Developing a Flight Delay Prediction Model using Machine Learning

Professional Readiness for Innovation, Employability and Entrepreneurship

Team ID: PNT2022TMID54451

Surya Prasath R-310619106153
Shivarajakrishnan P-310619106134
Suriya B -310619106150
Ravikanth V-310619106112
Manikandan C-310619106501

EASWARI ENGINEERING COLLEGE,RAMAPURAM,CHENNAI
(AUTONOMOUS)
B.E-ELECTRONICS AND COMMUNICATION ENGINEERING

INTRODUCTION

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.

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.

LITERATURE SURVEY

Existing problem

S.No	Author	Title of the paper	Year of publicati on	Algorithm/ Method	Results

1	Maryam Farshchian Yazdi1 , Seyed Reza Kamel2 , Seyyed Javad Mahdavi Chabok2 and Maryam Kheirabadi1	Flight delay prediction based on deep learning and Levenberg-Marquart algorithm	26 November 2020	Novel optimized forecasting model based on deep learning which engages LM algorithm.	Greater accuracy in forecasting fight delay compared to previous model called RNN
2	Kaiquan Cai, Yue Li, YiPing Fang, Yanbo Zhu,	A Deep Learning Approach for Flight Delay Prediction Through TimeEvolving Graphs	12 August 2021	Graph Convolutional Neural Network (GCN)	Through extensive experiments, it has been shown that the proposed approach

					outperforms benchmark methods with a satisfying accuracy improvement at the cost of acceptable execution time. The obtained results reveal that deep learning approaches based on graph structured inputs have great potential in the flight delay prediction problem.

4	Fan Liu, Jinlong Sun, Miao Liu, Jie Yang, Guan Gui	Generaliz ed Flight Delay Prediction Method Using Gradient Boosting Decision Tree	30 June 2020	Gradient boosting decision tree (GBDT) based model	Experimental results show that the proposed GBDTbased model can obtain higher prediction accuracy (87.72% for the binary classification) when handling a limited dataset.

2.2 References

- ✓ https://ieeexplore.ieee.org/document/8903554
- ✓ https://ieeexplore.ieee.org/document/9512525
- ✓ https://www.sciencedirect.com/science/article/abs/pii/S1270963821003321
- ✓ https://ieeexplore.ieee.org/document/9129110

2.3 Problem Statement Definition

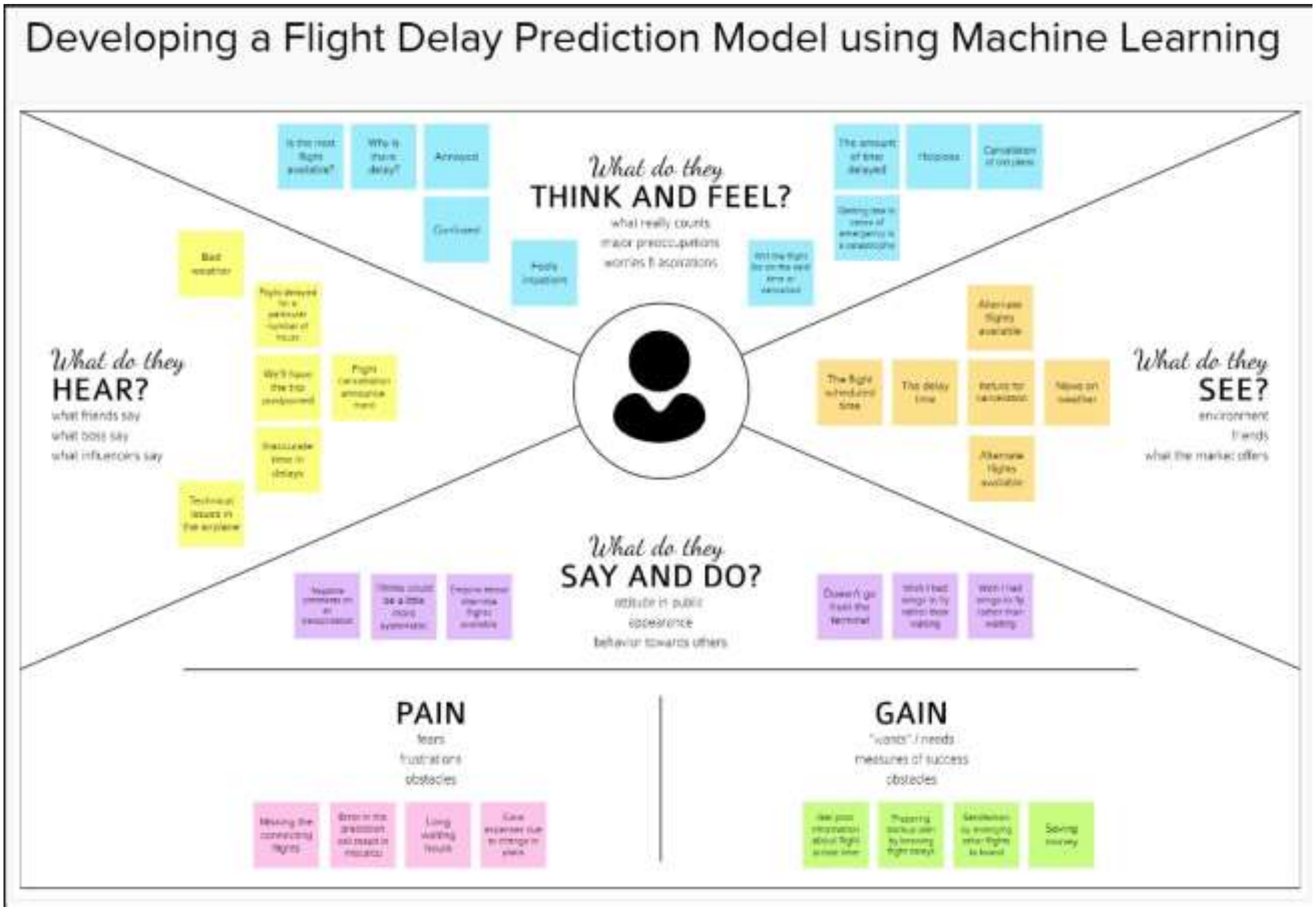
Problem Statement: Passenger flights getting delayed

Defining the Problem:

Question	Description
Who does the problem affect?	Passengers who are in emergency situation to reach destination
What is the issue?	An airline flight takes off and lands later than its scheduled time. A cancellation occurs when the airline does not operate the flight at all for a certain reason. However, depending on why the flight was canceled, finding seats on a new flight will also be difficult and may change travel plans
When does the issue occur?	The issue occurs when there are adverse weather conditions, if the flight is waiting for cargo or crew or if the airport staff strike.
Where is the issue occurring?	Airport
Why is it important that we fix the problem?	Flight delays cause inconvenience for both airline companies and passengers. They cause a decrease in efficiency, an increase in capital costs, reallocation of flight crew and aircraft, and additional crew expenses and require the consumption of extra labor, capital, and other inputs necessary in the process. Another impact of flight delay can be a risk which represents dissatisfaction of passengers and their loss in time.

