# FINAL PROJECT REPORT

| | |
|---|---|
| Date | 19 November 2022 |
| Team ID | PNT2022TMID32830 |
| Project Name | Flight Delay Prediction Using Machine Learning |

## Preview/Index:

# I. Introduction

## 1.Project Overview

Airports are significant nodes of air transportation. The number of airport flight delays has been on increase in recent years. Delayed flights are defined by the Federal Aviation Administration when they arrive or depart more than 15 minutes later than scheduled. In 2019, the arrival delay rate is 19.2% and the departure delay rate is 18.18% in the United States. Flight delays can causemany negative effects, such as passengers' inconvenience, increased airport pressure, and airline losses. Effective flight delay prediction could provide support for flight plan and emergency plan formulation, reduce the economic loss, and alleviate the negative impact (1). Hence, a delay prediction turns out very important. International Civil Aviation Organization (ICAO) has enableda program called Air Traffic Flow Management (ATFM) with the objective of ensuring that the traffic volume is compatible with the capacities declared by aviation authorities in order to reduceground and en-route delays. Another application of traffic management is the Free Route Airspace(FRA) concept which consists of using the shortest routes possible in order to reduce flight time, CO2 emissions, and fuel waste. Moreover, several other models have been developed to solve delays problem based on probability, statistics, graph and network representations, operational research studies, and so on (2). The Related Work gives the recent works and research on this topic.

## 2.Purpose

The main purpose of this project is to early warning of flight delay to the customers, Airport Authorities and Airlines

# II. Literature Survey

## 1.Existing problem

Passengers and and airline wants a way to reduce economic loss, rushes, tension which caused by fight delay and make their travel happy.

## 2.References

The principle of the Stacking classification algorithm is introduced, the SMOTE algorithm is selected to process imbalanced datasets, and the Boruta algorithm is utilized for feature selection. There are five supervised machine learning algorithms in the first-level learner of Stacking including KNN, Random Forest, Logistic Regression, Decision Tree, and Gaussian Naive Bayes.The second-level learner is Logistic Regression. To verify the effectiveness of the proposed method, comparative experiments are carried out based on Boston Logan International Airport flight datasets from January to December 2019. Multiple indexes are used to comprehensively evaluate the prediction results, such as Accuracy, Precision, Recall, F1 Score, ROC curve, and AUC Score. The results show that the Stacking algorithm not only could improve the prediction accuracy but also maintains great stability (1). Multi Layer Perceptron (MLP) to train and test dataapplied. The neural network MLP was able to predict flight arrival delay with a coefficient of determination R2 of 0.9048, and the selective procedure achieved a time saving and a better R2 score of 0.9560. To enhance the reliability of the proposed method, the performance of the MLP was compared with that of Gradient Boosting (GB) and Decision Trees (DT). The result is that theMLP outperformed all existing benchmark methods (2). Here present the first data-driven systemicstudy of air transport delays in China, of their evolution and causes, based on 11 million flights between 2016 and 2018. A significant fraction of the delays can be explained by a few variables, e.g., weather conditions and traffic levels, the most important factors being the presence of thunderstorms and the season of the year. Remaining delays can often be explained by en-route weather phenomena or by reactionary delays. This study contributes towards a better understanding of delays and their prediction through a data-driven methodology, leveraging on statistics and data mining concepts (3). Recognize useful patterns of the flight delay from aviationdata and perform accurate delay prediction. The best result for flight delay prediction (five classes) using machine learning models is 89.07% (Multilayer Perceptron). A Convolution neural networkmodel is also built which is enlightened by the idea of pattern recognition and success of neural network method, showing a slightly better result with 89.32% prediction accuracy (4). Explores abroader scope of factors which may potentially influence the flight delay, and compares several machine learning-based models in designed generalized flight delay prediction tasks. Compared with the previous schemes, the proposed random forest-based model can obtain higher predictionaccuracy (90.2% for the binary classification) and can overcome the overfitting problem (5). Themodel demonstrated to reduce by 30% the take-off time prediction errors of the current system onehour before the time that flight is scheduled to depart from the parking position and presents an extension of the model, which overcomes this look-ahead time constraint and allows to improve take-off time predictions as early as the initial flight plan is received. In addition, a subset of the original set of input features has been meticulously selected to facilitate the implementation of the solution in an operational air traffic flow and capacity management system, while minimising theloss of predictive power. Finally, the importance and interactions of the input features arethoroughly analysed with additive feature attribution methods (6). The designed prediction tasks contain different classification tasks and a regression task. Experimental results show that long short-term memory (LSTM) is capable of handling the obtained aviation sequence data, but overfitting problem occurs in our limited dataset. Compared with the previous schemes, the proposed random forest-based model can obtain higher prediction accuracy (90.2% for the binaryclassification) and can overcome the overfitting problem (7).

## References

1. *Flight Delay Classification Prediction Based on.* Jia Yi, 1 Honghai Zhang ,. [ed.] Chi-HuaChen. Honghai Zhang; zhh0913@163.com : Wiley, 2021. p. 10.

2. *A Multilayer Perceptron Neural Network with Selective-Data.* Hajar Alla, Lahcen Moumoun
, and Youssef Balouki. [ed.] Jianping Gou. Settat, Morocco : Hindawi, 2021. Hindawi Scientific Programming. p. 12.

3. *Characterization and Prediction of Air Transport.* Massimiliano Zanin 1, * , Yanbo Zhu 2,3,Ran Yan 3. 2020, MDPI Journals, p. 15.

4. *Applying Machine Learning to Aviation Big Data.* Yushan Jiang, Yongxin Liu,Dahai Liu,Houbing Song. 2020. 2020 IEEE Intl Conf on Dependable,Intl Conf on Cloud and Big Data Computing. p. 8.

5. *FLIGHT DELAY PREDICTION USING MACHINE LEARNING.* Sarah Ajmeria, Srushti V,Prof. Kavitha S Patil. Bangalore, India : IJIREEICE, 2022. DOI: 10.17148/IJIREEICE.2022.10584. p. 5.

6. *An explainable machine learning approach to improve take-off time.* Ramon Dalmau, Franck Ballerini,Herbert Naessens,Seddik Belkoura. 2021, Journal of Air Transport Management, p. 12.

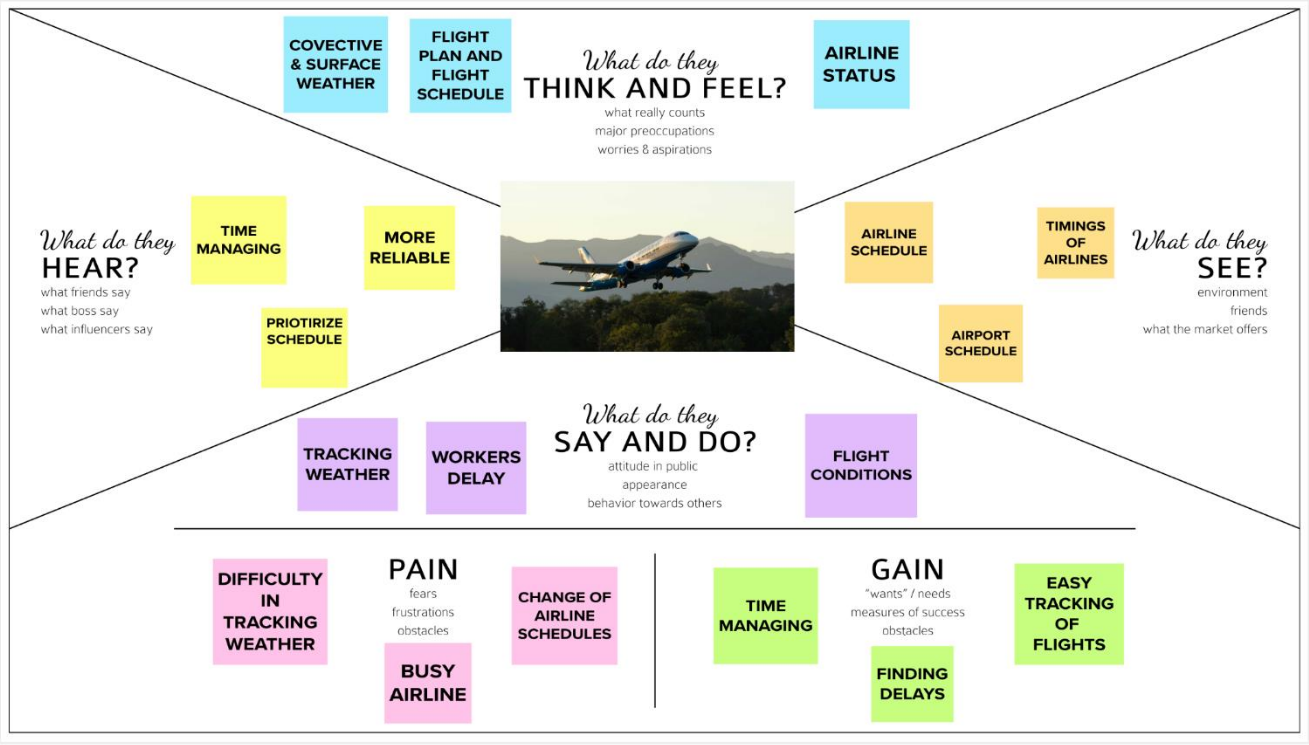
7. *Flight Delay Prediction Based on Aviation Big Data.* Guan Gui, Senior Member, IEEE, Fan Liu, Student Member, IEEE, Jinlong Sun, Member, IEEE,. 2020, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. XX, NO. XX, XXX 2015, p. 11.

## 3.Problem statement definition

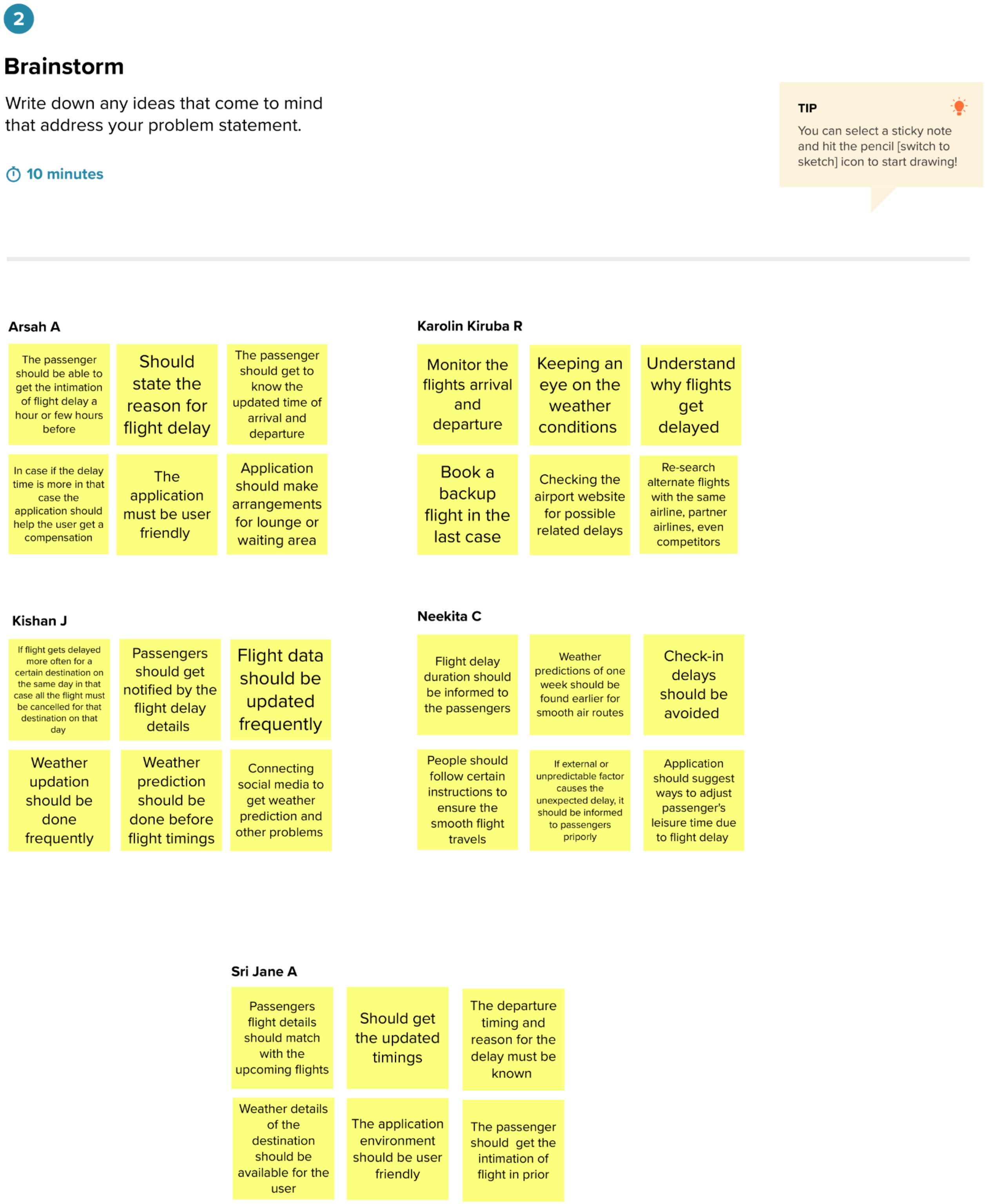
| Problem Statement (PS) | I am (Customer) | I'm trying to | But | Because | Which makes me feel |
|---|---|---|---|---|---|
| PS-1 | Passenger | Reach my destination with nice and happy travel experience | Sometimes rushes and tensions happen | Flight delay, diversion and cancelation | Frustrated and upset. |
| PS-2 | Airlines andAirport companies | Gives best and pleasant experience for passengers | Sometimes became a one of worst experience for passengers | Flight delay, diversion and cancelation causes economic losses and damage the brand value in market and society | Discourage, less effective and vague |

