PROJECT BASED EXPERIENTIAL LEARNING -
PROGRAM (NALAIYA THIRAN)

DEVELOPING A FLIGHT DELAY PREDICTION MODEL USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

TAMIL MANI P	(111719106158)
SARAVANA KUMAR P B	(111719106139)
SRIKANTH MU	(111719106150)
SURIYA RAAJ P	(111719106153)

TEAM ID : PNT2022TMID16204

DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING

R.M.K. ENGINEERING COLLEGE
(An Autonomous Institution)
R.S.M. Nagar, Kavaraipettai-601 206

NOVEMBER 2022

CONTENTS

1. CHAPTER 1: INTRODUCTION

Travelers have begun to favor air travel more and more over the past 20 years,primarily due to its quickness and occasional comfort. Both on the ground and in the air, as a result, have experienced amazing growth. Massive amounts of ground and airborne aircraft delays have also been brought on by an increase in air traffic. Large economic and environmental losses are the result of these delays. The model's primary goal is to correctly forecast flight delays in order to improve aircraft operations and reduce delays.

1.1 PROJECT OVERVIEW

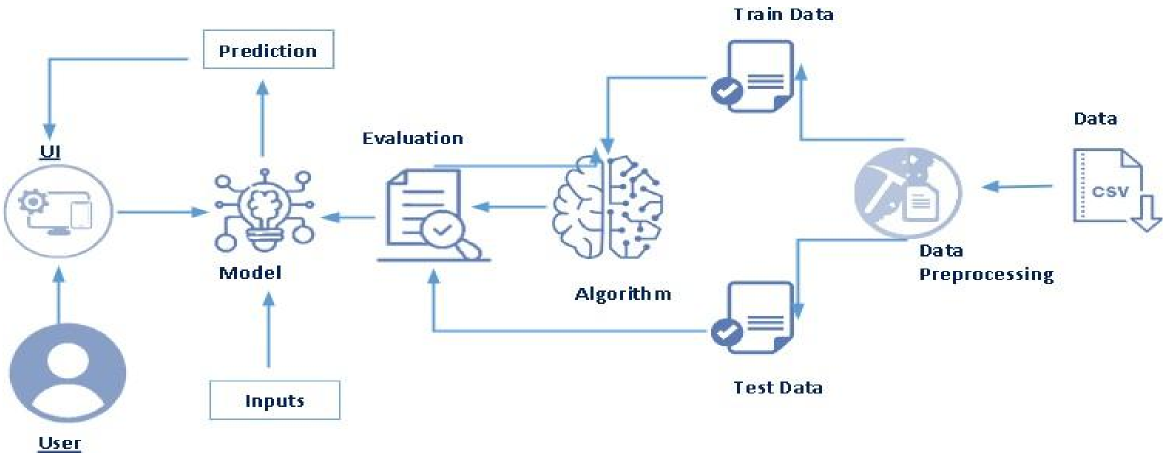


Figure 1.1. Technical Architecture

Flight arrival delays can be predicted using a machine learning algorithm. Rows of feature vectors, such as departure date, delay, travel time between the two airports, and scheduled arrival time, provide the input to our algorithm. The decision tree classifier is then used to determine whether or not the flight arrival will be delayed. When there is more than a 15-minute gap between the scheduled and actual arrival timings, a flight is deemed to be delayed. For various figures of merit, we contrast the decision tree classifier with logistic regression and a straightforward neural network.

1.2. PURPOSE

The main goal of this project is to predict the flight delay using machine learning algorithms. Flight planning is one of the difficulties in the industrial environment because there are many unpredictabilities. One such condition is the incidence of delays, which can result from a variety of causes and impose significant expenses on airlines, operators, and passengers. Delays in departure can be brought on by inclement weather, seasonal and holiday demands, airline policies, technical issues with airport infrastructure, baggage handling, and mechanical equipment, and a buildup of delays from earlier flights. Hence Predicting flight delays can improve airline operations and passenger satisfaction, which will result in a positive impact on the economy.

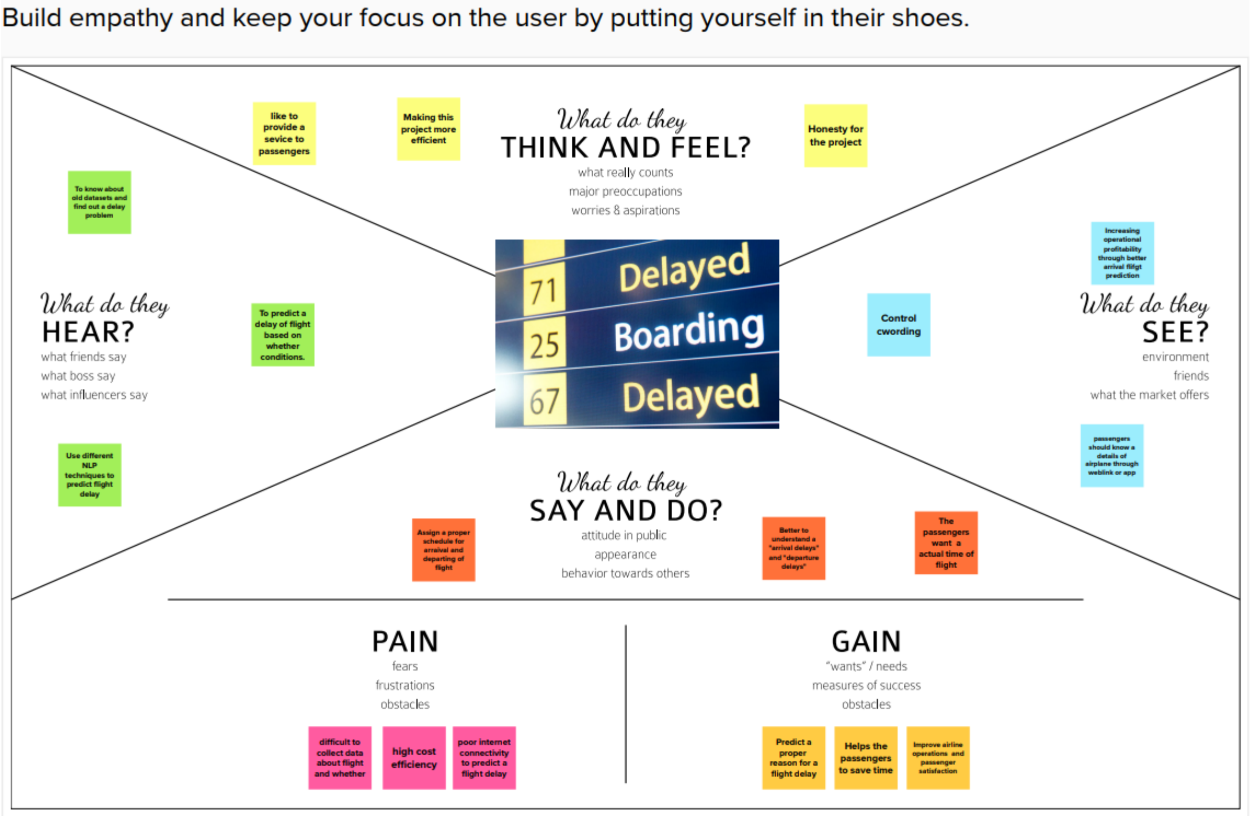
2. LITERATURE SURVEY

S. NO	TITLE	AUTHOR	ABSTRACT	DRAWBACKS
1.	Flight Delay Prediction	Alice Sternberg, Jorge Soares, Diego Carvalho, Eduardo Ogasawara	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. Here , proposed 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 , also present a timeline of significant works that depicts relationships between flight delay prediction problems and research trends to address them.	Dimensions were not directly related to the type of problem, but to the scope of application. This characteristic is notable in this case. Attributes such as weather, capacity, and demand were characteristics of airport or enroute airspace scopes. On the other hand, airlines schedules indicated scopes that considered airlines elements. It was also observed several ensembles of different dimensions, showing that prediction models may be improved through the selection of different attributes.
2.	Flight Delay Prediction System	Mrs Yogita Borse, Dhruvin Jain, Shreyash Sharma, Viral Vora, Aakash Zaveri	Flight Planning is one of the challenges in industrial world which faces many uncertain conditions. One such condition is delay occurrence, which stems from various factors and imposes considerable costs on airlines, operators, and travelers. Delays in departure can occur due to bad weather conditions, seasonal and holiday demands, airline policies, technical issue such as problems in airport facilities, luggage handling and mechanical apparatus, and accumulation of delays from preceding flights. Here in flight delay prediction system based on the weather parameters which can result in delays. The system considers the temperature, humidity, rain in mm, visibility and month number as important parameters for prediction of delay.	Results in this system is not so accurate. 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 which is not considered in this Prediction System.
3.	Flight Delay Prediction	Vishrut Raj , Viran Raj, Satyam Singh , Adityanath Mishra	Fight delay prediction is fundamental to establish the more efficient airline business. The development of accurate prediction models for flight delays became cumbersome due to the complexity of air transportation system, the number of	This model only included one-year data due to our computation capability, as more years of data included, the prediction could be

			methods for prediction, and the deluge of flight data. The paper presents a thorough literature review of approaches used to build flight delay prediction model. Airlines delays make immense loss for business field as well as in budget loss for a country. Flight delays hurt airlines, airports, and passengers. Here, proposing a machine learning algorithms like Linear regression Techniques. The aim of this research work is to predict Flight Delay, which is highest economy producing field for many countries and among many transportations this one is fastest and comfort, so to identify and reduce flight delays, can dramatically reduce the flight delays to saves huge number of turnovers, using machine-learning algorithms. Flight delays could always be annoying, especially in the case when the period of delay was so long that there was even a danger to miss the next flight. However, if there was a way to predict whether there would be a delay or even better – how long the delay could be, then people could make earlier preparation to reschedule following flights in an earlier manner.	easier. In addition, some other related information such as airplane type, e.g., detailed weather data specific to airport were not included. Therefore, researchers could try to collect more related data and deploy better computational powers to build a better mode.

CHAPTER 3: IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



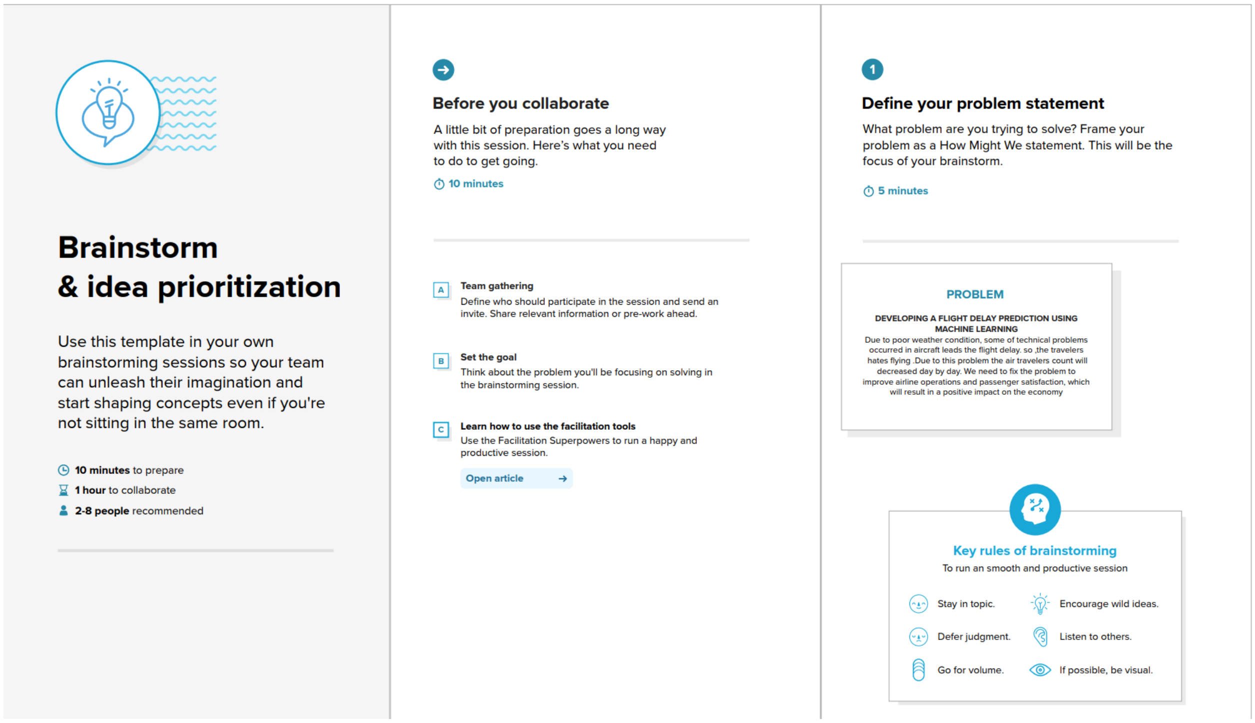
Build empathy and keep your focus on the user by putting yourself in their shoes.

