**FLIGHT DELAY PREDICTION FOR AVIATION INDUSTRY USING**

**MACHINE LEARNING**

**TEAM ID: NM2023TMID21593**

**PERUMAL.R(F2220EA4D8FC4F3BC0C42FB5794D997A)**

**CHANDRU.K(6839AC81DE40060A23F4DC1A99E47E4E)**

**PRABHAKARAN.S(7389001B69CD6869E094A4C4C071778D)**

**MAGESHBOOBATHI.V(1E7BFD335CE3FD7ABA94CBAA2B2733E6)**

**SUBMITTED FOR THE PROJECT UNDER THE NAAN MUDHALVAN – SMARTINTERNZ PROGRAM**

**INTRODUCTION**

### 1.1 Project Overview

Due to its quickness and occasional comfort, air travel has become more and more popular among tourists during the past 20 years. The result has been a spectacular increase in land traffic and air traffic. Massive levels of aircraft delays on the ground and in the air have also been brought on by an increase in air traffic. There have been significant monetary and environmental losses as a result of these delays. To optimize flight operations and reduce delays, the model's primary goal is to estimate flight delays accurately.

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 Support Vector Machine 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.

### 1.2 Project Purpose

Flight delay forecasting can enhance airline operations and passenger happiness, which will boost the economy. Comparing the effectiveness of machine learning classification systems for predicting flight delays is the major objective. Flight delays are unavoidable, and they significantly affect the carriers' profits and losses. The traveler's ability to plan ahead and avoid wasting vital time can be greatly aided by this delay prediction. For airlines, estimating flight delays correctly is essential since the data may be used to boost client happiness and revenue for airline agencies.