IDEATION & PROPOSED SYSTEM

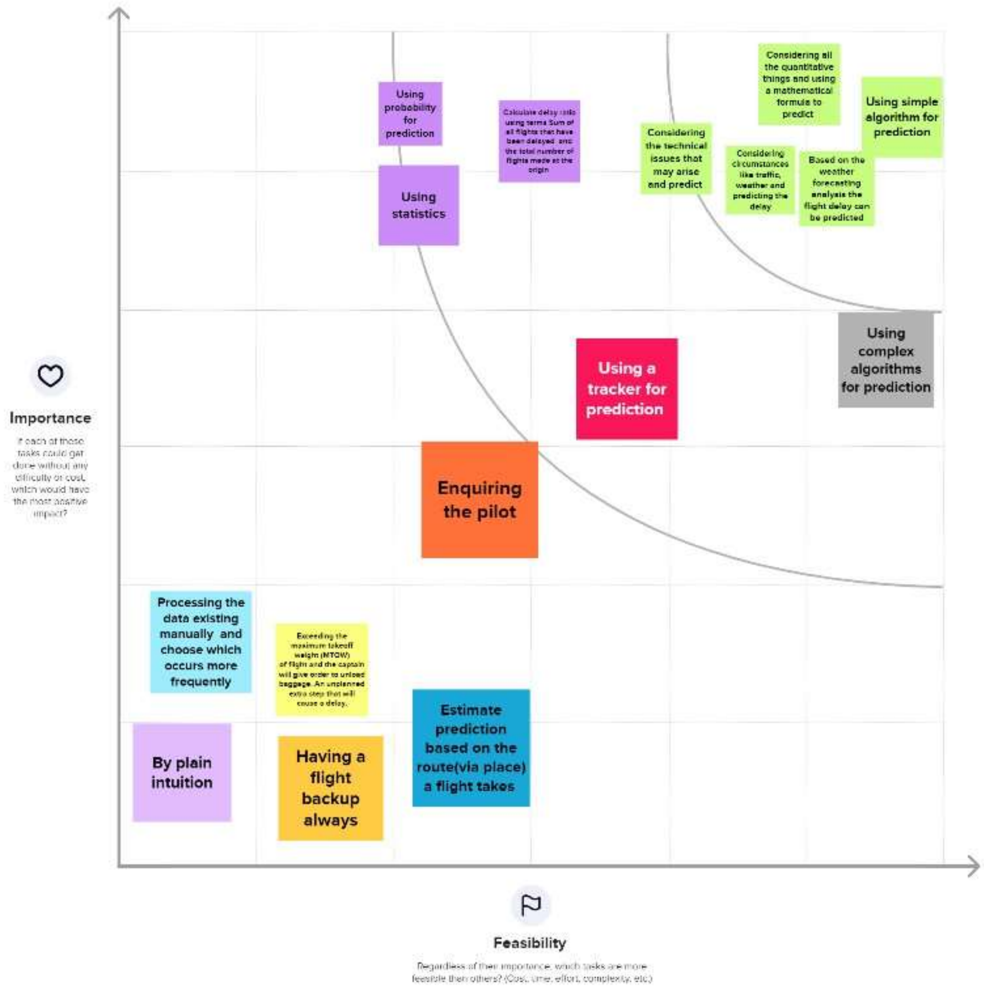
Empathy Map Canvas


Developing a Flight Delay Prediction Model using Machine Learning
What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Is the next flight available?
Why is there delay?
Annoyed
Confused
Feels impatient
Will the flight be on the said time or cancelled
The amount of time delayed
Hopeless
Cancellation of our plans
Getting late in cases of emergency is a catastrophe
What do they HEAR?
what friends say
what boss say
what influencers say
Bad weather
Flight delayed for a particular number of hours
We'll have the trip postponed
Flight cancellation announcement
Inaccurate time in delays
Technical issues in the airplane
What do they SEE?
environment
friends
what the market offers
Alternate flights available
The flight scheduled time
The delay time
Refund for cancellation
News on weather
Alternate flights available
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
Things could be a little more systematic
Enquire about alternate flights available
Doesn't go from the terminal
Wish I had wings to fly rather than waiting
Wish I had wings to fly rather than waiting
PAIN
fears
frustrations
obstacles
Missing the connecting flights
Error in the prediction will result in
Long waiting hours
Cost expenses due to change in plans
GAIN
"wants" / needs
measures of success
obstacles
Satisfaction by arranging other flights to board
Saving money

Empathy Map Canvas

Ideation & Brainstorming

Brainstorming


Using probability for prediction
Calculate delay ratio using terms Sum of all flights that have been delayed and the total number of flights made at the origin
Considering all the quantitative things and using a mathematical formula to predict
Using simple algorithm for prediction
Considering the technical issues that may arise and predict
Considering circumstances like traffic, weather and predicting the delay
Based on the weather forecasting analysis the flight delay can be predicted
Using statistics
Using complex algorithms for prediction
Using a tracker for prediction
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Enquiring the pilot
Processing the data existing manually and choose which occurs more frequently
Exceeding the maximum takeoff weight (MTOW) of flight and the captain will give order to unload baggage. An unplanned extra step that will cause a delay.
Estimate prediction based on the route(via place) a flight takes
By plain intuition
Having a flight backup always
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

Prioritise

Proposed Solution

PROBLEM STATEMENT

For several reasons, flying is substantially more exciting for an ordinary traveller than utilizing other modes of transportation. The first is that it is the fastest way of transportation, followed by the comfort it gives in that it is a somewhat quieter and well air-conditioned mode of transit. In addition, it is a more organized style of traveling since staff members support and assist visitors in every way possible. Land traffic and aviation travel have dramatically increased as a result. A rise in air traffic has also resulted in extremely high levels of aircraft delays both on the ground and in the air.

SOLUTION DESCRIPTION

The main objective of the model is to accurately estimate airplane delays. The entire system is a web application. We employ HTML, CSS, and JavaScript to make the application user interactive. The flight delay is predicted using a ML model. This model accepts inputs as departure date, departure delay, travel time between the two airports, and other data. By turning the ML model to a pickle file, it is linked to a web application. The ML model here uses random forest as its prediction method. Because of its rapid prediction, impressiveness, excellence with high dimensionality, etc., random forest is selected.

CUSTOMER SATISFACTION

An accurate estimation of flight delay can help to increase customer satisfaction and incomes of airline agencies. The delay information can reduce anxiety and tension among passengers thus, saving their time and improving their travel experience, which is significant to enhance passengers' loyalty to an airline.

FINANCIAL BENEFIT MODEL

Predicting flight delays prevents reallocation of flight crew and aircraft and additional crew expenses. This avoids consumption of extra labour, capital and other inputs necessary in the process thereby saving the capital costs. The proposed solution is also a low-cost model and the customers are not charged for the service they receive.

SCALABILITY

The model's ability to increase its performance is possible by building it more accurately with the use of several classification models and selecting the best accurate model among them. Also, through integration of these models, an optimized hybrid model can be obtained in order to result in more scalability. Deploying the ML model into cloud also makes it easy for enterprises to experiment with the model capabilities and scale up. Placing a finished flight prediction model into a live environment can be used for its intended purpose and it is integrated with Flask, so that they can be accessed by end users.

PROBLEM STATEMENTS

Problem Statements

Adam is a user, who needs to **take an layover flight**, and **wants to know the approximate hours** of delay, **so that he can take an alternate flight**

Jimmy is a normal user who **wants to get entertained in some way** so that **he doesn't feel bored.**

Nikita is a normal user who **wants to know how much time is it going to be delayed** so that she can **visit nearby places if possible.**

Rita has the flight scheduled at night and wants to **know if accomodation be provided** so she **doesnt miss the day's sleep.**

Problem Statements

Alice is a **business professional** having an important client meeting, needs to **know the ETD**, so she can **make alternate accomodations**

Robert is in middle of an important event and wants to **know if his flight would be delayed,** so that **he can stay till the event completes.**

Joy is the incharge for inventory at an airport and **wants to know if a flight gets delayed**, so that he could **allot the inventory/resources planned for this fight to some other flight** when there are just enough resources.

