**1. INTRODUCTION**

1.1 PROJECT OVERVIEW

Over the last twenty years, air travel has been increasingly preferred among travelers, mainly because of its speed and in some cases comfort. This has led to phenomenal growth in air traffic and on the ground. An increase in air traffic growth has also resulted in massive levels of aircraft delays on the ground and in the air. These delays are responsible for large economic and environmental losses. The main objective of the model is to predict flight delays accurately in order to optimize flight operations and minimize delays.

Using a machine learning model, we can predict flight arrival delays. The input to our algorithm is rows of feature vector like departure date, departure delay, distance between the two airports, scheduled arrival time etc. We then use decision tree classifier to predict if the flight arrival will be delayed or not. A flight is considered to be delayed when difference between scheduled and actual arrival times is greater than 15 minutes. Furthermore, we compare decision tree classifier with logistic regression and a simple neural network for various figures of merit. Category: Machine Learning

Skills Required: Python, Python Web Frame Works, Python For Data Visualization, Data Preprocessing Techniques, Machine Learning, IBM Cloud, IBM Watson Studio, Python-Flask

1.2 PURPOSE

The aim of this project is developing a Flight Delay Prediction Model Using Machine Learning. The purpose of this project is to look at the approaches used to build models for predicting flight delays that occur due to bad weather conditions.

Average aircraft delay is regularly referred to as an indication of airport capacity. Flight delay is a prevailing problem in this world. A few factors responsible for the flight delays like runway construction to excessive traffic are rare, but bad weather seems to be a common cause. Some flights are delayed because of the reactionary delays, due to the late arrival of the previous flight. It hurts airports, airlines, and affects a company's marketing strategies as companies rely on customer loyalty to support their frequent flying programs.

**2. LITERATURE SURVEY**

Since two decades, rapid growth in air traffic is observed due to comfort, flexibility, and speed. Every year, huge amount loss is noticed due to delay of flights in as per the reports of FAA (Federal Aviation Administration).According to Federal authorities if delay is more than 3 hours for domestic flights and more than 4 hours for International flights the airlines companies have to pay penalty. To avoid the paying of penalty to customer the airlines companies want to maintain a continues relationship among them. Air transportation provides services in the aviation sector and creates wider socioeconomic settlement through its potential to enable convinced types of actions in a local market. In addition, the economic impact of flight delays for domestic flights in the is probable to be more than $19 Billion per year to the airlines and over $41 Billion per year to the national economy.

2.1 EXISTING PROBLEM

Flight delays has become a very important subject for air transportation all over the world because of the associated financial loses that the aviation industry is going through. According to data from the Bureau of Transportation Statistics (BTS) of the United Stated, over 20% of US flights were delayed during 2018, which resulted in a severe economic.

These delays not only cause inconveniences to the airlines but also to the passengers. The result is an increase in travel time which increases the expenses associated with food and lodging and ultimately causes stress among passengers. The airlines are victims of extra costs associated to their crews, aircraft repositioning, fuel consumption while trying to reduce elapse times, and many others which together tamper their reputation and often result in a loss of demand by passengers.

2.2 REFERENCE

[1] The main concern of the researchers and analysts is to predict the reasons for flight delays and for that they have put in their efforts on collecting data about flight and the weather. Mohamed et al.

[2] Have studied the pattern of arrival delay for non-stop domestic flights at the Orlando International Airport. They focused primarily on the cyclic variations that happen in the air travel demand and the weather at that particular airport. In Shervin et al.’s work

[3] Their motive of research is to propose an approach that improves the operational performance without hampering or effecting the planned cost. Adrian et al.

[4] Have created a data mining model which enables the flight delays by observing the weather conditions. They have used WEKA and R to build their models by selecting different classifiers and choosing the one with the best results. They have used different machine learning techniques like logistic regression Analysis classifier. 4 Choi et al.

[5] Have focused on overcoming the effects of the data imbalancing caused during data training. They have used techniques like Decision Trees, AdaBoost, and K-Nearest Neighbors for predicting individual flight delays. A binary classification was performed by the model to predict the scheduled flight delay. Schaefer et al.

[6] Has done a sentiment analysis and opinion mining that analyzes people’s opinions, sentiments, and studies their behavior. The output of the research is a feature-based opinion summary which is also known as sentiment classification.

2.3PROBLEM STATEMENT DEFINITION

My case study was about LaGuardia Airport in New York, Logan International Airport in Boston, San Francisco International Airport in San Francisco, and O’Hare International Airport in Chicago, which are four major airports in the United States of America. But we focused the idea and research on LaGuardia International Airport. Compared with the data produced by all airports in USA, the data which we gathered was very limited, but it gave us a great direction on how weather plays a part in flight delays. In this project, the goal is to use exploratory analysis and to build machine learning models to predict airline departure and arrival delays.