What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
like to provide a sevice to passengers
Making this project more efficient
Honesty for the project
To know about old datasets and find out a delay problem
What do they
HEAR?
what friends say
what boss say
what influencers say
To predict a delay of flight based on whether conditions.
Use different NLP techniques to predict flight delay
71 Delayed
25 Boarding
67 Delayed
Control cwording
Increasing operational profitability through better arrival flifgt prediction
What do they
SEE?
environment
friends
what the market offers
passengers should know a details of airplane through weblink or app
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
Assign a proper schedule for arraival and departing of flight
Better to understand a "arrival delays" and "departure delays"
The passengers want a actual time of flight
PAIN
fears
frustrations
obstacles
difficult to collect data about flight and whether
high cost efficiency
poor internet connectivity to predict a flight delay
GAIN
"wants" / needs
measures of success
obstacles
Predict a proper reason for a flight delay
Helps the passengers to save time
Improve airline operations and passenger satisfaction

Figure 3.1. Empathy Map

3.2. IDEATION AND BRAINSTORMING

Step 1 - Team Gathering, Collaboration and Selecting the Problem Statement



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

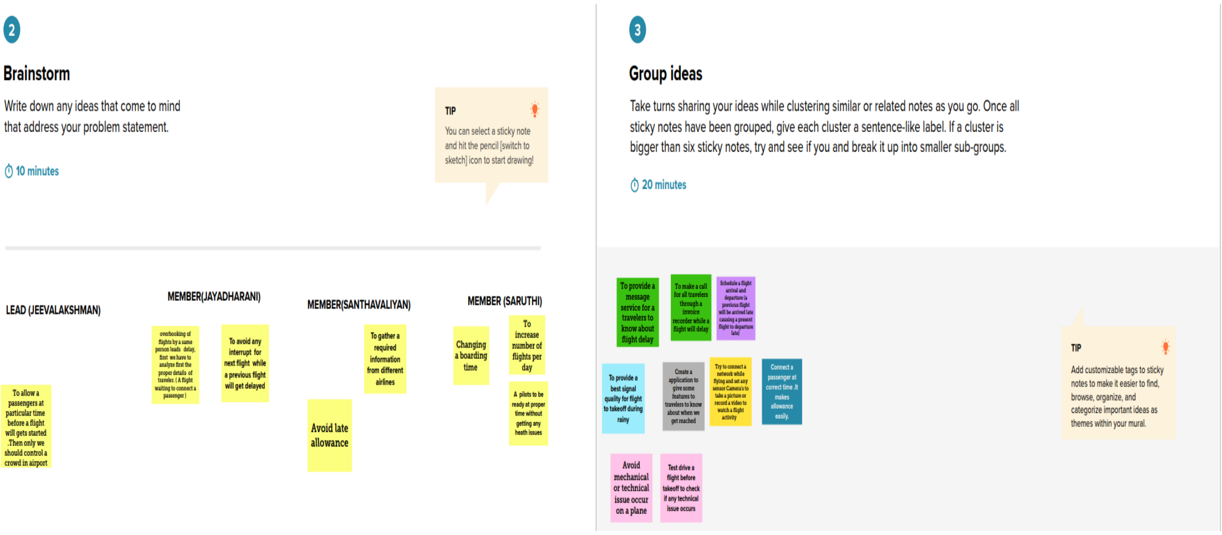
PROBLEM

DEVELOPING A FLIGHT DELAY PREDICTION USING MACHINE LEARNING
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

Key rules of brainstorming
To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

Step 2 - Brainstorm, Idea Listing and Grouping



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP
You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

LEAD (JEEVALAKSHMAN)

To allow a passengers at particular time before a flight will gets started .Then only we should control a crowd in airport

MEMBER(JAYADHARANI)

overbooking of flights by a same person leads delay, first we have to analyze first the proper details of traveler. (A flight waiting to connect a passenger)

To avoid any interrupt for next flight while a previous flight will get delayed

MEMBER(SANTHAVALIYAN)

To gather a required information from different airlines

Avoid late allowance

MEMBER (SARUTHI)

Changing a boarding time

To increase number of flights per day

A pilots to be ready at proper time without getting any heath issues

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

20 minutes

To provide a message service for a travelers to know about flight delay

To make a call for all travelers through a invoice recorder while a flight will delay

Schedule a flight arrival and departure (a previous flight will be arrived late causing a present flight to departure late)

To provide a best signal quality for flight to takeoff during rainy

Create a application to give some features to travelers to know about when we get reached

Try to connect a network while flying and set any sensor Camera's to take a picture or record a video to watch a flight activity

Connect a passenger at correct time .It makes allowance easily.

Avoid mechanical or technical issue occur on a plane

Test drive a flight before takeoff to check if any technical issue occurs

TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

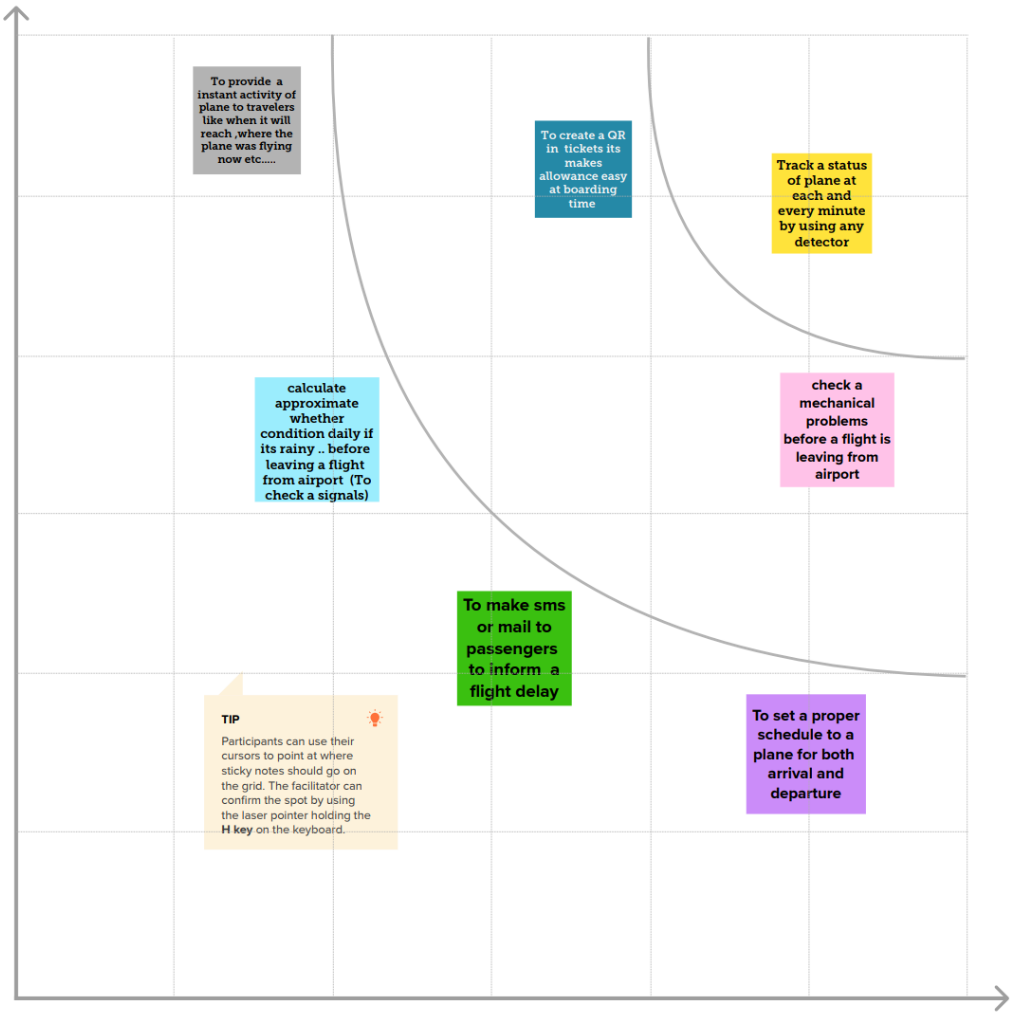
Step 3 - Idea Prioritization

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes



To provide a instant activity of plane to travelers like when it will reach ,where the plane was flying now etc.....

To create a QR in tickets its makes allowance easy at boarding time

Track a status of plane at each and every minute by using any detector

calculate approximate whether condition daily if its rainy .. before leaving a flight from airport (To check a signals)

check a mechanical problems before a flight is leaving from airport

To make sms or mail to passengers to inform a flight delay

To set a proper schedule to a plane for both arrival and departure

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

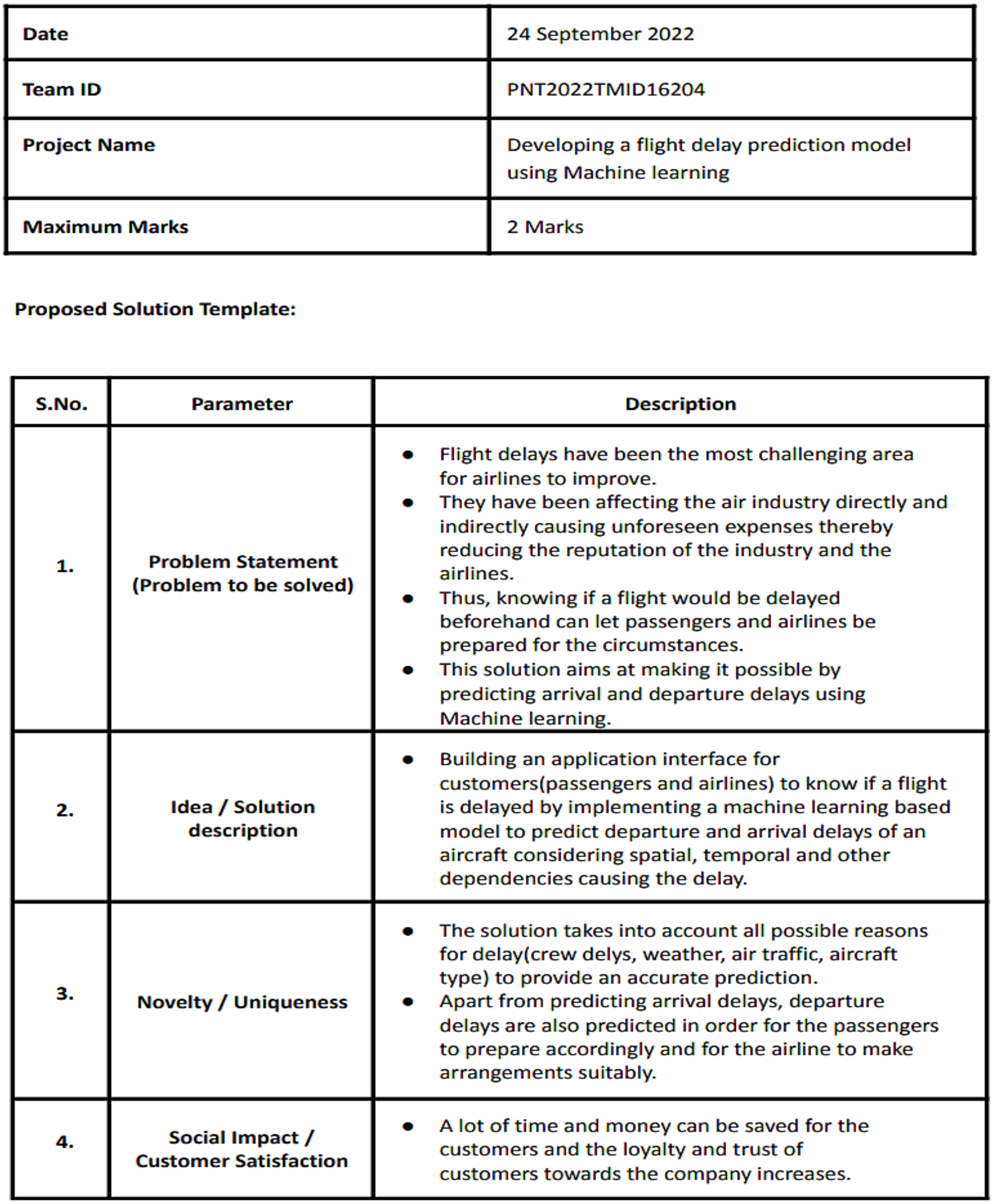
Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3. PROPOSED SOLUTION



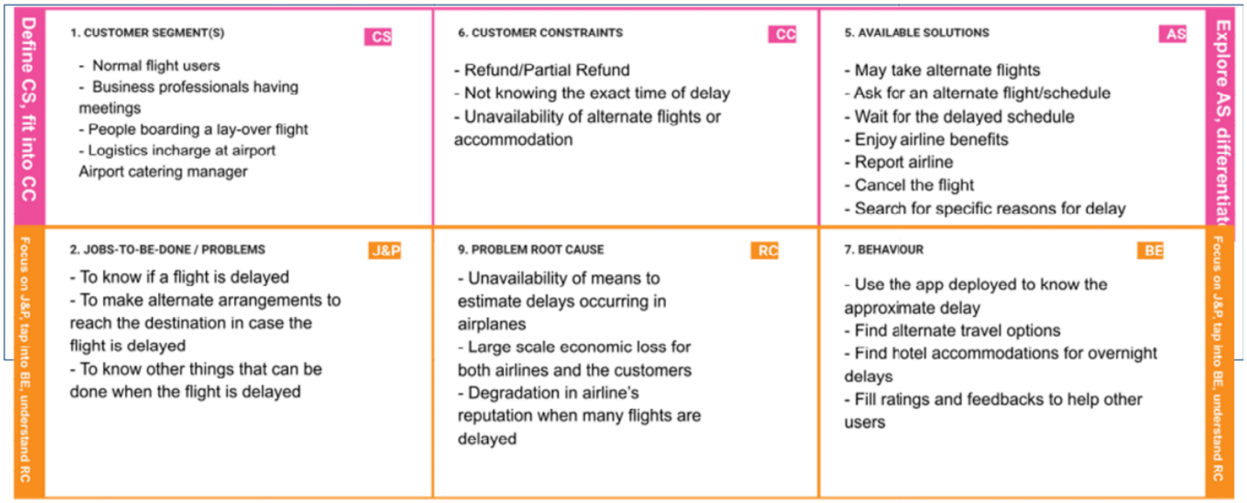
Date	24 September 2022
Team ID	PNT2022TMID16204
Project Name	Developing a flight delay prediction model using Machine learning
Maximum Marks	2 Marks

Proposed Solution Template:

S.No.	Parameter	Description
1.	**Problem Statement (Problem to be solved)**	● Flight delays have been the most challenging area for airlines to improve. ● They have been affecting the air industry directly and indirectly causing unforeseen expenses thereby reducing the reputation of the industry and the airlines. ● Thus, knowing if a flight would be delayed beforehand can let passengers and airlines be prepared for the circumstances. ● This solution aims at making it possible by predicting arrival and departure delays using Machine learning.
2.	**Idea / Solution description**	● Building an application interface for customers(passengers and airlines) to know if a flight is delayed by implementing a machine learning based model to predict departure and arrival delays of an aircraft considering spatial, temporal and other dependencies causing the delay.
3.	**Novelty / Uniqueness**	● The solution takes into account all possible reasons for delay(crew delys, weather, air traffic, aircraft type) to provide an accurate prediction. ● Apart from predicting arrival delays, departure delays are also predicted in order for the passengers to prepare accordingly and for the airline to make arrangements suitably.
4.	**Social Impact / Customer Satisfaction**	● A lot of time and money can be saved for the customers and the loyalty and trust of customers towards the company increases.

		● Improves airline operations by letting the company prepare in advance to adversaries (like crew illness, timeouts, rescheduling) leading to passenger satisfaction which will result positively on the economy and brand value.
5.	**Business Model (Revenue Model)**	● Business to Consumer model ● The solution is a low-cost airline model planned to be created as an application with which the consumers can interact directly to know the details of their flight. ● It follows a non-monetary revenue model where the consumers aren't charged for what they get but are asked to provide their flight details and ratings which can be used to improve the model and shared with the airline in return for the airline's flight data.
6.	**Scalability of the Solution**	● The present solution is drafted with the aim of experimenting with airlines based out of the United States of America. ● If there is a possibility to acquire data of a broader region (say North America, other continents), then the solution can be developed to benefit a wider range of people. ● International flight dependencies in both temporal and spatial focus can be derived from that data to provide more accurate predictions. ● Presence of ADS-B data can further increase the efficiency of the system making it reach a global audience and live time tracking of flights.