Joy also would **wish to know if a flight is delayed** and by how much so that he could **arrange for its required resources well in advance**.

Problem Statements

Sharma is the head of the public relations department and needs to **know if a flight gets delayed**, so that he could make arrangements for **notifying the passengers** regarding the delay through the airline's application.

Stefan is the manager of the catering team of the airlines and needs to **know if a flight gets delayed** so that he could make **arrangements for serving food and refreshments** to the passengers.

REQUIREMENT ANALYSIS

Functional requirement

FR No.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	User Registration	Registration through form Registration through email
FR-2	User Confirmation	OTP verification for confirmation
FR-3	Dashboard and Search	Search for flights by entering the flight details Get to know the details of the flight and other flights
FR-4	View flight details	View the details of the flight View if there were any delays previously
FR-5	Display prediction results	Based on the input given by the user predict the delay Display the prediction results to the user

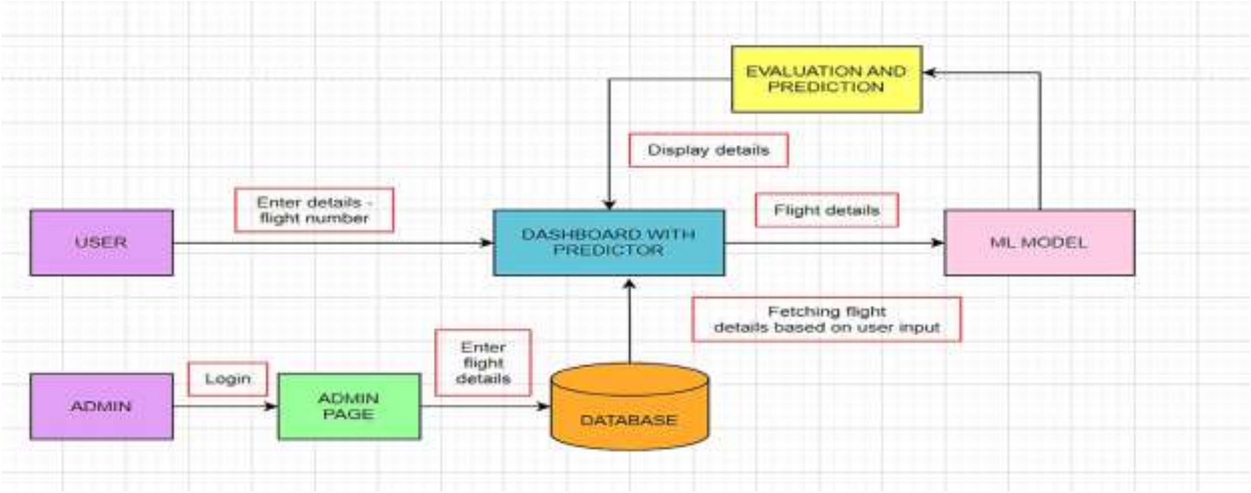
Non-Functional requirements

NFR No.	**Non-Functional Requirement**	**Description**
NFR-1	Usability	Compatible with all browsers
NFR-2	Security	The application will be designed in such a way that the user data won’t be let out and is secure

NFR-3	Reliability	The application will be fast enough and be run on cloud servers to ensure there is no DoS
NFR-4	Performance	High accuracy predictions
NFR-5	Availability	Available 24/7 as it is deployed in cloud
NFR-6	Scalability	Scalable to a very large extent. The only limit is the amount invested to buy cloud services.

PROJECT DESIGN

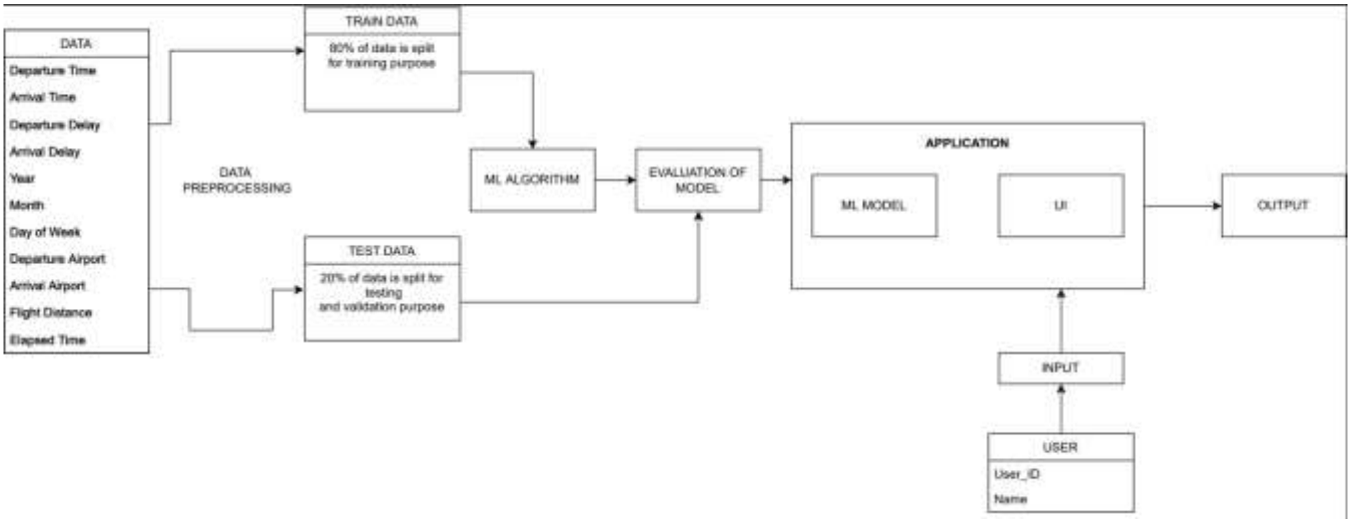
Data Flow Diagrams



Data Flow Diagrams

Solution & Technical Architecture

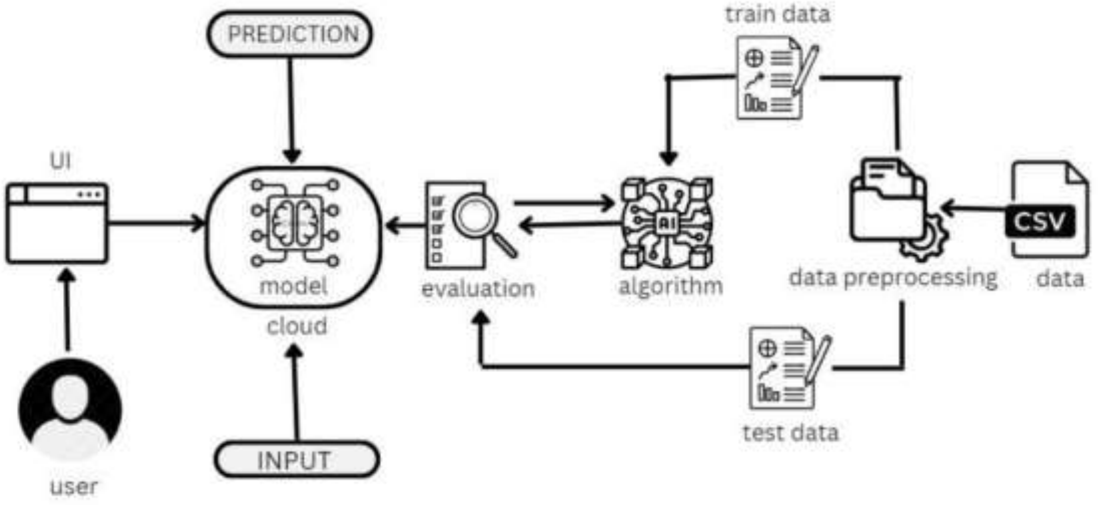
Solution Architecture



Solution Architecture

Technical Architecture

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2.