**IDEATION & PROPOSED SYSTEM**

### 2.1 Empathy Map Canvas

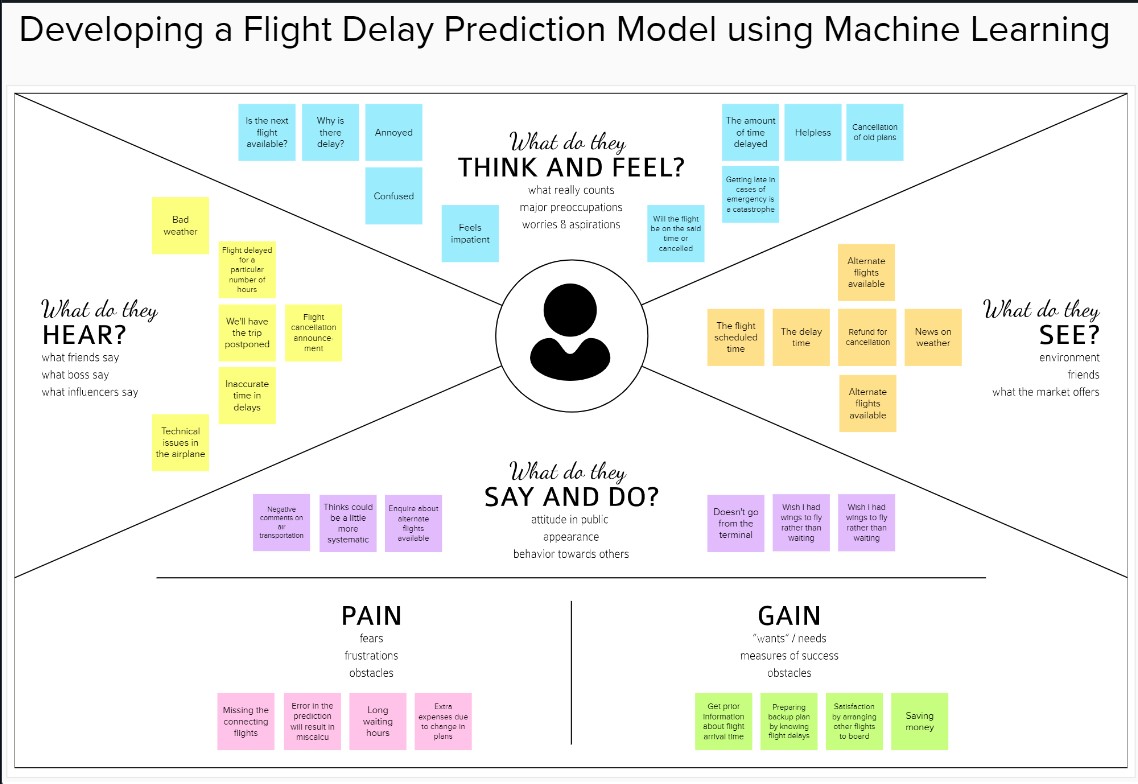


Figure 2.1 Empathy Map Canvas

### 2.2 Ideation & Brainstorming

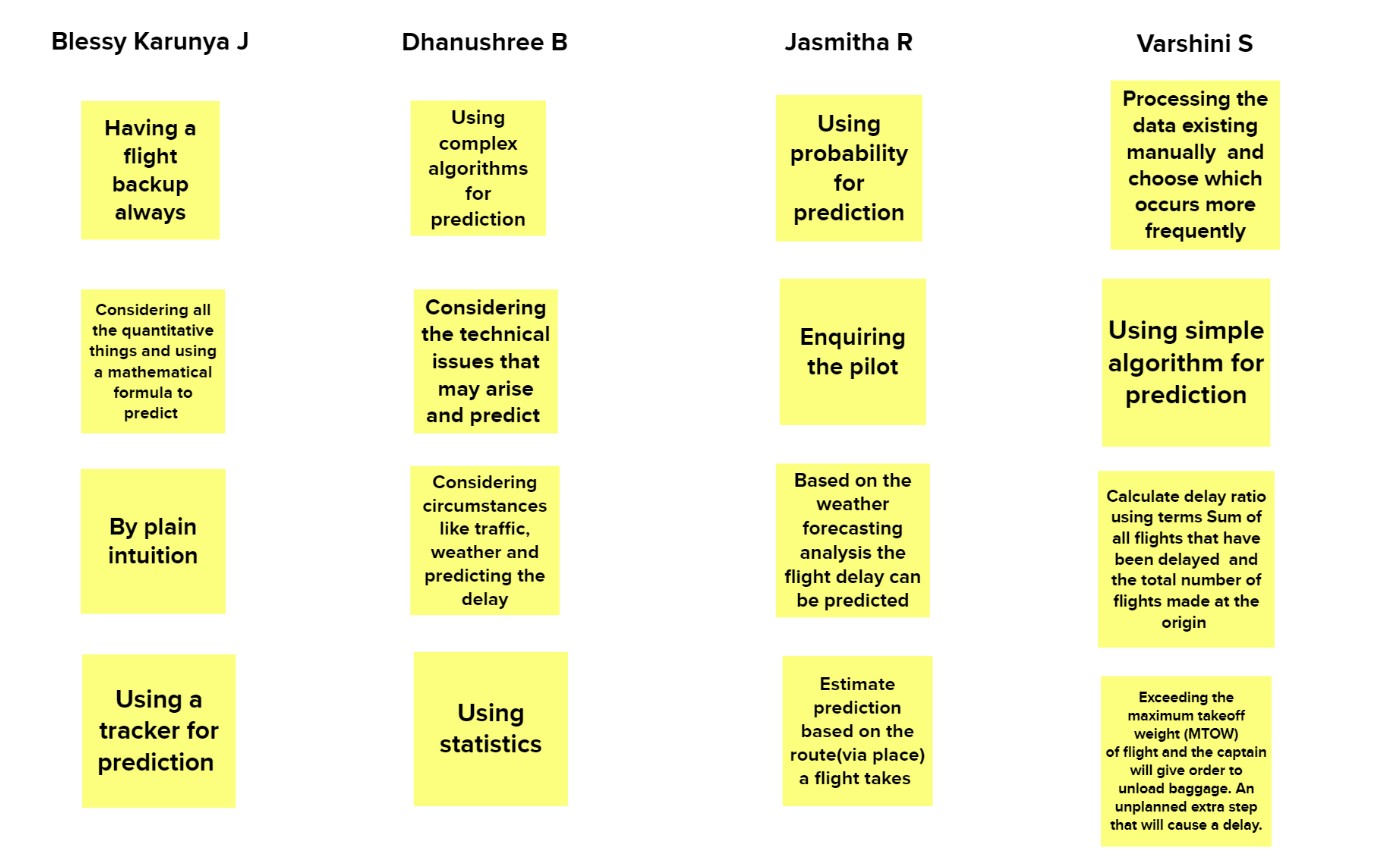


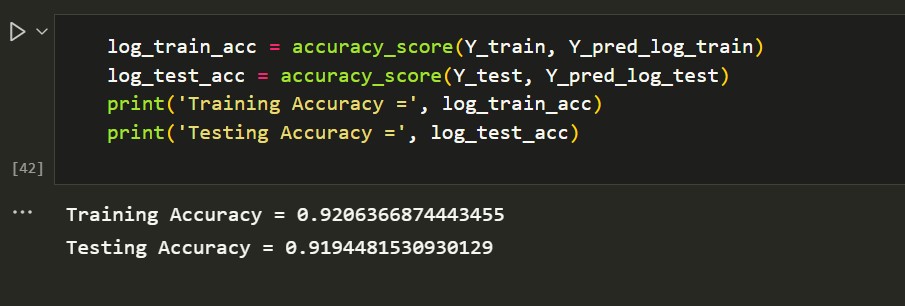
Figure 2.2 Brainstorming

**RESULTS**

### 3.1 Performance metrics

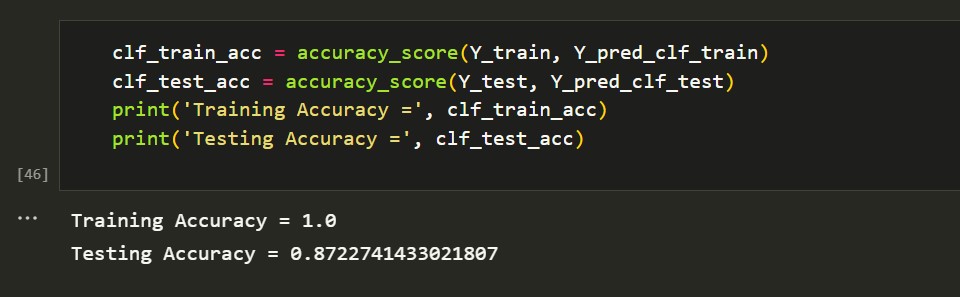
**Model: Logistic Regression performance values**

There is no big variation in the training and testing accuracy. Therefore, the Logistic Regression model is not overfit or underfit.



