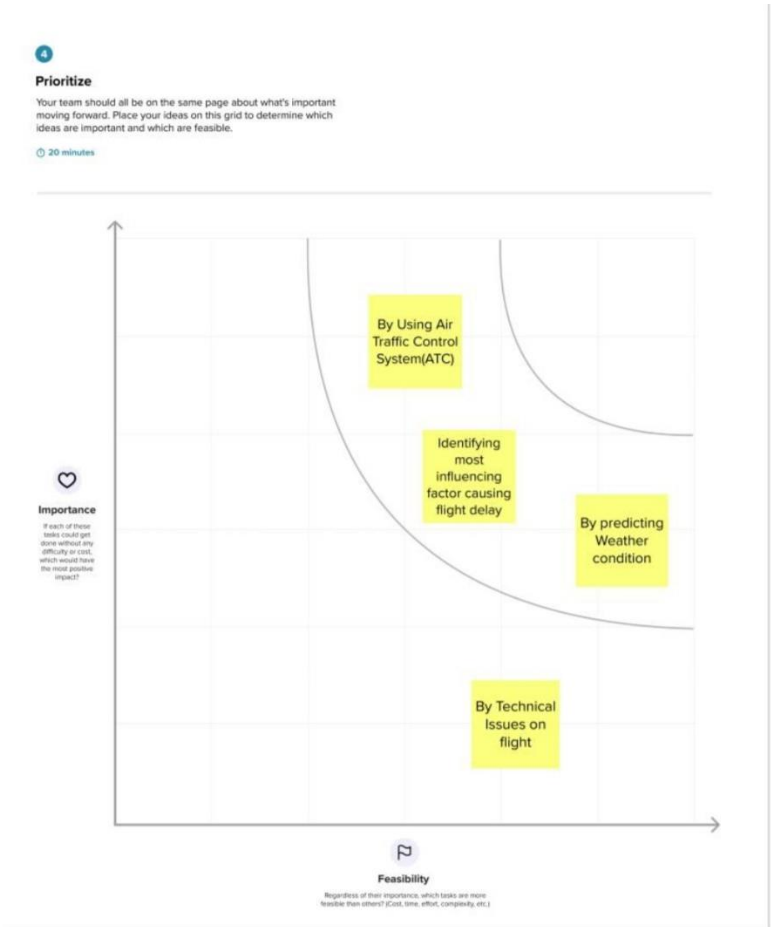
1. Identify the most influencing factors causing flight delays
2. Predict if a specific flight will be delayed or not,
3. Estimate the magnitude and impact in case of a delay.

Step-1: Team Gathering, Collaboration and Select the Problem Statement

Step-2: Brainstorm, Idea Listing and Grouping



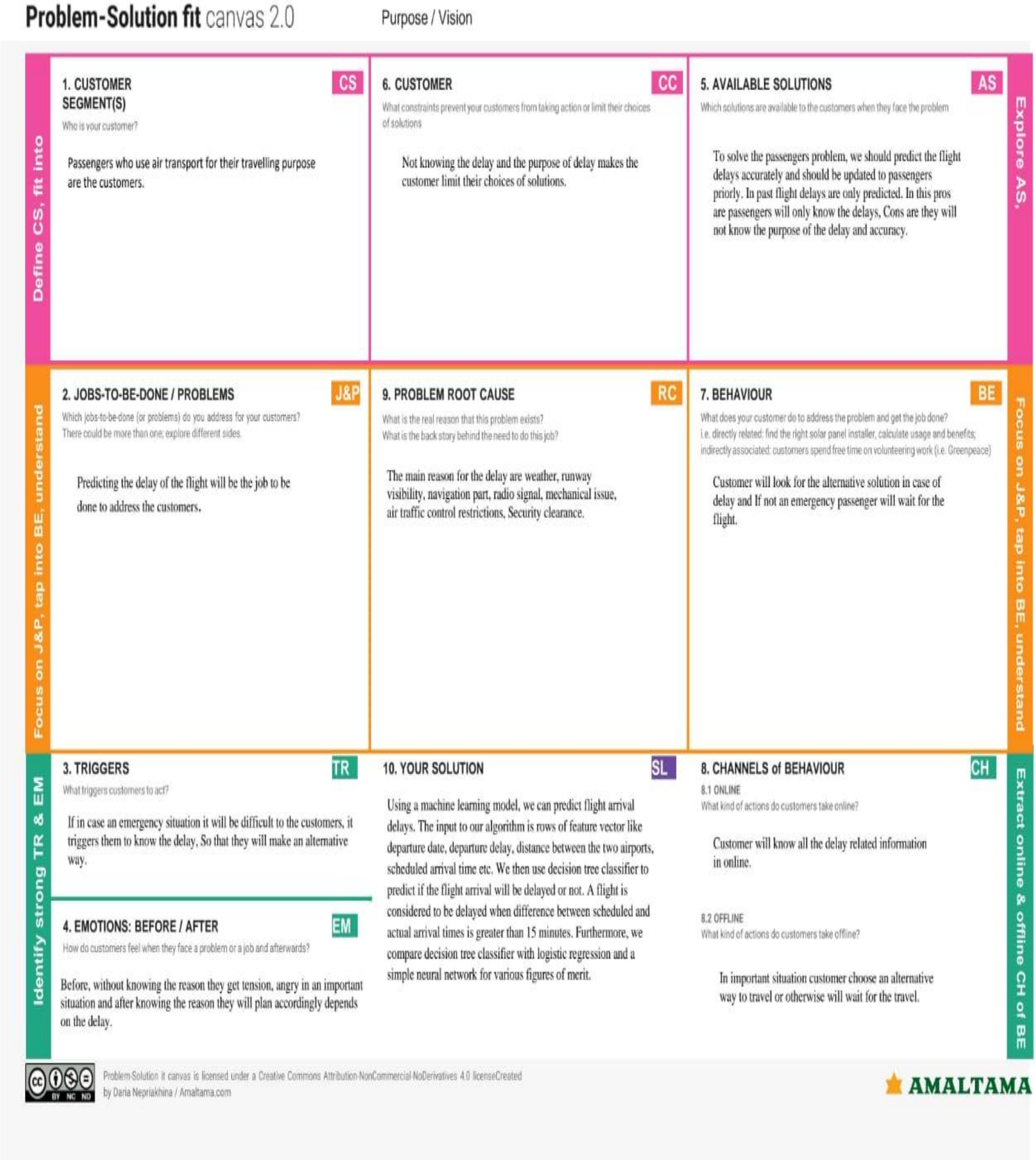
Step-3: Idea Prioritization :



3.3 Proposed solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	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 isto predict flight delays accurately in order to optimize flight operations and minimize delays.
2.	Idea / Solution description	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.
3.	Novelty / Uniqueness	we compare decision tree classifier with logistic regression and a simple neural network for various figures of merit.
4.	Social Impact / Customer Satisfaction	Time management will be the social impact and passengers can plan accordingly.
5.	Business Model (Revenue Model)	➢ Low-cost airline business model. ➢ B2C business Model.
6.	Scalability of the Solution	Any type of flight delays can be known and it provide maximum accuracy.

3.4 Problem solution fit:



Problem-Solution fit canvas 2.0

Purpose / Vision

Define CS, fit into

1. CUSTOMER SEGMENT(S) CS

Who is your customer?

Passengers who use air transport for their travelling purpose are the customers.

6. CUSTOMER CC

What constraints prevent your customers from taking action or limit their choices of solutions

Not knowing the delay and the purpose of delay makes the customer limit their choices of solutions.

5. AVAILABLE SOLUTIONS AS

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

To solve the passengers problem, we should predict the flight delays accurately and should be updated to passengers priorly. In past flight delays are only predicted. In this pros are passengers will only know the delays, Cons are they will not know the purpose of the delay and accuracy.

Explore AS,

Focus on J&P, tap into BE, understand

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.

Predicting the delay of the flight will be the job to be done to address the customers.

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?

The main reason for the delay are weather, runway visibility, navigation part, radio signal, mechanical issue, air traffic control restrictions, Security clearance.

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)

Customer will look for the alternative solution in case of delay and If not an emergency passenger will wait for the flight.

Focus on J&P, tap into BE, understand

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act?

If in case an emergency situation it will be difficult to the customers, it triggers them to know the delay, So that they will make an alternative way.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?

Before, without knowing the reason they get tension, angry in an important situation and after knowing the reason they will plan accordingly depends on the delay.

10. YOUR SOLUTION SL

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.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online?

Customer will know all the delay related information in online.

8.2 OFFLINE
What kind of actions do customers take offline?

In important situation customer choose an alternative way to travel or otherwise will wait for the travel.