Technical Architecture

Table-1: Components & Technologies:

S.No.	**Component**	**Description**	**Technology**
1.	User Interface	How the user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Using Python's regularization approach with Regression Analysis to create predictions about future delays	Python
3.	Application Logic-2	Build, run and manage AI models	IBM Watson Machine Learning service
4.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.

5.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
6.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
7.	External API-1	Defines communication	
between customer and the administration	Flask (Python), etc.		
8.	Machine Learning Model	To predict flight delay model	Object Recognition Model, etc.
9	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: local host server on which flask runs Cloud Server Configuration: Cloud object storage	Local, Cloud Foundry, Kubernetes, etc.
Table-2: Application Characteristics:			
S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the opensource frameworks used	Flask(python)
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.

3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Flask or ML
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Flask or ML
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of	
Cache, use of CDN's) etc.	Flask or ML		

PROJECT PLANNING AND SCHEDULING

Sprint Planning & Estimation

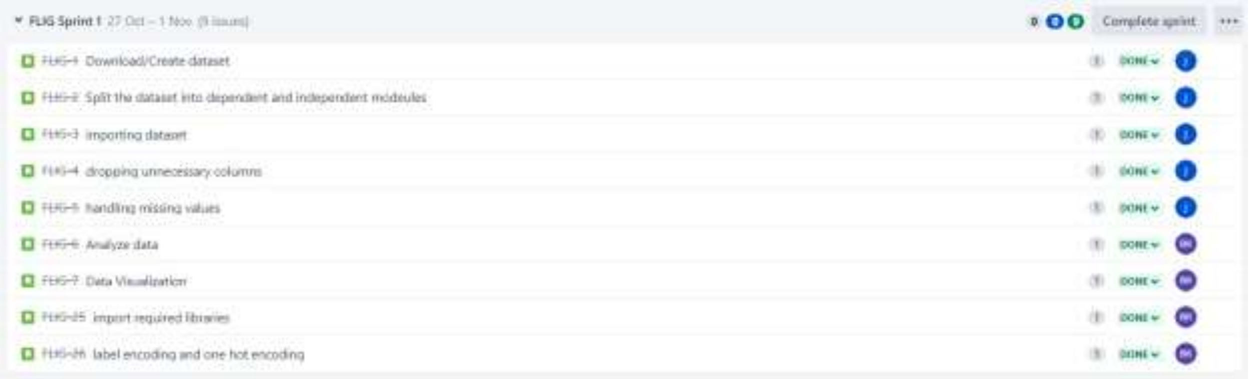
Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**

Sprint-1	Registration	USN-1	As a user, I can register for the applicati on by entering my email, password, and confirmi ng my password.	2	High	Surya Prasath R
Sprint-2	Confirmation email	USN-2	As a user, I will receive confirmation email once I have registered for the application.	2	Medium	Suriya B
Sprint-1	User Login	USN-3	As a user, I can log into the applicati on by entering email & password	2	High	Ravikanth V

Sprint-2	Admin panel	USN-4	As an admin, I can authenticate the registration and login credentia ls of the passengers.	2	High	Shivarajakrishnan P
Sprint-3	Arrival and Departure time of flights	USN-5	As a user, I can find all the details of a specific flight with its number or name.	2	High	Surya Prasath R
Sprint-3		USN-6	As a user, I can find exactly how long the flight will be delayed.	2	High	Suriya B
Sprint-4	Helpdesk	USN-7	As a customer care executive, I can provide airline details for contact.	1	Medium	Shivarajakrishnan p

Sprint-4		USN-8	As a passenger, I am able to find available alternative flights to the destination.	1	High	Suriya B Shivarajakrishnan P Ravikanth V Surya aprasath R Manikandan C
Sprint-4	Feedback	USN-9	As a user, I can provide my suggestions and feedback for the improvement of the application.	2	Medium	Manikandan C

Reports from JIRA



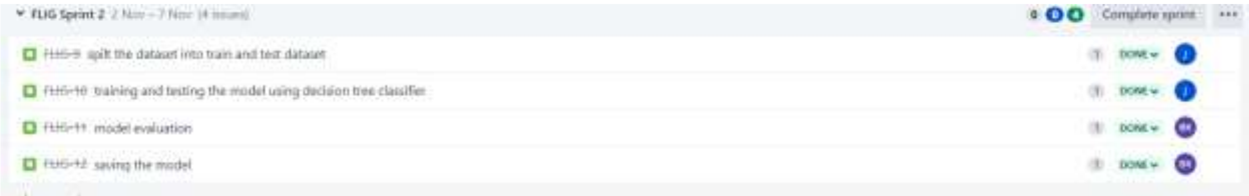
FLIG Sprint 1 27 Oct – 1 Nov (9 issues)

Complete sprint

- ~~FLIG-1~~ Download/Create dataset DONE
- ~~FLIG-2~~ Split the dataset into dependent and independent modeules DONE
- ~~FLIG-3~~ importing dataset DONE
- ~~FLIG-4~~ dropping unnecessary columns DONE
- ~~FLIG-5~~ handling missing values DONE
- ~~FLIG-6~~ Analyze data DONE
- ~~FLIG-7~~ Data Visualization DONE
- ~~FLIG-25~~ import required libraries DONE
- ~~FLIG-26~~ label encoding and one hot encoding DONE

Tasks to be performed in Sprint 1

The above figure displays the various tasks to be performed in sprint 1. The goal of the sprint 1 is to prepare data for model training. There are nine issues that need to be addressed in the sprint 1. The story points of each issue is mentioned in the above diagram.



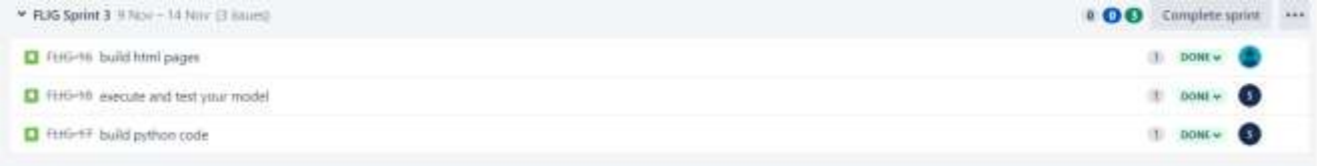
FLIG Sprint 2 2 Nov – 7 Nov (4 issues)

Complete sprint

- ~~FLIG-9~~ spilt the dataset into train and test dataset DONE
- ~~FLIG-10~~ training and testing the model using decision tree classifier DONE
- ~~FLIG-11~~ model evaluation DONE
- ~~FLIG-12~~ saving the model DONE

Tasks to be performed in Sprint 2

The above figure displays the various tasks to be performed in sprint 2. The goal of the sprint 2 is to train and save the model. There are four issues that need to be addressed in the sprint 2. The story points of each issue is mentioned in the above diagram.