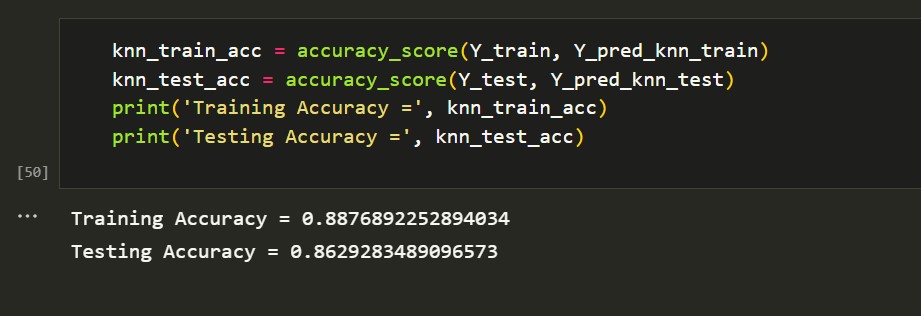
#### Model: Decision tree Classifier

There is a variation in the training and testing accuracy. The Decision tree classifier model is overfit. So, it is not chosen for best results.



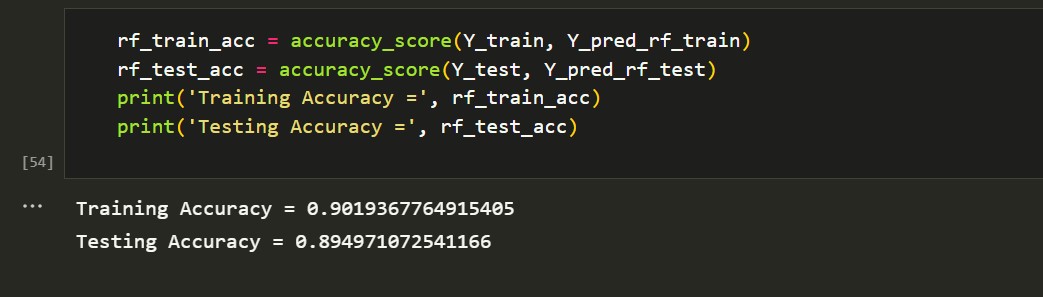
#### Model: KNN Classifier

There is no big variation in the training and testing accuracy. Therefore, the KNN Classifier model is not overfit or underfit.