Extract online & offline CH of BE

AMALTAMA

4.REQUIREMENT ANALYISIS

4.1 Functional requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	• Registration through Form • Registration through Gmail
FR-2	User Confirmation	• Confirmation via Email • Confirmation via Password
FR-3	Log in	• Log in with registered user details and enter the application.
FR-4	Delay Prediction	• User can search details for flight that can be apply prediction • Prediction results have shown the web page.
FR-5	Support	• Support option mainly focus on user related queries and facing issues solution.
FR-6	Expertness of Delay Prediction	• Using strong prediction formulas and predicting exponentially.
FR-7	Feedback	• Get user feedback about application difficulties and good responses.
FR-8	Log Out	• Log out from the application after process.

4.2 Non-Functional requirments:

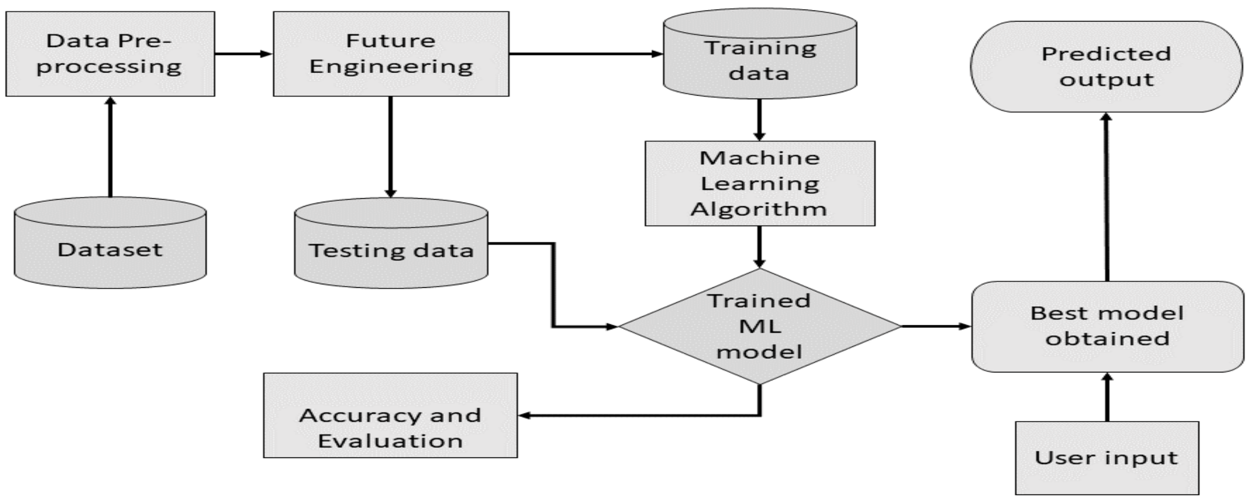
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This application gives good experience and easily understandable.
NFR-2	**Security**	User data should be stored in IBM Cloud so, highlysecurable.
NFR-3	**Reliability**	This application high priority of prediction and highreliability by deploying in IBM Watson.
NFR-4	**Performance**	Application can do better performance in every time, and also user can access any time.
NFR-5	**Availability**	Data should be stored in IBM Cloud so, available anytime to process this application.
NFR-6	**Scalability**	This application runs based on prediction (Machine Learning) so, it had high scalability.

5.PROJECT DESIGN

5.1 Data flow diagram:

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



5.2 User Stories

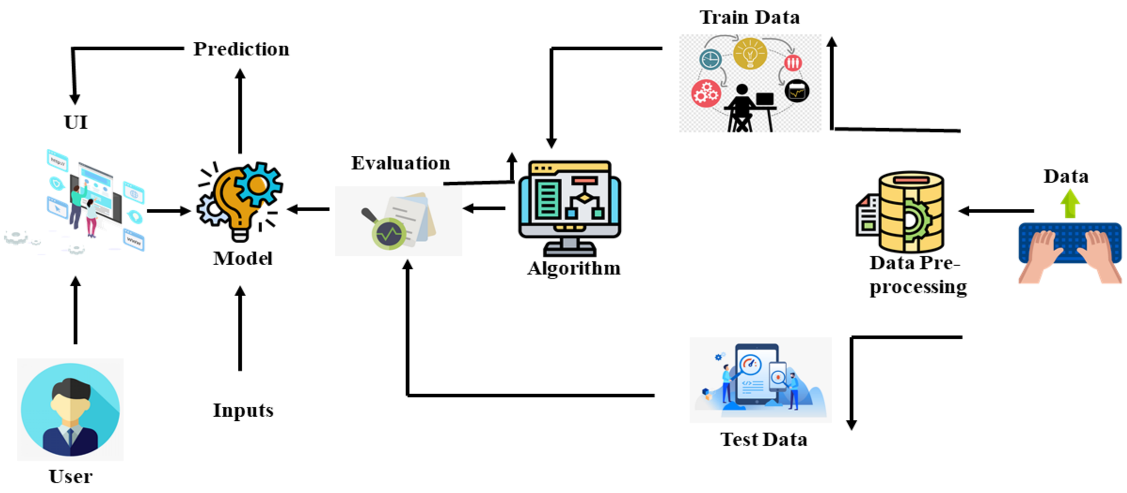
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail	I can register & login by providing Gmail with access permission	Medium	Sprint-2
	Login	USN-4	As a user, I can register for the application by entering mail and password	I can login using my registered mail id and password	High	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
	Dashboard	USN-5	As a user, I can access the dashboard which provide data to predict flight delay.	Can provide valid input data	High	Sprint-2
	Logout	USN-6	As a user, I can logout by clicking logoutbutton	I can logout my accountfrom website	High	Sprint-1
	Prediction	USN-7	As a user, I can prediction result through dashboard by integrated ML Model	I can get prediction by giving valid input	High	Sprint-3
Customer Care / support	Query/ complaint raise	USN-8	As a user, I can raise Query or complaintabout technical issues	If raised query valid or truethen resolve and response, else explain the missing understanding	Medium	Sprint-4
	Feedback/ rating	USN-9	As a user, I can give feedback and rating tothe application	Support team accept the feedback , try to improve application.	Medium	Sprint-4
Administrator	Maintain	USN-10	Administrator maintain the database andoverall application	Punctual maintenance	High	Sprint-4
Developer	Testing	USN-11	As a developer, I test the application which Ihave developed	I test the application for checking errors and rectifyit	High	Sprint-1,2,3,4

5.3 Solution architecture:

Solution architecture is a complex process – with many sub-processes – that bridgesthe gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of thesoftware to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed,and delivered.



5.4 Technology Stack:

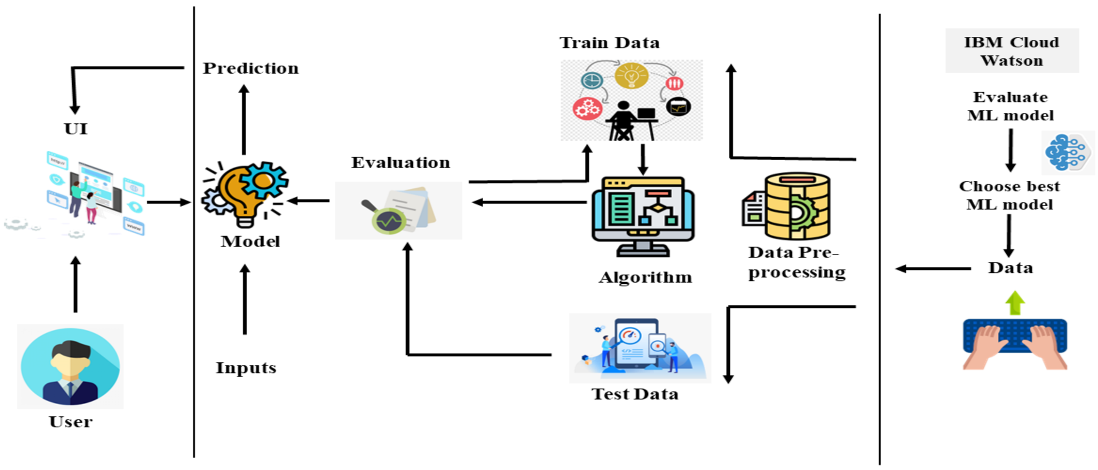


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application web UI.	HTML, CSS.
2.	Application Logic-1	Develop, train and finding best ML model	Python
3.	Application Logic-2	Deploy best ML model	IBM Watson
4.	Application Logic-3	Integrating ML model with flask	IBM Watson, Flask, Python
5.	Database	Structured data	MySQL
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.