# III. Ideation & Proposed Solution

## 1.Empathy Map Canvas



# 2.Ideation & Brainstorming



2

## Brainstorm

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

10 minutes

TIP

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

**Arsah A**

- The passenger should be able to get the intimation of flight delay a hour or few hours before
- Should state the reason for flight delay
- The passenger should get to know the updated time of arrival and departure
- In case if the delay time is more in that case the application should help the user get a compensation
- The application must be user friendly
- Application should make arrangements for lounge or waiting area

**Karolin Kiruba R**

- Monitor the flights arrival and departure
- Keeping an eye on the weather conditions
- Understand why flights get delayed
- Book a backup flight in the last case
- Checking the airport website for possible related delays
- Re-search alternate flights with the same airline, partner airlines, even competitors

**Kishan J**

- If flight gets delayed more often for a certain destination on the same day in that case all the flight must be cancelled for that destination on that day
- Passengers should get notified by the flight delay details
- Flight data should be updated frequently
- Weather updation should be done frequently
- Weather prediction should be done before flight timings
- Connecting social media to get weather prediction and other problems

**Neekita C**

- Flight delay duration should be informed to the passengers
- Weather predictions of one week should be found earlier for smooth air routes
- Check-in delays should be avoided
- People should follow certain instructions to ensure the smooth flight travels
- If external or unpredictable factor causes the unexpected delay, it should be informed to passengers priporly
- Application should suggest ways to adjust passenger's leisure time due to flight delay

**Sri Jane A**

- Passengers flight details should match with the upcoming flights
- Should get the updated timings
- The departure timing and reason for the delay must be known
- Weather details of the destination should be available for the user
- The application environment should be user friendly
- The passenger should get the intimation of flight in prior

3

## Group ideas

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

20 minutes

The passenger should be able to get the intimation of flight delay a hour or few hours before

The passenger should get the intimation of flight in prior

Passengers should get notified by the flight delay details

The passenger should get to know the updated time of arrival and departure

Flight delay duration should be informed to the passengers

Should get the updated timings

The application environment should be user friendly

The application must be user friendly

TIP

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

Should state the reason for flight delay

Understand why flights get delayed

The departure timing and reason for the delay must be known

Monitor the flights arrival and departure

Flight data should be updated frequently

Weather updation should be done frequently

Weather prediction should be done before flight timings

Weather details of the destination should be available for the user

Weather predictions of one week should be found earlier for smooth air routes

Keeping an eye on the weather conditions

In case if the delay time is more in that case the application should help the user get a compensation

Book a backup flight in the last case

Re-search alternate flights with the same airline, partner airlines, even competitors

Application should suggest ways to adjust passenger's leisure time due to flight delay

People should follow certain instructions to ensure the smooth flight travels

Passengers flight details should match with the upcoming flights

Application should make arrangements for lounge or waiting area

If external or unpredictable factor causes the unexpected delay, it should be informed to passengers priporly

If flight gets delayed more often for a certain destination on the same day in that case all the flight must be cancelled for that destination on that day

Checking the airport website for possible related delays

Check-in delays should be avoided

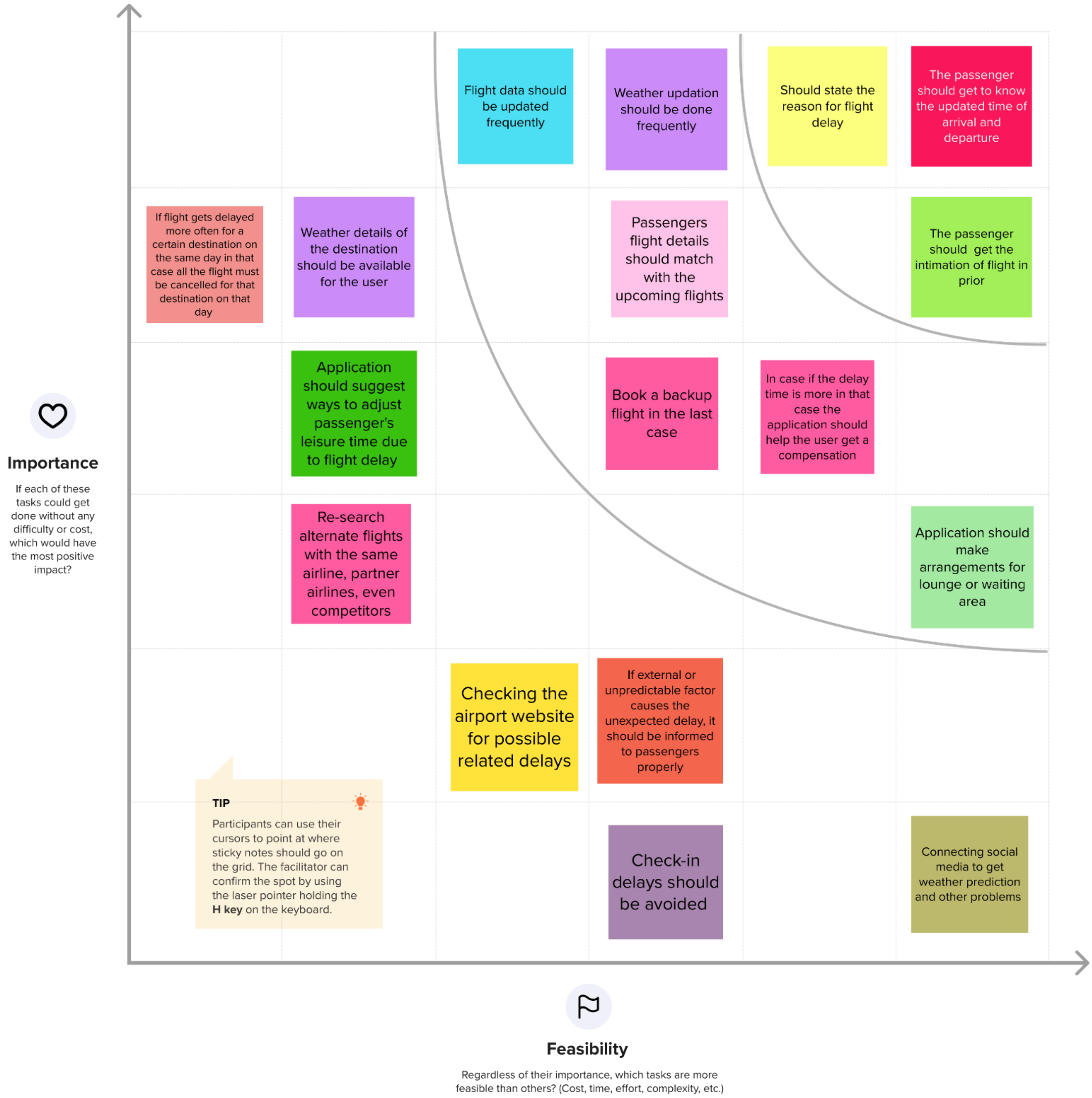
Connecting social media to get weather prediction and other problems

4

## Prioritize

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

20 minutes



Flight data should be updated frequently

Weather updation should be done frequently

Should state the reason for flight delay

The passenger should get to know the updated time of arrival and departure

If flight gets delayed more often for a certain destination on the same day in that case all the flight must be cancelled for that destination on that day

Weather details of the destination should be available for the user

Passengers flight details should match with the upcoming flights

The passenger should get the intimation of flight in prior

Application should suggest ways to adjust passenger's leisure time due to flight delay

Book a backup flight in the last case

In case if the delay time is more in that case the application should help the user get a compensation

Re-search alternate flights with the same airline, partner airlines, even competitors

Application should make arrangements for lounge or waiting area

Checking the airport website for possible related delays

If external or unpredictable factor causes the unexpected delay, it should be informed to passengers properly