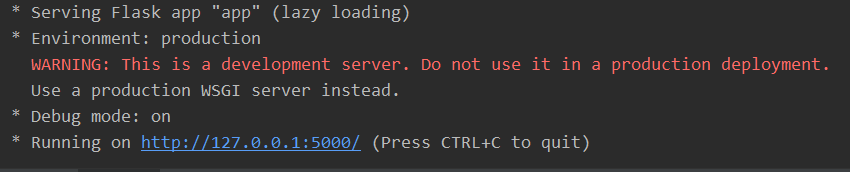
#### Model: Random Forest Classifier

There is no big variation in the training and testing accuracy. Therefore, the Random Forest Classifier model is not overfit or underfit.



On comparing the four models built, based on the performance metrics it is clear that logistic regression gives the highest performance. Hence, that model is chosen for deployment.

**Output of the Project**









## 10ADVANTAGES & DISADVANTAGES

### 4.1Advantages

1. Accuracy: Machine learning algorithms can analyze large amounts of data and identify patterns that may not be immediately apparent to human analysts. This can lead to more accurate predictions of flight delays, which can help airlines and airports plan and prepare more effectively.
2. Real-time Analysis: Machine learning models can analyze data in real-time, which means they can provide up-to-the-minute information about flight delays. This can help passengers make more informed decisions about their travel plans and help airlines and airports respond quickly to changes in flight schedules.
3. Improved Efficiency: By predicting flight delays in advance, airlines and airports can allocate resources more efficiently, which can lead to cost savings and improved operational efficiency.
4. Customer Satisfaction: Providing accurate and timely information about flight delays can help improve customer satisfaction. Passengers are less likely to be frustrated or upset if they are informed of delays in advance and provided with alternative travel options.
5. Reduced Costs: By predicting flight delays and scheduling maintenance activities in advance, airlines can reduce the cost of unscheduled maintenance activities and minimize the impact of unplanned downtime.

### 4.2Disadvantages

1. Data Quality: The accuracy of machine learning algorithms depends on the quality and quantity of data used for training. If the data used for training is incomplete or inaccurate, the predictions may not be reliable.
2. Overreliance on Predictions: While machine learning models can provide accurate predictions, there is always a degree of uncertainty associated with them. Overreliance on these predictions can lead to poor decision-making, especially if the model is not calibrated correctly.
3. Complexity: Machine learning models can be complex and difficult to understand, especially for non-technical users. This can make it challenging to explain the rationale behind predictions and to troubleshoot issues.
4. Cost: Developing and maintaining machine learning models can be expensive, requiring significant investment in hardware, software, and skilled personnel. This can be a barrier to adoption for smaller airlines and airports.
5. Limited Coverage: Machine learning models may be less effective in predicting flight delays in regions or airports with limited historical data or unique environmental factors, such as airports with frequent weather disruptions.Top of Form

**APPLICATIONS**

* Passenger Information: One of the primary applications of flight delay prediction is to provide passengers with accurate information about their flights. Passengers can be informed in advance of possible delays, which can help them plan their travel accordingly.
* Airport Operations: Flight delays can have a cascading effect on airport operations. Machine learning models can help airport operators plan and allocate resources based on predicted delays. For example, if a model predicts that a particular flight is going to be delayed, the airport can plan to have more staff available to handle the additional passengers.
* Airline Operations: Airlines can use machine learning models to optimize their operations and reduce the impact of delays on their business. For example, by predicting delays in advance, airlines can adjust their flight schedules, assign alternate routes, and even change aircraft to minimize the disruption caused by delays.
* Maintenance and Repair: Machine learning models can also be used to predict maintenance and repair requirements for aircraft. By analyzing data from previous flights, models can predict when an aircraft is likely to require maintenance or repair, which can help airlines plan and schedule maintenance activities in advance.
* Weather Forecasting: Weather is one of the primary causes of flight delays. Machine learning models can be trained to analyze weather data and predict the impact of weather conditions on flights. This can help airlines and airports plan ahead and minimize the impact of weather-related delays.