3.4. PROBLEM SOLUTION FIT



Define CS, fit into CC

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

1. CUSTOMER SEGMENT(S) CS - Normal flight users - Business professionals having meetings - People boarding a lay-over flight - Logistics incharge at airport Airport catering manager	6. CUSTOMER CONSTRAINTS CC - Refund/Partial Refund - Not knowing the exact time of delay - Unavailability of alternate flights or accommodation	5. AVAILABLE SOLUTIONS AS - May take alternate flights - Ask for an alternate flight/schedule - Wait for the delayed schedule - Enjoy airline benefits - Report airline - Cancel the flight - Search for specific reasons for delay
2. JOBS-TO-BE-DONE / PROBLEMS J&P - To know if a flight is delayed - To make alternate arrangements to reach the destination in case the flight is delayed - To know other things that can be done when the flight is delayed	9. PROBLEM ROOT CAUSE RC - Unavailability of means to estimate delays occurring in airplanes - Large scale economic loss for both airlines and the customers - Degradation in airline's reputation when many flights are delayed	7. BEHAVIOUR BE - Use the app deployed to know the approximate delay - Find alternate travel options - Find hotel accommodations for overnight delays - Fill ratings and feedbacks to help other users

Explore AS, differentiat

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

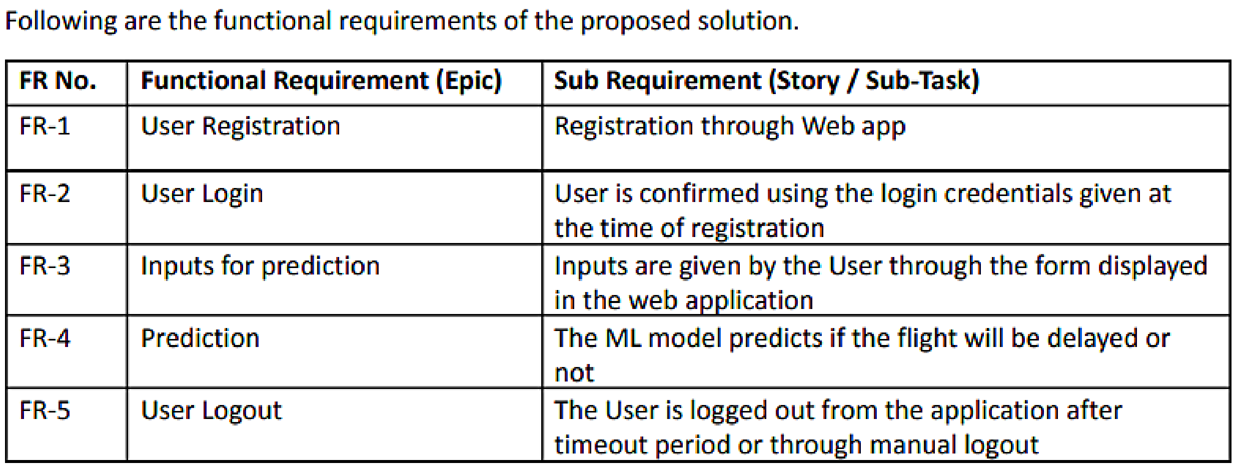
Identify strong TR & EM

3. TRIGGERS TR - Cancellation of flights - Extreme boredom - Guilt of wasting time - Thought of missing important meetings - Missing layover flight - Uncertainty in deciding if the flight is delayed when they start late for the airport	10. YOUR SOLUTION SL - The aim is to develop an application that predicts flight delays using a supervised machine learning model (a decision tree classifier) with the data of flights and delays so far and estimate the time of delay taking spatial dependencies of flights into account.	8.CHANNELS of BEHAVIOUR CH 8.1 ONLINE - Check if a particular flight will be delayed and the estimated time of arrival - Giving ratings and feedbacks for various flights so as to improve the app's performance in predicting further delays - Check for other specific reasons for delay
4. EMOTIONS: BEFORE / AFTER EM Before: - Worried - About missing important events - About missing layover flights - If the flight is gonna be canceled - Frustrated - About the unexpected delay/cancellation - Not knowing the news of delay beforehand - About the weather - Bored - Don't know how to make use of time After: - Gets to enjoy the airline benefits - Stays relaxed after getting a proper updatefrom the airline - Relieved if an alternate solution can be found		8.2 OFFLINE - Finding alternate travel routes in the airport - Hotels near the airport can be visit for overnight stays during delays

Identify strong TR & EM

CHAPTER 4
REQUIREMENT ANALYSIS

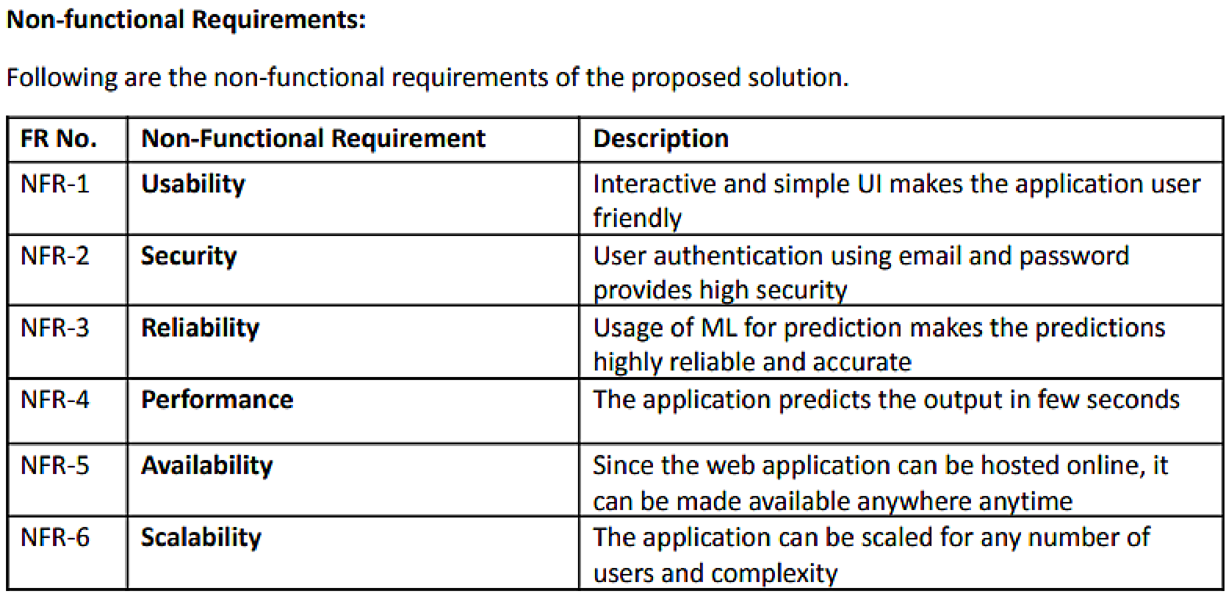
4.1. FUNCTIONAL REQUIREMENT



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 Web app
FR-2	User Login	User is confirmed using the login credentials given at the time of registration
FR-3	Inputs for prediction	Inputs are given by the User through the form displayed in the web application
FR-4	Prediction	The ML model predicts if the flight will be delayed or not
FR-5	User Logout	The User is logged out from the application after timeout period or through manual logout

4.2. NON-FUNCTIONAL REQUIREMENTS



Non-functional Requirements:

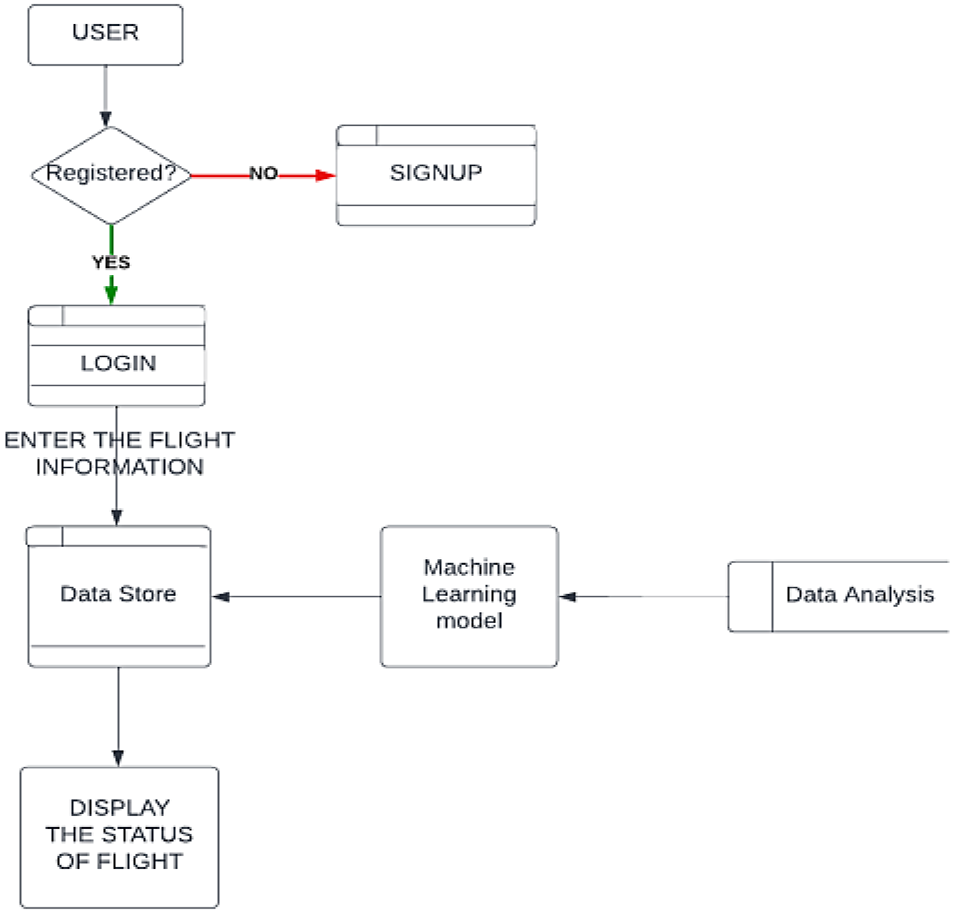
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Interactive and simple UI makes the application user friendly
NFR-2	**Security**	User authentication using email and password provides high security
NFR-3	**Reliability**	Usage of ML for prediction makes the predictions highly reliable and accurate
NFR-4	**Performance**	The application predicts the output in few seconds
NFR-5	**Availability**	Since the web application can be hosted online, it can be made available anywhere anytime
NFR-6	**Scalability**	The application can be scaled for any number of users and complexity

CHAPTER 5
PROJECT DESIGN

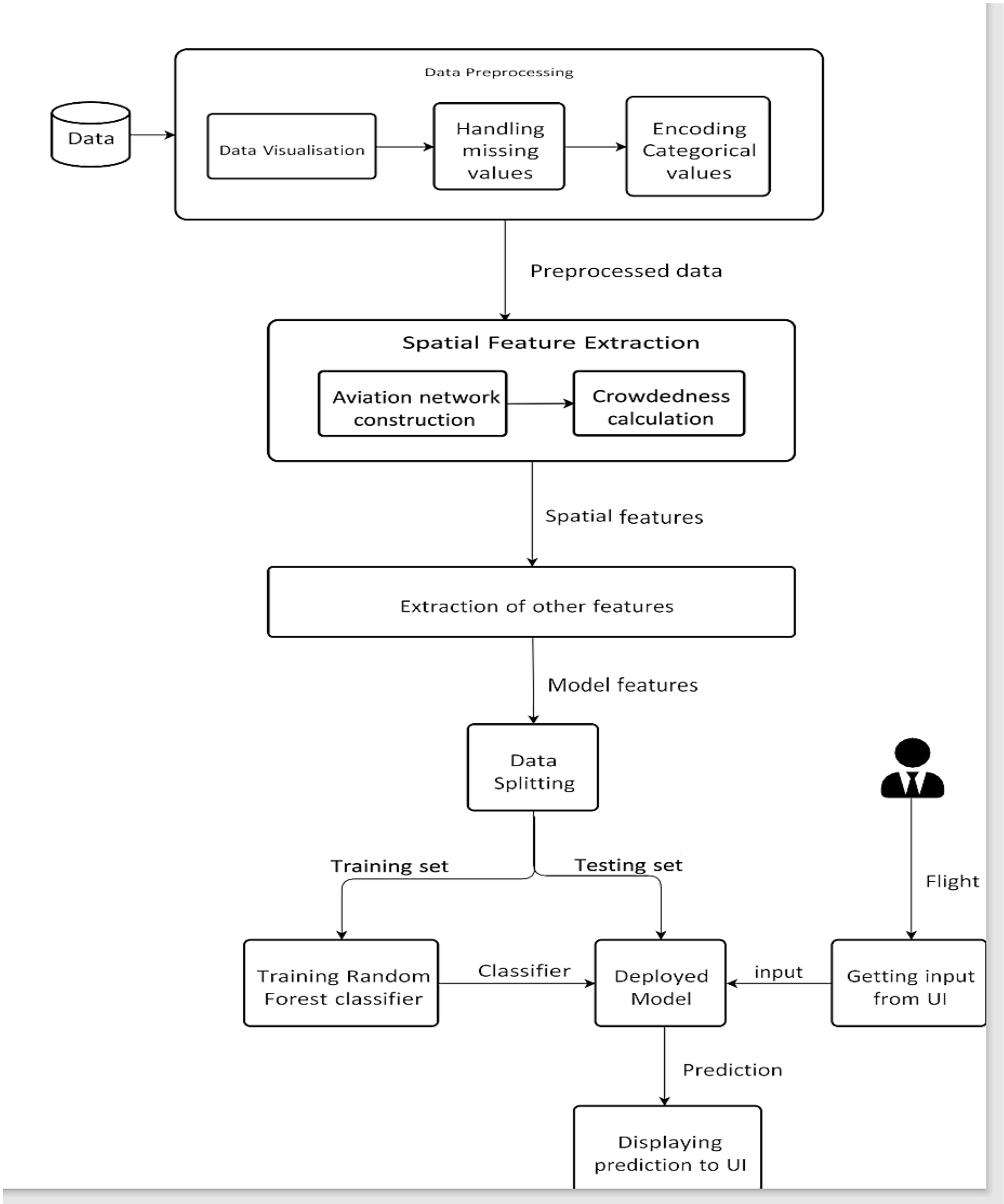
5.1. DATA FLOW DIAGRAMS

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.