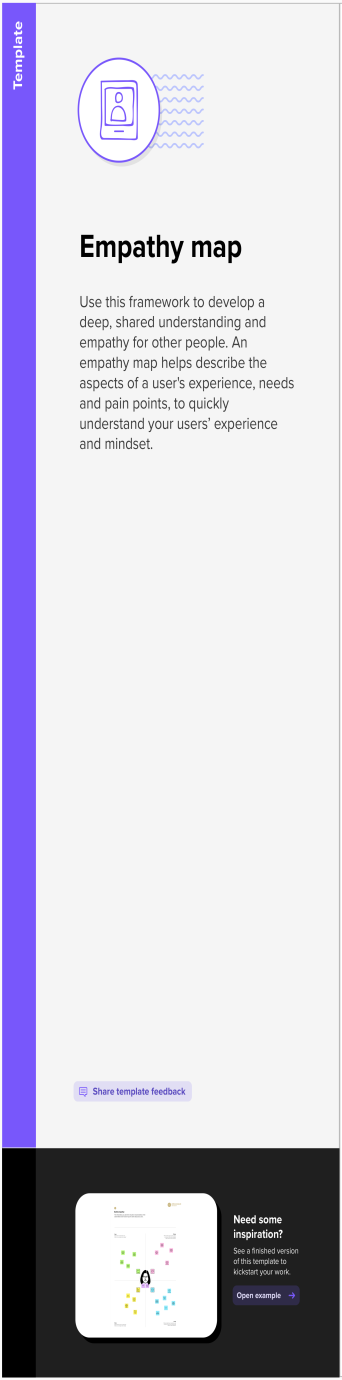
**3. IDEATION & PROPOSED SOLUTION**

3.1 EMPATHY MAP CANVAS

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user personal, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community. The empathy map represents a principal user and helps teams better understand their motivations, concerns, and user experience.

**The 4 Attributes of Empathy:**

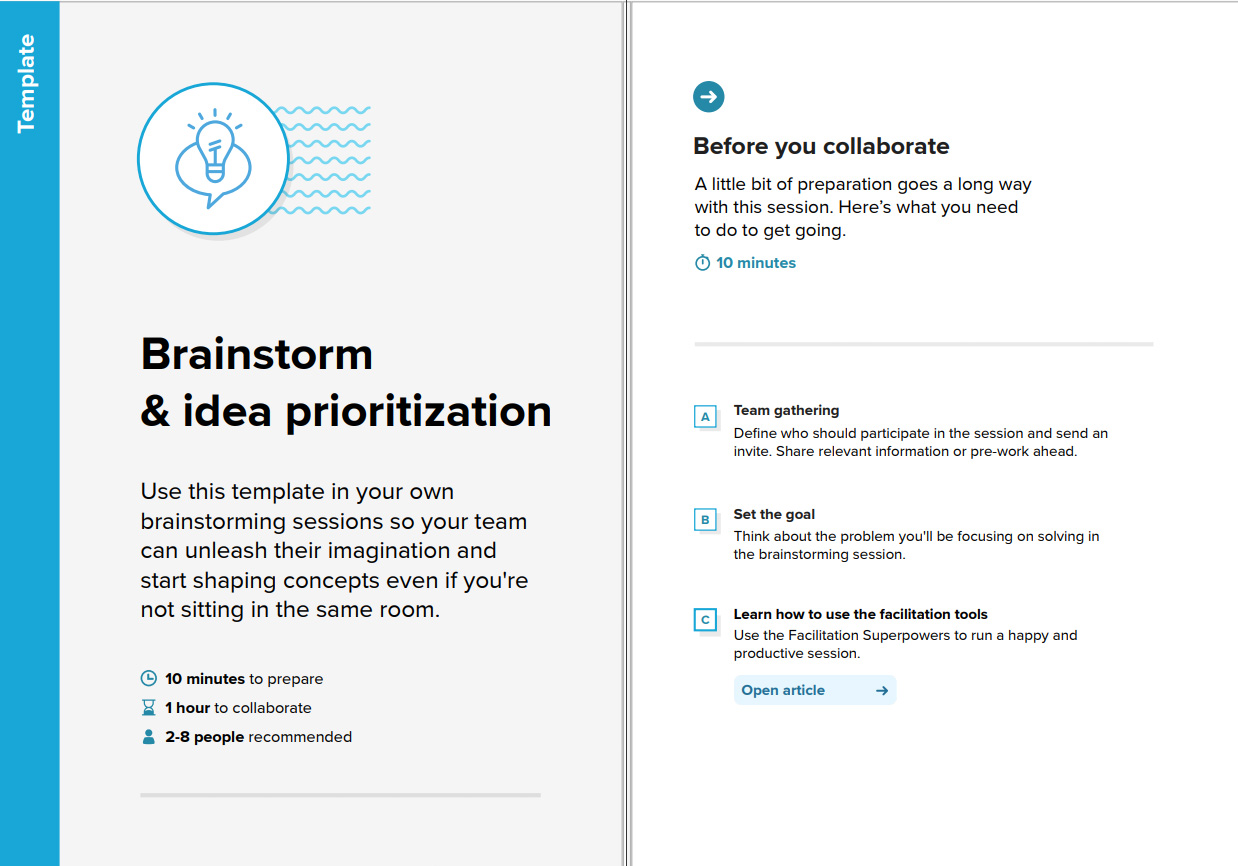
1. Perspective taking.
2. Staying out of judgment.
3. Recognizing emotion in another person.
4. Communicating the understanding of another person's emotions.