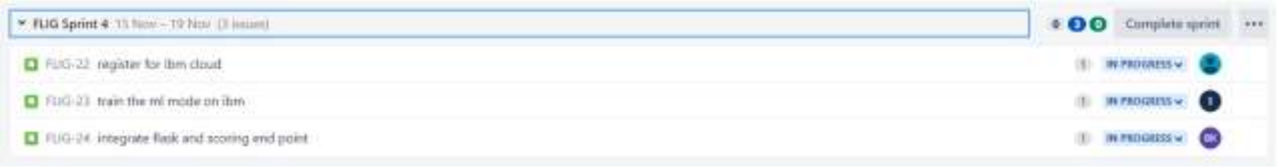
FLIG Sprint 3 9 Nov – 14 Nov (3 issues)

Complete sprint

- ~~FLIG-16~~ build html pages DONE
- ~~FLIG-18~~ execute and test your model DONE
- ~~FLIG-17~~ build python code DONE

Tasks to be performed in Sprint 3

The above figure displays the various tasks to be performed in sprint 3. The goal of the sprint 3 is to build the application and execute the model. There are three issues that need to be addressed in the sprint 3. The story points of each issue are mentioned in the above diagram.



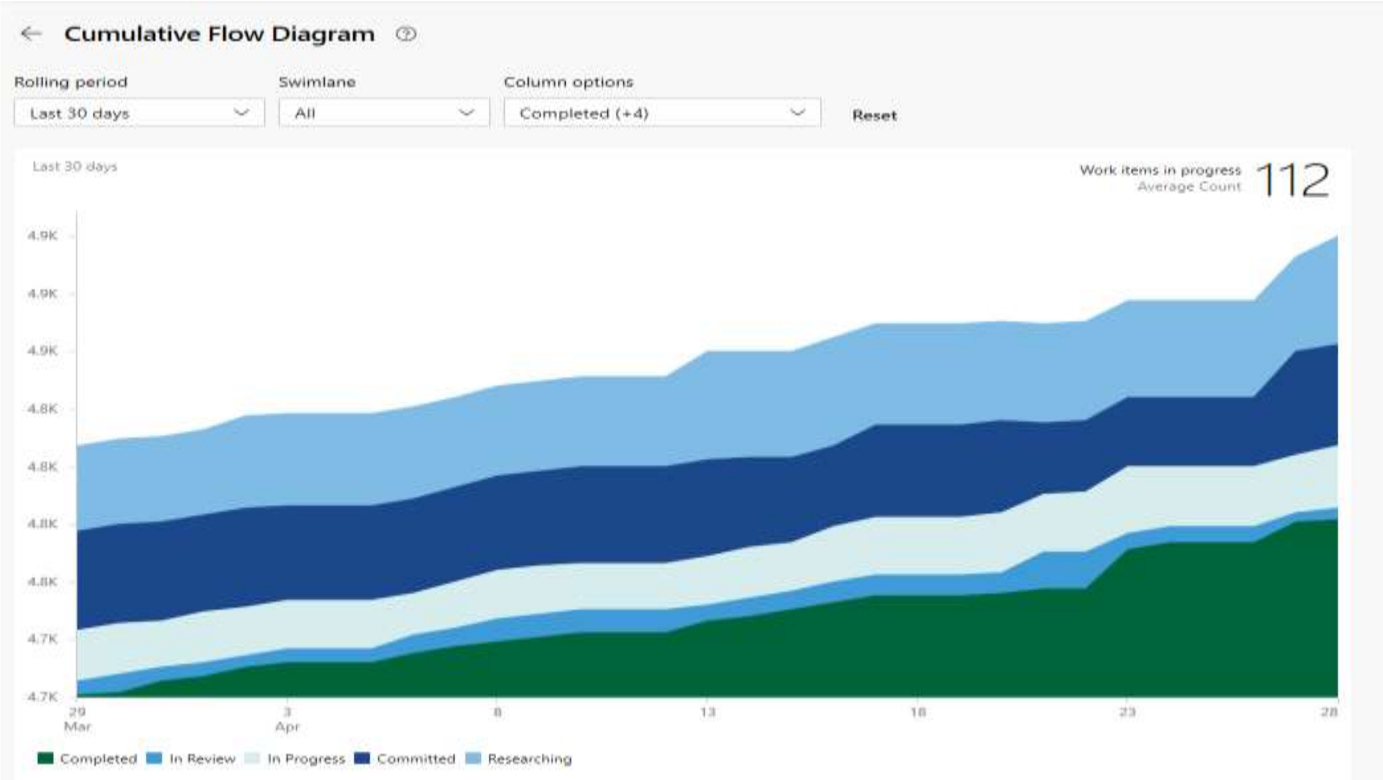
FLIG Sprint 4 15 Nov – 19 Nov (3 issues) 0 3 0 Complete sprint ...

FLIG-22	register for ibm cloud	1	IN PROGRESS
FLIG-23	train the ml mode on ibm	1	IN PROGRESS
FLIG-24	integrate flask and scoring end point	1	IN PROGRESS

Tasks to be performed in Sprint 4

The above figure displays the various tasks to be performed in sprint 4. The goal of the sprint 4 is to integrate the model with the application. There are three issues that need to be addressed in the sprint 3. The story points of each issue are mentioned in the above diagram.

Cumulative flow



Cumulative Flow Diagram

Rolling period: Last 30 days

Swimlane: All

Column options: Completed (+4)

Reset

Work items in progress Average Count 112

Cumulative flow diagram

CODING & SOLUTIONING

We completed four sprints—Sprint 1, Sprint 2, Sprint 3 and Sprint 4—during the project development phase. A sprint is a predetermined amount of time in Agile product development during which particular tasks must be finished and prepared for review.

Sprint 1

The dataset has been downloaded. The features are analysed and visualized and data has been cleaned and pre-processed using techniques like encoding. The independent and dependent variables are then identified and the dataset is split into train and test sets. Several machine learning algorithms have been applied for classification like logistic regression, decision tree classifier, KNN classifier, random forest classifier and it is found that logistic regression gives the highest accuracy, so it is used for deployment. The model is then dumped into a pickle file.

Sprint 2

We had done building HTML files, writing Python code, and running the application during Sprint 2. The source code is attached in the appendix for reference.

Sprint 3

We then asked users to enter numerical and selection data and tested for many inputs and checked the correctness of the result during sprint 3.