## CONCLUSION

Flight delays not only anger and disturb air travelers' plans, but they also reduce efficiency, raise capital costs, reallocate flight crews and aircraft, and add to crew costs.

The goal of the flight delay prediction model is to forecast aircraft delays caused as a lot of passengers have become dependent on flights these days for their mode of transportation. The dataset which has all important information about the flights and its delay is made use for developing the model. A lot of steps are performed right from importing the data, then pre-processing it till training and testing the model. First the necessary packages were imported then the missing values in the data were handled and it was checked for outliers and then one hot encoding was performed and scaling was done. Then the data was split and given for training.

Four different models were used for training and out of it the best one was chosen based on the performance metric which is the Logistic regression model. Once the model was built it was integrated along with the Flask framework so that the users can enter their flight details and see if the flight would be on time or get delayed. Then this model is trained and deployed in the IBM Cloud.

As a result, anticipating delays can enhance airline operations and passenger satisfaction, which will benefit the economy and bring a positive impact.

## 12 FUTURE SCOPE

1. More Accurate Predictions: Advances in machine learning techniques and access to larger datasets can help improve the accuracy of flight delay predictions. This will allow airlines and airports to better plan and allocate resources, reducing the impact of delays on passengers and operations.
2. Integration with Other Systems: Machine learning models can be integrated with other airport and airline systems, such as maintenance, scheduling, and logistics systems, to provide a more holistic view of operations. This can help airlines and airports optimize their operations and reduce the impact of delays.
3. Personalization: Machine learning can help provide personalized information to passengers about their flights, including the likelihood of delays and alternative travel options. This can help improve customer satisfaction and loyalty.
4. Real-time Analysis: Real-time analysis of flight data using machine learning models can help airlines and airports respond quickly to changes in flight schedules and minimize the impact of delays on passengers.
5. Use of Unstructured Data: Machine learning models can be trained to analyze unstructured data, such as social media and news reports, to identify potential causes of flight delays, such as weather events or air traffic control issues.

**APPENDIX**

**8.1 Source code:**

***8.1.1.1Flask file:***

### 8.1.1.1.1 app.py

from flask import Flask, render\_template, request

import pandas as pd

import joblib

import numpy as np

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

@app.route('/')

def home():

return render\_template('Flightdelay.html')

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

def predict():

fl\_num = int(request.form.get('fno'))

month = int(request.form.get('month'))

dayofmonth = int(request.form.get('daym'))

dayofweek = int(request.form.get('dayw'))

sdeptime = request.form.get('sdt')

adeptime = request.form.get('adt')

arrtime = int(request.form.get('sat'))

depdelay = int(adeptime) - int(sdeptime)

inputs = list()

inputs.append(fl\_num)

inputs.append(month)

inputs.append(dayofmonth)

inputs.append(dayofweek)

if (depdelay< 15):

inputs.append(0)

else:

inputs.append(1)

inputs.append(arrtime)

origin = str(request.form.get("org"))

dest = str(request.form.get("dest"))

if(origin=="ATL"):

a=[1,0,0,0,0]

inputs.extend(a)

elif(origin=="DTW"):

a=[0,1,0,0,0]

inputs.extend(a)

elif(origin=="JFK"):

a=[0,0,1,0,0]

inputs.extend(a)

elif(origin=="MSP"):

a=[0,0,0,1,0]

inputs.extend(a)

elif(origin=="SEA"):

a=[0,0,0,0,1]

inputs.extend(a)

if(dest=="ATL"):

b=[1,0,0,0,0]

inputs.extend(b)

elif(dest=="DTW"):

b=[0,1,0,0,0]

inputs.extend(b)

elif(dest=="JFK"):

b=[0,0,1,0,0]

inputs.extend(b)

elif(dest=="MSP"):

b=[0,0,0,1,0]

inputs.extend(b)

elif(dest=="SEA"):

b=[0,0,0,0,1]

inputs.extend(b)

prediction = preprocessAndPredict(inputs)