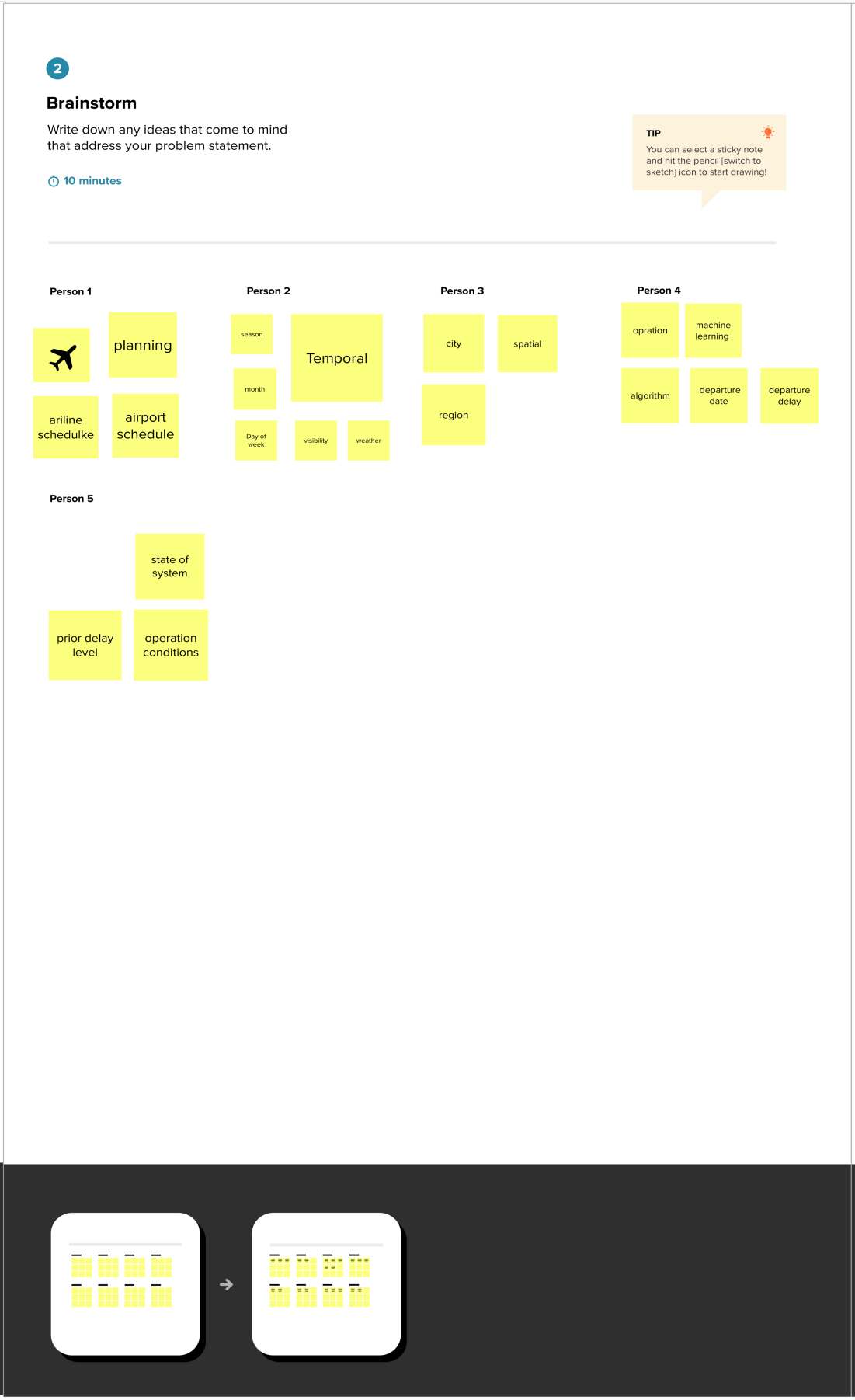
3.2 IDEATION & BRAINSTORMING

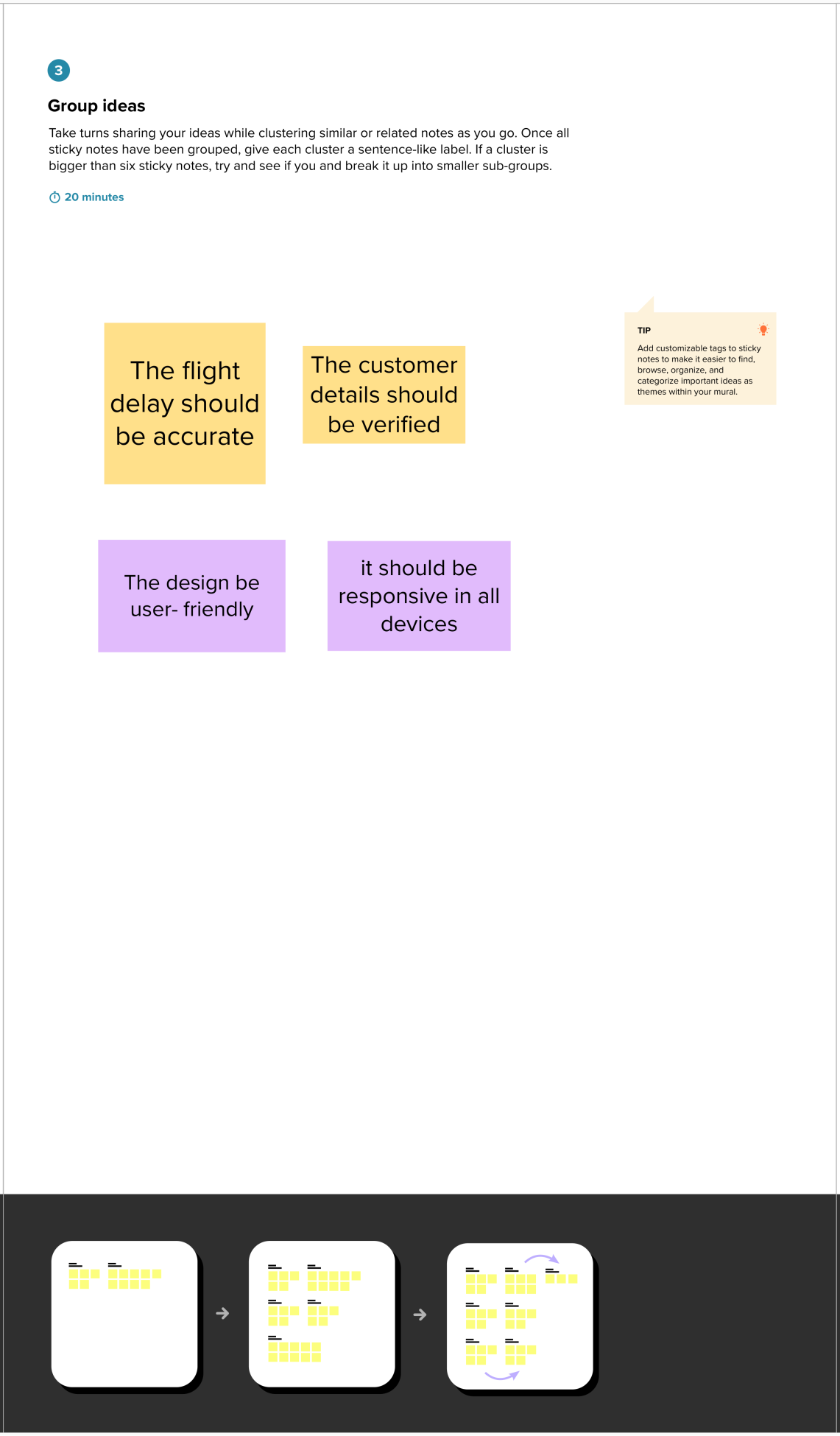
**Benefits of Brainstorming:**

1. Provides a quick and easy class activity. Brainstorming sessions can be effectively used in the classroom.
2. Contributes to classroom collective power.
3. Creates a student-centered activity.
4. Supports learning in a relaxed environment.
5. Strengthens problem-based learning.
6. Encourages creative thought**.**