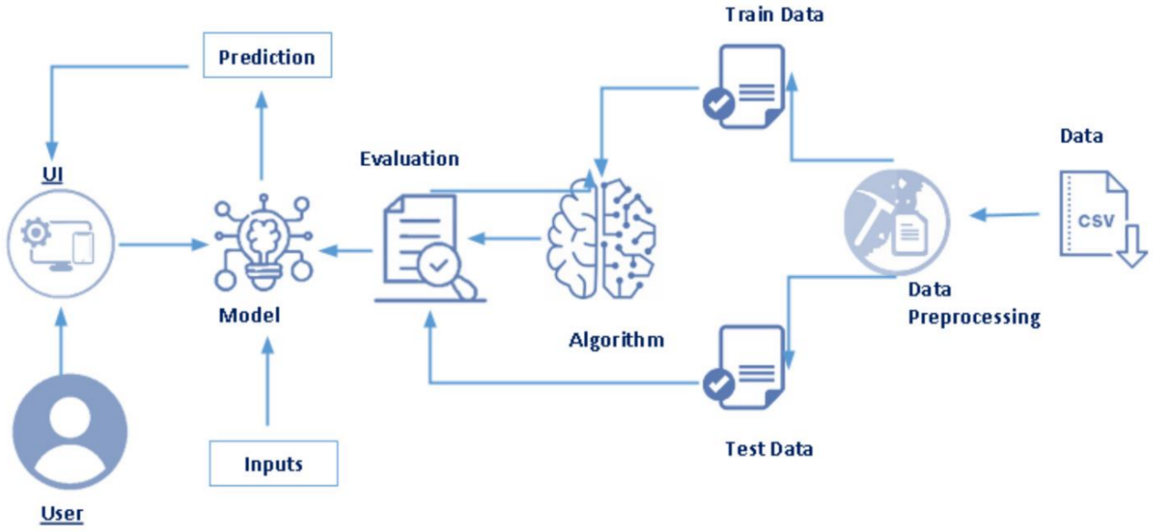


.

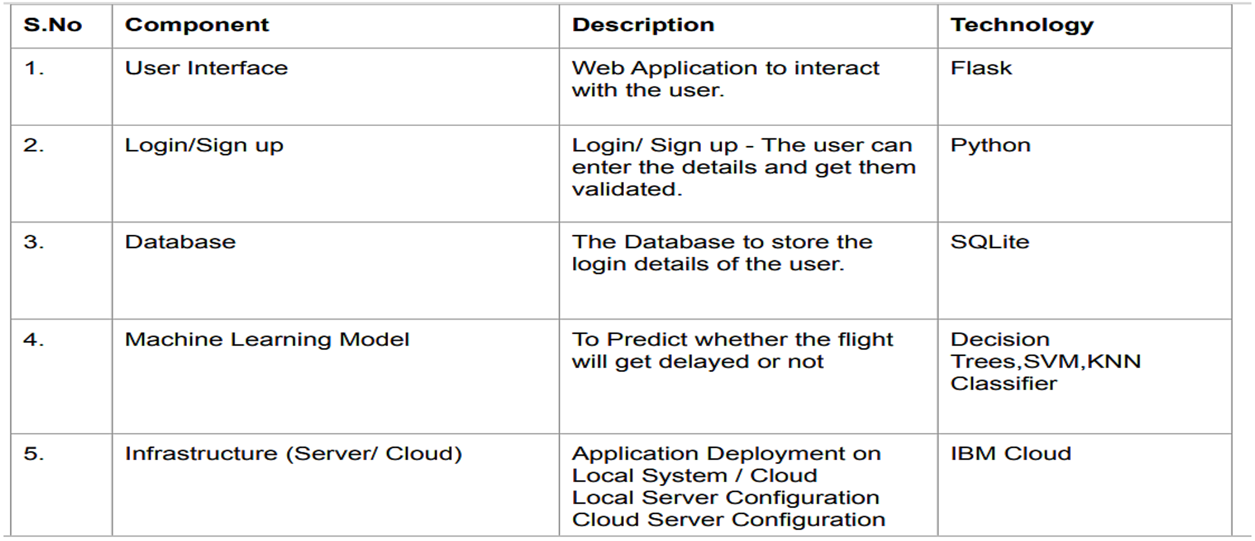
5.2. SOLUTION & TECHNICAL ARCHITECTURE



Technology Stack

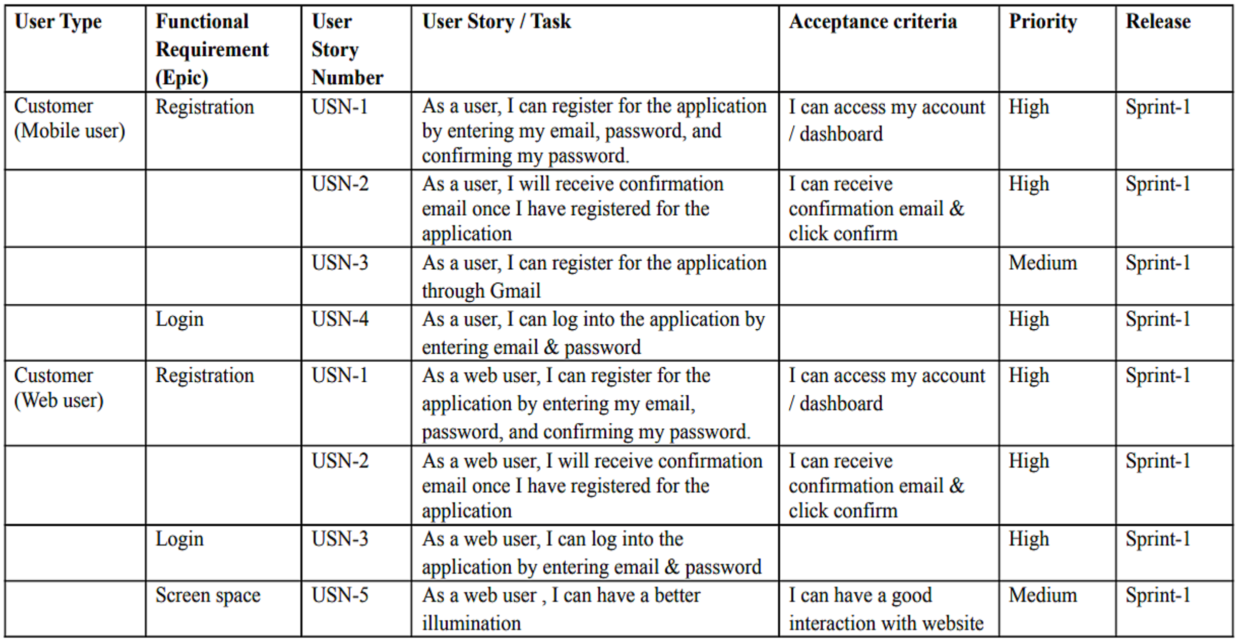


Components and Technologies

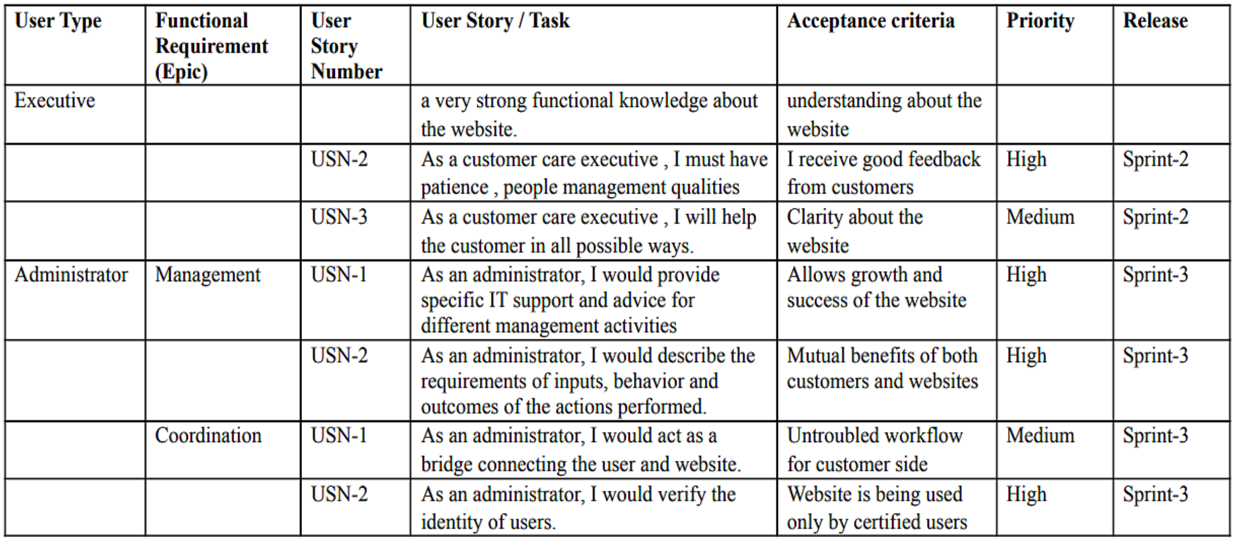


S.No	Component	Description	Technology
1.	User Interface	Web Application to interact with the user.	Flask
2.	Login/Sign up	Login/ Sign up - The user can enter the details and get them validated.	Python
3.	Database	The Database to store the login details of the user.	SQLite
4.	Machine Learning Model	To Predict whether the flight will get delayed or not	Decision Trees,SVM,KNN Classifier
5.	Infrastructure (Server/ Cloud)	Application Deployment on Local System / Cloud Local Server Configuration Cloud Server Configuration	IBM Cloud

5.3. User Stories



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering email & password		High	Sprint-1
Customer (Web user)	Registration	USN-1	As a web 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 web user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
	Login	USN-3	As a web user, I can log into the application by entering email & password		High	Sprint-1
	Screen space	USN-5	As a web user , I can have a better illumination	I can have a good interaction with website	Medium	Sprint-1

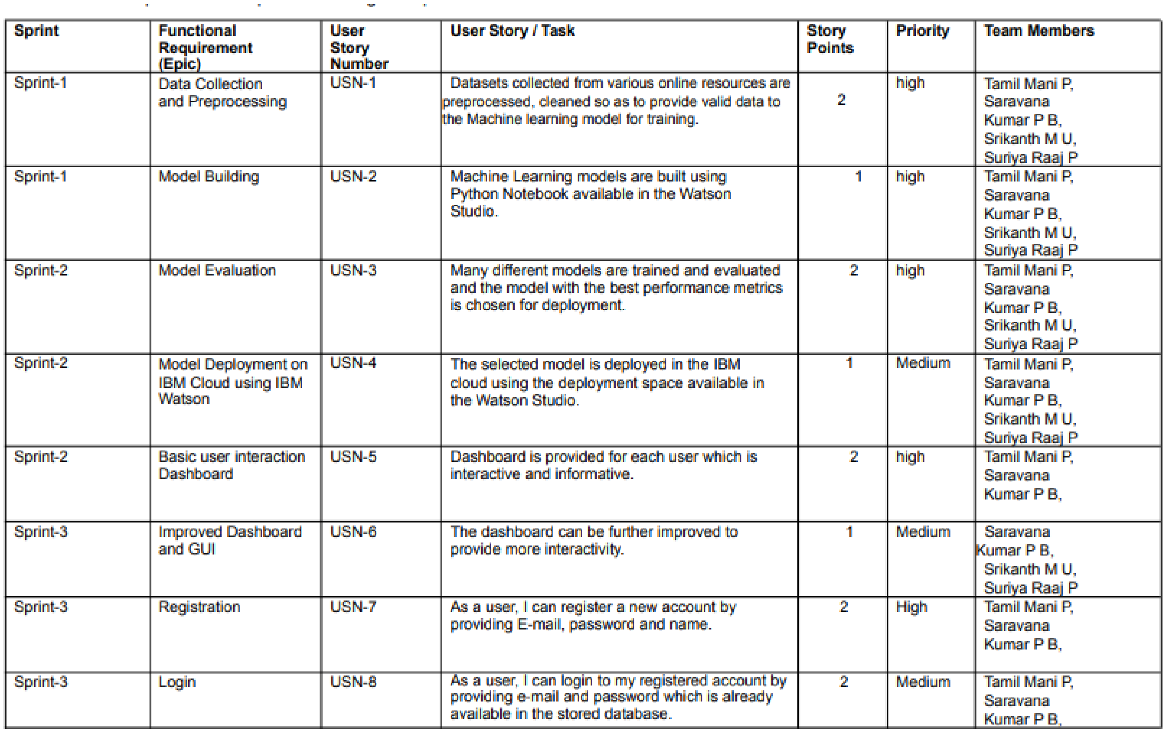


User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Executive			a very strong functional knowledge about the website.	understanding about the website		
		USN-2	As a customer care executive , I must have patience , people management qualities	I receive good feedback from customers	High	Sprint-2
		USN-3	As a customer care executive , I will help the customer in all possible ways.	Clarity about the website	Medium	Sprint-2
Administrator	Management	USN-1	As an administrator, I would provide specific IT support and advice for different management activities	Allows growth and success of the website	High	Sprint-3
		USN-2	As an administrator, I would describe the requirements of inputs, behavior and outcomes of the actions performed.	Mutual benefits of both customers and websites	High	Sprint-3
	Coordination	USN-1	As an administrator, I would act as a bridge connecting the user and website.	Untroubled workflow for customer side	Medium	Sprint-3
		USN-2	As an administrator, I would verify the identity of users.	Website is being used only by certified users	High	Sprint-3

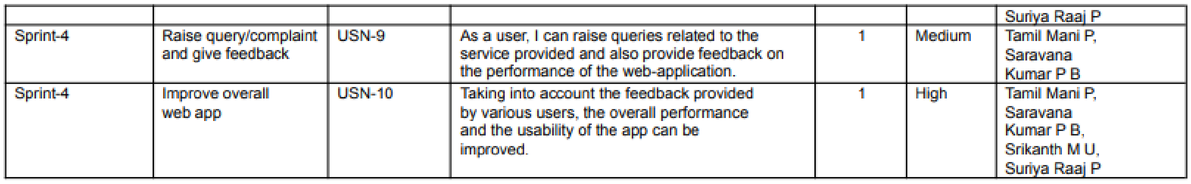
CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection and Preprocessing	USN-1	Datasets collected from various online resources are preprocessed, cleaned so as to provide valid data to the Machine learning model for training.	2	high	Tamil Mani P, Saravana Kumar P B, Srikanth M U, Suriya Raaj P
Sprint-1	Model Building	USN-2	Machine Learning models are built using Python Notebook available in the Watson Studio.	1	high	Tamil Mani P, Saravana Kumar P B, Srikanth M U, Suriya Raaj P
Sprint-2	Model Evaluation	USN-3	Many different models are trained and evaluated and the model with the best performance metrics is chosen for deployment.	2	high	Tamil Mani P, Saravana Kumar P B, Srikanth M U, Suriya Raaj P
Sprint-2	Model Deployment on IBM Cloud using IBM Watson	USN-4	The selected model is deployed in the IBM cloud using the deployment space available in the Watson Studio.	1	Medium	Tamil Mani P, Saravana Kumar P B, Srikanth M U, Suriya Raaj P
Sprint-2	Basic user interaction Dashboard	USN-5	Dashboard is provided for each user which is interactive and informative.	2	high	Tamil Mani P, Saravana Kumar P B,
Sprint-3	Improved Dashboard and GUI	USN-6	The dashboard can be further improved to provide more interactivity.	1	Medium	Saravana Kumar P B, Srikanth M U, Suriya Raaj P
Sprint-3	Registration	USN-7	As a user, I can register a new account by providing E-mail, password and name.	2	High	Tamil Mani P, Saravana Kumar P B,
Sprint-3	Login	USN-8	As a user, I can login to my registered account by providing e-mail and password which is already available in the stored database.	2	Medium	Tamil Mani P, Saravana Kumar P B,