#Pass prediction to prediction template

print(inputs)

return render\_template('/result.html', prediction = prediction)

def preprocessAndPredict(inputs):

test\_data = np.array(inputs).reshape((1,16))

model=open('E:\\NM Smartinternz\\Flight Delay\\flight.pkl','rb')

trained\_model = joblib.load(model)

df = pd.DataFrame(data=test\_data[0:, 0:], columns=['FL\_NUM', 'MONTH', 'DAY\_OF\_MONTH', 'DAY\_OF\_WEEK', 'DEP\_DEL15', 'CRS\_ARR\_TIME', 'ORIGIN\_ATL', 'ORIGIN\_DTW', 'ORIGIN\_JFK', 'ORIGIN\_MSP', 'ORIGIN\_SEA', 'DEST\_ATL', 'DEST\_DTW', 'DEST\_JFK', 'DEST\_MSP', 'DEST\_SEA'])

data = df.values

result = trained\_model.predict(data)

print(result)

return result

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

app.run(debug=True)

***8.1.1.2HTML JS and CSS files:***

#### 8.1.1.2.1 Flightdelay.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

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

<script src="{{url\_for('static', filename='styles/delaypredict.js')}}"></script>

<title>Flight Delay Prediction</title>

</head>

<body id="flight-form">

<h2 id="main-head" class="centered-head">FLIGHT DELAY PREDICTION</h2>

<imgsr c="{{url\_for('static', filename='styles/images/Flight.jpg')}}" id="bgimg">

<form name="flightForm" action="/result" method="POST" target="\_blank">

<div id="form-content">

<div id="block1">

<div class="detail-container">

<label for="fno" class="label-item">Enter the Flight Number</label>

<br>

<input type="number" id="fno" name="fno" class="text-input">

</div>

<div class="detail-container">

<label for="month" class="label-item">Month</label>

<br>

<input type="number" id="month" name="month" class="text-input" onblur="checkValid('month');" placeholder="Enter the Month Number">

<div class="alert-text" id="month-valid">Enter a valid month between 1 to 12.</div>

</div>

<div class="detail-container">

<label for="daym" class="label-item">Day of Month</label>

<br>

<input type="number" id="daym" name="daym" class="text-input" onblur="checkValid('daym');">

<div class="alert-text" id="daym-valid">Enter a valid day of month.</div>

</div>

<div class="detail-container">

<label for="dayw" class="label-item">Day of Week</label>

<br>

<input type="number" id="dayw" name="dayw" class="text-input" onblur="checkValid('dayw');">

<div class="alert-text" id="dayw-valid">Enter a valid day between 1 to 7.</div>

</div>

<div class="detail-container">

<label for="org" class="label-item">Origin</label>

<br>

<select id="org" name="org" class="select-input">

<option value="ATL" class="option-item">ATL</option>

<option value="SEA" class="option-item">SEA</option>

<option value="DTW" class="option-item">DTW</option>

<option value="MSP" class="option-item">MSP</option>

<option value="JFK" class="option-item">JFK</option>

</select>

</div>

<div class="detail-container">

<label for="dest" class="label-item">Destination</label>

<br>

<select id="dest" name="dest" class="select-input" onblur="checkValid('dest');">

<option value="ATL" class="option-item">ATL</option>

<option value="SEA" class="option-item">SEA</option>

<option value="DTW" class="option-item">DTW</option>

<option value="MSP" class="option-item">MSP</option>

<option value="JFK" class="option-item">JFK</option>

</select>

<div class="alert-text" id="dest-valid">Enter different Origin and Destination.</div>

</div>

<div class="detail-container">

<label for="sdt" class="label-item">Scheduled Departure Time</label>

<br>

<input type="number" id="sdt" name="sdt" class="text-input" onblur="checkValid('sdt');" placeholder="Enter in the format HHMM">

<div class="alert-text" id="sdt-valid">Enter a valid time between 500 to 2359.</div>

</div>

<div class="detail-container">

<label for="sat" class="label-item">Scheduled Arrival Time</label>

<br>

<input type="number" id="sat" name="sat" class="text-input" onblur="checkValid('sat');" placeholder="Enter in the format HHMM">

<div class="alert-text" id="sat-valid">Enter a valid time between 500 to 2359.</div>

</div>

<div class="detail-container">

<label for="adt" class="label-item">Actual Departure Time</label>

<br>

<input type="number" id="adt" name="adt" class="text-input" onblur="checkValid('adt');" placeholder="Enter in the format HHMM">