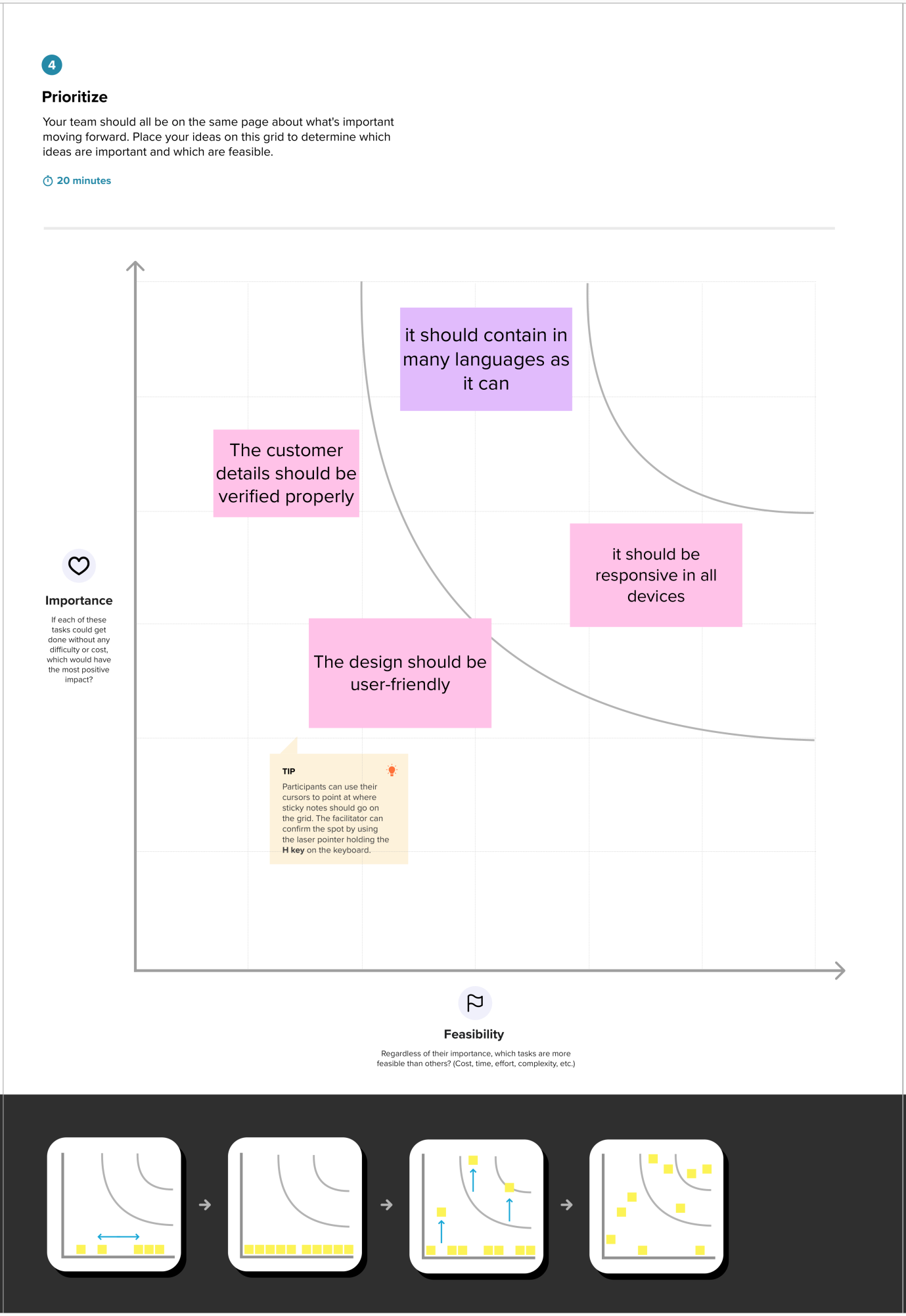


Step-2: Brainstorm, Idea Listing and Grouping





Step-3: Idea Prioritization



3.3 Define the Problem Statements

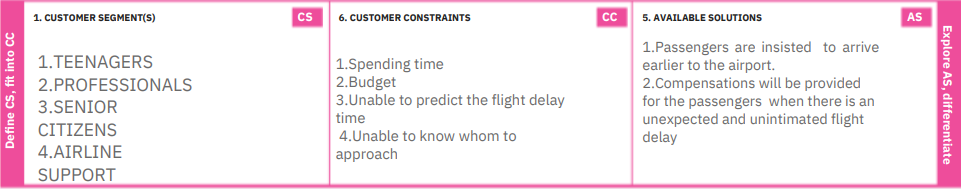


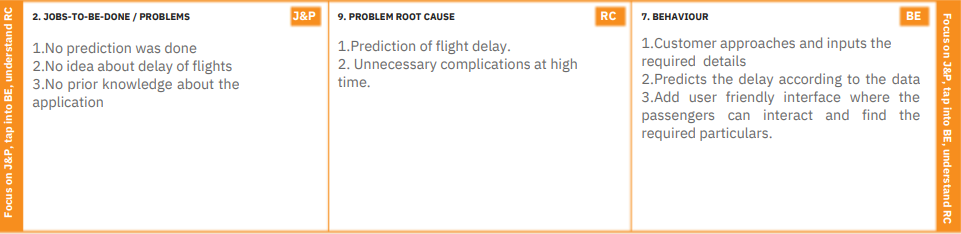
3.4 PROPOSED SOLUTION

As discussed, weather condition plays an important role in proper and timely functioning of flights. We propose a flight delay prediction system which focuses mainly on predicting delay of a flight based on the weather situation. To make the system more scalable it is necessary to choose an algorithm which considers all the parameters to be independent. Essentially supervised learning could be a learning that within which we tend to teach or train the machine exploitation data which is well tagged which means some data is already labeled with correct answer. After that, machine is given new set data of so machine learning algorithmic rule analyses the coaching knowledge and produces an correct outcome from tagged data Using machine learning approach, the labeled data gives it authenticity. logistic regression model is one of the algorithm which is proven to be efficient for real time prediction as well as the fact that it considers every attribute to be independent from each other makes it an apt algorithm for the concerned project The proposed system takes the city of departure. It then returns the predicted weather data using an API key and passes the data into the algorithm. The attributes considered for calculations and taken by the