TIP

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

Check-in delays should be avoided

Connecting social media to get weather prediction and other problems

**Importance**

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

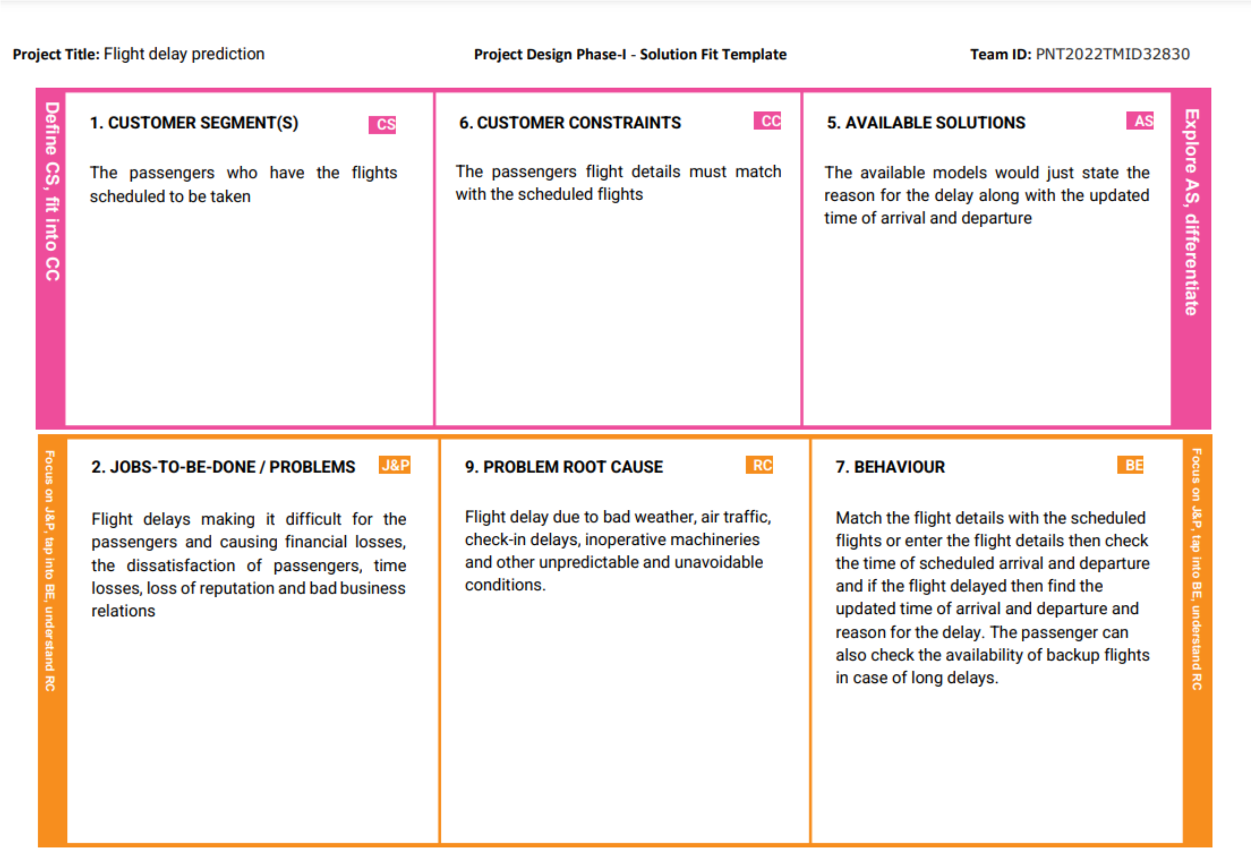
**Feasibility**

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

## 3.Proposed Solution

| S.No. | Parameter | Description |
|---|---|---|
| 1. | Problem Statement (Problem to be solved) | Developing A Flight Delay Prediction ModelUsing Machine Learning |
| 2. | Idea / Solution description | A model which will predict the delay in arrival and departure of the flight in order to optimize flight operations and minimize the delays |
| 3. | Novelty / Uniqueness | The model would state the reason for the delay to passenger and will also check the alternate routes and the availability of seats in backup flights |
| 4. | Social Impact / Customer Satisfaction | With the help of this model the passengers would be able to manage the time in a efficient and effective way |
| 5. | Business Model (Revenue Model) | |
| 6. | Scalability of the Solution | This model can cater only up to a certain extent due to limited data set but with the availability of live data set and proper machine learning algorithms we can increase the accuracy in near future |

## 4. Problem Solution Fit



**Project Title:** Flight delay prediction

**Project Design Phase-I - Solution Fit Template**

**Team ID:** PNT2022TMID32830

Define CS, fit into CC

**1. CUSTOMER SEGMENT(S)** CS

The passengers who have the flights scheduled to be taken

**6. CUSTOMER CONSTRAINTS** CC

The passengers flight details must match with the scheduled flights

**5. AVAILABLE SOLUTIONS** AS

The available models would just state the reason for the delay along with the updated time of arrival and departure

Explore AS, differentiate

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

**2. JOBS-TO-BE-DONE / PROBLEMS** J&P

Flight delays making it difficult for the passengers and causing financial losses, the dissatisfaction of passengers, time losses, loss of reputation and bad business relations

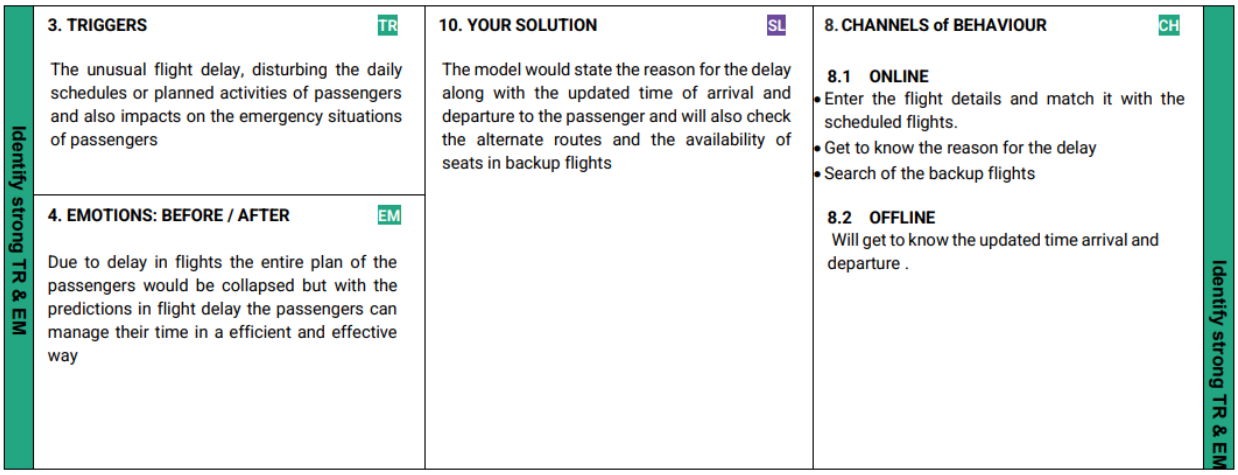
**9. PROBLEM ROOT CAUSE** RC

Flight delay due to bad weather, air traffic, check-in delays, inoperative machineries and other unpredictable and unavoidable conditions.

**7. BEHAVIOUR** BE

Match the flight details with the scheduled flights or enter the flight details then check the time of scheduled arrival and departure and if the flight delayed then find the updated time of arrival and departure and reason for the delay. The passenger can also check the availability of backup flights in case of long delays.

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



Identify strong TR & EM

**3. TRIGGERS** TR

The unusual flight delay, disturbing the daily schedules or planned activities of passengers and also impacts on the emergency situations of passengers

**4. EMOTIONS: BEFORE / AFTER** EM

Due to delay in flights the entire plan of the passengers would be collapsed but with the predictions in flight delay the passengers can manage their time in a efficient and effective way

**10. YOUR SOLUTION** SL

The model would state the reason for the delay along with the updated time of arrival and departure to the passenger and will also check the alternate routes and the availability of seats in backup flights

**8. CHANNELS of BEHAVIOUR** CH

**8.1 ONLINE**

- Enter the flight details and match it with the scheduled flights.
- Get to know the reason for the delay
- Search of the backup flights

**8.2 OFFLINE**

Will get to know the updated time arrival and departure .

Identify strong TR & EM

# IV. Requirement Analysis

## 1.Functional Requirements:

Following are the functional requirements of the proposed solution.

| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
|---|---|---|
| FR-1 | User Registration | Registration through Form Registration through Gmail |
| FR-2 | User Confirmation | Confirmation via EmailConfirmation via OTP |
| FR-3 | Log In | Log in with give user credentials |
| FR-4 | Support | Support option provided for queries andcontact customer support team |
| FR-5 | Prediction of delay | Requesting for prediction by providingdetails of flight<br>Shows prediction results |
| FR-6 | Trust ability of prediction | Gives the confidence percentage abouttheir prediction |
| FR-7 | Notify User | Notify user with the prediction, if theywish<br>Notify user about flight arrival before 1hour |

| | | |
|---|---|---|
| FR-8 | Get Feedback | Get feedback about their experience<br>Request give rating |
| FR-9 | Log Out | Log out from the application |

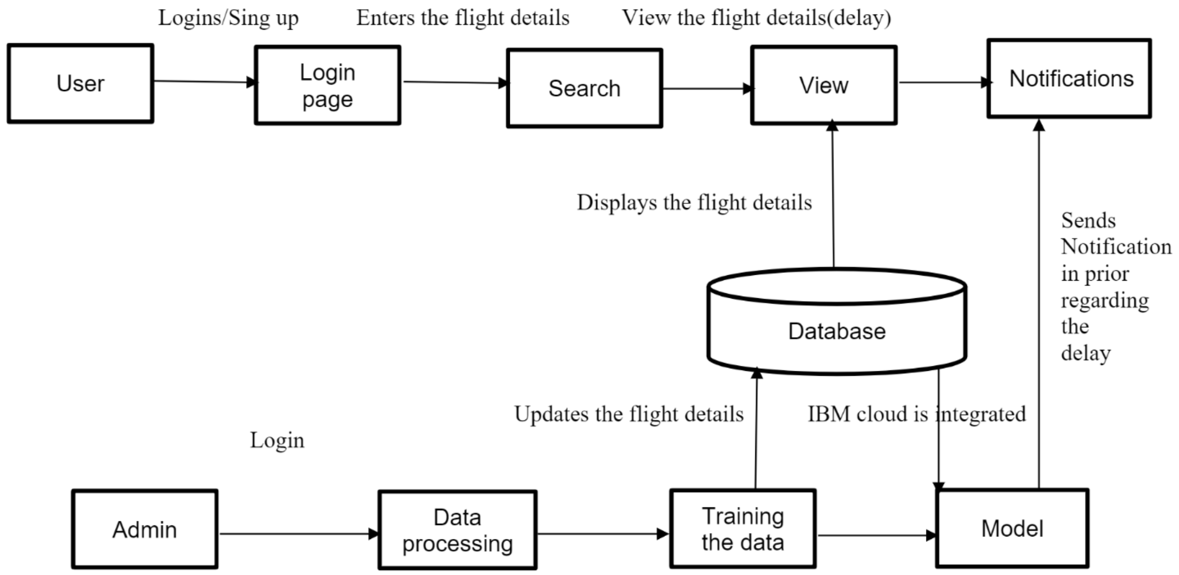
## 2.Non-functional Requirements:

| FR No. | Non-Functional Requirement | Description |
|---|---|---|
| NFR-1 | **Usability** | Web app is provided with smooth and user-friendly GUI. |
| NFR-2 | **Security** | Data security of user is ensured with IBM Cloud sec secured login credentials<br>. |
| NFR-3 | **Reliability** | This web app have reliability by deploying in IBM Watson. |
| NFR-4 | **Performance** | 50 request per second is handled. |
| NFR-5 | **Availability** | 99% avail with the help of IBM Cloud. |
| NFR-6 | **Scalability** | It had high scalability by having ability to extendthe when request came. |

# V. Project Design:

## 1.Data Flow Diagrams

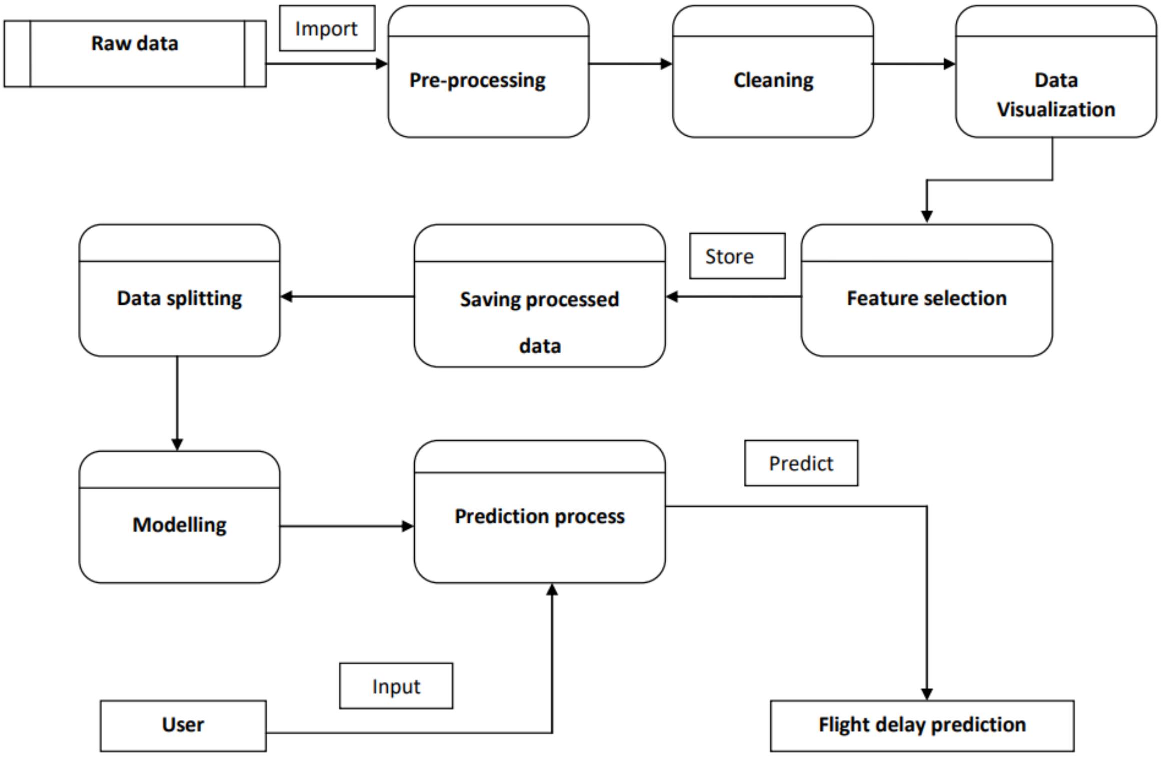
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.