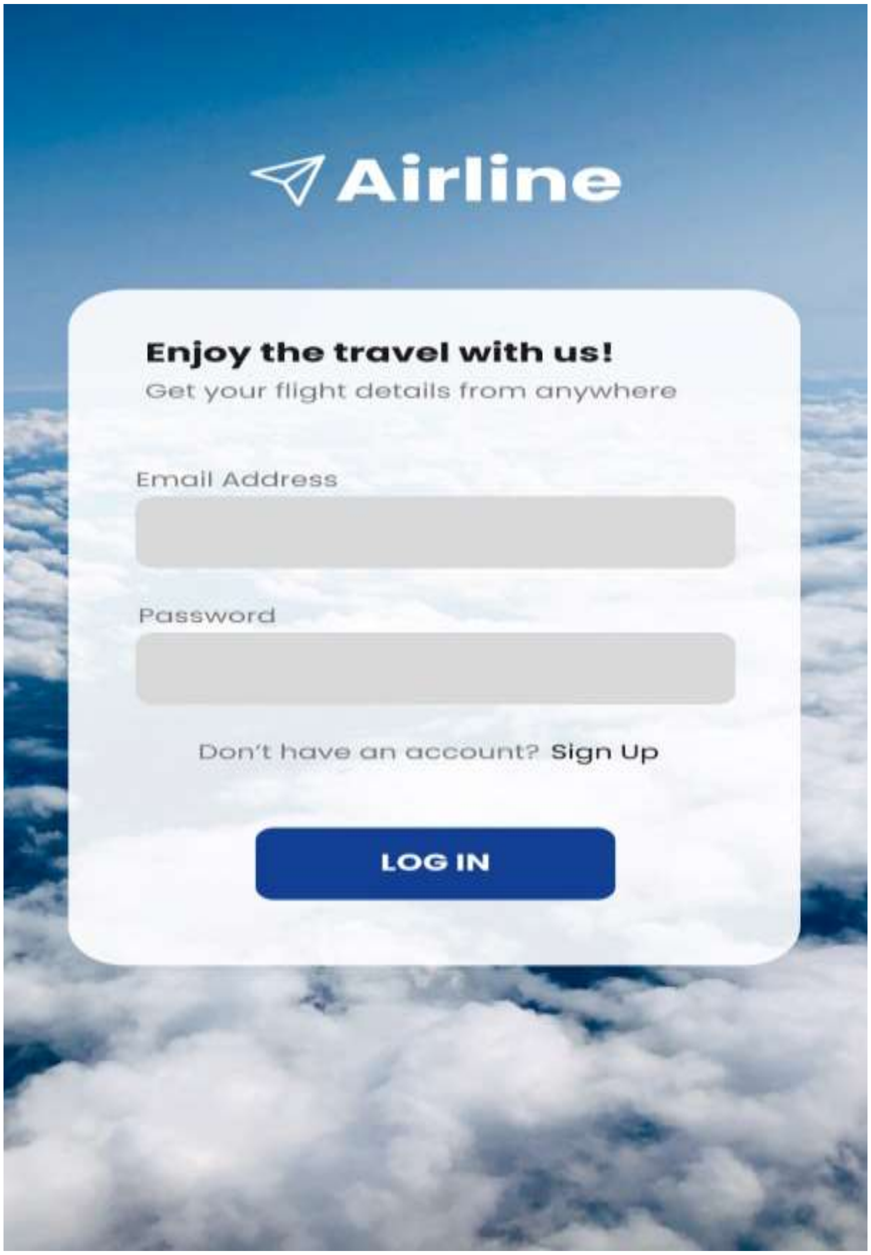
Sprint 4

We trained the model on IBM where we will register for IBM cloud, train the ML model on IBM and integrate flask with scoring end point. Registered on IBM cloud and activated Watson machine learning, cloud storage and Watson studio then trained the ML model on IBM using API KEY during sprint 4.

TESTING

Test Cases

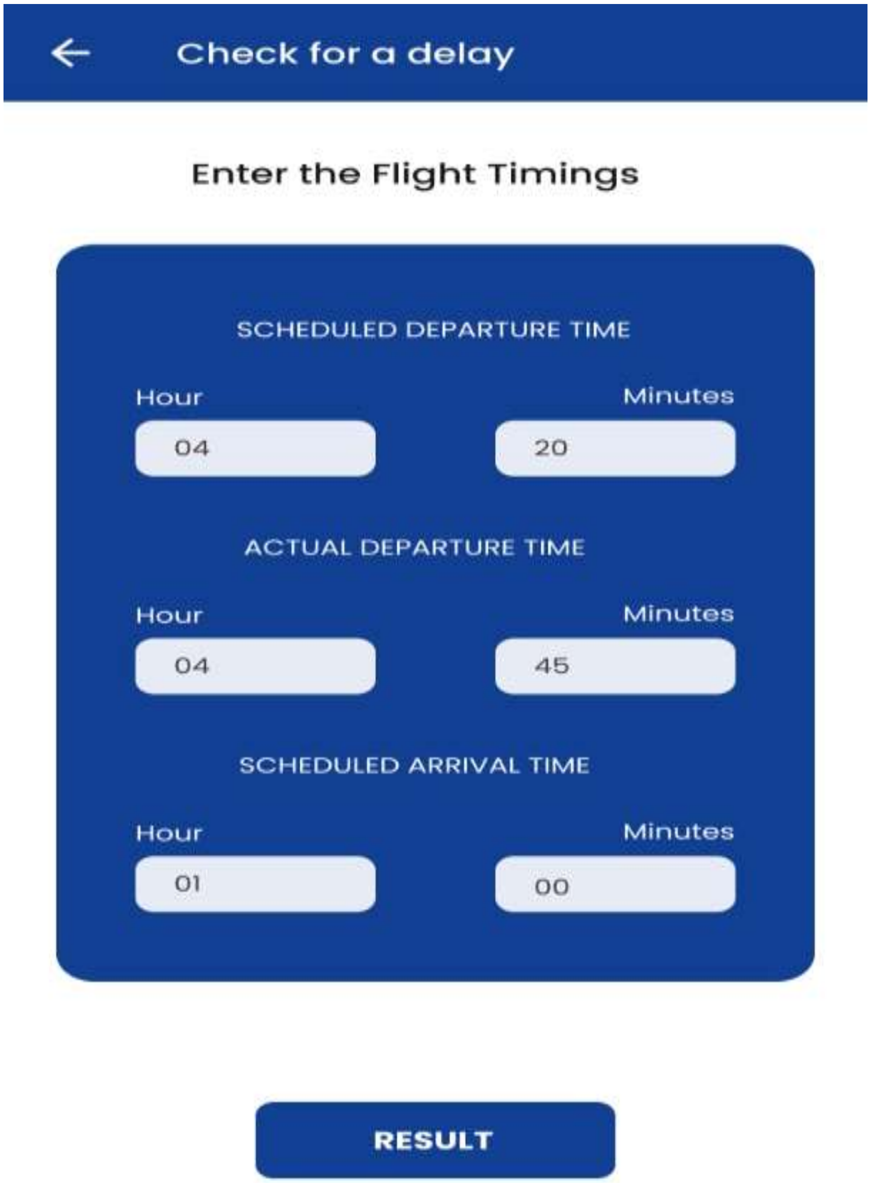
Test case 1: The designed application predicts if the flight is delayed.



Enjoy the travel with us!

Get your flight details from anywhere

Test case 2: The designed application predicts if the flight is not delayed.