API are as follows weather, temperature, Visibility and Month number. As discussed that machine learning is based on having a set of correct labeled data form which the algorithm bases its prediction. We use a csv file for storing that data as a flat file format is easier to edit, update and retrieve it for calculations.

3.5 PROBLEM SOLUTION FIT

The work presented in and approach the optimization problem differently. focus on choosing the shortest path for an aircraft with the existing data, applying their method to all the aircrafts on the ground while making corrections one-the-fly in case of an interaction with another aircraft. The objective of this study is to show that the Ant algorithm can optimize taxi paths, hence taxi-times. in focus on setting the taxi-time for an aircraft, and then choosing the right taxi-route to minimize interactions with other aircrafts.







**4. REQUIREMENT ANALYSIS**

4.1 FUNCTIONAL REQUIREMENT

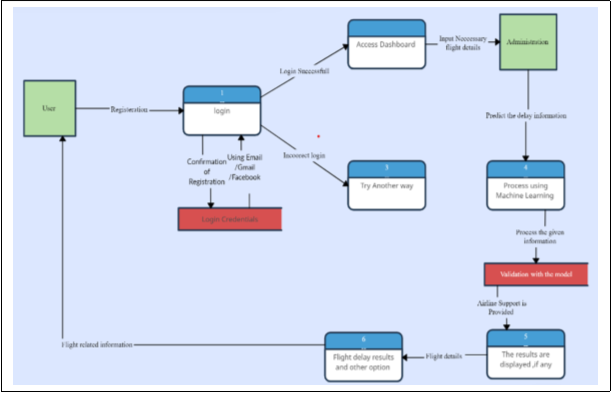
|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | Registration through Form  Registration through Gmail  Registration through LinkedIN |
| FR-2 | User Confirmation | Confirmation via Email  Confirmation via OTP |
| FR-3 | Flight code validation | Validation through airline database |
| FR-4 | Delay prediction | Prediction result through Gmail Prediction result through Message |
| FR-5 | Airline Support | Collaboration with the airlines customer support |
| FR-6 | Algorithm for prediction | Train the data Predict the data |