Logins/Sing up
Enters the flight details
View the flight details(delay)
User
Login page
Search
View
Notifications
Displays the flight details
Sends Notification in prior regarding the delay
Database
Updates the flight details
IBM cloud is integrated
Login
Admin
Data processing
Training the data
Model

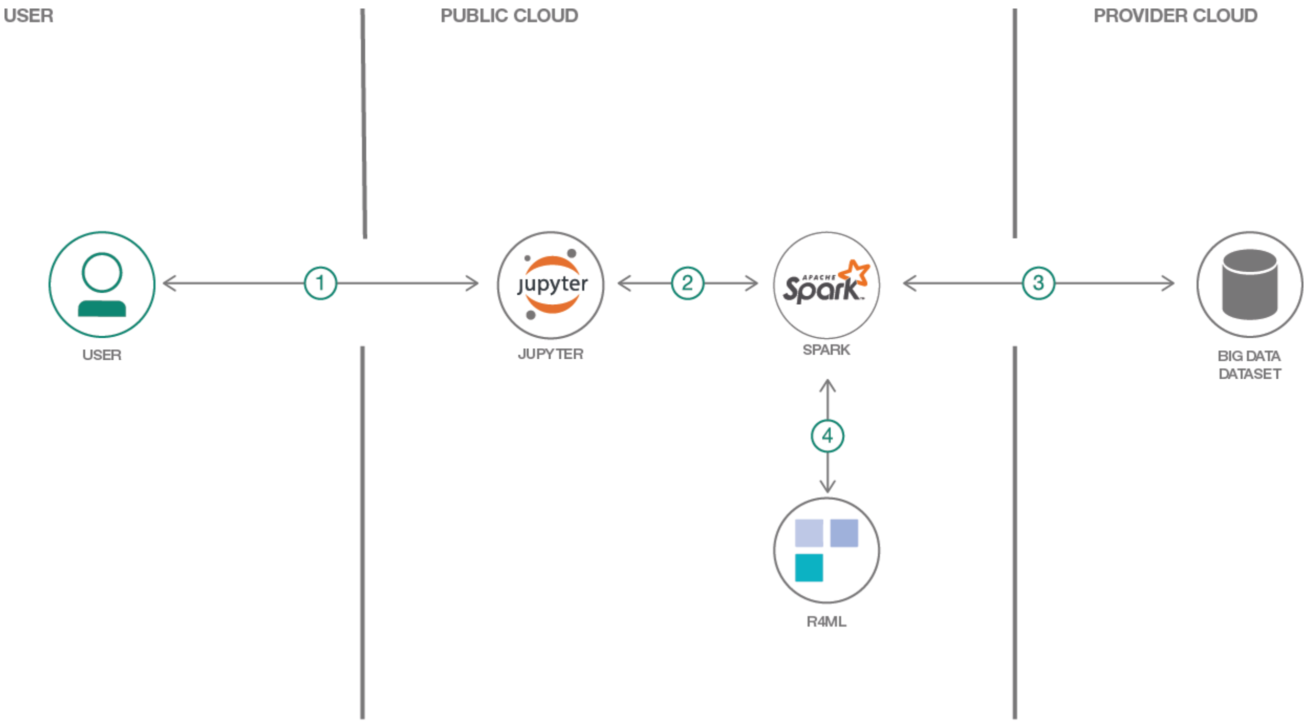
**User Stories:**

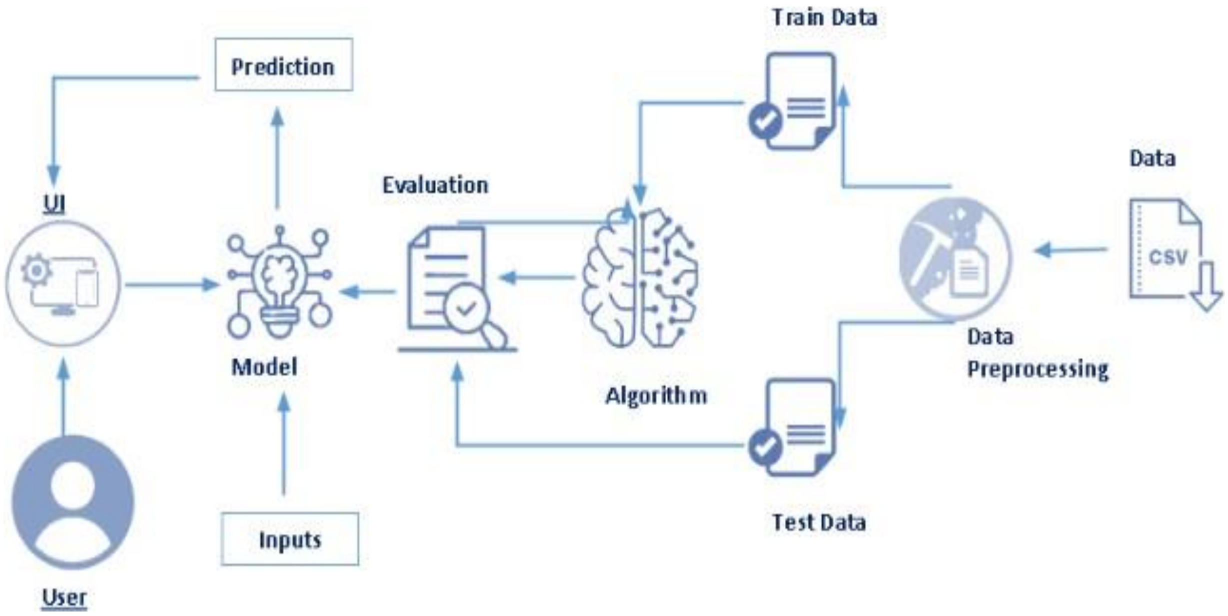
# 2. Solution & Technological Architecture

## Solution Architecture



## Technical Architecture





## Application Characteristics:

| S.No | Characteristics | Description | Technology |
|---|---|---|---|
| **1.** | Open-Source Frameworks | Flask, Scipy, Jupiter Notebook | microframework |
| **2.** | Scalable Architecture | 3 – tier, Micro-services | Relational database, cloud, GUI |
| **3.** | Availability | distributed servers | IBM Cloud |
| **4.** | Performance | 100 per sec | IBM Watson App Service |

## 3.User Stories

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptanc ecriteria | Priority | Release |
|---|---|---|---|---|---|---|
| Customer (Web user) | Registration | USN-1 | As a user, I can register for the application by entering my email, password, | I can access my account / dashboard | High | Sprint-1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | and confirming my password. | | | |
| | | USN-2 | As a user, I will receive confirmation email once I have registered for the application | I can receive confirmation email & click confirm | High | Sprint-1 |
| | | USN-3 | As a user, I can register for the application through Gmail | I can register and login by providing Gmail with access permissions | Medium | Sprint-2 |
| | Login | USN-4 | As a user, I can log into the application by entering email & password | I can login using my registered email and password(login credentials) | High | Sprint-1 |
| | Dashboard | USN-5 | As a user, I can access my dashboard which give way to provide data to predict flight delay | I can provide valid Input data | High | Sprint-2 |
| | Logout | USN-6 | As a user, I can logout by clicking logout button | I can logout my account from website | High | Sprint-1 |
| | Prediction | USN-7 | As a user, I can prediction result through dashboard by integratedML Model | I can get prediction by giving valid input | High | Sprint-3 |
| Customer Care / support | Query/ complaint raise | USN-8 | As a user, I can raise Query or complaint about technical issues | If raised query valid or true then resolve and response, else explain the missing understanding | Medium | Sprint-4 |
| | Feedback/ rating | USN-9 | As a user, I can give feedback and rating tothe application | Support team accept the feedback , try to improve application | Medium | Sprint-4 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Administrator | Maintain | USN-10 | Administrator maintain the databaseand overall application | Punctual maintenance | High | Sprint-4 |
| Developer | Testing | USN-11 | As a developer, I test the application which I have developed | I test the application for checking errors and rectify it | High | Sprint- 1,2,3,4 |

# VI. Project Planning & Scheduling

## 1.Sprint Planning & Estimation

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-1 | Data Collection and Pre-processing | USN-1 | As a user, I can't interact anything. Waiting is user's task. User can listen the relationship exist between the various attributes of data by presentation of developer | 2 | high | Arsah A |
| Sprint-1 | Model Building | USN-2 | As a user, I can predict flight delay by various developed ML models by console | 1 | high | Arsah A |
| Sprint-2 | Model Evaluation | USN-3 | As a user, I can predict flight delay by best Model in various developed ML model by console | 2 | high | Karolin Kiruba R |
| Sprint-2 | Model Deployment on<br>IBM Cloud using IBM Watson | USN-4 | As a user, I can use the model by requesting the deployed model on Cloud | 1 | Medium | Karolin kiruba R |

| | | | | | | |
|---|---|---|---|---|---|---|
| Sprint-2 | Basic user interaction Dashboard | USN-5 | As a user, I can use the model or prediction from model by interacting with dashboard | 2 | high | Karolin Kiruba R |
| Sprint-3 | Improved Dashboard and GUI | USN-6 | As a user, I can use the model or prediction from model by interacting with improved dashboard | 1 | Medium | Kishan I |
| Sprint-3 | Registration | USN-7 | As a user, I can register for the application by entering my email, password, and confirming my password. | 2 | High | Kishan I |
| Sprint-3 | Registration | USN-7 | As a user, I can register for the application by entering my email, password, and confirming my password. | 2 | High | Kishan I |
| Sprint-3 | Login | USN-8 | As a user, I can log into the application by entering email & password and I can register .login to the application through Gmail | 2 | Medium | Sri Jane A |
| Sprint-4 | Raise query/complaint and give feedback | USN-9 | As a user, I can raise complaint or query and give feedback | 1 | Medium | Neekitha C |
| Sprint-4 | Improve overall web app | USN-10 | As a user, I can user revised and improved version of web application | 1 | High | Neekitha C |

## 2.Sprint Delivery Schedule

| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed (as on Planned End Date) | Sprint Release Date(Actual) |
|---|---|---|---|---|---|---|
| Sprint-1 | 20 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 20 | 31 Oct 2022 |

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

## Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

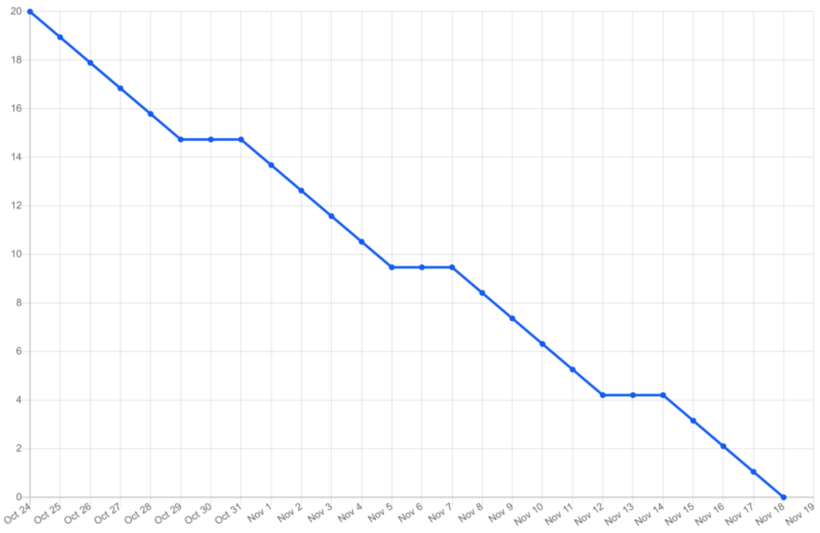
(Sprint – 1 ) Velocity = 3/20 = 0.15

( Sprint – 2 ) Velocity = 3/20 = 0.15

( Sprint – 3 ) Velocity = 7/20 = 0.35

( Sprint – 4 ) Velocity = 2/20 = 0.1

## Burndown chart:



# VII. Coding & Solutioning

## SPRINT-1:

### Outline:

1. Data Pre-processing
2. EDA/Data Analysis
3. Feature Engineering
4. Model Building
5. Saving Best Model

### Required Libraries:

| | |
|---|---|
| Pandas | - Data Pre-processing |
| Numpy | - Data Pre-processing, Analysis |
| Matplotlib | - Visualization |
| Seaborn | - Visualization |
| Imblearn | - Balancing Data |
| Sklearn | - Model Building |
| Pickle | - Model saving |

### Software/Tools:

- Anaconda – Jupyter Notebook
- Used Language Python

## Data Pre-processing:

### Data Collection:

Dataset is collected from the IBM career smartinternz portal in Guided Project.