7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	Machine Learning Model	Purpose of Machine Learning Model is to predict the delay.	Object Recognition Model,Decision tree classifier.
10.	Infrastructure (Server / Cloud)	Application Deployment on Cloud	Cloud Foundry, Kubernetes.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask, scipy, Jupiter Notebook	microframework
2.	Scalable Architecture	3 – tier, Micro-services	Relational database, cloud, GUI
3.	Availability	application distributed servers	IBM Cloud
4.	Performance	Performance of the application	IBM Watson App services

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Use the below template to create product backlog and sprint schedule

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 by entering my email, password, and confirming my password.	2	High	Sridhar K V Sriram S M Tamilarasan D Vittal J S
Sprint-1	Login	USN-2	As a user, I can log into the application by entering email & password and I can register login to the application through Gmail	2	Medium	Sridhar K V Sriram S M Tamilarasan D
Sprint-2	Basic user interaction Dashboard	USN-3	As a user, I can log into the application byentering email & password	1	Medium	Sridhar K V Sriram S M Tamilarasan D Vittal J S
Sprint-2	Improved Dashboard and GUI	USN-4	As a user, I can use the model or predictionfrom model by interacting with improved dashboard	2	High	Sridhar K VSriram S M
Sprint-3	Model Building	USN-5	As a user, I will receive confirmationemail once I have registered for the application	1	Medium	Sridhar K V Sriram S M Tamilarasan D Vittal J S
Sprint-3	Data Collection and Pre-processing	USN-6	As a user, I can't interact anything. Waiting is user's task. User can listenthe relationship exist between the various attributes of data by presentation of developer	2	High	Sridhar K V Sriram S M Vittal J S
Sprint-3	Model Deployment on IBM Cloud usingIBM Watson	USN-7	As a user, Ican register for the application through Gmail	2	High	Sridhar K V Sriram S M Tamilarasan D Vittal J S
Sprint-4	Raise query/complaint andgive feedback	USN-8	As a user, I can raise complaint or queryand give feedback	1	Medium	Sridhar K V Sriram S M Tamilarasan D Vittal J S

Sprint-4	Improve overall webapp	USN-9	As a user, I can user revised and improved version of web application	2	High	Sridhar K V Sriram S M Tamilarasan D Vittal J S

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	20	6 Days	24 Oct 2022	29 Oct 2022	20	31 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	07 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

7. CODING AND SOLUTIONS

App.py:

```
import email
from email import message
from importlib.resources import contents
from tkinter import S
from turtle import title
from flask import Flask, redirect,render_template, request,session, url_for, Flask
from pyexpat import model
from werkzeug.utils import secure_filename
import ibm_db
from flask_mail import Mail, Message
from markupsafe import escape
from flask import Flask,render_template,request
import requests
# NOTE: you must manually set API_KEY below using information retrieved from your
IBM Cloud account.
API_KEY = "A3SrnPK-7Z8jLS9Zlcmmm-B7lFWjGtRjuPmhXXjpCvQM"
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 = b'_5#y2L"F4Q8z\n\Xec]/'

mail = Mail(app)

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=6667d8e9-9d4d-4ccb-ba32-
```

```
21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30376;SECURITY
=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=trm74992;PWD=7EVyzBSou
gGI2vwn",",")
print(conn)
print("connection successful...")
@app.route('/', methods = ['GET','POST'])
def signup():
return render_template('signup.html')


@app.route('/login', methods=['GET','POST'])
def login():
return render_template('login.html')


@app.route('/logout')
def logout():
return render_template('login.html')


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


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


@app.route('/register', methods=['GET', 'POST'])
def register():
if request.method == 'POST':
uname = request.form['uname']
```

```
mail = request.form['email']
phone = request.form['phone']
password = request.form['password']

sql = "SELECT * FROM customer WHERE email=?"
stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt,1,mail)
ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)

if account:
return render_template('index.html', msg="You are already a member, please login using
your details....")

else:
insert_sql = "INSERT INTO customer VALUES (?,?,?,?)"
prep_stmt = ibm_db.prepare(conn, insert_sql)
ibm_db.bind_param(prep_stmt, 1, uname)
ibm_db.bind_param(prep_stmt, 2, mail)
ibm_db.bind_param(prep_stmt, 3, phone)
ibm_db.bind_param(prep_stmt, 4, password)
ibm_db.execute(prep_stmt)

return render_template('login.html', msg="Student Data saved successfuly..")



@app.route('/signin', methods=['GET', 'POST'])
def signin():
sec = ''
if request.method == 'POST':

mail = request.form['email']
password = request.form['password']
```

```
sql = f"select * from customer where email='{escape(mail)}' and password=
'{escape(password)}'"
stmt = ibm_db.exec_immediate(conn, sql)
data = ibm_db.fetch_both(stmt)

if data:

session["mail"] = escape(mail)
session["password"] = escape(password)
return redirect(url_for('index'))

else:
return render_template('login.html',msg = "Invalid email/ Password or Not registered!!?")

return "not going to happen dickhead!!??"



@app.route('/prediction',methods=["POST"])
def predict():
if request.method=="POST":
name=request.form["name"]
month=request.form["month"]
if(int(month)>12):
ans="Please Enter the correct Month"
return render_template("index.html" ,y=ans)

dayofmonth=request.form["dayofmonth"]
if(int(dayofmonth)>31):
ans="Please Enter the correct Day of Month"
return render_template("index.html" ,y=ans)

dayofweek=request.form["dayofweek"]
if(int(dayofweek)>7):
ans="Please Enter the correct Day of Week"
```

```
return render_template("index.html" ,y=ans)


origin=request.form["origin"]
destination=request.form['destination']

if(origin==destination):
ans="Origin airport and destination airport can't be same"
return render_template("index.html" ,y=ans)

if(origin=="msp"):
origin1,origin2,origin3,origin4,origin5=0,0,0,1,0
if(origin=="dtw"):
origin1,origin2,origin3,origin4,origin5=0,1,0,0,0
if(origin=="jfk"):
origin1,origin2,origin3,origin4,origin5=0,0,1,0,0
if(origin=="sea"):
origin1,origin2,origin3,origin4,origin5=0,0,0,0,1
if(origin=="alt"):
origin1,origin2,origin3,origin4,origin5=1,0,0,0,0



if(destination=="msp"):
destination1,destination2,destination3,destination4,destination5=0,0,0,1,0
if(destination=="dtw"):
destination1,destination2,destination3,destination4,destination5=0,1,0,0,0
if(destination=="jfk"):
destination1,destination2,destination3,destination4,destination5=0,0,1,0,0
if(destination=="sea"):
destination1,destination2,destination3,destination4,destination5=0,0,0,0,1
if(destination=="alt"):
destination1,destination2,destination3,destination4,destination5=1,0,0,0,0
```

```
depthr=request.form['depthr']
deptmin=request.form['deptmin']
if(int(depthr)>23 or int(deptmin)>59):
ans="Please enter the correct Departure time"
return render_template("index.html" ,y=ans)
else:
dept=depthr+deptmin

actdepthr=request.form['actdepthr']
actdeptmin=request.form['actdeptmin']
if(int(actdepthr)>23 or int(actdeptmin)>59):
ans="Please enter the correct Actual Departure time"
return render_template("index.html" ,y=ans)
else:
actdept=actdepthr+actdeptmin



arrtimehr=request.form['arrtimehr']
arrtimemin=request.form['arrtimemin']
if(int(arrtimehr)>23 or int(arrtimemin)>59):
ans="Please enter the correct Arrival time"
return render_template("index.html" ,y=ans)
else:
arrtime=arrtimehr+arrtimemin


if((int(actdept)-int(dept))<15):
dept15=0
else:
dept15=1

print(dept15)
```

```
total=[[int(month),int(dayofmonth),int(dayofweek),int(origin1),int(origin2),int(origin3),int(or
igin4),int(origin5),int(destination1),int(destination2),int(destination3),int(destination4),int(de
stination5),int(dept),int(actdept),int(dept15),int(arrtime)]]
print(total)
# NOTE: manually define and pass the array(s) of values to be scored in the next line
payload_scoring = {"input_data": [{"fields":
["int(month)","int(dayofmonth)","int(dayofweek)","int(origin1)","int(origin2)","int(origin3)",
"int(origin4)","int(origin5)","int(destination1)","int(destination2)","int(destination3)","int(des
tination4)","int(destination5)","int(dept)","int(actdept)","int(dept15)","int(arrtime)"],
"values": total}]}

response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/5b2670ac-b4ed-4173-a575-
bf3383144c03/predictions?version=2022-11-15', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")
print(response_scoring.json())

pred = response_scoring.json()
value = pred['predictions'][0]['values'][0][0]

print(value)
if value==0:
ans="THE FLIGHT WILL BE ON TIME"
else:
ans="THE FLIGHT WILL BE DELAYED"

return render_template("results.html" ,y=ans)

if _name_ == "_main_":
app.run(debug=True)
```