						Suriya Raaj P
Sprint-4	Raise query/complaint and give feedback	USN-9	As a user, I can raise queries related to the service provided and also provide feedback on the performance of the web-application.	1	Medium	Tamil Mani P, Saravana Kumar P B
Sprint-4	Improve overall web app	USN-10	Taking into account the feedback provided by various users, the overall performance and the usability of the app can be improved.	1	High	Tamil Mani P, Saravana Kumar P B, Srikanth M U, Suriya Raaj P

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

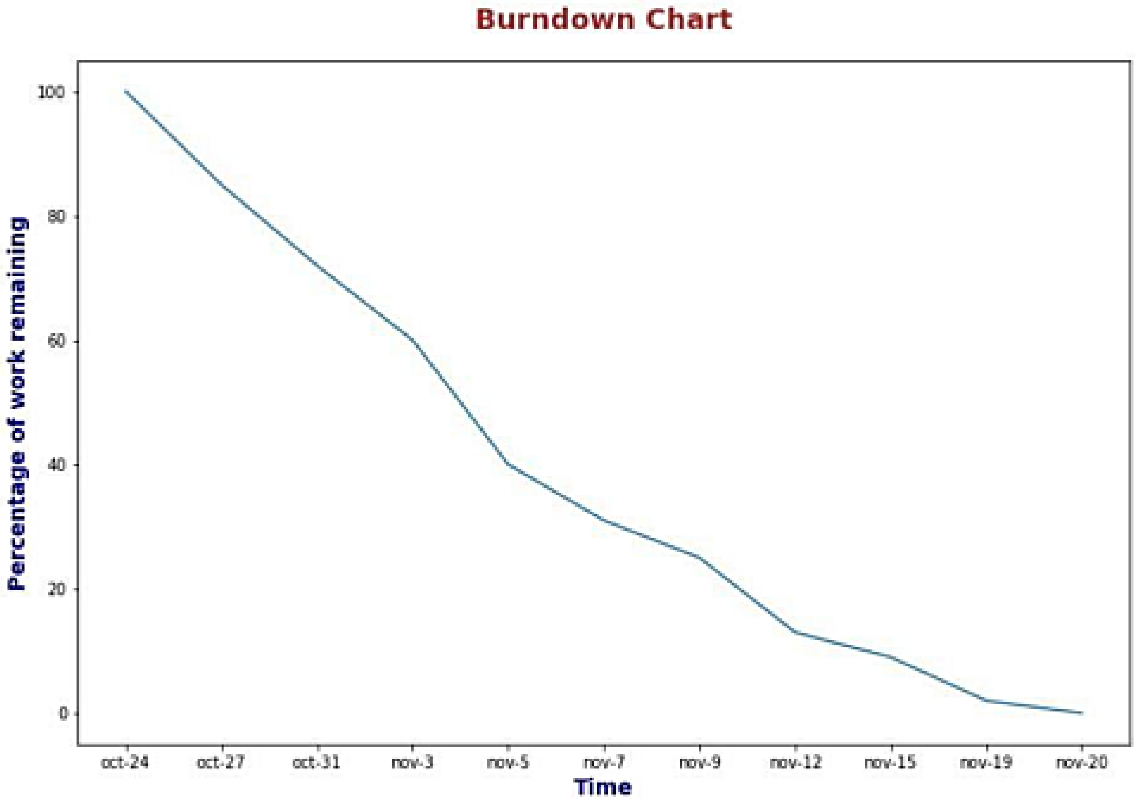
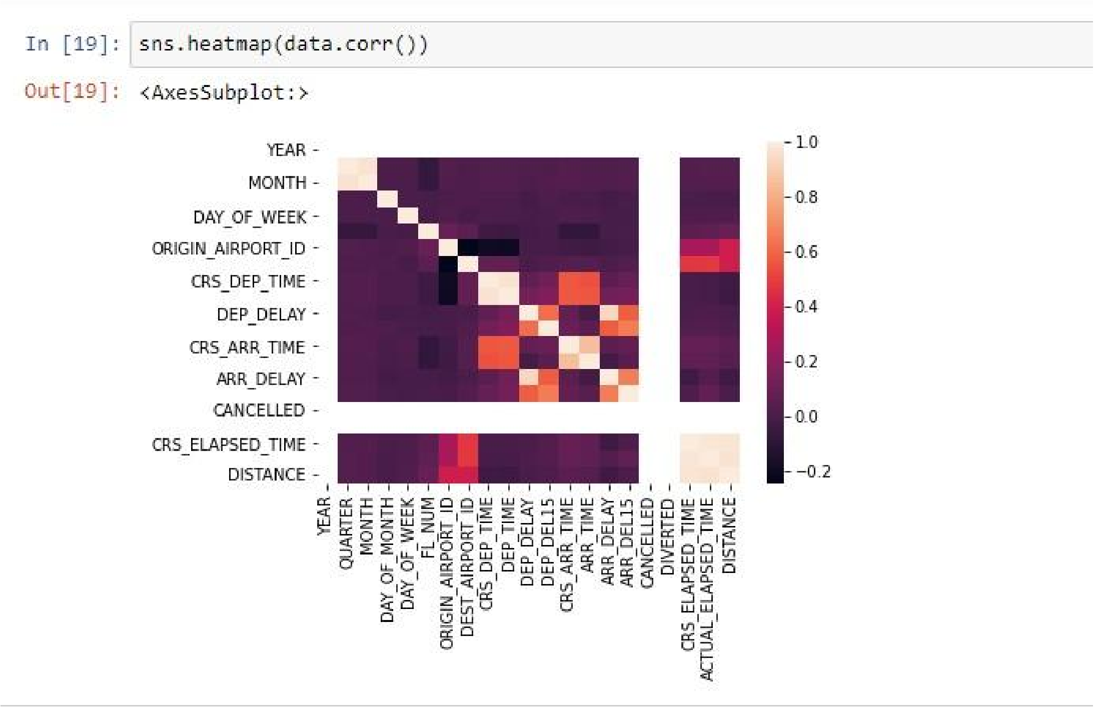


Figure 6.1 - Burndown Chart

CHAPTER 7
CODING AND SOLUTIONING

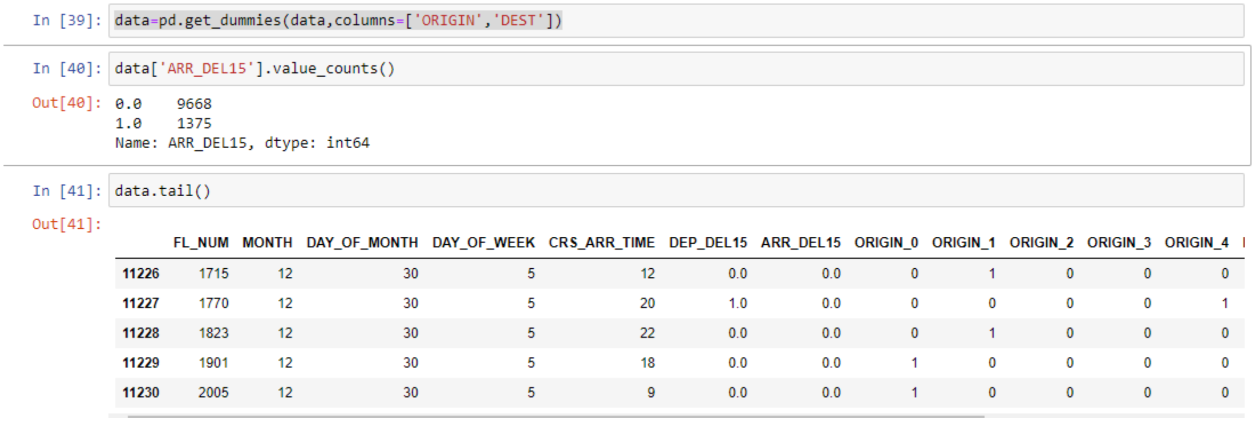
7.1 FEATURE 1 - CORRELATION BETWEEN THE VARIABLES IN THE DATASET



```
In [19]: sns.heatmap(data.corr())

Out[19]: <AxesSubplot:>
```

7.2

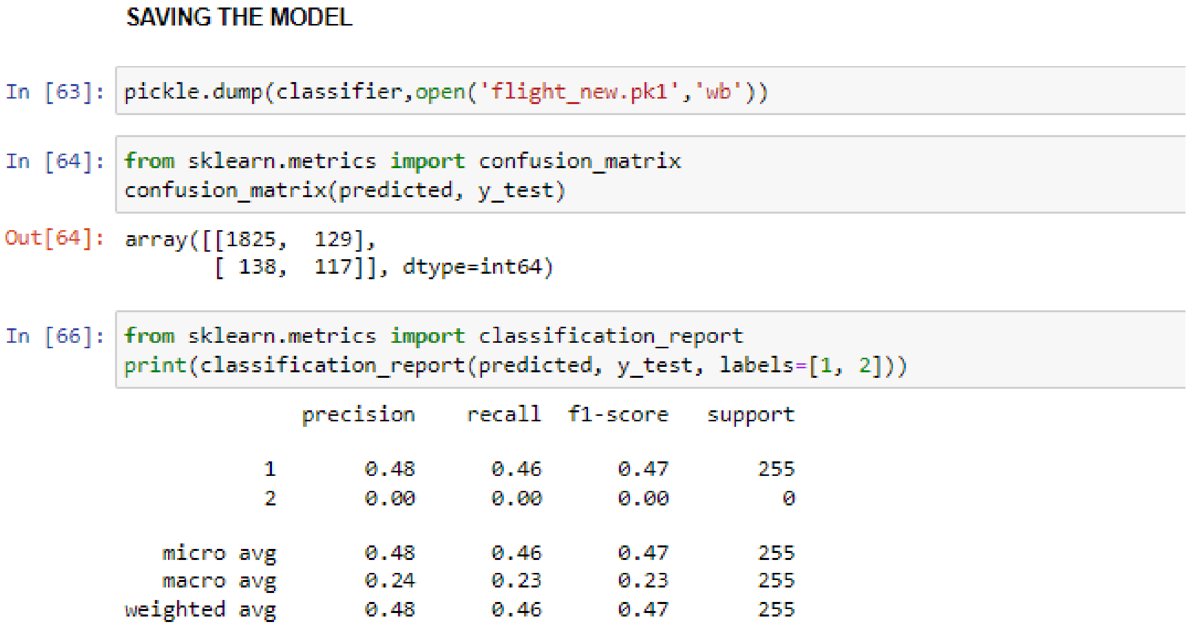
FEATURE 2 - ONE HOT ENCODING



```
In [39]: data=pd.get_dummies(data,columns=['ORIGIN','DEST'])

In [40]: data['ARR_DEL15'].value_counts()

Out[40]: 0.0    9668
         1.0    1375
         Name: ARR_DEL15, dtype: int64

In [41]: data.tail()

Out[41]:
```

	FL_NUM	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	CRS_ARR_TIME	DEP_DEL15	ARR_DEL15	ORIGIN_0	ORIGIN_1	ORIGIN_2	ORIGIN_3	ORIGIN_4
11226	1715	12	30	5	12	0.0	0.0	0	1	0	0	0
11227	1770	12	30	5	20	1.0	0.0	0	0	0	0	1
11228	1823	12	30	5	22	0.0	0.0	0	1	0	0	0
11229	1901	12	30	5	18	0.0	0.0	1	0	0	0	0
11230	2005	12	30	5	9	0.0	0.0	1	0	0	0	0

7.3 FEATURE 3 - SAVING THE MODEL WEIGHTS FOR DEPLOYMENT



SAVING THE MODEL

```
In [63]: pickle.dump(classifier,open('flight_new.pk1','wb'))

In [64]: from sklearn.metrics import confusion_matrix
         confusion_matrix(predicted, y_test)

Out[64]: array([[1825,  129],
                [ 138,  117]], dtype=int64)

In [66]: from sklearn.metrics import classification_report
         print(classification_report(predicted, y_test, labels=[1, 2]))

                      precision    recall  f1-score   support

                   1       0.48      0.46      0.47       255
                   2       0.00      0.00      0.00         0

           micro avg       0.48      0.46      0.47       255
           macro avg       0.24      0.23      0.23       255
        weighted avg       0.48      0.46      0.47       255
```

7.4 FEATURE 4 - FLASK INTERFACE - UI

```
from flask import Blueprint, render_template,request,redirect,url_for
from flask_login import login_required, current_user
from . import db
```

```
import requests
import flask
from flask_cors import CORS
from datetime import datetime

API_KEY = "b1papptuFebhE9mB86BRaPcjkCS3jwsVV_69I5w3os7E"
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}

main = Blueprint('main', __name__)
@main.route('/')
def index():
    return render_template('index.html')




@main.route('/profile')
@login_required
def profile():
    return render_template('profile.html', name=current_user.name)

@main.route('/prediction')
@login_required
def prediction():
    return render_template('prediction.html')

@main.route('/prediction',methods=['POST'])
```

```
@login_required
def prediction_post():
    departure_date=request.form['date_flight']
    departure_time=request.form['time_flight']
    departure_date_lis=departure_date.split('-')
    departure_date_str=departure_date_lis[2]+"/"+departure_date_lis[1]+"/"+departure_date_lis[0]
    origin=request.form['source']
    destination=request.form['destination']
    departure_date_time=departure_date_str+" "+departure_time
    try:
        departure_date_time_parsed = datetime.strptime(departure_date_time, '%d/%m/%Y
%H:%M:%S')
except ValueError as e:
        return 'Error parsing date/time - {}'.format(e)

    month = departure_date_time_parsed.month
    day = departure_date_time_parsed.day
    day_of_week = departure_date_time_parsed.isoweekday()
    hour = departure_date_time_parsed.hour

    origin = origin.upper()
    destination = destination.upper()
    X= [[month, day, day_of_week, hour, 1 if origin == 'ATL' else 0, 1 if origin == 'DTW' else 0,
    1 if origin == 'JFK' else 0, 1 if origin == 'MSP' else 0, 1 if origin == 'SEA' else 0,
    1 if destination == 'ATL' else 0, 1 if destination == 'DTW' else 0, 1 if destination == 'JFK' else 0,
    1 if destination == 'MSP' else 0, 1 if destination == 'SEA' else 0 ]]
    print(X)


    #predict= model.predict(X)[0]
    #print(predict)
    pred=['Flight is on Time','Flight is Delayed']
    payload_scoring = {"input_data": [{"field":
[['MONTH','DAY','DAY_OF_WEEK','CRS_DEP_TIME','ORIGIN_ATL',

'ORIGIN_DTW','ORIGIN_JFK','ORIGIN_MSP','ORIGIN_SEA','DEST_ATL','DEST_DTW','DEST_JFK'
,'DEST_MSP','DEST_SEA']], "values": X}]}
    response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/82a07ea5-a22b-4882-acc3-
7edb67a61b88/predictions?version=2022-11-15', json=payload_scoring,
    headers={'Authorization': 'Bearer ' + mltoken})
    print(response_scoring)
    predictions = response_scoring.json()
    print(predictions)
    predict = int(predictions['predictions'][0]['values'][0][0])
    predict_str=pred[predict]
    print("Final prediction :",predict_str)

    # showing the prediction results in a UI# showing the prediction results in a UI
    return render_template('output.html', predict_str=predict_str)

@main.route('/output')
@login_required
def predict_again():
    return render_template('prediction.html')
```

Explanation:

The above code will be able to get the details of the flight from the user in the respective text fields created using the HTML, scale the inputs and give the inputs to the model which has been developed already. The predictions are shown in another HTML page.