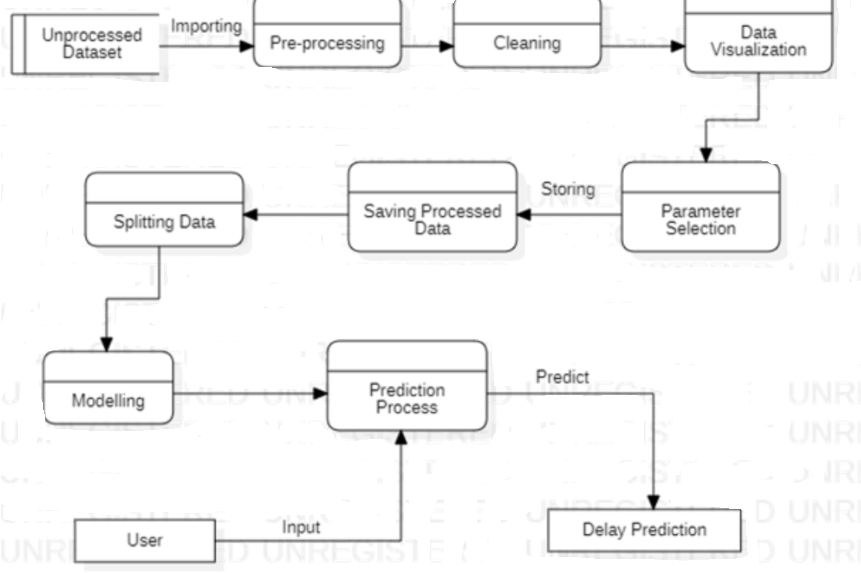
4.2 NON-FUNCTIONAL REQUIREMENTS

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | The UX/UI will be user-friendly and will be highly responsive |
| NFR-2 | **Security** | The data security will be ensured by using appropriate security measures |
| NFR-3 | **Reliability** | The application performs for all proper data efficiently |
| NFR-4 | **Performance** | The appliation response time will be faster and the data will be processed at a higher speed |
| NFR-5 | **Availability** | The application is highly compatabile in any devices. |
| NFR-6 | **Scalability** | The application is designed in such a way that it can handle ,if any high traffic occurs |

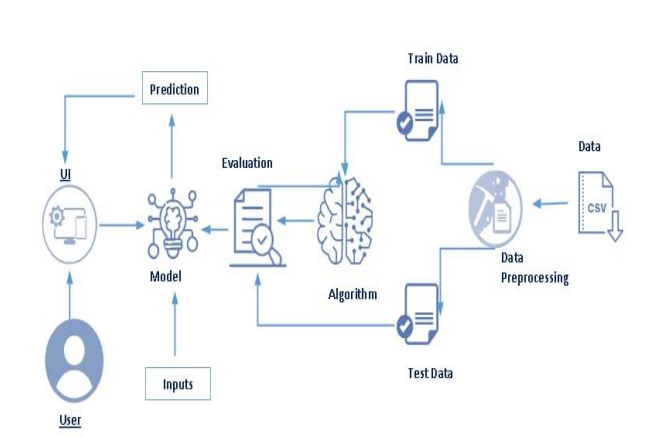
**5.PROJECT DESIGN**

5.1 DATA FLOW DIAGRAMS

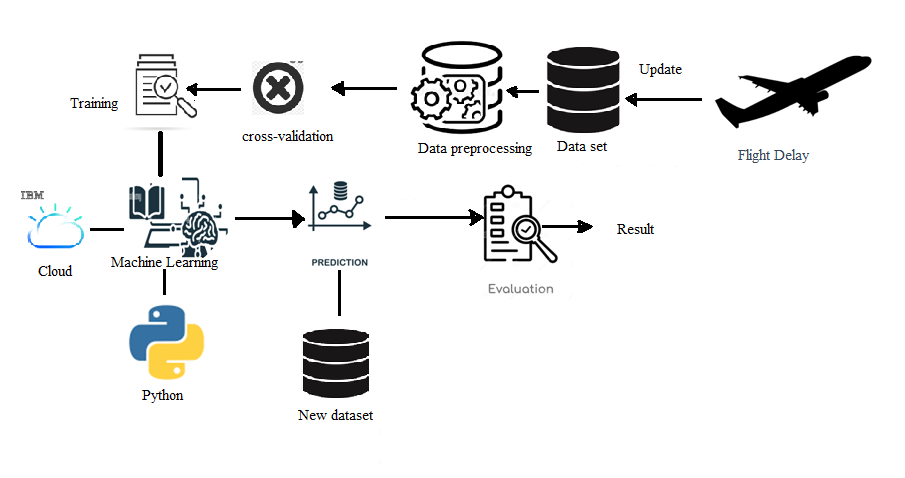
****



5.2 SOLUTION & TECHNICAL ARCHITECTURE

****

TECHNICAL ARCHITECTURE:

****

* 1. USER STORIES



****

1. **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 | Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | 2 | High | K.Swathi |
| Sprint-1 | User confirmation | USN-2 | As a user, I will receive confirmation email once I have registered for the application | 1 | Medium | S.Vinothini |
| Sprint-1 | Login | USN-3 | As a user, I can lig into the aapplicationby entering email and password | 1 | High | A.Abitha |
| Sprint-2 | Analysis | USN-4 | As a user, I can analyse the dataset | 2 | Medium | S.Sumithra |
| Sprint-3 | Develop and Train | USN-5 | As a user, I can develop and train the model predict the flight delay | 2 | High | k.swathi |
| Sprint-4 | Application |  | Shows the flight details | 2 | High | S.Vinothini |

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** |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint-1 | 20 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 20 | 29 Oct 2022 |
| Sprint-2 | 20 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 20 | 05 Nov 2022 |
| Sprint-3 | 20 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 20 | 07 Nov 2022 |
| Sprint-4 | 20 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 20 | 19 Nov 2022 |

**7. CODING & SOLUTIONING**

7.1 FEATURE 1

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

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

<!-- The above 3 meta tags \*must\* come first in the head; any other head content must come \*after\* these tags -->

<title>Flight Delay Prediction</title>

<!-- Google font -->

<link href="https://fonts.googleapis.com/css?family=Lato:400,700" rel="stylesheet">

<!-- Bootstrap -->

<linktype="text/css"rel="stylesheet"href="{{url\_for('static',filename='css/bootstrap.min.css') }}" />