INDEX.HTML:

```
<!DOCTYPE html>

<head>
</head>
<link rel="stylesheet" type="text/css" href="{{ url_for('static', filename= 'css/main.css') }}"
/>

<link rel="stylesheet" href="../static/style.css">

<body>
<form name="register" class="form-group" action="/prediction" method="post">
<center>

<ul>
<li style="color:white;font-weight: bolder;font-size: 45px;">Flight Delay Prediction
</li>
<li style="float:right"><a class="active" href=" {{ url_for('index') }}">Home</a></li>
<li style="float:right"><a class="active" href="{{ url_for('logout') }}">Log out</a></li>
<li style="float:right"><a class="active" href="{{ url_for('about') }}">About</a></li>
</ul>
<table>
<tr>
<td class="col-25"> <label>Enter Flight Number</label></td>
<td class="col-75"> <input class="cols" type="text" placeholder="Enter Flight Number"
required name="name">
</td>
</tr>
<tr>
<td class="col-25"><label>Month</label></td>
<td class="col-75"><input class="cols" type="text" placeholder="Month" required
name="month"></td>
</tr>
<tr>
```

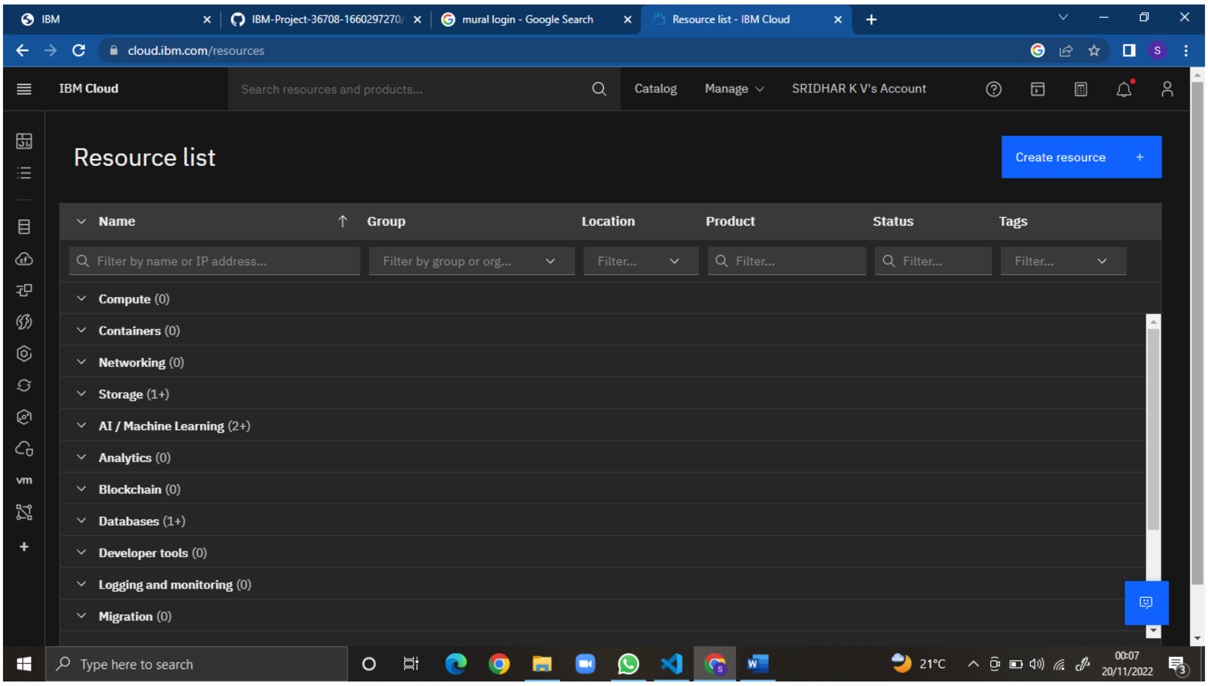
```
<td class="col-25"><label> Day of month</label></td>
<td class="col-75"> <input class="cols" type="text" placeholder="Day of month" required
name="dayofmonth"><br>
</td>
</tr>
<tr>
<td class="col-25"><label>Day of week</label></td>
<td class="col-75"> <input class="cols" type="text" placeholder="Day of week" required
name="dayofweek"><br>
</td><br>
</tr>
<tr>
<td class="col-25"><label>Origin</label></td>
<td class="col-75">
<div class="select">
<select id="country" required name="origin" placeholder="select origin">
<option value="" disabled selected>Select origin</option>
<option value="alt" selected style="color: black;">ATL - Hartsfield-Jackson Atlanta
International
</option>
<option value="dtw" style="color: black;">DTW - Detroit Metropolitan Wayne
County</option>
<option value="Sea" style="color: black;">SEA - Seattle-Tacoma International</option>
<option value="msp" style="color: black;">MSP - Minneapolis−Saint Paul
International</option>
<option value="jfk" style="color: black;">JFK - John F. Kennedy International</option>
</select>
</div>
</td>
</tr>

<tr>
<td class="col-25"><label>Destination</label></td>
<td class="col-75">
```

```
<select id="region" required name="destination">
<option value="alt" selected style="color: black;">ATL - Hartsfield-Jackson Atlanta
International</option>
<option value="dtw" style="color: black;">DTW - Detroit Metropolitan Wayne
County</option>
<option value="sea" style="color: black;">SEA - Seattle-Tacoma International</option>
<option value="msp" style="color: black;">MSP - Minneapolis-Saint Paul
International</option>
<option value="jfk" style="color: black;">JFK - John F. Kennedy International</option>
</select>
</td>
<tr>
<td class="col-25"><label>Scheduled Departure Time</label> </td>
<td class="col-75"><label>Hour : </label>
<input placeholder="00" class="cols" type="text" required name="depthr" />
<label>Minutes : </label>
<input placeholder="00" class="cols" type="text" required name="deptmin" />
</td>
</tr>
<tr>
<td class="col-25"><label>Actual Departure Time </label> </td>
<td class="col-75"><label>Hour : </label>
<input placeholder="00" class="cols" type="text" required name="actdepthr" />
<label>Minutes : </label>
<input placeholder="00" class="cols" type="text" required name="actdeptmin" />
</td>
</tr>
<tr>
<td class="col-25"><label>Scheduled Arrival Time </label> </td>
<td class="col-75"><label>Hour : </label>
<input placeholder="00" class="cols" type="text" required name="arrtimehr" />
<label>Minutes : </label>
<input placeholder="00" class="cols" type="text" required name="arrtimemin" />
</td>
```

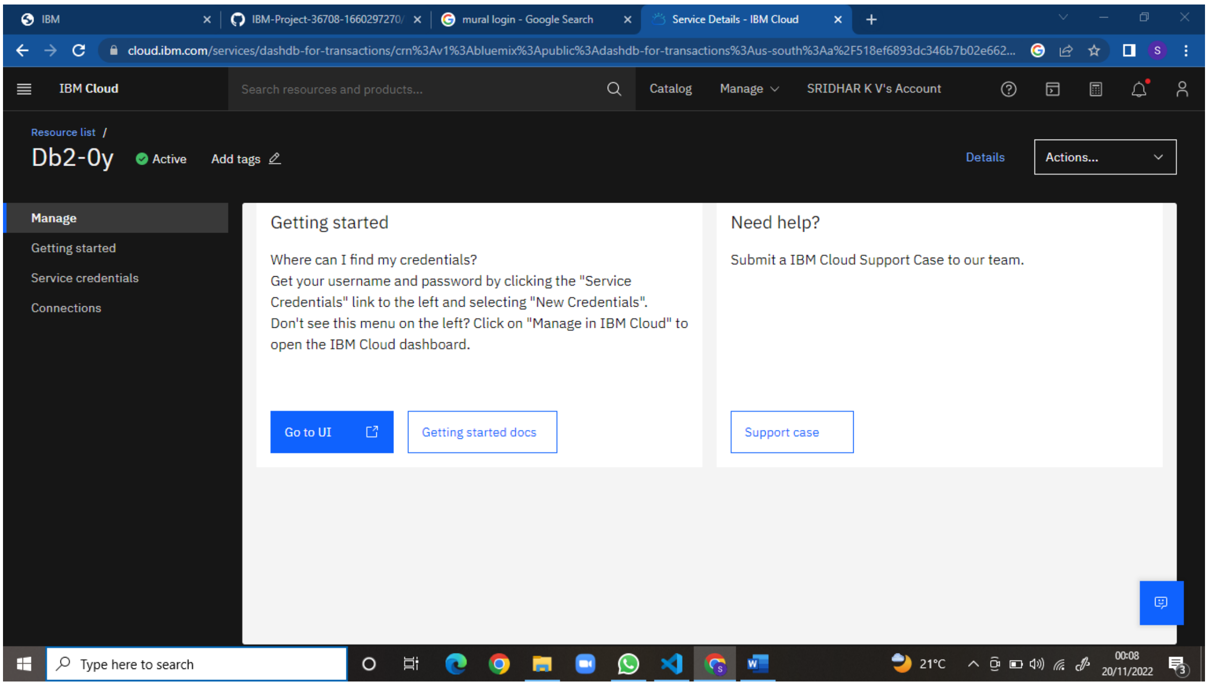
```
</tr>
</table>
<center> <br><button class="glow-button" type="submit"
name="Predict">Predict</button></center>
</form>
<p>{{y}}</p>
</center>
</body>
</html>
```