7.5 FEATURE 5 - HTML PAGES FOR FRONTEND DESIGN

prediction.html page

```
{% extends "base.html" %}
{% block content%}


<div class="column is-4 is-offset-4">
    <h3 class="title">Enter the details of your scheduled flight</h3>
    <div class="box">
        <form method="POST" action="" class="">
            <p>Enter the Date of the Flight:</p>
            <div class="input-field">
                <input name="date_flight" input type='date'  required>

            </div>
            <p>Enter the Departure Time in 24 hour format:</p>

            <input name="time_flight" input type='time' value='13:45:00' step='1'  required>

            <p>Enter the Source:</p>
        <!--input name="source" required-->

            <select name="source" id="source" class="form-select" >
                <option value="ATL">ATL-Hartsfield-Jackson Atlanta International Airport,
Atlanta</option>
                <option value="JFK">JFK-John F kennedy International Airport, New York</option>
                <option value="DTW">DTW-Detroit Metropolitan Wayne County Airport-Detroit,
Michigan</option>
                <option value="MSP">MSP-Minneapolis−Saint Paul International Airport,
Minnesota</option>
        <option value="SEA">SEA-Seattle-Tacoma International Airport, Washington</option>
      </select>
      <br>
      <p>Enter the Destination:</p>
      <select name="destination" id="source" class="form-select">
        <option value="ATL">ATL-Hartsfield-Jackson Atlanta International Airport,
Atlanta</option>
        <option value="JFK">JFK-John F kennedy International Airport, New York</option>
        <option value="DTW">DTW-Detroit Metropolitan Wayne County Airport-Detroit,
Michigan</option>
        <option value="MSP">MSP-Minneapolis−Saint Paul International Airport,
Minnesota</option>
        <option value="SEA">SEA-Seattle-Tacoma International Airport, Washington</option>
      </select>
<br>
      <br>
      <!--button class="button is-block is-info is-large is-fullwidth" type="submit" >Submit</button--
>
      <button class="btn btn-link" type="submit">Predict</button>
  </form>
  </div>
</div>
<style>
  input{
    -webkit-text-fill-color:gray;
  }
  .checkbox{
    -webkit-text-fill-color: gray;
  }
  button{
        background-color: black  !important;
    }
    button a:hover{
        color:aqua !important;
    }
    option{
        color:black;
        background-color:white;
    }
    option :hover{
        background-color:aqua !important;
    }
    p{
        text-align:left;
}
</style>
{% endblock %}
```

Explanation:

The above code will be able to get the details of the flight from the user in the respective text fields created using the form and provides it to the back end flask framework.

ouput.html Page

```
{% extends "base.html" %}
{% block content %}
<h2>{{predict_str}}</h2>
<form action="/prediction">
    <button class="btn btn-link"  type="submit"><a hef="/prediction">Predict Again</a></button>
```

```
</form>

<style>
    input{
        -webkit-text-fill-color:gray;
    }
    .checkbox{
        -webkit-text-fill-color: gray;
    }
    button{
        background-color: black  !important;
    }
    button a:hover{
        color:aqua !important;
    }
    option{
        color:black;
        background-color:white;
    }
    option :hover{
        background-color:aqua !important;
    }
</style>
{% endblock %}
```

Explanation:

The above page loads when the user submits the form and provides the output predicted classification as whether the flight is delayed or not.

CHAPTER 8
TESTING

8.1 TEST

User No	Flight No	Month	Day of month	Day of week	Origin	Destination	Scheduled Departure Time	Scheduled Arrival Time	Actual Departure Time	Actual Inputs
1	1232	1	1	1	ATL	MSP	1905	2305	1945	Delayed
2	1399	1	1	1	ATL	SEA	1805	2410	1855	Delayed
3	2351	1	2	3	ATL	DTW	1305	2305	1305	Not Delayed
4	2637	2	1	3	DTW	ATL	1500	2410	1505	Not Delayed

8.2 USER ACCEPTANCE TESTING

User No	Flight No	Month	Day Of Month	Day Of Week	Origin	Destination	Scheduled Departure Time	Scheduled Arrival Time	Actual Departure Time	Actual Output	Predicted Output	Correctness
1	1232	1	1	1	ATL	MSP	1905	2305	1945	Delayed	Delayed	Correct
2	1399	1	1	1	ATL	SEA	1805	2410	1855	Delayed	Delayed	Correct

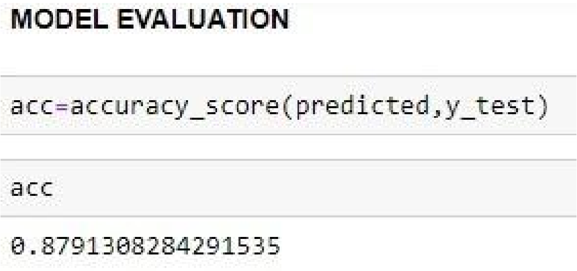
3	2351	1	2	3	ATL	DTW	1305	2305	1305	Not Delayed	Not Delayed	Correct
4	2637	2	1	3	DTW	ATL	1500	2410	1505	Not Delayed	Not Delayed	Correct

CHAPTER 9

RESULTS

9.1 PERFORMANCE METRICS

Training Accuracy



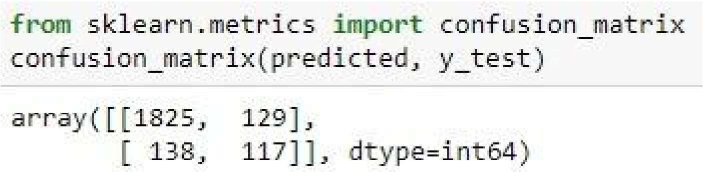
MODEL EVALUATION

```
acc=accuracy_score(predicted,y_test)
```

```
acc
```

```
0.8791308284291535
```

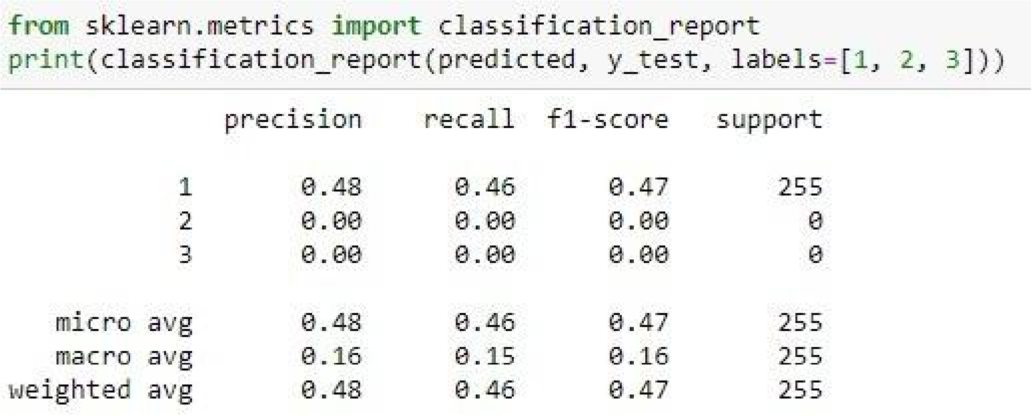
Confusion Matrix



```
from sklearn.metrics import confusion_matrix
confusion_matrix(predicted, y_test)
```

```
array([[1825,  129],
       [ 138,  117]], dtype=int64)
```

Classification Model



```
from sklearn.metrics import classification_report
print(classification_report(predicted, y_test, labels=[1, 2, 3]))
```

```
              precision    recall  f1-score   support

           1       0.48      0.46      0.47       255
           2       0.00      0.00      0.00         0
           3       0.00      0.00      0.00         0

   micro avg       0.48      0.46      0.47       255
   macro avg       0.16      0.15      0.16       255
weighted avg       0.48      0.46      0.47       255
```

CHAPTER 10
ADVANTAGES AND DISADVANTAGES

Advantages

- Passengers can identify easily if their flights will be delayed or not.
- Passengers become aware of the delays early and can be composed in crunch situations.
- The current status of the flight can also be tracked.
- Helps passengers plan their journey prior to their travel.

Disadvantages

- Biased prediction due to the presence of outliers.
- Passengers might miss their flights due to incorrect prediction of the model that the flight will be delayed.
- Incorrect prediction leads to various confusions including the loss of capital for

the trave agency.

CHAPTER 11
CONCLUSION

The created web-application makes use of Machine Learning model trained on the dataset of previous flights and predicts the status of the flights in the current time. Using our Flight Delay Predictor application, the users can be benefitted by getting know the delay status of their scheduled flights.

Since the application is web based, any person can access and get the information at anytime anywhere.

CHAPTER 12
FUTURE SCOPE

Based on data analysis between the years 2016-2018, this project was made. There is a sizable dataset accessible from 1987 to 2020, but managing a larger dataset necessitates extensive preprocessing and purification of the data Therefore, adding a larger dataset is a part of this project's future effort. Preprocessing a bigger dataset can be done in a variety of methods, such as establishing a Spark cluster on a computer or using cloud services like AWS and Azure. Now that deep learning has advanced, we can also employ neural networks algorithms to analyze aviation and meteorological data. Neural networks employ a form of pattern matching.

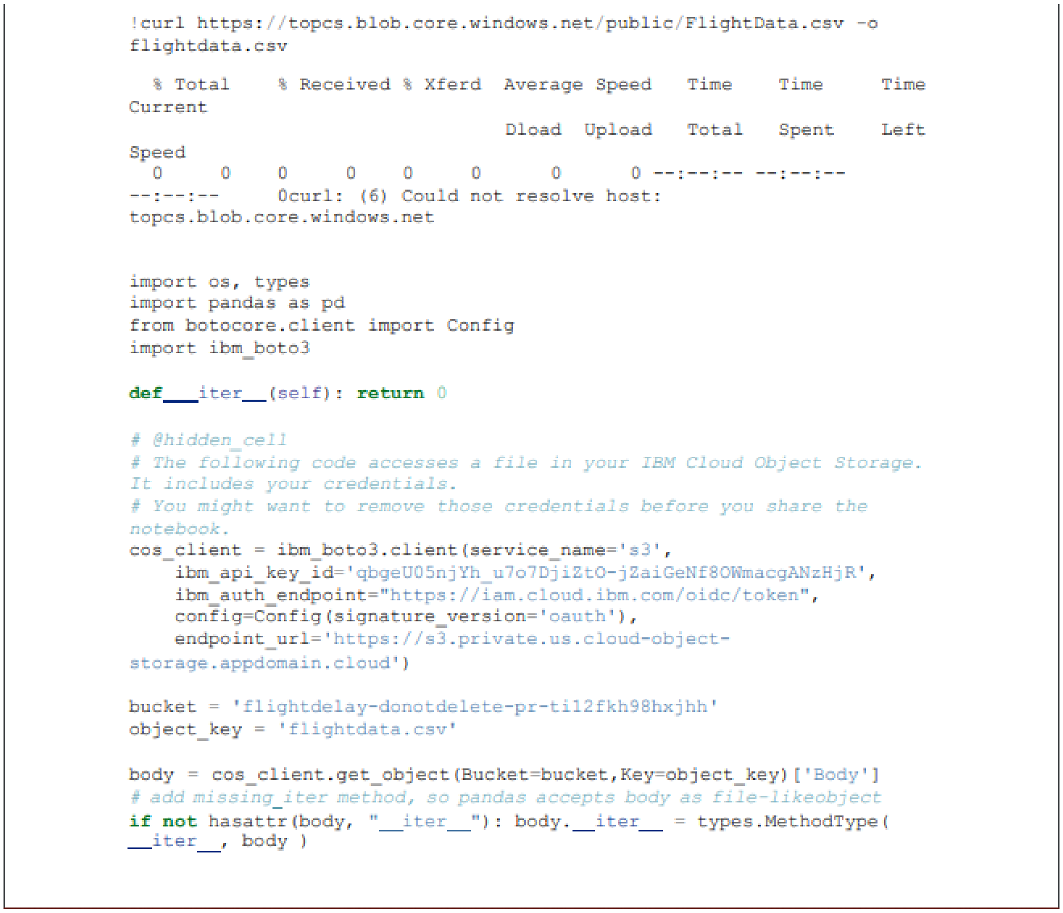
The project's focus is primarily on flight and weather data for specific airports of USA, but we can also include data from other nations like China, Europe, Russia and also

India. We can broaden the project's scope by including flight information from international flights rather than just domestic flights.