Check for a delay

Enter the Flight Timings

SCHEDULED DEPARTURE TIME

Hour: 04
Minutes: 20

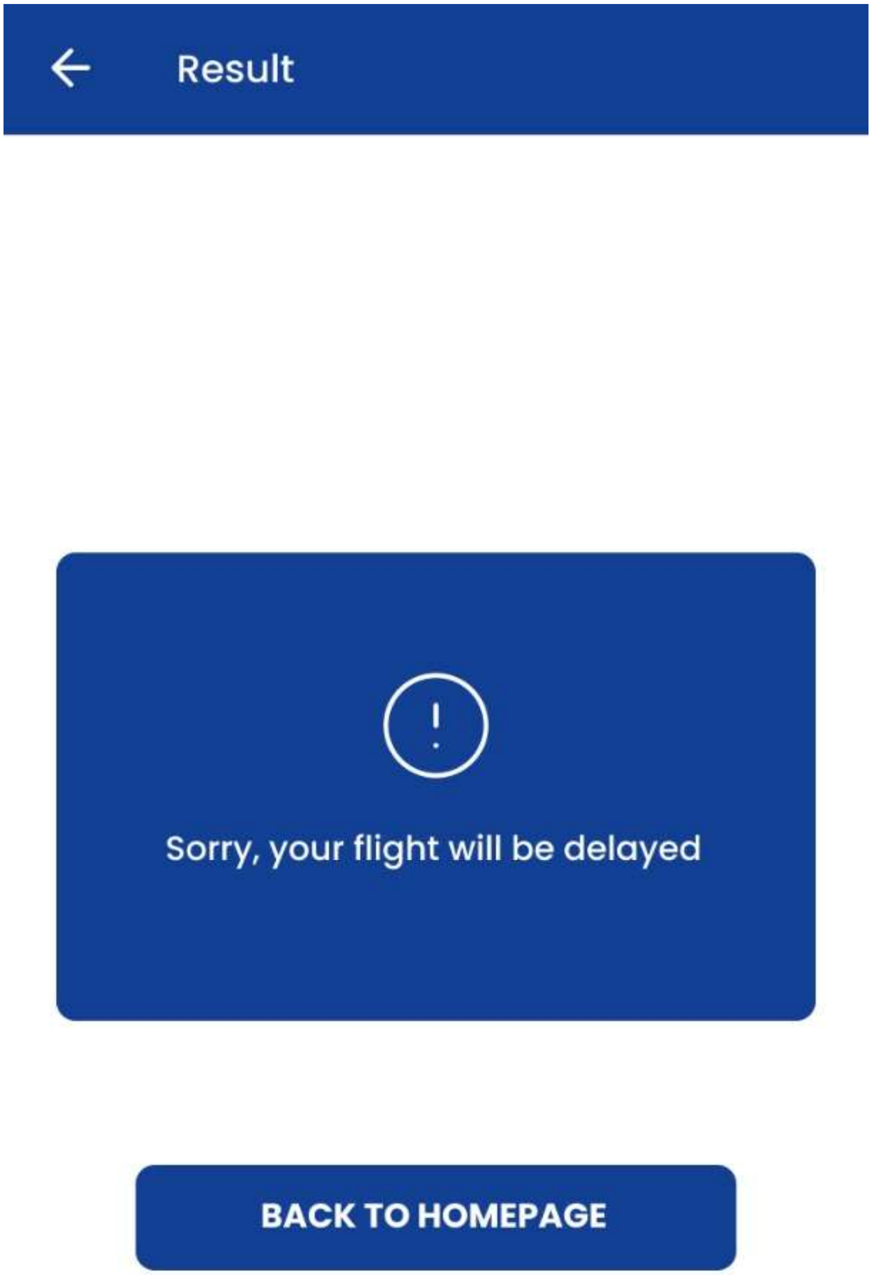
ACTUAL DEPARTURE TIME

Hour: 04
Minutes: 45

SCHEDULED ARRIVAL TIME

Hour: 01
Minutes: 00

Deployment testing phase: DONE



Sorry, your flight will be delayed

Can the user be able to make use of cloud-deployed ML model and run it in local machine using the API key? YES

The below table shows the number of test cases that have passed, failed and untested.

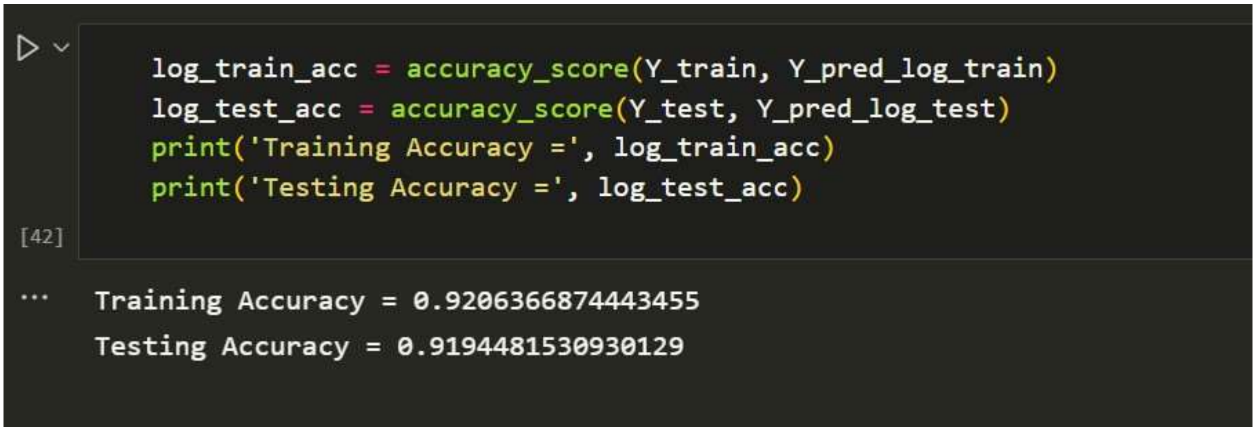
Section	**Total cases**	**Not Tested**	**Fail**	**Pass**
Home Screen	1	0	0	1
User input	4	0	0	4
Flight delay testing	2	0	0	2
No flight delay testing	2	0	0	2
Version control	1	0	0	1

RESULTS

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.

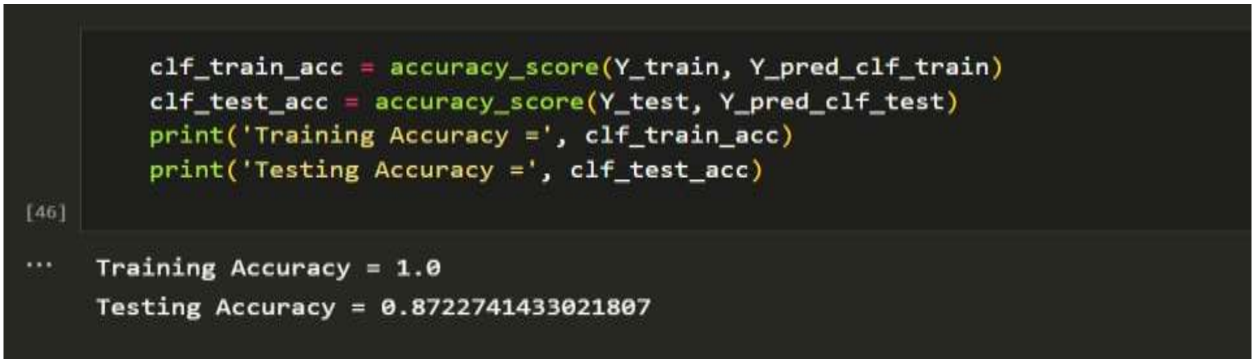


```
log_train_acc = accuracy_score(Y_train, Y_pred_log_train)
log_test_acc = accuracy_score(Y_test, Y_pred_log_test)
print('Training Accuracy =', log_train_acc)
print('Testing Accuracy =', log_test_acc)
```

```
Training Accuracy = 0.9206366874443455
Testing Accuracy = 0.9194481530930129
```

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.