### Dataset description:

The dataset contains 31 variables with various data types such as string, object, time, integer, float.

```
Data columns (total 31 columns):

#   Column                    Non-Null Count  Dtype

  --  -------             ---------------  ------
 0  YEAR   11231 non-null  int64
 1  QUARTER                       11231 non-null  int64
 2  MONTH                         11231 non-null  int64
 3  DAY_OF_MONTH                  11231 non-null  int64
 4  DAY_OF_WEEK                   11231 non-null  int64
 5  UNIQUE_CARRIER                11231 non-null  object
 6  TAIL_NUM                      11231 non-null  object
 7  FL_NUM                        11231 non-null  int64
 8  ORIGIN_AIRPORT_ID             11231 non-null  int64
 9  ORIGIN                        11231 non-null  object
10  DEST_AIRPORT_ID               11231 non-null  int64
11  DEST    11231 non-null  object
12  CRS_DEP_TIME                  11231 non-null  object
13  CRS_DEP_TIME.1                11231 non-null  int64
14  DEP_TIME                      11124 non-null  object
15  DEP_TIME.1                    11124 non-null  float64
16  DEP_DELAY                     11124 non-null  float64
17  DEP_DEL15                     11124 non-null  float64
18  CRS_ARR_TIME                  11231 non-null  object
19  CRS_ARR_TIME.1                11231 non-null  int64
20  ARR_TIME                      11116 non-null  object
21  ARR_TIME.1                    11116 non-null  float64
22  ARR_DELAY                     11043 non-null  float64
23  ARR_DEL15                     11043 non-null  float64
24  CANCELLED                     11231 non-null  int64
25  DIVERTED                      11231 non-null  int64
26  CRS_ELAPSED_TIME1             11231 non-null  object
27  ACTUAL_ELAPSED_TIME1  11231 non-null  object
28  CRS_ELAPSED_TIME              11231 non-null  int64
29  ACTUAL_ELAPSED_TIME  11043 non-null  float64
30  DISTANCE      11231 non-null int64dtypes: float64(7), int64(14), object(
```

### Columns Description:

Dest means Destination Airport.

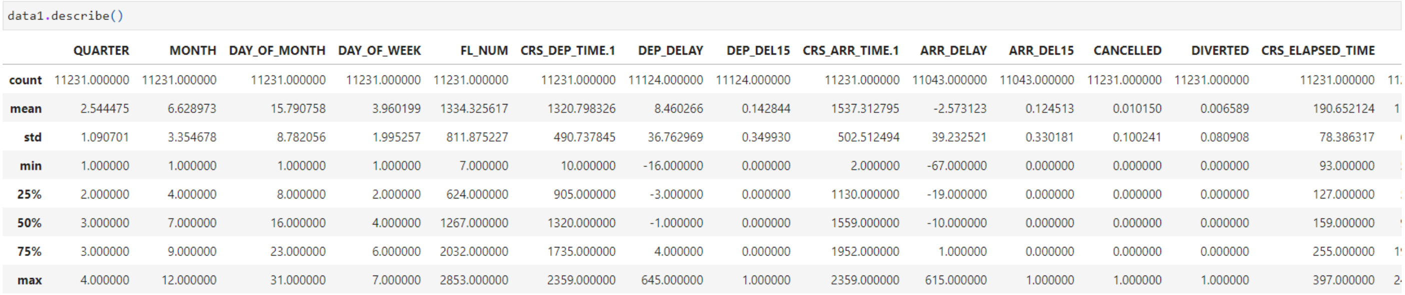
Crs_dep_time and crs_arr_time is planned departure and arrival time.Crs_elapsed _time is estimated travel time as per plan.

Arr_time and dep_time are actual arrival and departure time.Actual_elapsed_time is actual travelled time

To pre-process our dataset, we need to import above mentioned required libraries, then importdata using  pandas.

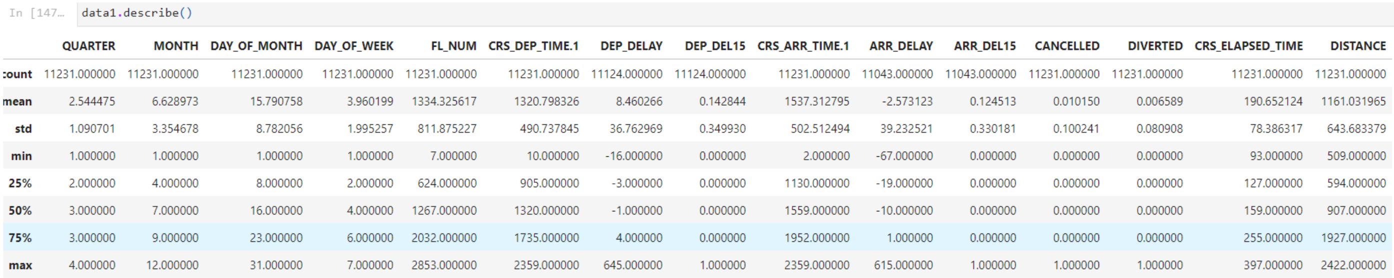
This data does not contain any duplicated values and null values except in arrival , departuretime columns, because these left empty when flights are cancelled.

## Descriptive Analytics:



```
data1.describe()
```

| | QUARTER | MONTH | DAY_OF_MONTH | DAY_OF_WEEK | FL_NUM | CRS_DEP_TIME.1 | DEP_DELAY | DEP_DEL15 | CRS_ARR_TIME.1 | ARR_DELAY | ARR_DEL15 | CANCELLED | DIVERTED | CRS_ELAPSED_TIME |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 11231.000000 | 11231.000000 | 11231.000000 | 11231.000000 | 11231.000000 | 11231.000000 | 11124.000000 | 11124.000000 | 11231.000000 | 11043.000000 | 11043.000000 | 11231.000000 | 11231.000000 | 11231.000000 |
| **mean** | 2.544475 | 6.628973 | 15.790758 | 3.960199 | 1334.325617 | 1320.798326 | 8.460266 | 0.142844 | 1537.312795 | -2.573123 | 0.124513 | 0.010150 | 0.006589 | 190.652124 |
| **std** | 1.090701 | 3.354678 | 8.782056 | 1.995257 | 811.875227 | 490.737845 | 36.762969 | 0.349930 | 502.512494 | 39.232521 | 0.330181 | 0.100241 | 0.080908 | 78.386317 |
| **min** | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 7.000000 | 10.000000 | -16.000000 | 0.000000 | 2.000000 | -67.000000 | 0.000000 | 0.000000 | 0.000000 | 93.000000 |
| **25%** | 2.000000 | 4.000000 | 8.000000 | 2.000000 | 624.000000 | 905.000000 | -3.000000 | 0.000000 | 1130.000000 | -19.000000 | 0.000000 | 0.000000 | 0.000000 | 127.000000 |
| **50%** | 3.000000 | 7.000000 | 16.000000 | 4.000000 | 1267.000000 | 1320.000000 | -1.000000 | 0.000000 | 1559.000000 | -10.000000 | 0.000000 | 0.000000 | 0.000000 | 159.000000 |
| **75%** | 3.000000 | 9.000000 | 23.000000 | 6.000000 | 2032.000000 | 1735.000000 | 4.000000 | 0.000000 | 1952.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 255.000000 |
| **max** | 4.000000 | 12.000000 | 31.000000 | 7.000000 | 2853.000000 | 2359.000000 | 645.000000 | 1.000000 | 2359.000000 | 615.000000 | 1.000000 | 1.000000 | 1.000000 | 397.000000 |

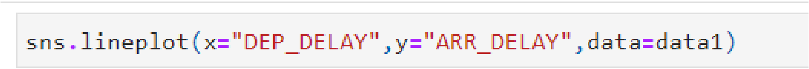


```
In [147… data1.describe()
```

| | QUARTER | MONTH | DAY_OF_MONTH | DAY_OF_WEEK | FL_NUM | CRS_DEP_TIME.1 | DEP_DELAY | DEP_DEL15 | CRS_ARR_TIME.1 | ARR_DELAY | ARR_DEL15 | CANCELLED | DIVERTED | CRS_ELAPSED_TIME | DISTANCE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 11231.000000 | 11231.000000 | 11231.000000 | 11231.000000 | 11231.000000 | 11231.000000 | 11124.000000 | 11124.000000 | 11231.000000 | 11043.000000 | 11043.000000 | 11231.000000 | 11231.000000 | 11231.000000 | 11231.000000 |
| **mean** | 2.544475 | 6.628973 | 15.790758 | 3.960199 | 1334.325617 | 1320.798326 | 8.460266 | 0.142844 | 1537.312795 | -2.573123 | 0.124513 | 0.010150 | 0.006589 | 190.652124 | 1161.031965 |
| **std** | 1.090701 | 3.354678 | 8.782056 | 1.995257 | 811.875227 | 490.737845 | 36.762969 | 0.349930 | 502.512494 | 39.232521 | 0.330181 | 0.100241 | 0.080908 | 78.386317 | 643.683379 |
| **min** | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 7.000000 | 10.000000 | -16.000000 | 0.000000 | 2.000000 | -67.000000 | 0.000000 | 0.000000 | 0.000000 | 93.000000 | 509.000000 |
| **25%** | 2.000000 | 4.000000 | 8.000000 | 2.000000 | 624.000000 | 905.000000 | -3.000000 | 0.000000 | 1130.000000 | -19.000000 | 0.000000 | 0.000000 | 0.000000 | 127.000000 | 594.000000 |
| **50%** | 3.000000 | 7.000000 | 16.000000 | 4.000000 | 1267.000000 | 1320.000000 | -1.000000 | 0.000000 | 1559.000000 | -10.000000 | 0.000000 | 0.000000 | 0.000000 | 159.000000 | 907.000000 |
| **75%** | 3.000000 | 9.000000 | 23.000000 | 6.000000 | 2032.000000 | 1735.000000 | 4.000000 | 0.000000 | 1952.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 255.000000 | 1927.000000 |
| **max** | 4.000000 | 12.000000 | 31.000000 | 7.000000 | 2853.000000 | 2359.000000 | 645.000000 | 1.000000 | 2359.000000 | 615.000000 | 1.000000 | 1.000000 | 1.000000 | 397.000000 | 2422.000000 |

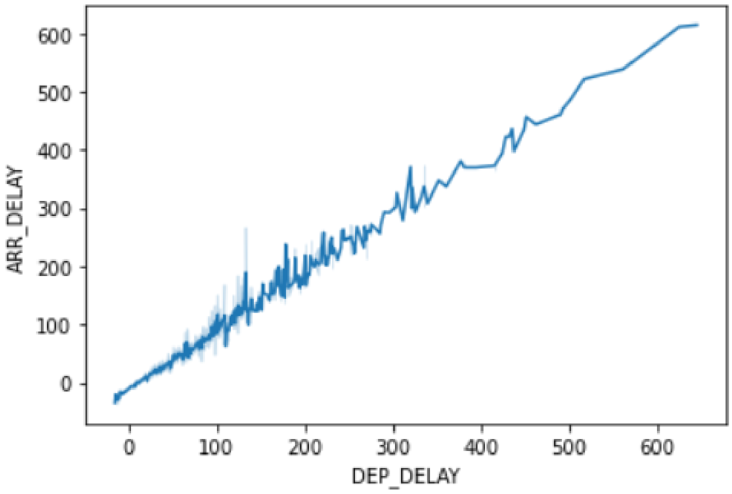
## Data Analysis and Visualisation:

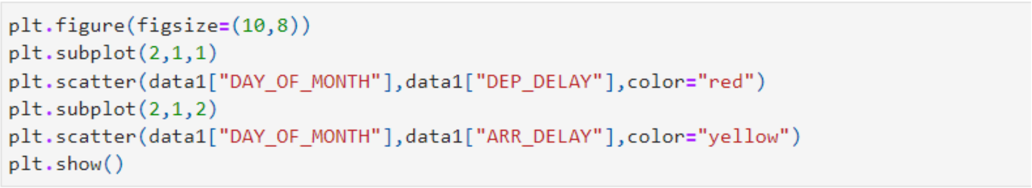
This graph shows the positive trend and strong binding between arrival and departure delay



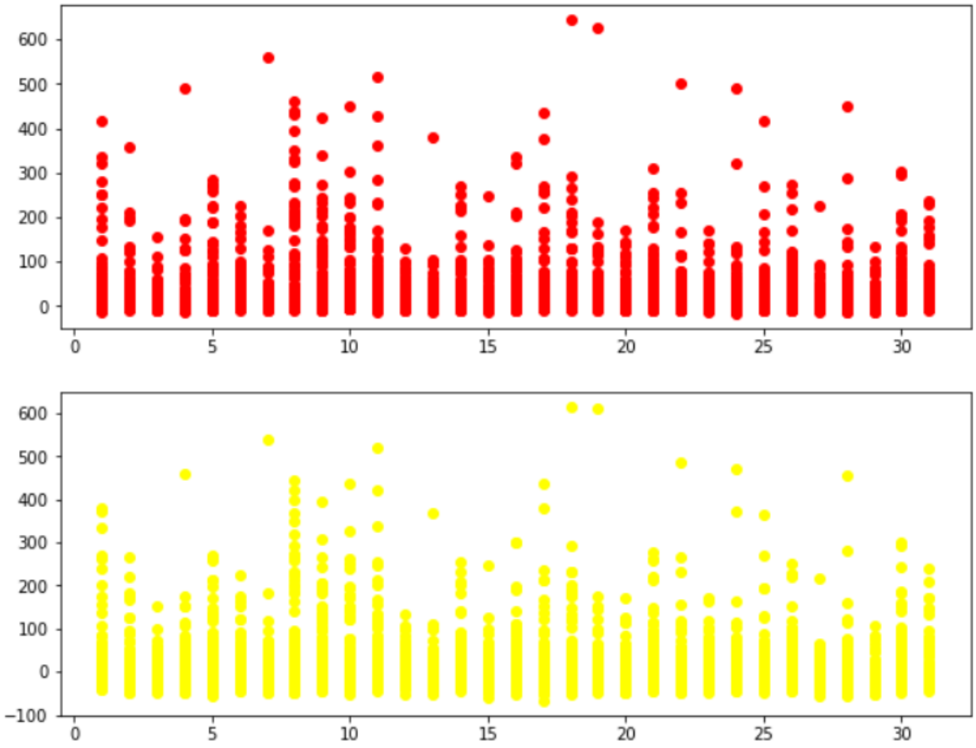
```
sns.lineplot(x="DEP_DELAY",y="ARR_DELAY",data=data1)
```

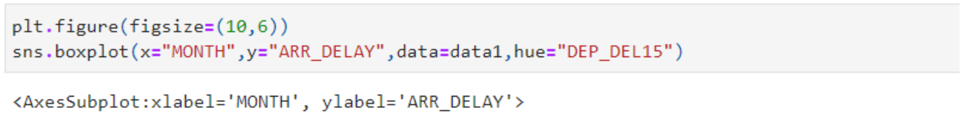
```
<AxesSubplot:xlabel='DEP_DELAY', ylabel='ARR_DELAY'>
```





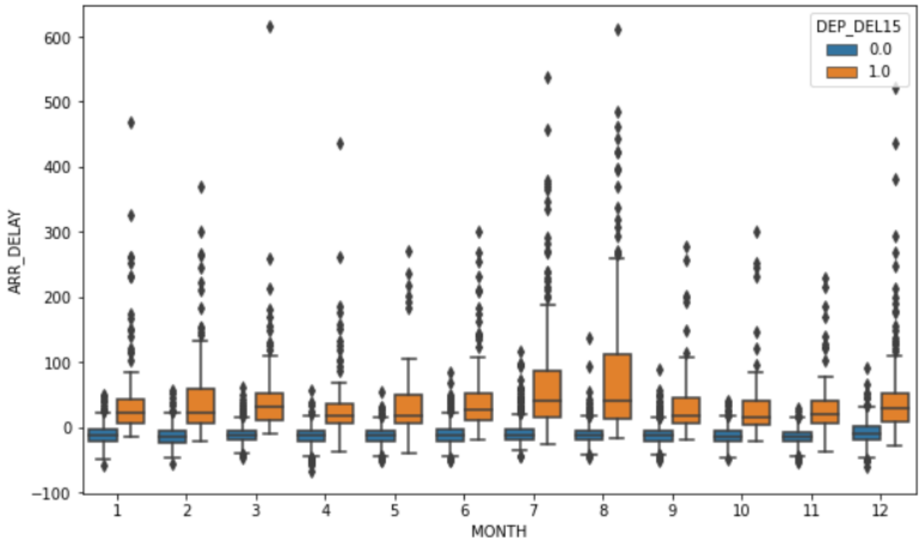
```
plt.figure(figsize=(10,8))
plt.subplot(2,1,1)
plt.scatter(data1["DAY_OF_MONTH"],data1["DEP_DELAY"],color="red")
plt.subplot(2,1,2)
plt.scatter(data1["DAY_OF_MONTH"],data1["ARR_DELAY"],color="yellow")
plt.show()
```



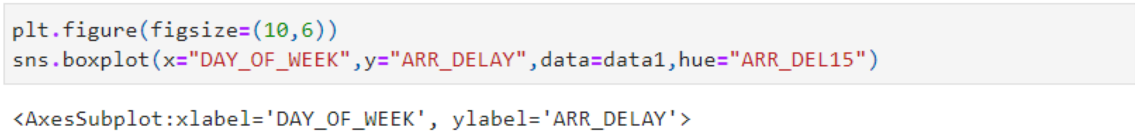


```
plt.figure(figsize=(10,6))
sns.boxplot(x="MONTH",y="ARR_DELAY",data=data1,hue="DEP_DEL15")
```

```
<AxesSubplot:xlabel='MONTH', ylabel='ARR_DELAY'>
```

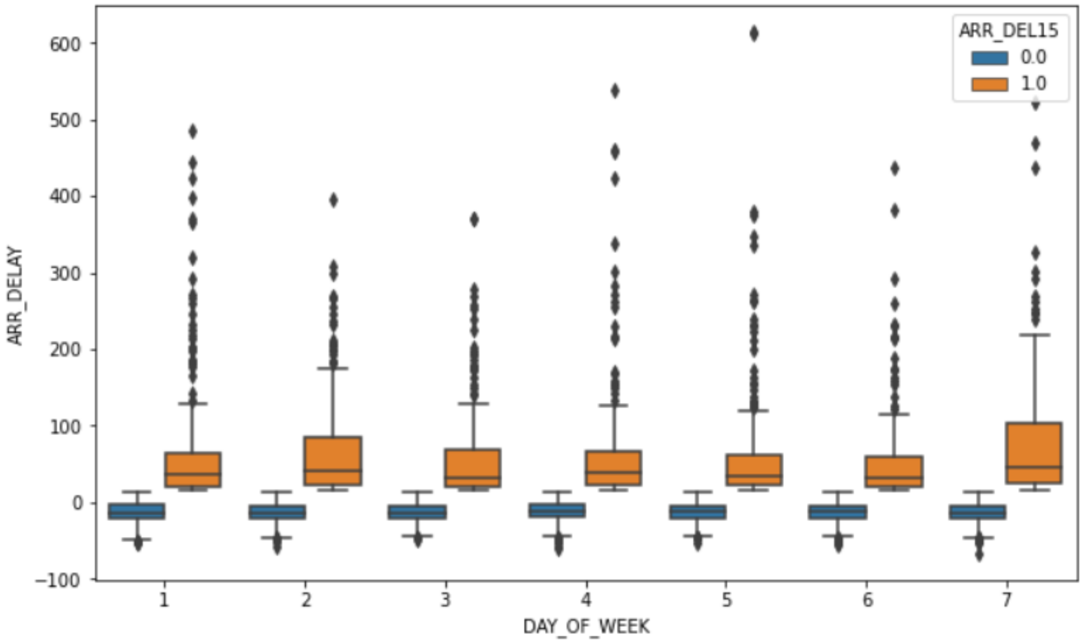


This above picture shows the relationship between day of month and delays



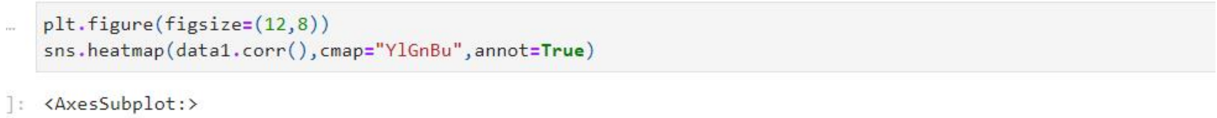
```
plt.figure(figsize=(10,6))
sns.boxplot(x="DAY_OF_WEEK",y="ARR_DELAY",data=data1,hue="ARR_DEL15")
```

```
<AxesSubplot:xlabel='DAY_OF_WEEK', ylabel='ARR_DELAY'>
```



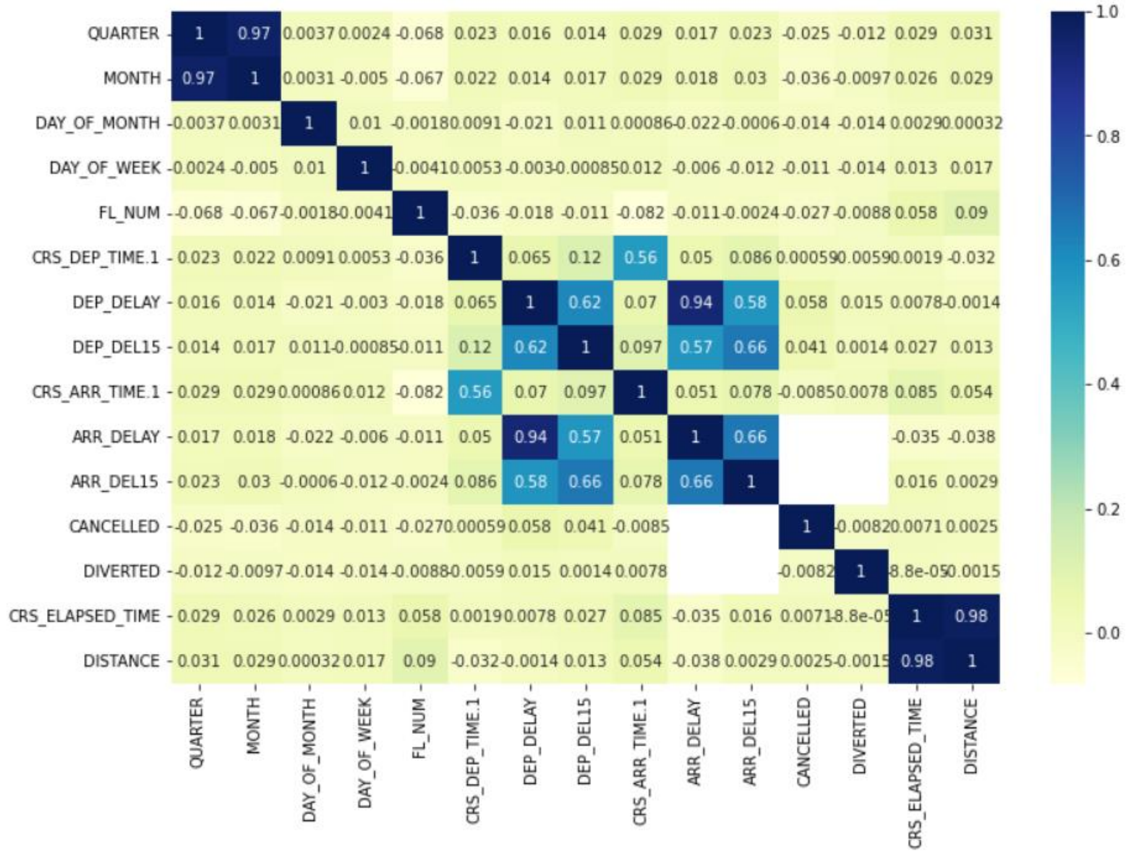
This above boxplot shows the trends of days of the week and delay, Monday and Saturday hadhigh delays.