8. IBM CLOUD SERVICES

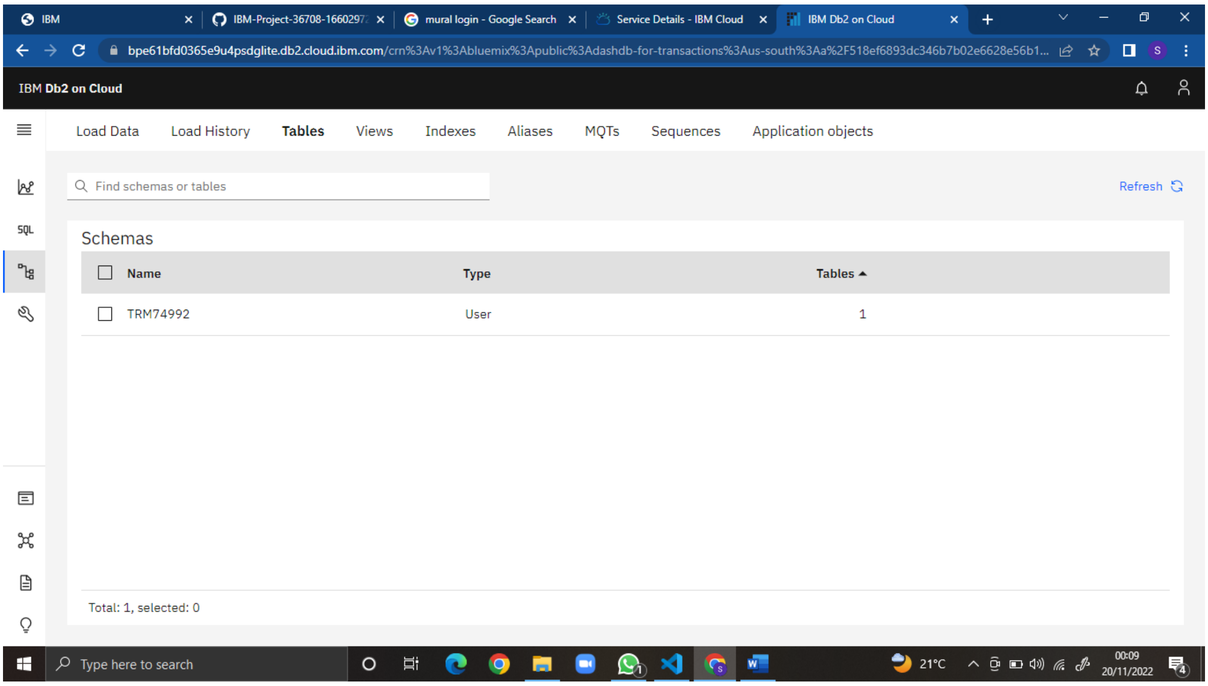
8.1 IBM CLOUD:



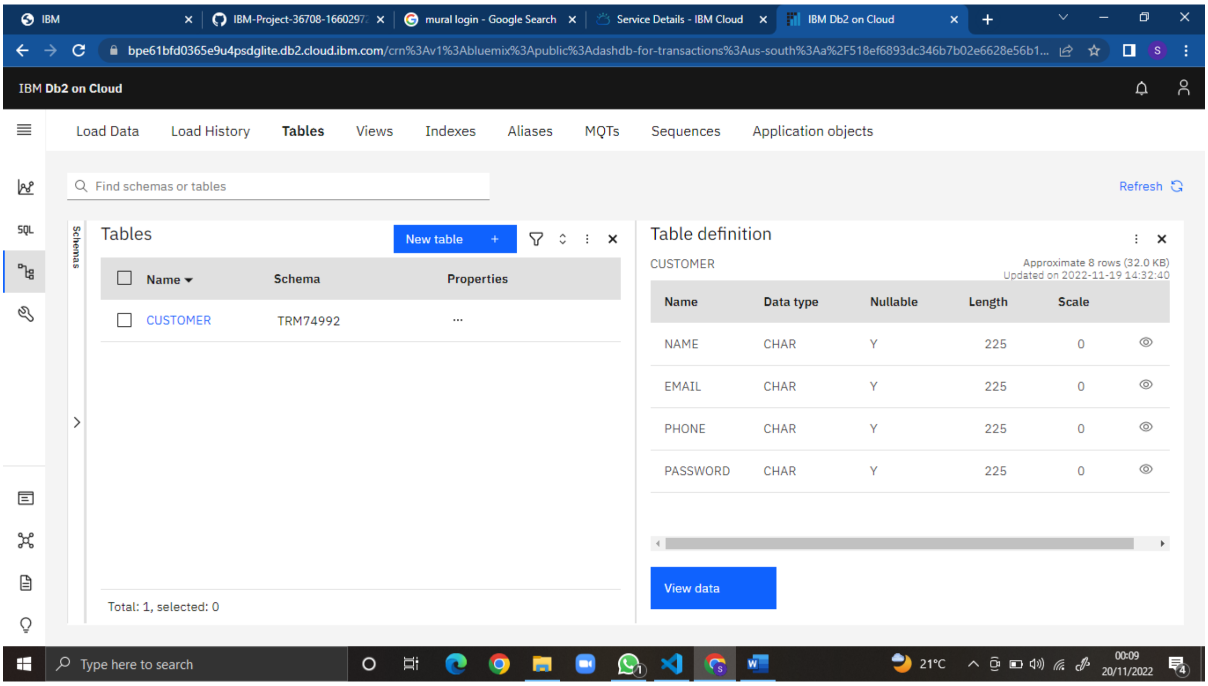
8.2 IBM DATABASE:



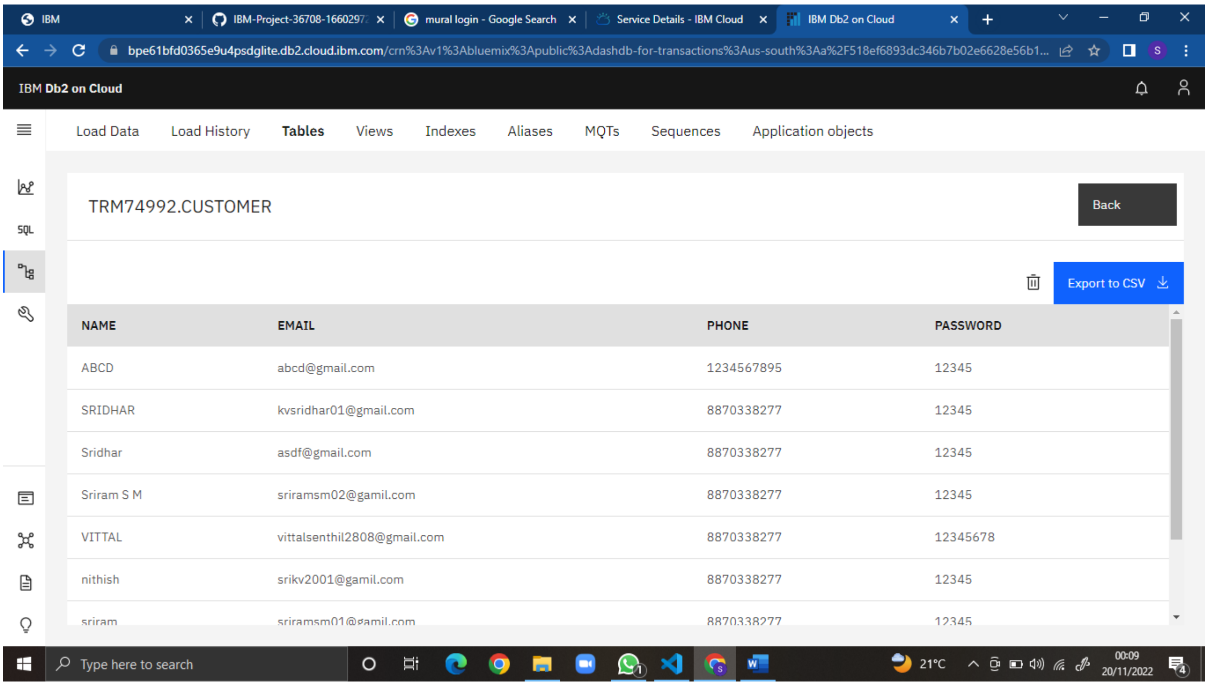
8.3 IBM SCHEMA:



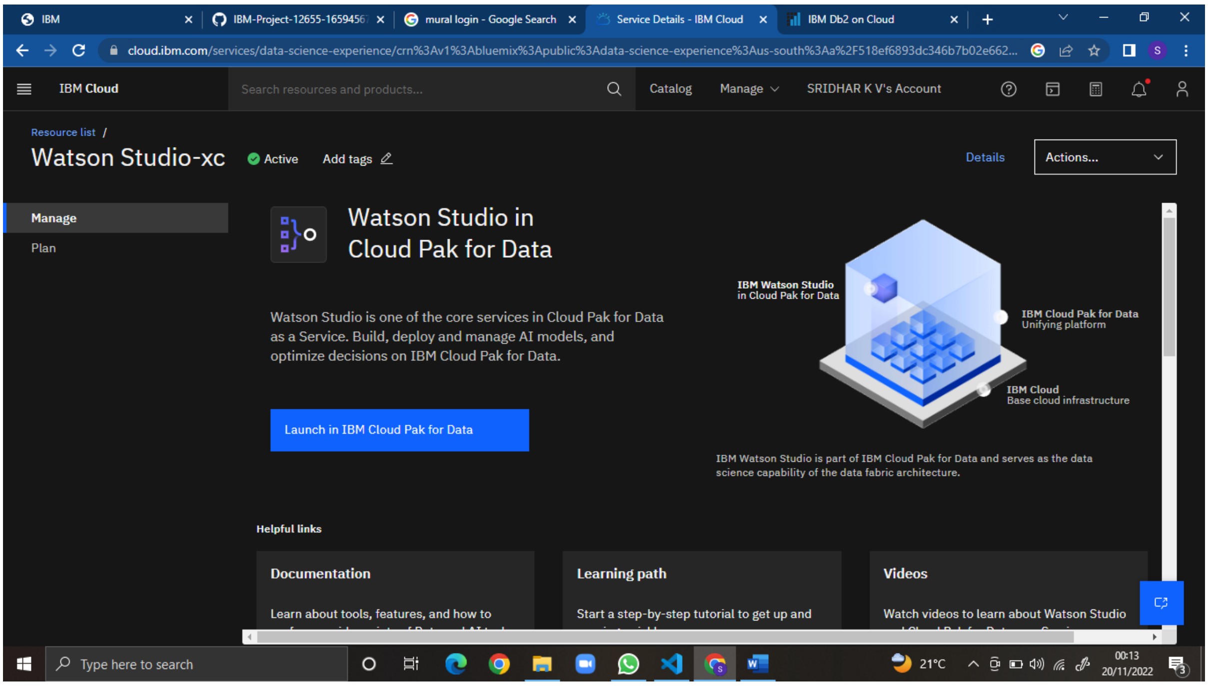
8.4 IBM DATABASE TABLE:



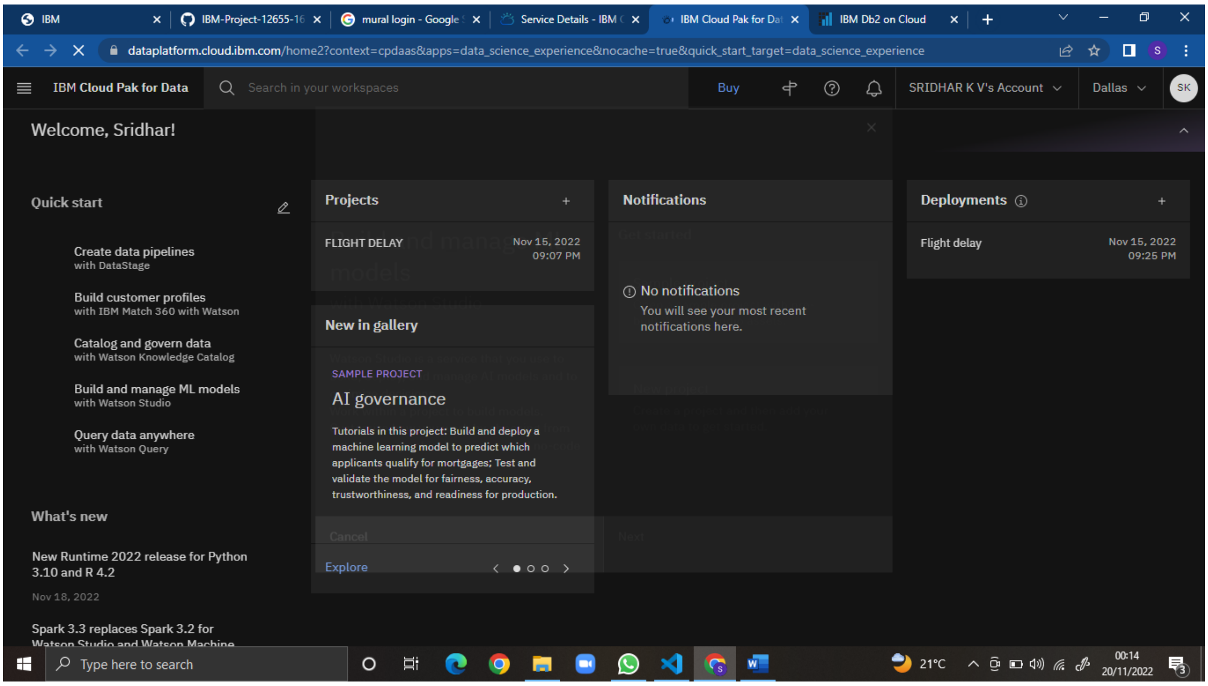
8.5 IBM USER DATA



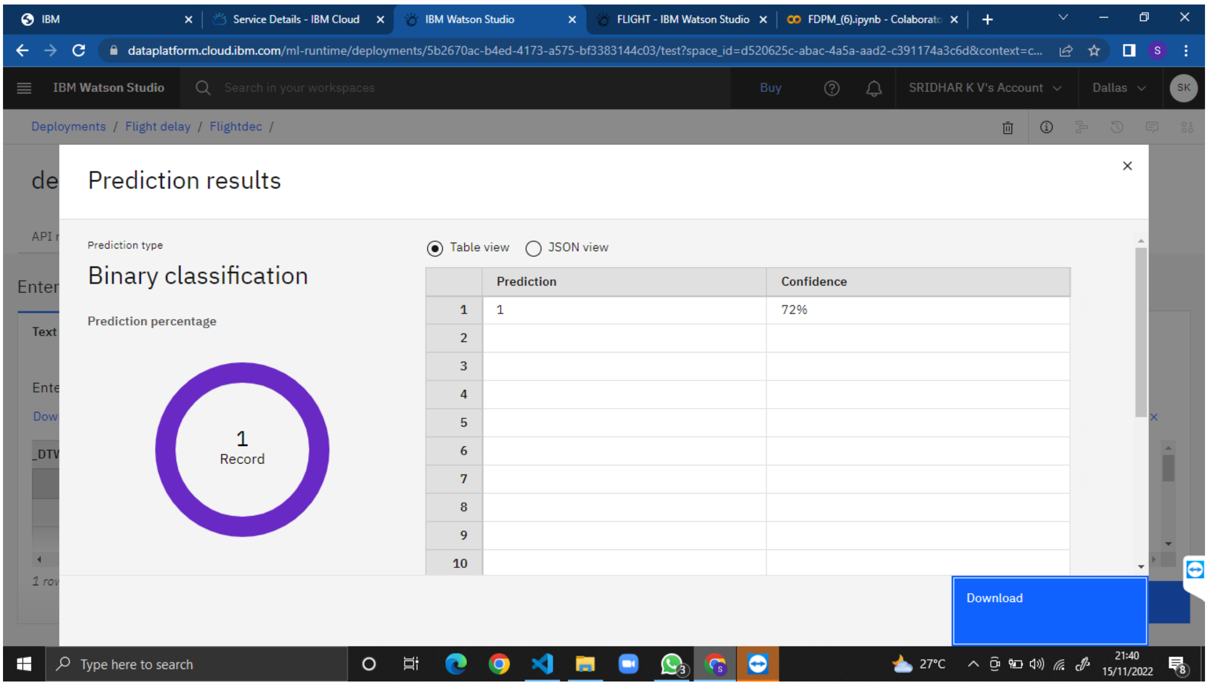
8.6 IBM WATSON STUDIO:



8.7 IBM PROJECT DEPLOYMENT

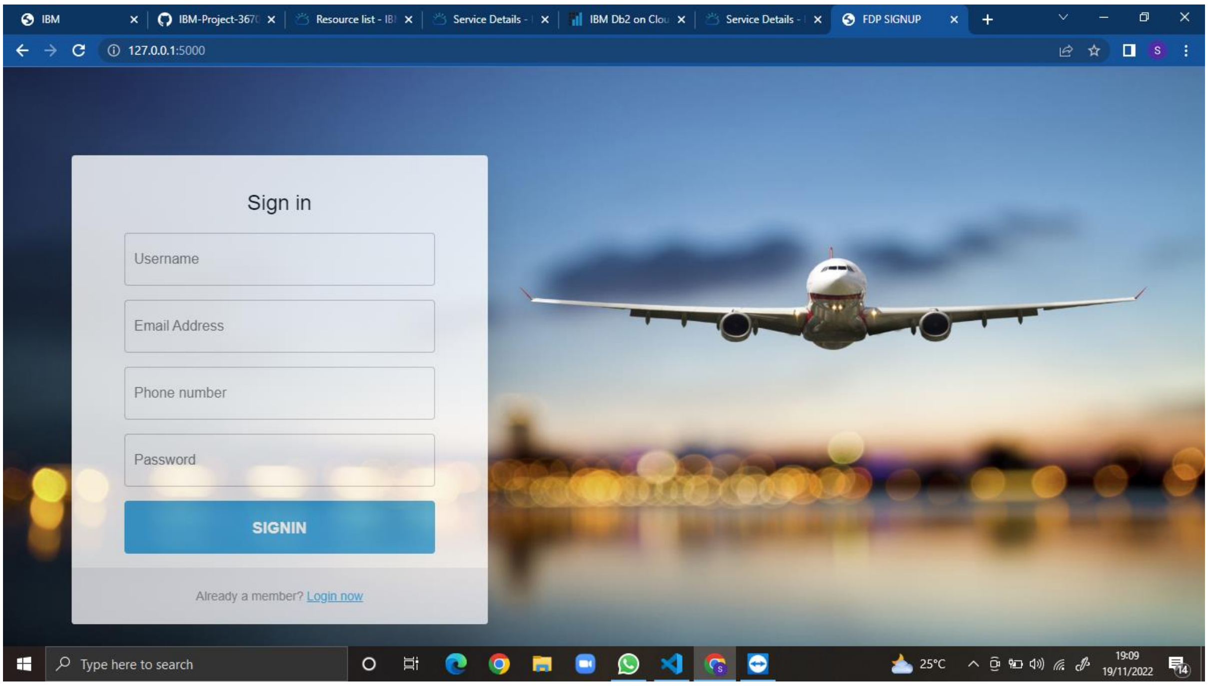


8.8 PREDICTION RESULT

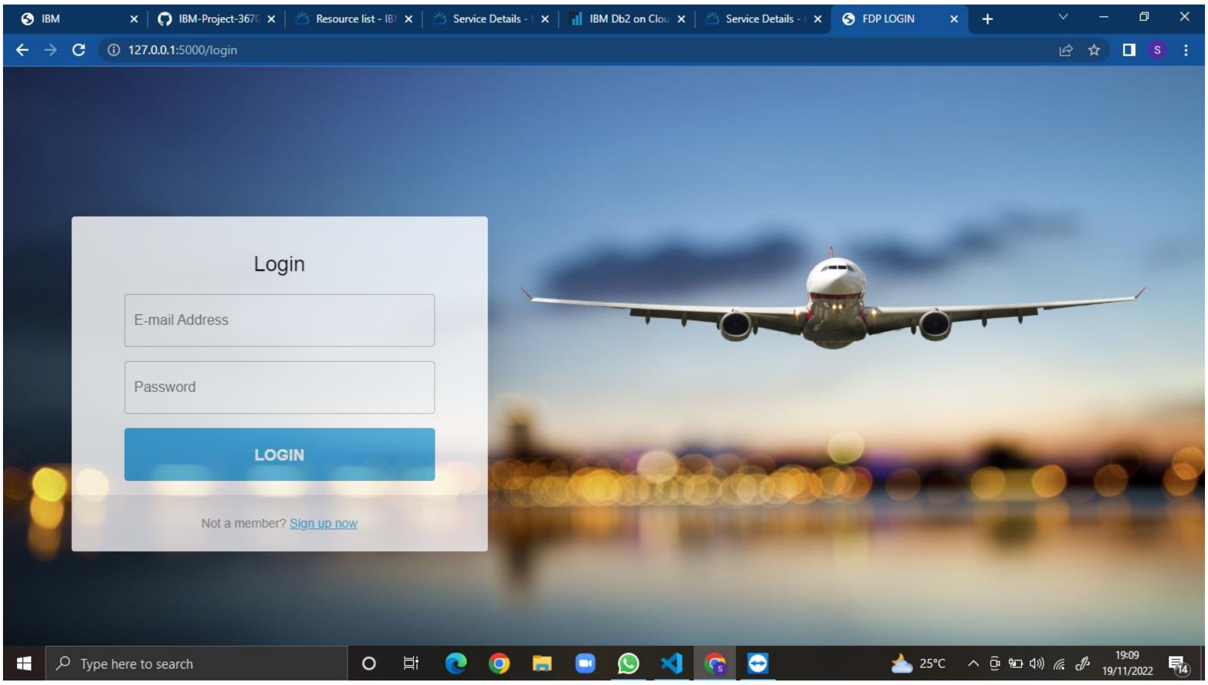


9.OUTPUTS AND SCREENSHOTS

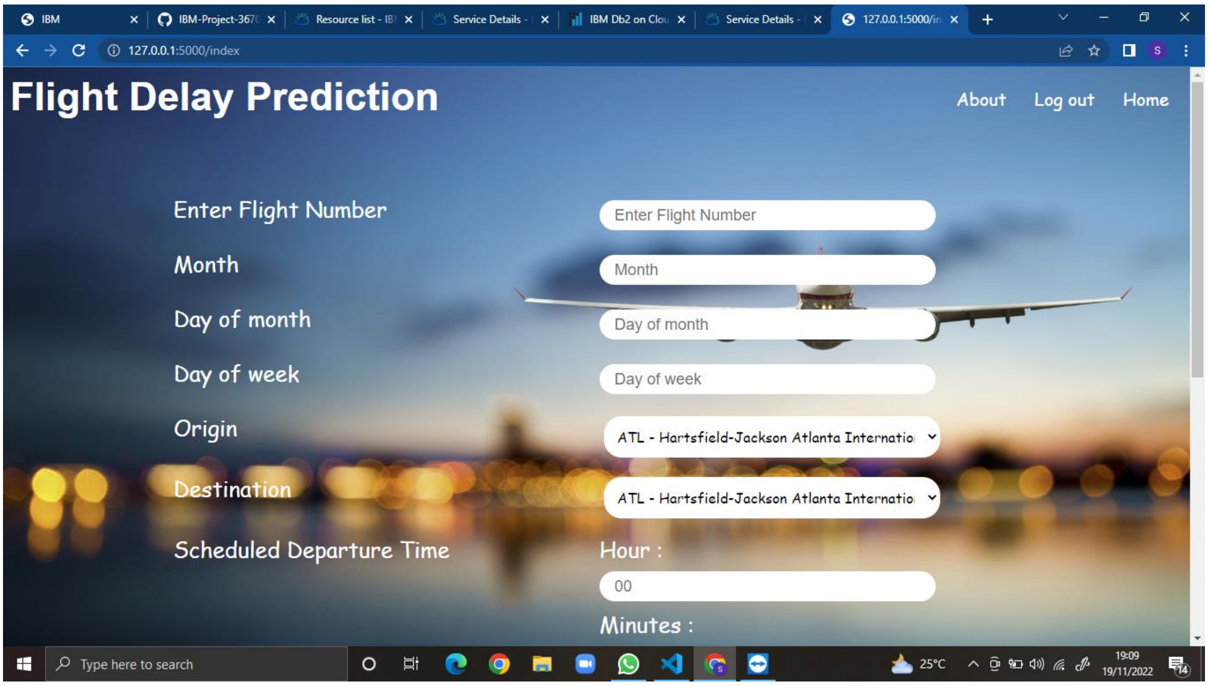
9.1 SIGNUP PAGE :