<!-- Custom stlylesheet -->

<link type="text/css" rel="stylesheet" href="{{ url\_for('static', filename='css/style.css') }}" />

<style>

#booking

{

font-family: 'Lato', sans-serif;

background: url(../static/background.jpg);

background-size: cover;

background-position: center;

color: #191a1e;

}

</style>

<!--<img src="{{url\_for('static', filename='background.png')}}" />-->

<!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->

<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

<![endif]-->

</head>

<body>

<div id="booking" class="section">

<div class="section-center">

<div class="container">

<div class="row">

<div class="col-md-4">

<div class="booking-cta">

<h1>Flight Delay Prediction</h1>

<p></p>

<div class="container">

<!-- {{ prediction\_text }} -->

{% if prediction\_text == 0 %}

<h2>The flight is not delayed</h2>

{% elif prediction\_text == 1 %}

<h2>The flight is delayed</h2>

{% endif %}

<div class="col-md-4">

<div class="form-group">

<span class="form-label">Month</span>

<input type="text" class="form-control" name="month" placeholder="Enter month" required="true">

<span class="select-arrow"></span>

</div>

</div>

<!Month>

<div class="col-md-4">

<div class="form-group">

<span class="form-label">Date</span>

<input type="text" class="form-control" name="day" placeholder="Enter date" required="true">

<span class="select-arrow"></span>

</div>

</div>

<! Date>

</div>

<!--Year,Month,Date end-->

<div class="row">

<div class="col-md-6">

<div class="form-group">

<span class="form-label">Select an Airline</span>

<!-- <input class="form-control" type="date" required> -->

<select class="form-control" name="carrier">

</select>

</div>

</div>

<!Airline>

</div>

<!--Airline end-->

<div class="row">

<div class="col-md-6">

<div class="form-group">

<span class="form-label">Flying from</span>

<!--<input class="form-control" type="text" placeholder="City or airport">-->

<select class="form-control" name="origin">

<option value="EWR">Newark Liberty International Airport(EWR)</option>

<option value="JFK">John F. Kennedy International Airport(New York International Airport)(JFK)</option>

<option value="LGA">LaGuardia Airport(Marine Air Terminal)(LGA)</option>

</select>

</div>

</div>

<!Flying from>

<div class="col-md-6">

<div class="form-group">

<span class="form-label">Flying to</span>

<!-- <input class="form-control" type="text" placeholder="City or airport"> -->

<select class="form-control" name="dest">

<option value="ATL">Hartsfield-Jackson Atlanta InternationalAirport(ATL)</option>

<option value="ORD">Chicago O'Hare International Airport(ORD)</option>

<option value="LAX">Los Angeles International Airport(LAX)</option>

<option value="BOS">Gen. Edward Lawrence Logan International Airport(BOS)</option>

<option value="MCO">Orlando International Airport(MCO)</option>

<option value="STL">St. Louis International Airport at Lambert Field(STL)</option>

<option value="MDW">Chicago Midway International Airport(MDW)</option>

<option value="SEA">Seattle-Tacoma International Airport(SEA)</option>

<option value="HNL">Honolulu International Airport(HNL)</option>

<option value="LGB">Long Beach AirportÂ (Daugherty Field)(LGB)</option>

<option value="SAT">San Antonio International Airport(SAT)</option>

<option value="TYS">McGhee Tyson Airport(TYS)</option>

</select>

</div>

</div>

<!Flying to>

</div>

<!--Flying to,from end-->

<div class="form-btn">

<button class="submit-btn">Predict</button>

</div>

<!Button>

</form>

<!--Form end-->

</div>

<!Booking form>

</div>

</div>

</body>

</html>

**CSS FILE**

.section {

position: relative;

height: 100vh;}

.section .section-center {

position: absolute;

top: 50%;

left: 0;

right: 0;

-webkit-transform: translateY(-50%);

transform: translateY(-50%);}

#booking {

font-family: 'Lato', sans-serif;

background-image:url('{{url\_for('static', filename='img/background.png') }}');

background-size: cover;

background-position: center;

color: #191a1e;}

.booking-form {position: relative;background: #fff;max-width: 642px;width: 100%;margin: auto;padding: 45px 25px 25px;border-radius: 4px;

-webkit-box-shadow: 0px 0px 10px -5px rgba(0, 0, 0, 0.4);

box-shadow: 0px 0px 10px -5px rgba(0, 0, 0, 0.4);}

.booking-form .form-group {

position: relative;

margin-bottom: 20px;}

.booking-form .form-control {

background-color: #fff;

height: 65px;

.booking-form .form-control::placeholder {

color: #dfe5e9;}

.booking-form .form-control:focus {

background: #f9fafb;}

.booking-form input[type="date"].form-control:invalid {

color: #dfe5e9;}

-moz-appearance: none;

appearance: none;}

.booking-form select.form-control+.select-arrow {

position: absolute;

right: 6px;

bottom: 6px;

width: 32px;

line-height: 32px;

height: 32px;

display: block;

-webkit-transform: rotate(90deg);

transform: rotate(90deg);}

.booking-form .submit-btn {

color: #fff;

background-color: #4fa3e3;

border-radius: 4px;

text-transform: uppercase

**8.TESTING**

8.1 TEST CASES

A test case has components that describe input, action and an expected response, in order to determine if a feature of an application is working correctly.A test case is a set of instructions on “HOW” to validate a particular test objective/target, which when followed will tell us if the expected behavior of the system is satisfied or not.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **Scenario** | **Input** | **Excepted output** | **Actual output** |
| 1 | User login | User name and password | Login | Login success |
| 2 | Login success | Upload Flight delay prediction | Detecting object and analyze for prediction | Details are stored in a database. |