<div class="alert-text" id="adt-valid">Enter a valid time between 500 to 2359.</div>

</div>

</div></div>

<div id="submit-button">

<input type="submit" value="Submit" id="submit" class="button" onclick="validateForm()">

</div>

</form>

</body>

</html>

### 8.1.1.2.2 styles.css

body {

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

margin: 0;

}

.content {

padding: 10px;

display: block;

}

.content-head {

text-align: center;

font-weight: bold;

font-size: 36px;

}

.button {

background-color: #1C55A2;

color: aliceblue;

padding: 10px;

border-radius: 10px;

border-color: #0E0E0F;

border-width: 1.5px;

}

.button a {

color: aliceblue;

text-decoration: none;

font-weight: bold;

}

#feedback-button {

margin-top: 10px;

}

#feedback-button-section {

text-align: center;

}

#bgimg {

position: fixed;

z-index: -1;

width: 100%;

height: 100%;

padding: 0;

margin: 0;

top: 0;

opacity: 0.9;

}

.centered-head {

text-align: center;

color: #3700ff;

font-weight: bold;

}

.label-item {

color: #f4f6f8;

font-weight: bold;

}

.detail-container {

padding-bottom: 10px;

padding-top: 10px;

}

.text-input {

margin-top: 5px;

border-color: #1C55A2;

border-width: 1.5px;

width: 75%;

height: 20px;

padding-left: 5px;

padding-right: 5px;

padding-top: 2px;

padding-bottom: 2px;

}

.select-input {

margin-top: 5px;

border-color: #0E0E0F;

border-width: 1.5px;

width: 40%;

height: 30px;

background-color: #1C55A2;

color: aliceblue;

font-weight: bold;

cursor: pointer;

}

#form-content {

justify-content: space-evenly;

flex-direction: row;

}

#block1 {

display: block;

width: 40%;

padding: 20px;

}

#review {

height: 100px;

padding-top: 5px;

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

}

#submit-button {

padding-left: 20px;

}

#submit {

background-color: #00c3ff;

color: rgb(1, 7, 14);

font-weight: bold;

}

#submit:hover {

cursor: pointer;

}

.choose-item {

font-weight: 600;

}

input[type="radio"], input[type="checkbox"] {

cursor: pointer;

}

.alert-text {

color: rgb(255, 79, 47);

font-size: small;

padding-left: 10px;

display: none;

}

#### 8.1.1.2.3 delaypredict.js

function validateForm() {

var fno = document.forms["flightForm"]["fno"].value;

var month = document.forms["flightForm"]["month"].value;

var daym = document.forms["flightForm"]["daym"].value;

var dayw = document.forms["flightForm"]["dayw"].value;

var org = document.forms["flightForm"]["org"].value;

var dest = document.forms["flightForm"]["dest"].value;

var sdt = document.forms["flightForm"]["sdt"].value;

var sat = document.forms["flightForm"]["sat"].value;

var adt = document.forms["flightForm"]["adt"].value;

if (fno == "" || fno == null || month == "" || month == null || daym == "" || daym == null || dayw == "" || dayw == null || org == "" || org == null || dest == "" || dest == null || sdt == "" || sdt == null || sat == "" || sat == null || adt == "" || adt == null) {

alert("The given fields must be filled out");

event.preventDefault();

}

if(month<1 || month>12)

{

alert("Enter a valid month");

event.preventDefault();

}

if(month==2)

{

if(daym<1 || daym>=29)

{

alert("Enter a valid day of month");

event.preventDefault();

}

}

else if(month==1 || month==3 || month==5 || month==7 || month==8 || month==10 || month==12)

{

if(daym<1 || daym>31)

{

alert("Enter a valid day of month");

event.preventDefault();

}

}

else if(month==4 || month==6 || month==9 || month==11)

{

if(daym<1 || daym>30)

{

alert("Enter a valid day of month");

event.preventDefault();

}

}

if(dayw<1 || dayw>7)

{

alert("Enter a valid day of week");

event.preventDefault();

}

if(org==dest)

{

alert("Enter different origin and destination");

event.preventDefault();

}

if(sdt<500 || sdt>2400)

{

alert("Enter a valid Departure time between 500 to 2400");

event.preventDefault();