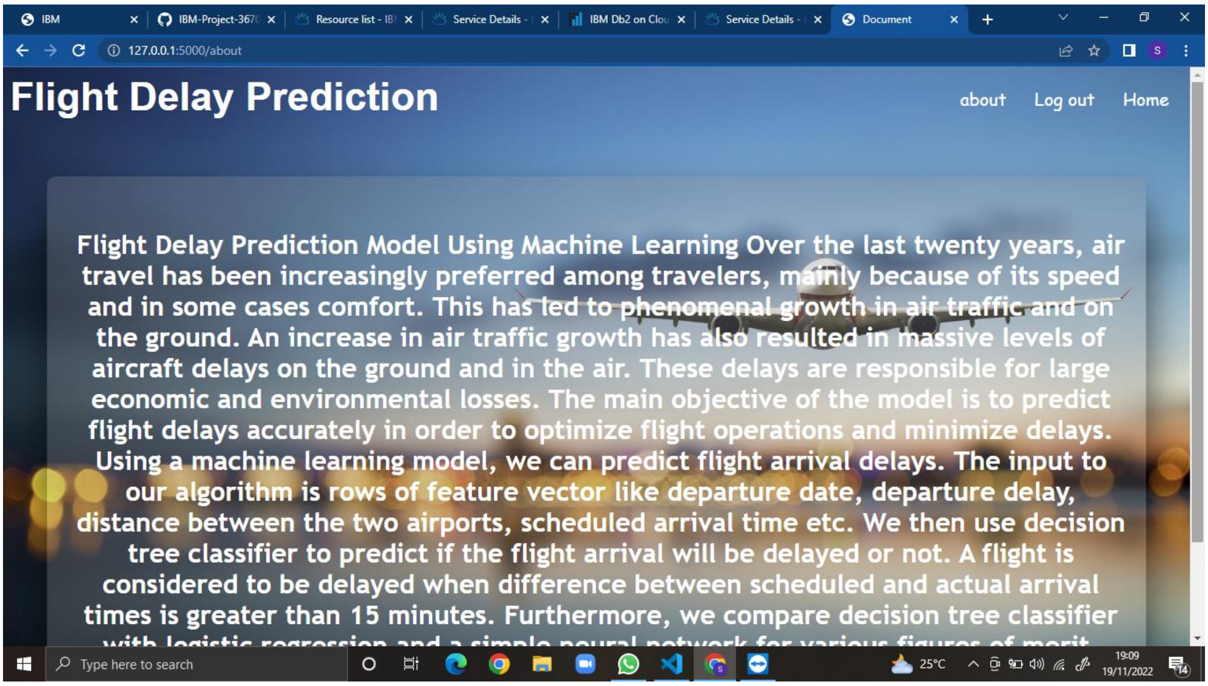
9.2 LOGIN PAGE:



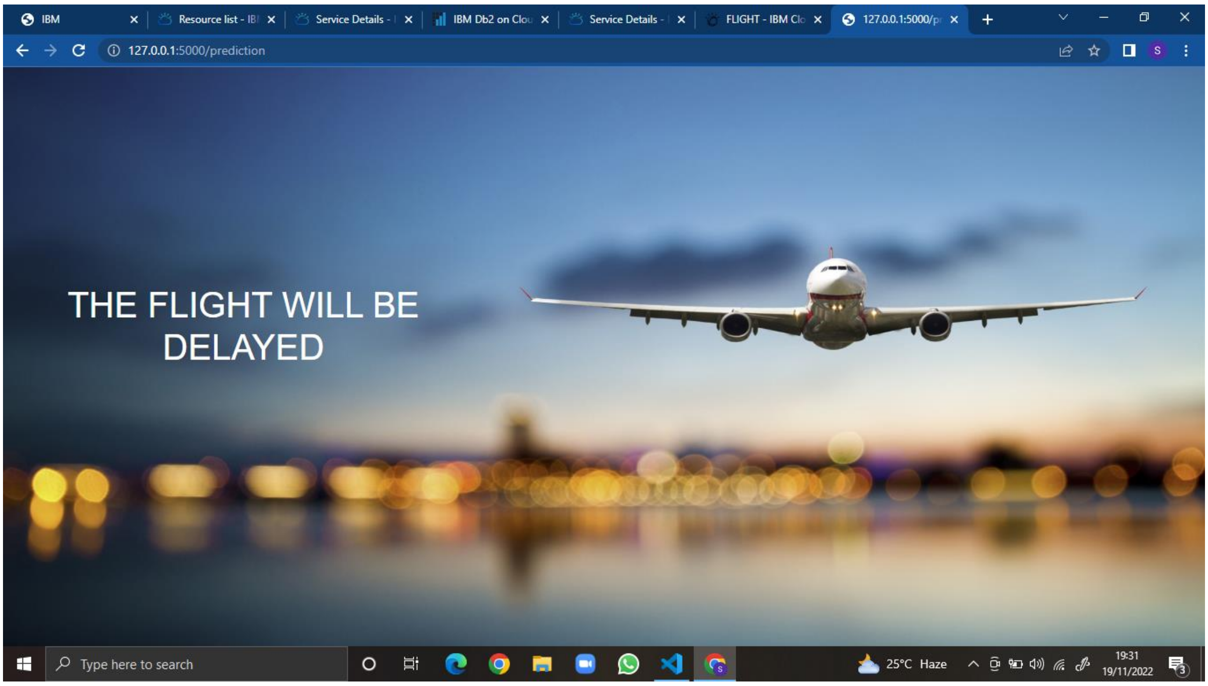
9.3 HOME PAGE :

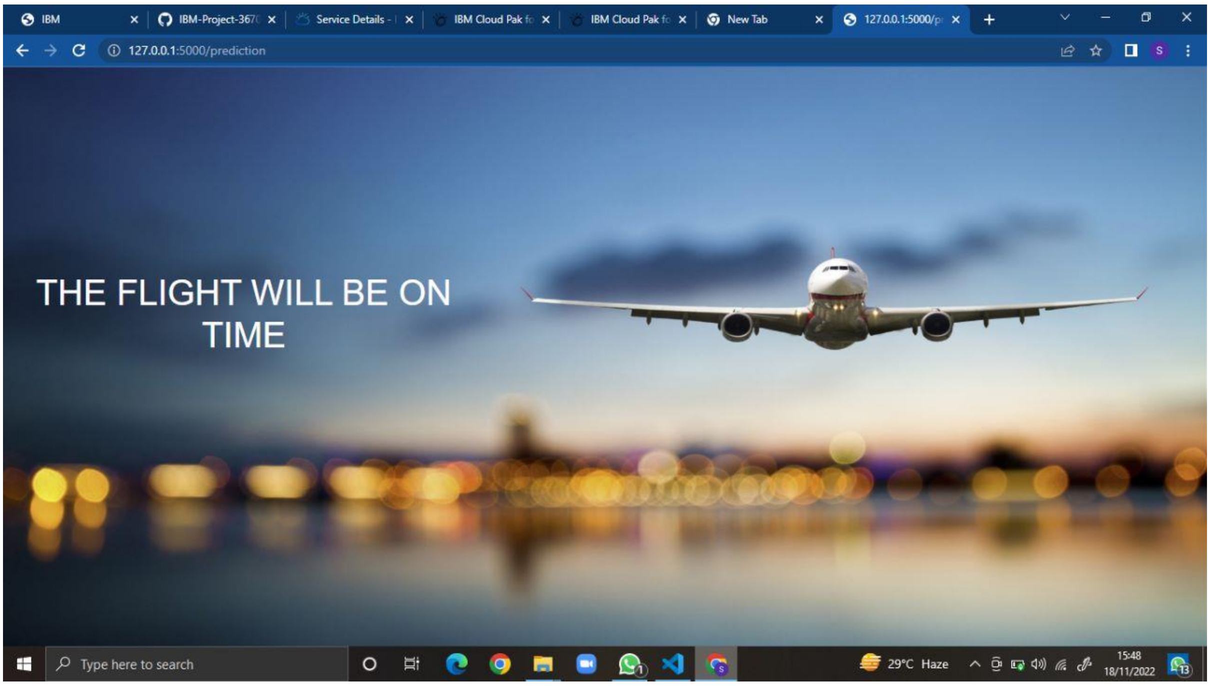


9.4 ABOUTUS PAGE :



9.5 RESULT PAGE:





10. ADVANTAGES & DISADVANTAGES

The model is fast in predicting and uses decision trees. The Decision tree classifier requires less training time and less data to train the model. The project's disadvantage is that the delay's extent is not predicted.

11. CONCLUSION

Predicting flight delays is on interesting research topic and required many attentions these years. Majority of research have tried to develop and expand their models in order to increase the precision and accuracy of predicting flight delays. Since the issue of flights being on-time is very important, flight delay prediction models must have high precision and accuracy. This report explains the implementation of the flight delay prediction system and the metrics used for testing the system. The flight delay prediction system is successfully deployed in the internet using render for viewing the webpage.

12. FUTURE SCOPE

Therefore, the future work of this project includes incorporating a larger dataset. There are many different ways to preprocess a larger dataset like running a Spark cluster over a server or using a cloud-based services like AWS and Azure to process the data.

With the new advancement in the field of deep learning, we can use Neural Networks algorithm on the flight and weather data. Neural Network works on the pattern matching methodology.

It is divided into three basic parts for data modelling that includes feed forward networks, feedback networks, and self organization network. Feed-forward and feedback networks are generally used in the areas of prediction, pattern recognition, associative memory, and optimization calculation, whereas self-organization networks are generally used in cluster analysis.

Neural Network offers distributed computer architecture with important learning abilities to represent nonlinear relationships. Also, the scope of this project is very much confined to flight and weather data of United States, but we can include more countries like China, India, and Russia. Expanding the scope of this project, we can also add the flight data from international flights and not just restrict our self to the domestic flights.

13. APPENDIX

GITHUB LINK: https://github.com/IBM-EPBL/IBM-Project-36708-1660297270

DEMO LINK: https://github.com/IBM-EPBL/IBM-Project-36708-1660297270/tree/main/Final%20Deliverables/DEMO%20VIDEO