8.2 USER ACCEPTANCE TESTING

This sort of testing is carried out by users, clients, or other authorized bodies to identify the requirements and operational procedures of an application or piece of software. The most crucial stage of testing is acceptance testing since it determines whether or not the customer will accept the application or programming. It could entail the application's U.I., performance, usability, and usefulness. It is also referred to as end-user testing, operational acceptance testing, and user acceptance testing (UAT)

**9. RESULTS**

9.1 PERFORMANCE METRICS

X=df\_new.iloc[:,:-1].values

y=df\_new.iloc[:,7].values

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.3,random\_state=0)

from sklearn.metrics import confusion\_matrix,accuracy\_score,classification\_report,roc\_curve,roc\_auc\_score,mean\_squared\_error

from sklearn.linear\_model import LinearRegression

regression=LinearRegression()

regression.fit(X\_train,y\_train)

LinearRegression()

y\_prediction=regression.predict(X\_test)

cutoff=0.7

y\_prediction\_classes=np.zeros\_like(y\_prediction)

y\_prediction\_classes[y\_prediction>cutoff]=1

y\_test\_classes=np.zeros\_like(y\_test)

y\_test\_classes[y\_test>cutoff]=1

**10. ADVANTAGES & DISADVANTAGES**

ADVANTAGE

* + 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.
  + predicting flight delays can improve airline operations and passenger satisfaction, which will result in a positive impact on the economy. In this study, the main goal is to compare the performance of machine learning classification algorithms when predicting flight delays.

DISADVANTAGE

* Flight delays not only irritate air passengers and disrupt their schedules but also cause a decrease in efficiency, an increase in capital costs, reallocation of flight crews and aircraft, and additional crew expenses

**11. CONCLUSION**

In this project, we were able to successfully apply machine learning algorithms to predict flight arrival-delay and show simple classifiers like decision tree and logistic regression can predict if a flight’s arrival will be delayed or not fairly accurately. For further work we like to further improve our models, perhaps with more training-data or deeper neural network, or both.Delay prediction is a natural progression to this work, considering amount of fuel wasted while taxiing. Accurate taxi-delay prediction requires taking airport runway and taxiway configurations in to consideration where very little work exists.

**12. FUTURE SCOPE**

This project is based on data analysis from year 2008. A large dataset is available from 1987-2008 but handling a bigger dataset requires a great amount of preprocessing and cleaning of the data. 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 to process the data. With the new advancement in the field of machine 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 modeling 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**

**Appy.py**

from flask import Flask, request, jsonify, render\_template, url\_for , request

import pickle

from sklearn.preprocessing import LabelEncoder

from sklearn.model\_selection import train\_test\_split

import pandas as pd

# Import dataset

df = pd.read\_csv('Data/Processed\_data15.csv')

# Label Encoding

le\_carrier = LabelEncoder()

df['carrier'] = le\_carrier.fit\_transform(df['carrier'])

le\_dest = LabelEncoder()

df['dest'] = le\_dest.fit\_transform(df['dest'])

le\_origin = LabelEncoder()

df['origin'] = le\_origin.fit\_transform(df['origin'])

# Converting Pandas DataFrame into a Numpy array

X = df.iloc[:, 0:6].values # from column(years) to column(distance)

y = df['delayed']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y, test\_size=0.25,random\_state=61) # 75% training and 25% test

app = Flask(\_\_name\_\_)

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

@app.route('/')

def home():

return render\_template('index.html')

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

def predict():

year = request.form['year']

month = request.form['month']

day = request.form['day']

carrier = request.form['carrier']

origin = request.form['origin']

dest = request.form['dest']

year = int(year)

month = int(month)

day = int(day)

carrier = str(carrier)

origin = str(origin)

dest = str(dest)

if year >= 2013:

x1 = [year,month,day]

x2 = [carrier, origin, dest]

x1.extend(x2)

df1 = pd.DataFrame(data = [x1], columns = ['year', 'month', 'date', 'carrier', 'origin', 'dest'])

df1['carrier'] = le\_carrier.transform(df1['carrier'])

df1['origin'] = le\_origin.transform(df1['origin'])

df1['dest'] = le\_dest.transform(df1['dest'])

x = df1.iloc[:, :6].values

ans = model.predict(x)

output = ans

return render\_template('index.html', prediction\_text=output)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=False)

# For mac, make 'app.run(debug=True**)'**

**Python**

# Import libraries

import pandas as pd

import pickle

from sklearn.preprocessing import LabelEncoder

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import train\_test\_split # Import train\_test\_split function

# Import dataset

df = pd.read\_csv('Data/Processed\_data15.csv')

# Label Encoding

le\_carrier = LabelEncoder()

df['carrier'] = le\_carrier.fit\_transform(df['carrier'])

le\_dest = LabelEncoder()

df['dest'] = le\_dest.fit\_transform(df['dest'])

le\_origin = LabelEncoder()

df['origin'] = le\_origin.fit\_transform(df['origin'])

# Converting Pandas DataFrame into a Numpy array

X = df.iloc[:, 0:6].values # from column(years) to column(distance)

y = df['delayed'].values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=18) # 70% training and 30% test

# For 75% Train and 25% test use Random state =809

#Create a Random Forest Classifier

clf = RandomForestClassifier(random\_state=18)

clf.fit(X\_train, y\_train)

# Saving model to disk

pickle.dump(clf,open('model.pkl','wb'))

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

DEMOVIDEO LINK: https://youtu.be/NvnHGQ2RaBo