}

if(sat<500 || sat>2400)

{

alert("Enter a valid Arrival time between 500 to 2400");

event.preventDefault();

}

if(sdt==sat)

{

alert("Departure and Arrival time must differ by atleast 1 hr");

event.preventDefault();

}

if(adt<500 || adt>2400)

{

alert("Enter a valid Departure time between 500 to 2400");

event.preventDefault();

}

}

function checkValid(element)

{

var obj = document.getElementById(element);

var valid\_obj = document.getElementById(element + "-valid");

if(element=='month')

{

if(obj.value<1 || obj.value>12)

{

obj.style.borderColor = "rgb(255, 79, 47)"

valid\_obj.style.display = "block";

}

else {

obj.style.borderColor = "#1C55A2";

valid\_obj.style.display = "none";

}

}

if(element=='daym')

{

var monobj = document.getElementById('month');

if(monobj.value==2)

{

if(obj.value<1 || obj.value>=29)

{

obj.style.borderColor = "rgb(255, 79, 47)"

valid\_obj.style.display = "block";

}

else {

obj.style.borderColor = "#1C55A2";

valid\_obj.style.display = "none";

}

}

else if(monobj.value==1 || monobj.value==3 || monobj.value==5 || monobj.value==7 || monobj.value==8 || monobj.value==10 || monobj.value==12)

{

if(obj.value<1 || obj.value>31)

{

obj.style.borderColor = "rgb(255, 79, 47)"

valid\_obj.style.display = "block";

}

else {

obj.style.borderColor = "#1C55A2";

valid\_obj.style.display = "none";

}

}

else if(monobj.value==4 || monobj.value==6 || monobj.value==9 || monobj.value==11)

{

if(obj.value<1 || obj.value>30)

{

obj.style.borderColor = "rgb(255, 79, 47)"

valid\_obj.style.display = "block";

}

else {

obj.style.borderColor = "#1C55A2";

valid\_obj.style.display = "none";

}

}

else

{

obj.style.borderColor = "rgb(255, 79, 47)"

valid\_obj.style.display = "block";

}

}

if(element=='dayw')

{

if(obj.value<1 || obj.value>7)

{

obj.style.borderColor = "rgb(255, 79, 47)"

valid\_obj.style.display = "block";

}

else {

obj.style.borderColor = "#1C55A2";

valid\_obj.style.display = "none";

}

}

if(element=='dest')

{

var origin\_obj = document.getElementById('org');

if(obj.value==origin\_obj.value)

{

obj.style.borderColor = "rgb(255, 79, 47)"

valid\_obj.style.display = "block";

}

else

{

obj.style.borderColor = "#1C55A2";

valid\_obj.style.display = "none";

}

}

if(element=='sdt')

{

if(obj.value<500 || obj.value>2400)

{

obj.style.borderColor = "rgb(255, 79, 47)"

valid\_obj.style.display = "block";

}

else {

obj.style.borderColor = "#1C55A2";

valid\_obj.style.display = "none";

}

}

if(element=='sat')

{

if(obj.value<500 || obj.value>2400)

{

obj.style.borderColor = "rgb(255, 79, 47)"

valid\_obj.style.display = "block";

}

else {

obj.style.borderColor = "#1C55A2";

valid\_obj.style.display = "none";

}

}

if(element=='adt')

{

if(obj.value<500 || obj.value>2400)

{

obj.style.borderColor = "rgb(255, 79, 47)"

valid\_obj.style.display = "block";

}

else {

obj.style.borderColor = "#1C55A2";

valid\_obj.style.display = "none";

}

}

}

#### 8.1.1.2.4 result.html

<!doctype html>

<html>

<head>

<title>Flight Delay Prediction-Result</title>

<link rel="stylesheet" href="{{ url\_for('static',filename='styles/result\_styles.css') }}">

</head>

<body>

<imgsrc="{{url\_for('static', filename='styles/images/Flight.jpg')}}" id="bgimg">

{% if prediction == 0.0 %}

<div class="pred\_result" id="result\_0">Your flight will likely be on time</div>

{% endif %}

{% if prediction == 1.0 %}

<div class="pred\_result" id="result\_1">Your flight is likely to be delayed</div>

{% endif %}

</body>

</html>

#### 8.1.1.2.5. Template