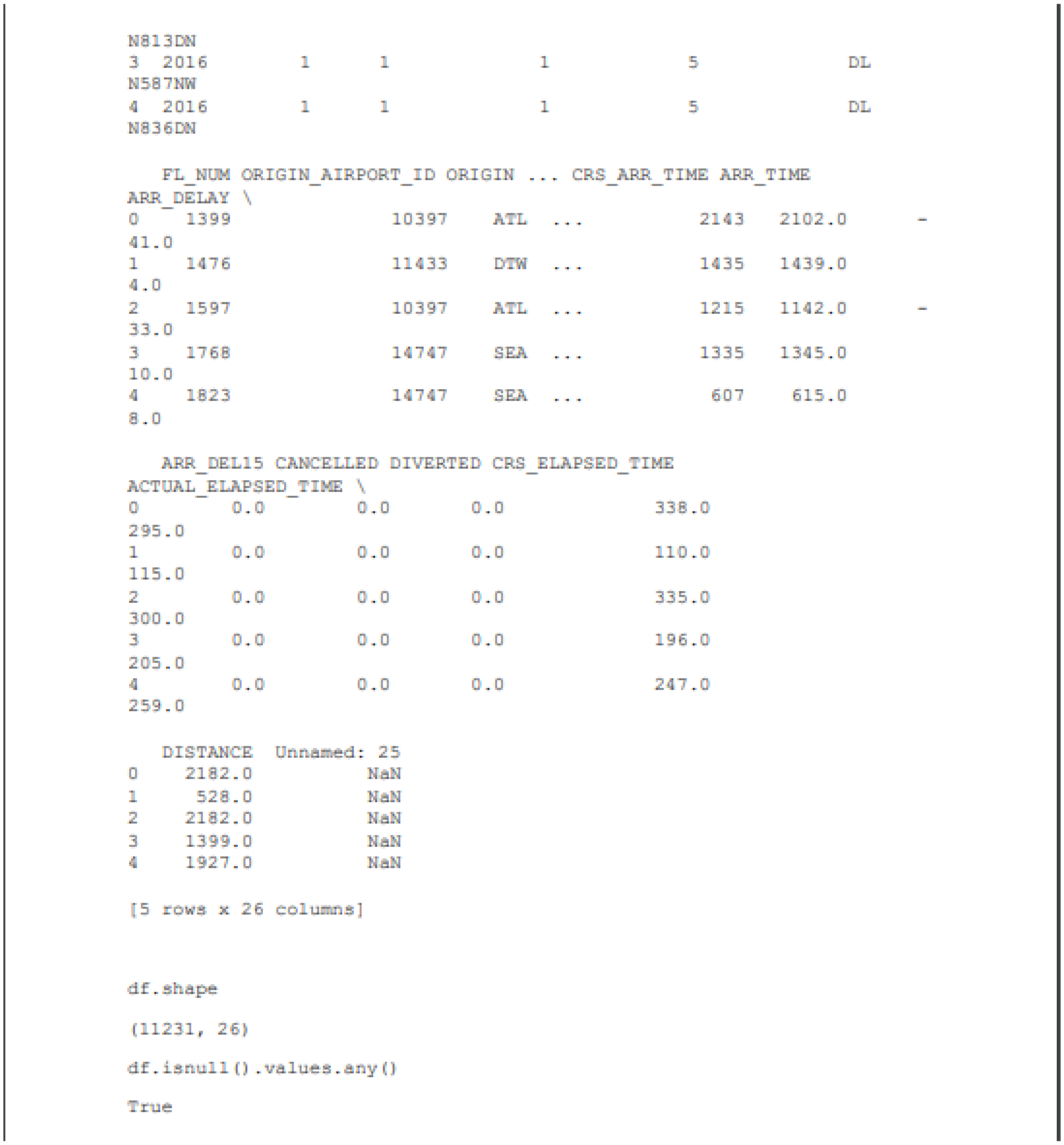
CHAPTER 13
APPENDIX

13.1 Source codes

13.1.2 Exploratory Data Analysis



```
!curl https://topcs.blob.core.windows.net/public/FlightData.csv -o
flightdata.csv

  % Total    % Received % Xferd  Average Speed   Time    Time     Time
Current
                                 Dload  Upload   Total   Spent    Left
Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:--
--:--:--     0curl: (6) Could not resolve host:
topcs.blob.core.windows.net


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='qbgeU05njYh_u7o7DjiZtO-jZaiGeNf8OWmacgANzHjR',
    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 = 'flightdelay-donotdelete-pr-ti12fkh98hxjhh'
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-likeobject
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType(
__iter__, body )
```



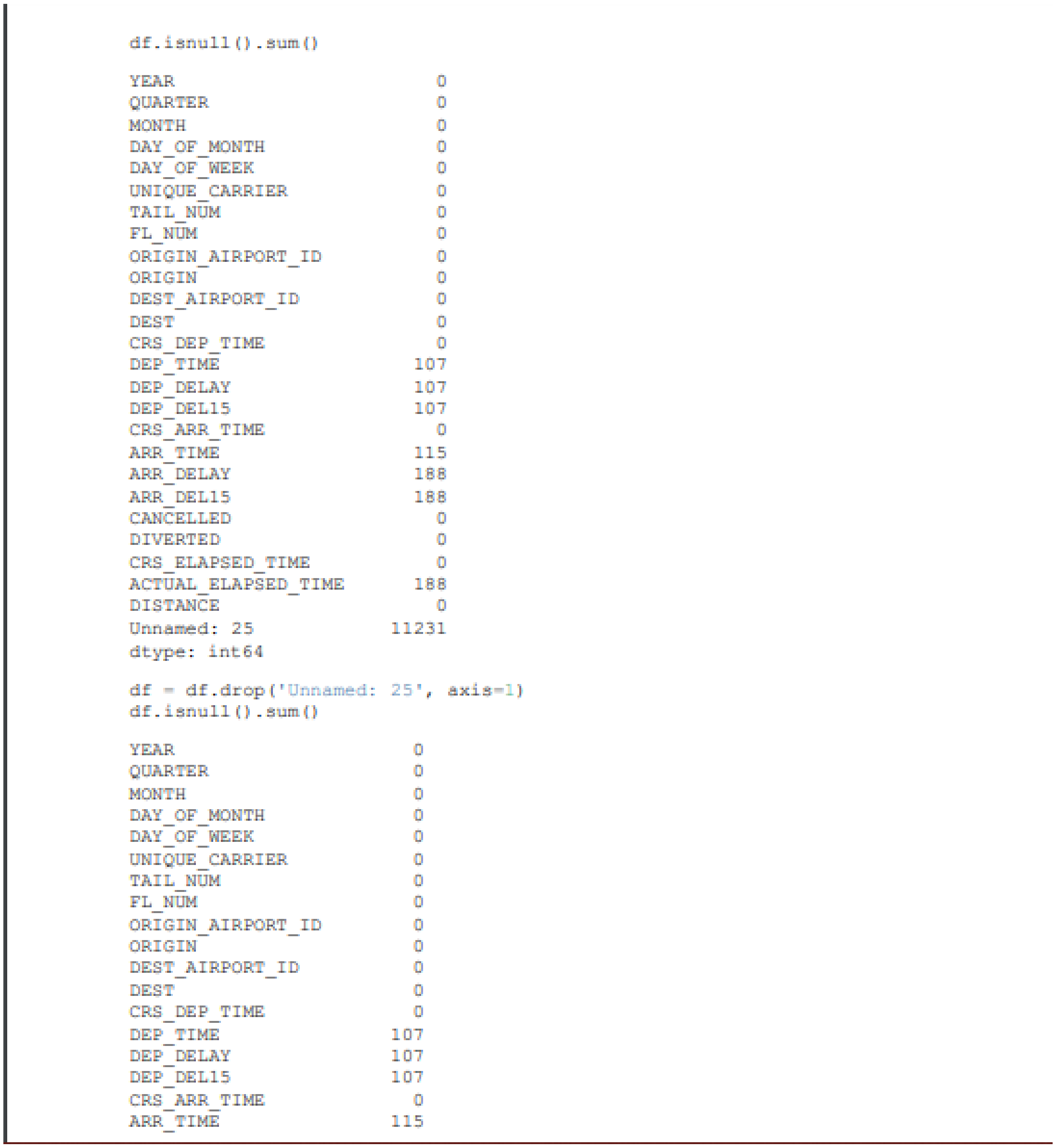
```
N813DN
3  2016          1      1             1            5             DL
N587NW
4  2016          1      1             1            5             DL
N836DN

   FL_NUM ORIGIN_AIRPORT_ID ORIGIN ... CRS_ARR_TIME ARR_TIME
ARR_DELAY \
0    1399              10397    ATL  ...          2143   2102.0      -
41.0
1    1476              11433    DTW  ...          1435   1439.0
4.0
2    1597              10397    ATL  ...          1215   1142.0      -
33.0
3    1768              14747    SEA  ...          1335   1345.0
10.0
4    1823              14747    SEA  ...           607    615.0
8.0

   ARR_DEL15 CANCELLED DIVERTED CRS_ELAPSED_TIME
ACTUAL_ELAPSED_TIME \
0        0.0        0.0       0.0             338.0
295.0
1        0.0        0.0       0.0             110.0
115.0
2        0.0        0.0       0.0             335.0
300.0
3        0.0        0.0       0.0             196.0
205.0
4        0.0        0.0       0.0             247.0
259.0

   DISTANCE  Unnamed: 25
0    2182.0          NaN
1     528.0          NaN
2    2182.0          NaN
3    1399.0          NaN
4    1927.0          NaN

[5 rows x 26 columns]
```

```
df.shape
```

```
(11231, 26)
```

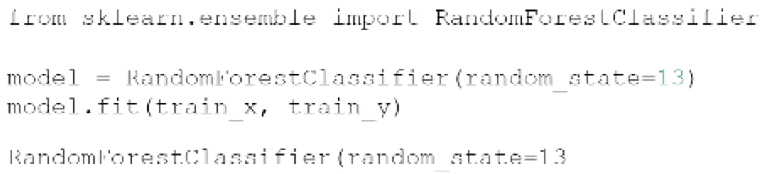
```
df.isnull().values.any()
```

```
True
```

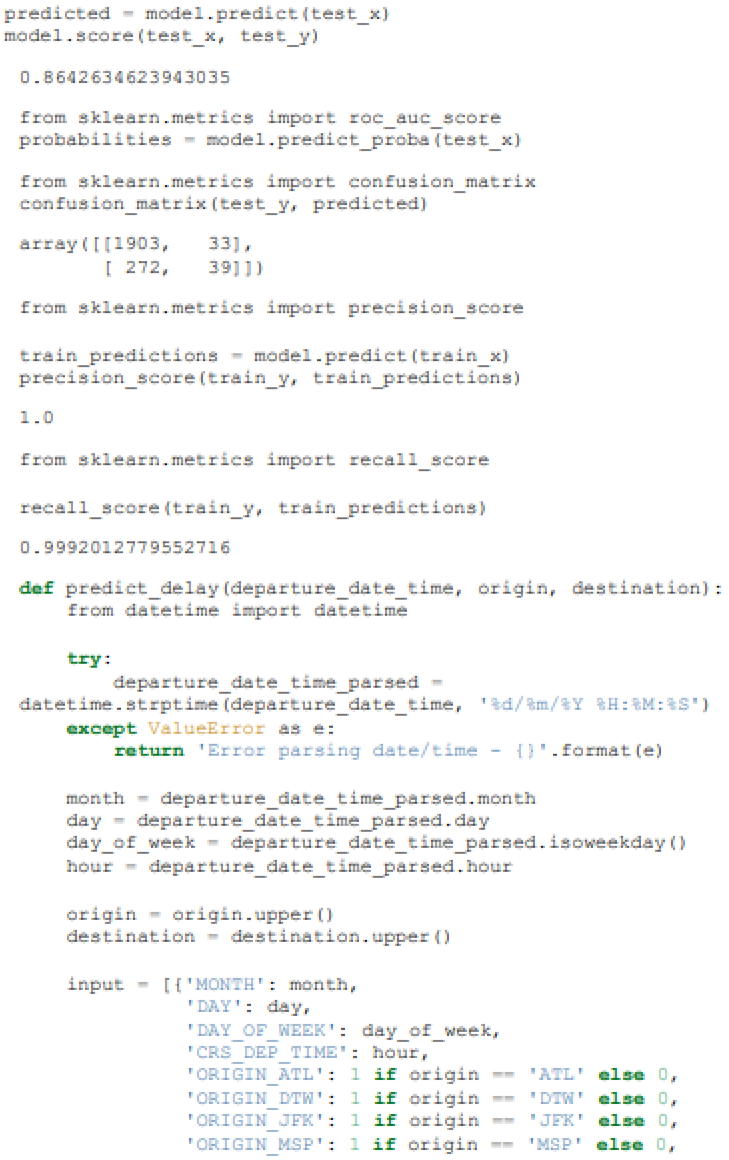


```
df.isnull().sum()

YEAR                         0
QUARTER                      0
MONTH                        0
DAY_OF_MONTH                 0
DAY_OF_WEEK                  0
UNIQUE_CARRIER               0
TAIL_NUM                     0
FL_NUM                       0
ORIGIN_AIRPORT_ID            0
ORIGIN                       0
DEST_AIRPORT_ID              0
DEST                         0
CRS_DEP_TIME                 0
DEP_TIME                   107
DEP_DELAY                  107
DEP_DEL15                  107
CRS_ARR_TIME                 0
ARR_TIME                   115
ARR_DELAY                  188
ARR_DEL15                  188
CANCELLED                    0
DIVERTED                     0
CRS_ELAPSED_TIME             0
ACTUAL_ELAPSED_TIME        188
DISTANCE                     0
Unnamed: 25              11231
dtype: int64

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

YEAR                       0
QUARTER                    0
MONTH                      0
DAY_OF_MONTH               0
DAY_OF_WEEK                0
UNIQUE_CARRIER             0
TAIL_NUM                   0
FL_NUM                     0
ORIGIN_AIRPORT_ID          0
ORIGIN                     0
DEST_AIRPORT_ID            0
DEST                       0
CRS_DEP_TIME               0
DEP_TIME                 107
DEP_DELAY                107
DEP_DEL15                107
CRS_ARR_TIME               0
ARR_TIME                 115
```