## Correlation between columns:



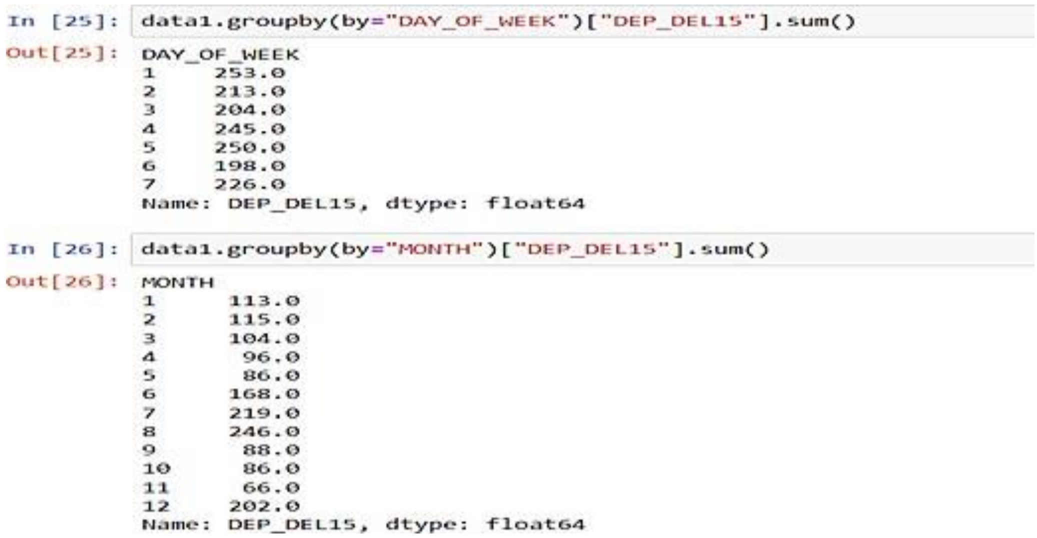
```
plt.figure(figsize=(12,8))
sns.heatmap(data1.corr(),cmap="YlGnBu",annot=True)
```

```
]: <AxesSubplot:>
```



## Feature Engineering:

We engineered Season from the month  according to the analysis



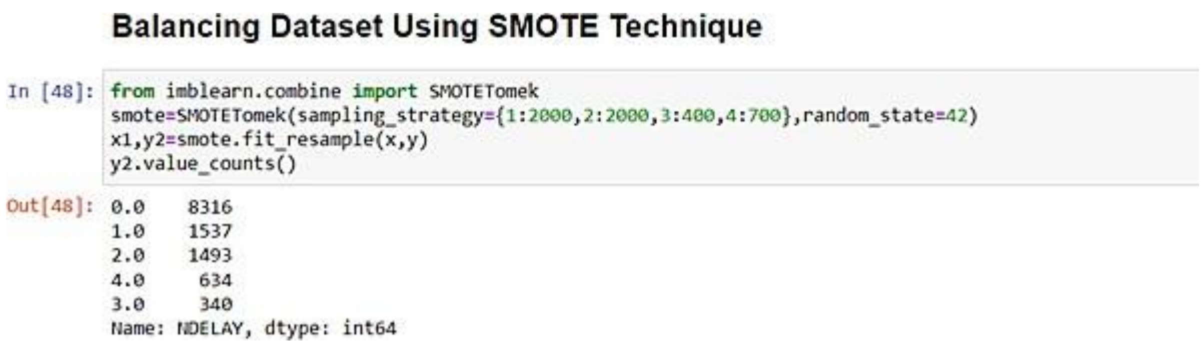
```
In [25]: data1.groupby(by="DAY_OF_WEEK")["DEP_DEL15"].sum()
Out[25]: DAY_OF_WEEK
         1     253.0
         2     213.0
         3     204.0
         4     245.0
         5     250.0
         6     198.0
         7     226.0
         Name: DEP_DEL15, dtype: float64

In [26]: data1.groupby(by="MONTH")["DEP_DEL15"].sum()
Out[26]: MONTH
         1      113.0
         2      115.0
         3      104.0
         4       96.0
         5       86.0
         6      168.0
         7      219.0
         8      246.0
         9       88.0
         10      86.0
         11      66.0
         12     202.0
         Name: DEP_DEL15, dtype: float64
```

Then Engineered NDELAY column from the summary of ARR_DEL15, DEP_DEL15, CANCELLED,DIVERTED columns.

Splitted NDELAY as dependenr column and others independent columns after removing unnecessary columns

## Data Balancing:

We balanced our using SMOTE technique which works based on KNN principle.



### Balancing Dataset Using SMOTE Technique

```
In [48]: from imblearn.combine import SMOTETomek
         smote=SMOTETomek(sampling_strategy={1:2000,2:2000,3:400,4:700},random_state=42)
         x1,y2=smote.fit_resample(x,y)
         y2.value_counts()

Out[48]: 0.0    8316
         1.0    1537
         2.0    1493
         4.0     634
         3.0     340
         Name: NDELAY, dtype: int64
```

Encoding Categorical columns into numerical columns:

We encoded ORGIN ,DEST into numerical columns.

## Model Buliding:

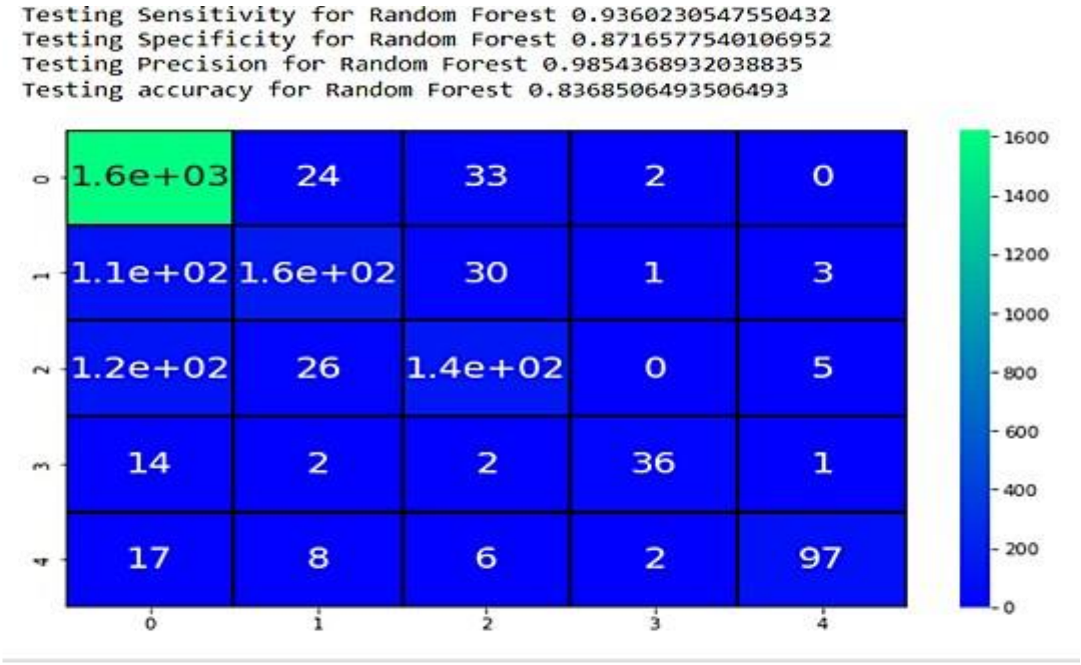
We builded

Decision Tree with 0.7536525974025974 Random Forest with 0.8368506493506493SVM with 0.6128246753246753

KNN with 0.7280844155844156

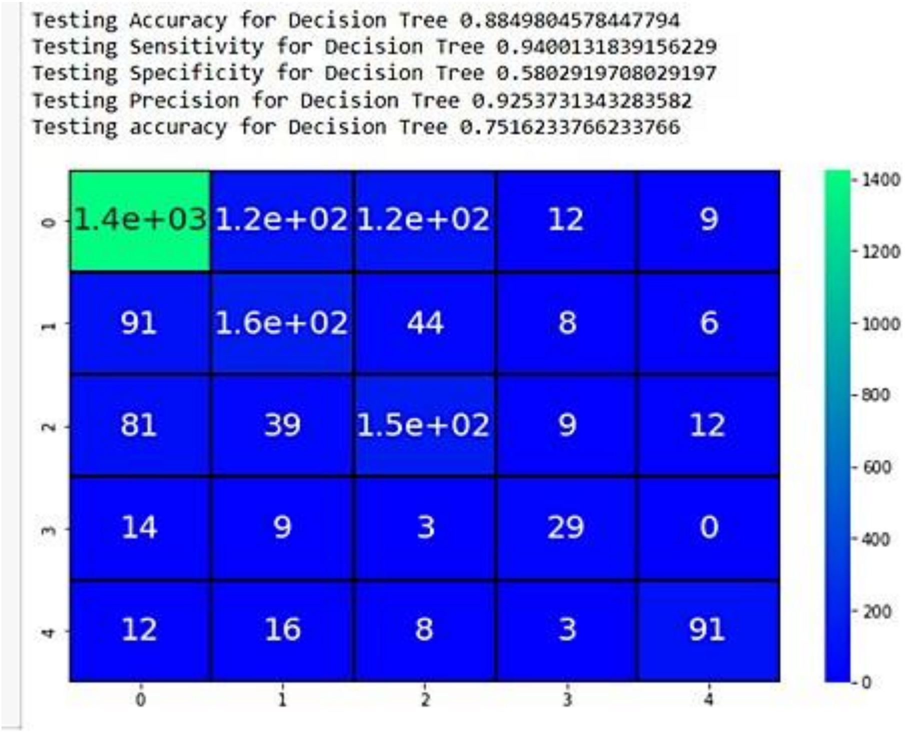
Logistic Regession with 0.6830357142857143

We will explore only Random Forest and Decision Tree which have high accuracyRandom Forest



```
Testing Sensitivity for Random Forest 0.9360230547550432
Testing Specificity for Random Forest 0.8716577540106952
Testing Precision for Random Forest 0.9854368932038835
Testing accuracy for Random Forest 0.8368506493506493
```

|   | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 1.6e+03 | 24 | 33 | 2 | 0 |
| 1 | 1.1e+02 | 1.6e+02 | 30 | 1 | 3 |
| 2 | 1.2e+02 | 26 | 1.4e+02 | 0 | 5 |
| 3 | 14 | 2 | 2 | 36 | 1 |
| 4 | 17 | 8 | 6 | 2 | 97 |

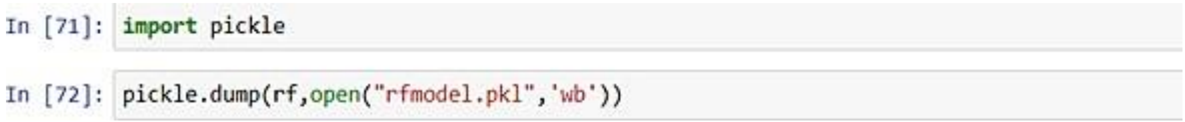


```
Testing Accuracy for Decision Tree 0.8849804578447794
Testing Sensitivity for Decision Tree 0.9400131839156229
Testing Specificity for Decision Tree 0.5802919708029197
Testing Precision for Decision Tree 0.9253731343283582
Testing accuracy for Decision Tree 0.7516233766233766
```

|   | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 1.4e+03 | 1.2e+02 | 1.2e+02 | 12 | 9 |
| 1 | 91 | 1.6e+02 | 44 | 8 | 6 |
| 2 | 81 | 39 | 1.5e+02 | 9 | 12 |
| 3 | 14 | 9 | 3 | 29 | 0 |
| 4 | 12 | 16 | 8 | 3 | 91 |

## Model Saving:

Random Forest gives the best accuracy then others , so we save random forest model usingpickle.



```
In [71]: import pickle

In [72]: pickle.dump(rf,open("rfmodel.pkl",'wb'))
```

## Conclusion:

In this sprint , we builded our model , evaluated and saved. In next sprint, we deploy our modelIBM cloud using IBM Watson and building Dashboard.