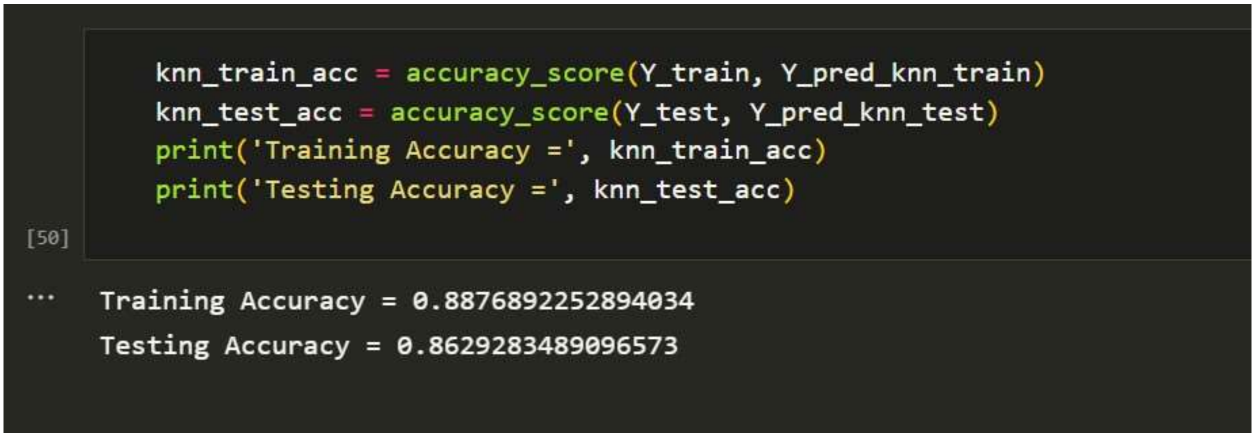


```
clf_train_acc = accuracy_score(Y_train, Y_pred_clf_train)
clf_test_acc = accuracy_score(Y_test, Y_pred_clf_test)
print('Training Accuracy =', clf_train_acc)
print('Testing Accuracy =', clf_test_acc)
```

```
Training Accuracy = 1.0
Testing Accuracy = 0.8722741433021807
```

Model: KNN Classifier

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

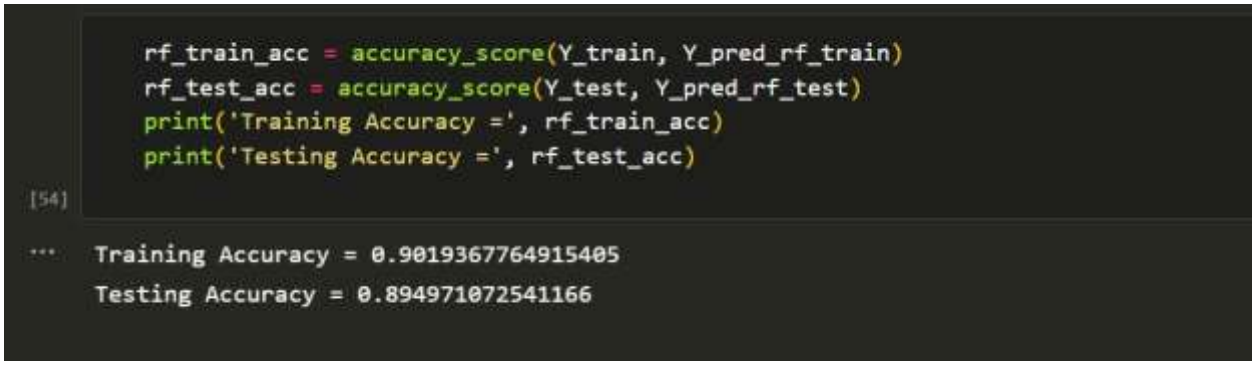


```
knn_train_acc = accuracy_score(Y_train, Y_pred_knn_train)
knn_test_acc = accuracy_score(Y_test, Y_pred_knn_test)
print('Training Accuracy =', knn_train_acc)
print('Testing Accuracy =', knn_test_acc)
```

```
Training Accuracy = 0.8876892252894034
Testing Accuracy = 0.8629283489096573
```

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.



```
[54]
rf_train_acc = accuracy_score(Y_train, Y_pred_rf_train)
rf_test_acc = accuracy_score(Y_test, Y_pred_rf_test)
print('Training Accuracy =', rf_train_acc)
print('Testing Accuracy =', rf_test_acc)
```

```
Training Accuracy = 0.9019367764915405
Testing Accuracy = 0.894971072541166
```

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.

ADVANTAGES & DISADVANTAGES

Advantages

- ✓ The application is fast and offers great accuracy in predicting the flight delay.
- ✓ Less maintenance is required.
- ✓ It is user friendly.
- ✓ It helps in reducing the tension of the passengers in knowing how long they will have to wait and lets passengers plan their schedule accordingly, thus in a way saving their time.

Disadvantages

- ✓ It requires an internet connection for the website to work.

CONCLUSION

Flight delays not only anger and disturb air travellers' 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.

FUTURE SCOPE

In the future, the application can be included with an user authentication model. Apart from checking if the flight would get delayed or not, their search history can be maintained and personalized flight recommendations can be done. A section where the users can give their feedback can also be implemented.

APPENDIX

Source code:

Flask file:

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_file = open('C:\\Users\\VARSHINI SANKAR\\Documents\\Sem 7\\Capstone
Design Project\\Local Deployment\\model.pkl', 'rb')
    trained_model = joblib.load(model_file)
    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)
```

HTML JS and CSS files:

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', file-name='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>
<img src="{{url_for('static', filename='styles/images/Flight.png')}}" 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 Num-ber</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>
        <div id="block2">
            <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>
```

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;
   opacity: 0.5;
   width: 100%;
   height: 100%;
   padding: 0;
   margin: 0;
   top: 0;
}
.centered-head {
   text-align: center;
   color: #1C55A2;
   font-weight: bold;
} .label-item
{
   color: #2E547F;
   font-weight: bold;
} .detail-container {
   padding-bottom: 10px;
   padding-top: 10px;
} .text-input
{
   margin-top: 5px;
   border-color: #1C55A2;     border-width: 1.5px;     border-radius: 10px;
   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;     border-
radius: 10px;
   width: 40%;
   height: 30px;
   background-color: #1C55A2;
   color: aliceblue;
   font-weight: bold;
```

```
    cursor: pointer;
}
#form-content {
    display: flex;
    justify-content: space-evenly;
    flex-direction: row;
    padding-left: 10%;
}
#block1 {
    display: block;
    width: 50%;
    padding: 20px;
}
#block2 {
    display: block;
    width: 50%;
    padding: 20px;
}
#review {
    height: 100px;
    padding-top: 5px;
    font-family: Arial, Helvetica, sans-serif;
}
#submit-button {
    text-align: center;
    align-items: center;
    display: block;
}
#submit {
    background-color: #1C55A2;
    color: aliceblue;
    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;
}
```

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";
        }
    }
}
```

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>
    <img src="{{url_for('static', filename='styles/images/Flight.png')}}" 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 de-layed</div>     {%
endif %}
</body>
</html>
```

GitHub link & Project Demo Link

GitHub link: https://github.com/IBM-EPBL/IBM-Project-42179-1660654606

Project Demo Link: https://www.mediafire.com/file/l1u53jpcoj2ib3h/2022-11-19_13-18-27-457.mp4/file