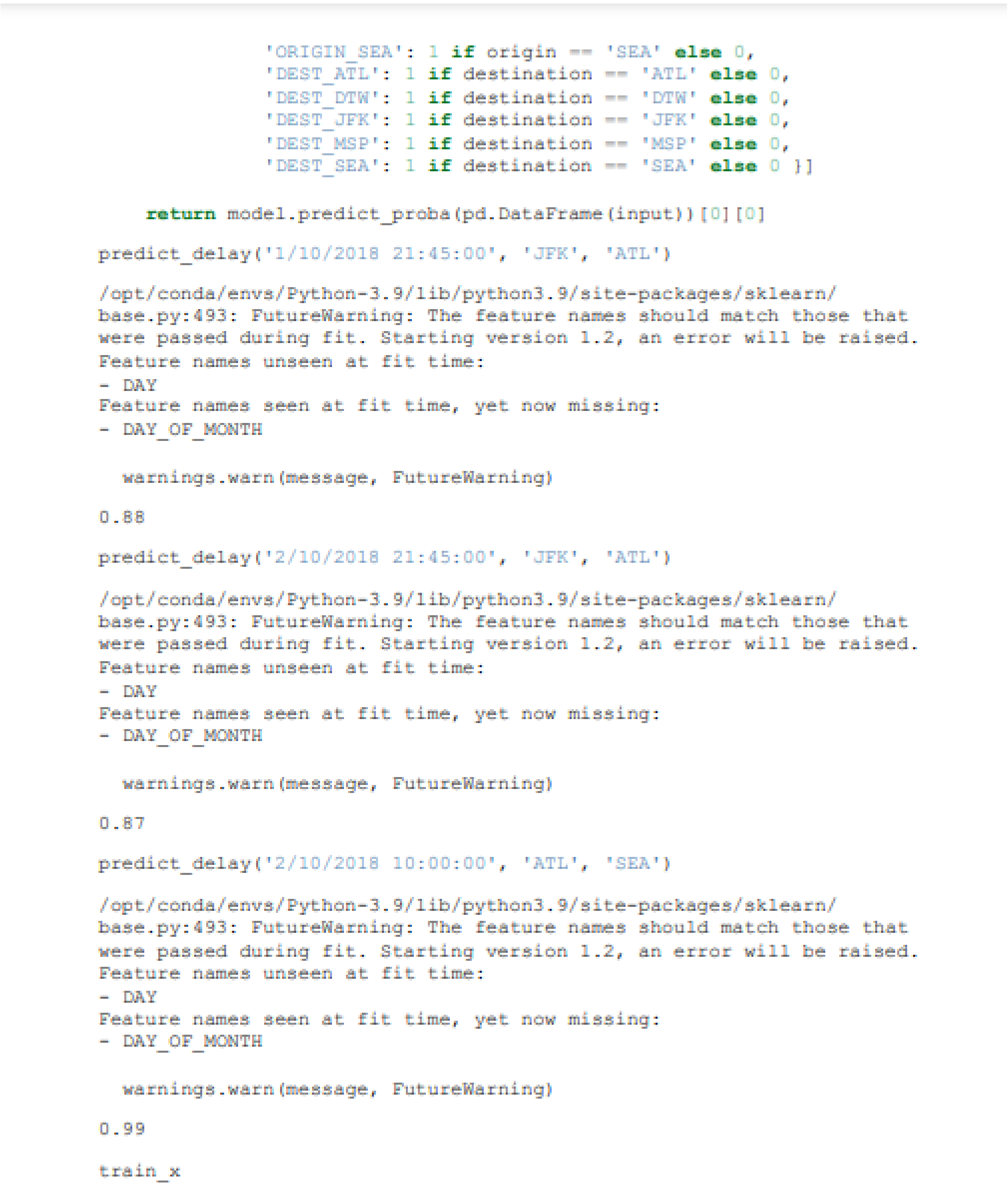
13.1.2 Train the ML Model



```
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(random_state=13)
model.fit(train_x, train_y)

RandomForestClassifier(random_state=13
```



```
predicted = model.predict(test_x)
model.score(test_x, test_y)

 0.8642634623943035

 from sklearn.metrics import roc_auc_score
 probabilities = model.predict_proba(test_x)

 from sklearn.metrics import confusion_matrix
 confusion_matrix(test_y, predicted)

 array([[1903,   33],
        [ 272,   39]])

 from sklearn.metrics import precision_score

 train_predictions = model.predict(train_x)
 precision_score(train_y, train_predictions)

 1.0

 from sklearn.metrics import recall_score

 recall_score(train_y, train_predictions)

 0.9992012779552716

 def predict_delay(departure_date_time, origin, destination):
     from datetime import datetime

     try:
         departure_date_time_parsed =
 datetime.strptime(departure_date_time, '%d/%m/%Y %H:%M:%S')
     except ValueError as e:
         return 'Error parsing date/time - {}'.format(e)

     month = departure_date_time_parsed.month
     day = departure_date_time_parsed.day
     day_of_week = departure_date_time_parsed.isoweekday()
     hour = departure_date_time_parsed.hour

     origin = origin.upper()
     destination = destination.upper()

     input = [{'MONTH': month,
               'DAY': day,
               'DAY_OF_WEEK': day_of_week,
               'CRS_DEP_TIME': hour,
               'ORIGIN_ATL': 1 if origin == 'ATL' else 0,
               'ORIGIN_DTW': 1 if origin == 'DTW' else 0,
               'ORIGIN_JFK': 1 if origin == 'JFK' else 0,
               'ORIGIN_MSP': 1 if origin == 'MSP' else 0,
```



```
                  'ORIGIN_SEA': 1 if origin == 'SEA' else 0,
                  'DEST_ATL': 1 if destination == 'ATL' else 0,
                  'DEST_DTW': 1 if destination == 'DTW' else 0,
                  'DEST_JFK': 1 if destination == 'JFK' else 0,
                  'DEST_MSP': 1 if destination == 'MSP' else 0,
                  'DEST_SEA': 1 if destination == 'SEA' else 0 }]

    return model.predict_proba(pd.DataFrame(input))[0][0]
```

```
predict_delay('1/10/2018 21:45:00', 'JFK', 'ATL')
```

```
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/sklearn/
base.py:493: FutureWarning: The feature names should match those that
were passed during fit. Starting version 1.2, an error will be raised.
Feature names unseen at fit time:
- DAY
Feature names seen at fit time, yet now missing:
- DAY_OF_MONTH

  warnings.warn(message, FutureWarning)

0.88
```

```
predict_delay('2/10/2018 21:45:00', 'JFK', 'ATL')
```

```
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/sklearn/
base.py:493: FutureWarning: The feature names should match those that
were passed during fit. Starting version 1.2, an error will be raised.
Feature names unseen at fit time:
- DAY
Feature names seen at fit time, yet now missing:
- DAY_OF_MONTH

  warnings.warn(message, FutureWarning)

0.87
```

```
predict_delay('2/10/2018 10:00:00', 'ATL', 'SEA')
```

```
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/sklearn/
base.py:493: FutureWarning: The feature names should match those that
were passed during fit. Starting version 1.2, an error will be raised.
Feature names unseen at fit time:
- DAY
Feature names seen at fit time, yet now missing:
- DAY_OF_MONTH

  warnings.warn(message, FutureWarning)

0.99
```

```
train_x
```

13.1.3 Mainpage – HTML Code

```
{% extends "base.html" %}
{% block content%}
```

```
<div class="column is-4 is-offset-4">
    <h3 class="title">Enter the details of your scheduled flight</h3>
    <div class="box">
        <form method="POST" action="" class="">
            <p>Enter the Date of the Flight:</p>
            <div class="input-field">
                <input name="date_flight" input type='date'  required>

            </div>
            <p>Enter the Departure Time in 24 hour format:</p>

            <input name="time_flight" input type='time' value='13:45:00' step='1'  required>

            <p>Enter the Source:</p>
        <!--input name="source" required-->

            <select name="source" id="source" class="form-select" >
                <option value="ATL">ATL-Hartsfield-Jackson Atlanta International Airport,
Atlanta</option>
                <option value="JFK">JFK-John F kennedy International Airport, New York</option>
                <option value="DTW">DTW-Detroit Metropolitan Wayne County Airport-Detroit,
Michigan</option>
                <option value="MSP">MSP-Minneapolis−Saint Paul International Airport,
Minnesota</option>
                <option value="SEA">SEA-Seattle-Tacoma International Airport, Washington</option>
            </select>
            <br>
            <p>Enter the Destination:</p>
            <select name="destination" id="source" class="form-select">
                <option value="ATL">ATL-Hartsfield-Jackson Atlanta International Airport,
Atlanta</option>
        <option value="JFK">JFK-John F kennedy International Airport, New York</option>
        <option value="DTW">DTW-Detroit Metropolitan Wayne County Airport-Detroit,
Michigan</option>
        <option value="MSP">MSP-Minneapolis−Saint Paul International Airport,
Minnesota</option>
        <option value="SEA">SEA-Seattle-Tacoma International Airport, Washington</option>
      </select>

      <br>
      <br>
      <!--button class="button is-block is-info is-large is-fullwidth" type="submit" >Submit</button--
>
      <button class="btn btn-link" type="submit">Predict</button>
  </form>
  </div>
</div>
<style>
  input{
    -webkit-text-fill-color:gray;
  }
  .checkbox{
    -webkit-text-fill-color: gray;
  }
  button{
    background-color: black  !important;
  }
  button a:hover{
    color:aqua !important;
  }
  option{
        color:black;
        background-color:white;
    }
    option :hover{
        background-color:aqua !important;
    }
    p{
        text-align:left;

    }
</style>
{% endblock %}
```

13.1.4 Prediction Output Page – HTML Code

```
{% extends "base.html" %}
{% block content %}
<h2>{{predict_str}}</h2>
<form action="/prediction">
    <button class="btn btn-link"  type="submit"><a hef="/prediction">Predict Again</a></button>
</form>

<style>
    input{
        -webkit-text-fill-color:gray;
    }
    .checkbox{
        -webkit-text-fill-color: gray;
    }
    button{
        background-color: black  !important;
    }
    button a:hover{
        color:aqua !important;
    }
    option{
        color:black;
        background-color:white;
    }
    option :hover{
        background-color:aqua !important;
    }
</style>
{% endblock %}
```

13.1.5 Flask Application

main.py

```
from flask import Blueprint, render_template,request,redirect,url_for
from flask_login import login_required, current_user
from . import db
import requests
import flask
from flask_cors import CORS
from datetime import datetime

API_KEY = "b1papptuFebhE9mB86BRaPcjkCS3jwsVV_69I5w3os7E"
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}
main = Blueprint('main', __name__)
@main.route('/')
def index():
 return render_template('index.html')

@main.route('/profile')
@login_required
def profile():
    return render_template('profile.html', name=current_user.name)

@main.route('/prediction')
@login_required
def prediction():
    return render_template('prediction.html')

@main.route('/prediction',methods=['POST'])
@login_required
def prediction_post():
    departure_date=request.form['date_flight']
    departure_time=request.form['time_flight']
    departure_date_lis=departure_date.split('-')
    departure_date_str=departure_date_lis[2]+"/"+departure_date_lis[1]+"/"+departure_date_lis[0]
    origin=request.form['source']
    destination=request.form['destination']
    departure_date_time=departure_date_str+" "+departure_time
    try:
        departure_date_time_parsed = datetime.strptime(departure_date_time, '%d/%m/%Y
%H:%M:%S')
    except ValueError as e:
        return 'Error parsing date/time - {}'.format(e)

    month = departure_date_time_parsed.month
    day = departure_date_time_parsed.day
    day_of_week = departure_date_time_parsed.isoweekday()
    hour = departure_date_time_parsed.hour

    origin = origin.upper()
    destination = destination.upper()
    X= [[month, day, day_of_week, hour, 1 if origin == 'ATL' else 0, 1 if origin == 'DTW' else 0,
    1 if origin == 'JFK' else 0, 1 if origin == 'MSP' else 0, 1 if origin == 'SEA' else 0,
    1 if destination == 'ATL' else 0, 1 if destination == 'DTW' else 0, 1 if destination == 'JFK' else 0,
    1 if destination == 'MSP' else 0, 1 if destination == 'SEA' else 0 ]]
    print(X)


    #predict= model.predict(X)[0]
    #print(predict)
    pred=['Flight is on Time','Flight is Delayed']
    payload_scoring = {"input_data": [{"field":
[['MONTH','DAY','DAY_OF_WEEK','CRS_DEP_TIME','ORIGIN_ATL',

'ORIGIN_DTW','ORIGIN_JFK','ORIGIN_MSP','ORIGIN_SEA','DEST_ATL','DEST_DTW','DEST_JFK'
,'DEST_MSP','DEST_SEA']], "values": X}]}
    response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/82a07ea5-a22b-4882-acc3-
7edb67a61b88/predictions?version=2022-11-15', json=payload_scoring,
    headers={'Authorization': 'Bearer ' + mltoken})
    print(response_scoring)
    predictions = response_scoring.json()
    print(predictions)
    predict = int(predictions['predictions'][0]['values'][0][0])
    predict_str=pred[predict]
    print("Final prediction :",predict_str)

    # showing the prediction results in a UI# showing the prediction results in a UI
    return render_template('output.html', predict_str=predict_str)
@main.route('/output')
@login_required
def predict_again():
    return render_template('prediction.html')
```

auth.py

```
from flask import Blueprint, render_template,redirect,url_for,request, flash
from werkzeug.security import generate_password_hash, check_password_hash
from . import db
from .models import User
from flask_login import login_user, login_required, logout_user

auth = Blueprint('auth', __name__)
@auth.route('/login')
def login():
    return render_template('login.html')


@auth.route('/login', methods=['POST'])
def login_post():
    # login code goes here
    email = request.form.get('email')
    password = request.form.get('password')
    remember = True if request.form.get('remember') else False

    user = User.query.filter_by(email=email).first()

    # check if the user actually exists
    # take the user-supplied password, hash it, and compare it to the hashed password in the database
    if not user or not check_password_hash(user.password, password):
        flash('Please check your login details and try again.')
        return redirect(url_for('auth.login')) # if the user doesn't exist or password is wrong, reload the page

    # if the above check passes, then we know the user has the right credentials
    login_user(user, remember=remember)
    return redirect(url_for('main.profile'))


@auth.route('/signup')
def signup():
    return render_template('sign_up.html')


@auth.route('/signup', methods=['POST'])
def signup_post():
    email = request.form.get('email')
    name = request.form.get('name')
    password = request.form.get('password')

    user = User.query.filter_by(email=email).first() # if this returns a user, then the email already exists in database
    if user: # if a user is found, we want to redirect back to signup page so user can try again
        flash('Email address already exists')
        return redirect(url_for('auth.signup')
.
    new_user = User(email=email, name=name, password=generate_password_hash(password,
method='sha256'))

    # add the new user to the database
    db.session.add(new_user)
    db.session.commit()
    return redirect(url_for('auth.login'))


@auth.route('/logout')
@login_required
def logout():
    logout_user()
    return redirect(url_for('main.index'))
```

Init.py

```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy
from flask_login import LoginManager

# init SQLAlchemy so we can use it later in our models
db = SQLAlchemy()


def create_app():
    app = Flask(__name__)

    app.config['SECRET_KEY'] = 'secret-key-goes-here'
    app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///db.sqlite'

    db.init_app(app)
    login_manager = LoginManager()
    login_manager.login_view = 'auth.login'
    login_manager.init_app(app)


    from . import models
    with app.app_context():
        db.create_all()
    from .models import User
    @login_manager.user_loader
    def load_user(user_id):
        # since the user_id is just the primary key of our user table, use it in the query for the user
        return User.query.get(int(user_id))

    # blueprint for auth routes in our app
    from .auth import auth as auth_blueprint
    app.register_blueprint(auth_blueprint)

    # blueprint for non-auth parts of app
    from .main import main as main_blueprint
    app.register_blueprint(main_blueprint)

    return app
```

models.py

```
from . import db
from flask_login import UserMixin
from . import db

class User(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True) # primary keys are required by SQLAlchemy
    email = db.Column(db.String(100), unique=True)
    password = db.Column(db.String(100))
    name = db.Column(db.String(1000))
```

13.2 GITHUB & PROJECT DEMO LINK

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-30699-1660155180

PROJECT DEMO LINK:

Team ID
PNT2022TMID16204

This is the homepage
Of our app