# SPRINT-2:

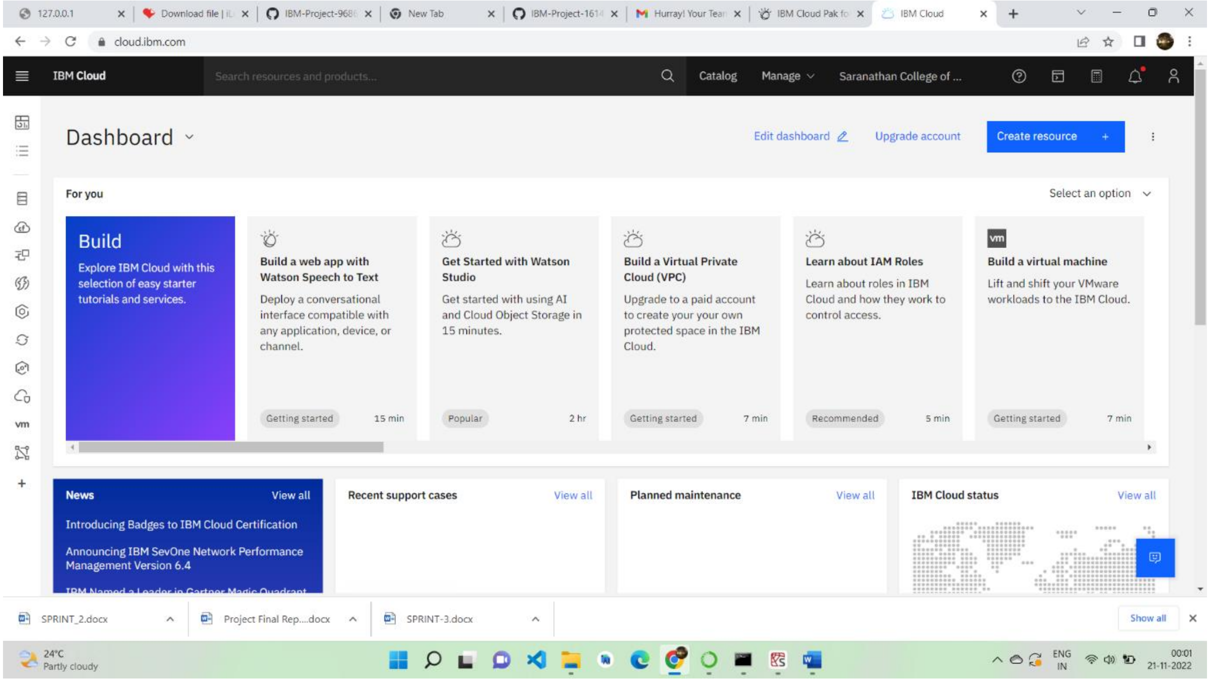
- **Creating IBM cloud account & Required Resources**
- **Deploy our model in IBM Watson**
- **Creating Dashboard using HTML/CSS**
- **Create web app and Hosting in falsk**
- **Testing web app**

## Creating IBM cloud account & Required Resources:

### Creating IBM cloud account:

Frist, need to create IBM Cloud account by using SI Mail Id and SI Password which is provided by IBMin profile.

Below dashboard of an account after created,



Dashboard

For you

Build

Explore IBM Cloud with this selection of easy starter tutorials and services.

Build a web app with Watson Speech to Text

Deploy a conversational interface compatible with any application, device, or channel.

Getting started 15 min

Get Started with Watson Studio

Get started with using AI and Cloud Object Storage in 15 minutes.

Popular 2 hr

Build a Virtual Private Cloud (VPC)

Upgrade to a paid account to create your your own protected space in the IBM Cloud.

Getting started 7 min

Learn about IAM Roles

Learn about roles in IBM Cloud and how they work to control access.

Recommended 5 min

Build a virtual machine

Lift and shift your VMware workloads to the IBM Cloud.

Getting started 7 min

News

Introducing Badges to IBM Cloud Certification

Announcing IBM SevOne Network Performance Management Version 6.4

IBM Named a Leader in Gartner Magic Quadrant

Recent support cases

Planned maintenance

IBM Cloud status

### Creating IBM Cloud Required Resources:

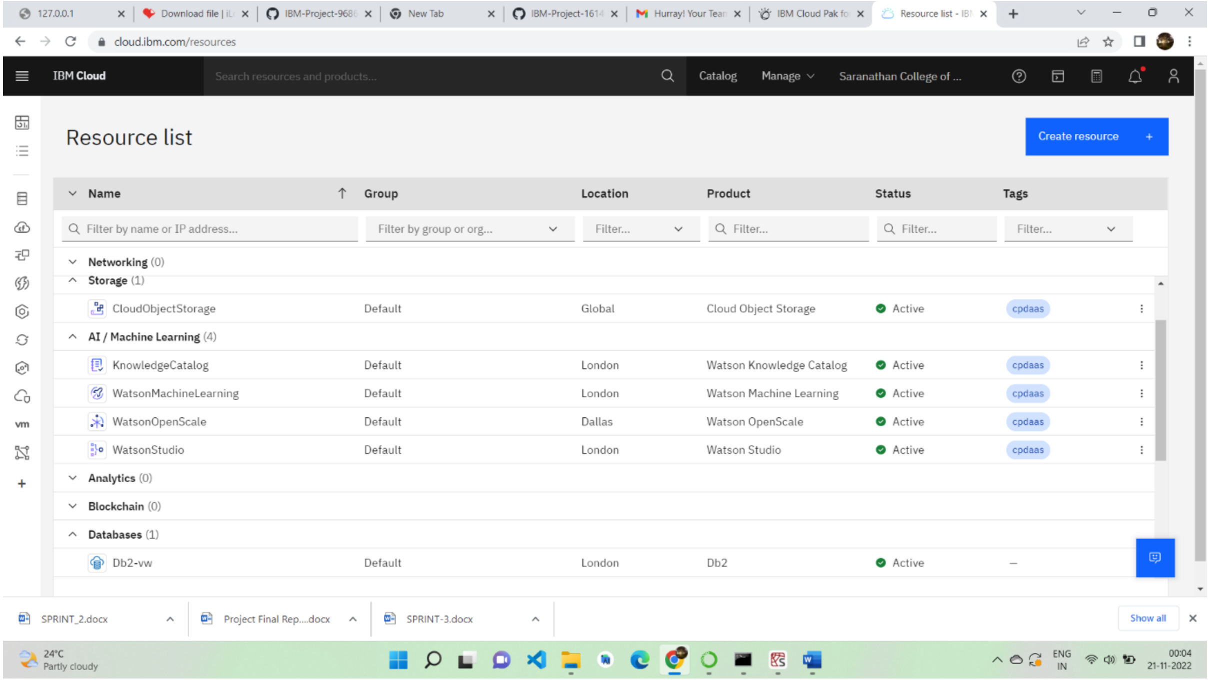
After creating IBM cloud account, to deploy ML model, need to create following resources such as,

Cloud Object Storage

Watson Machine Learning

Watson Studio

After created above resources Resource List of an account is displayed as follow,



Resource list

| Name | Group | Location | Product | Status | Tags |
|---|---|---|---|---|---|
| Networking (0) | | | | | |
| Storage (1) | | | | | |
| CloudObjectStorage | Default | Global | Cloud Object Storage | Active | cpdaas |
| AI / Machine Learning (4) | | | | | |
| KnowledgeCatalog | Default | London | Watson Knowledge Catalog | Active | cpdaas |
| WatsonMachineLearning | Default | London | Watson Machine Learning | Active | cpdaas |
| WatsonOpenScale | Default | Dallas | Watson OpenScale | Active | cpdaas |
| WatsonStudio | Default | London | Watson Studio | Active | cpdaas |
| Analytics (0) | | | | | |
| Blockchain (0) | | | | | |
| Databases (1) | | | | | |
| Db2-vw | Default | London | Db2 | Active | – |

All the resource are in active state.

All the required cloud resources are created successfully.

## Deploy our model in IBM Watson:

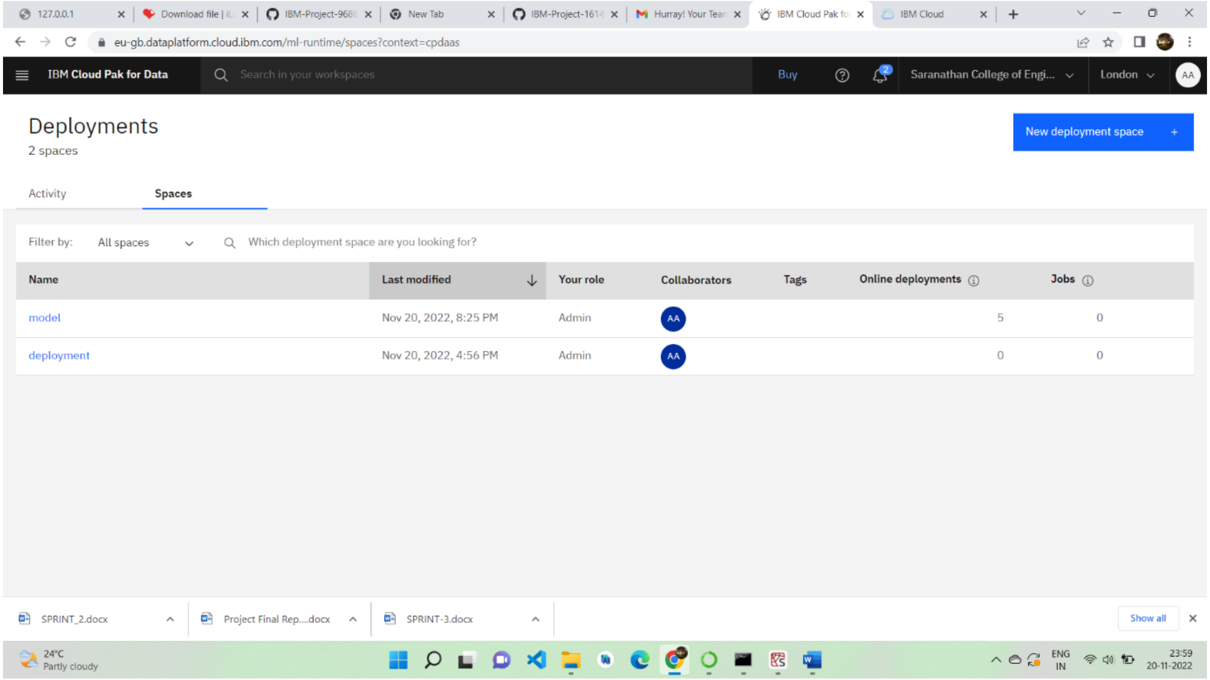
To deploy ML model in IBM cloud, need to create project in IBM Watson. After successful creation of project import .ipynb file of sprint-1 which ML models are build in Jupyter notebook.

Upload required datasets and import it.Deploy model using following code,

```
!pip install -U ibm-watson-machine-learning

from ibm_watson_machine_learning import APIClient

import json

import numpy as npwml_cred={

 "apikey":" IF-dWY9BFLrCPxwHr1kyXWh9AVRL4G4aDElRzF9TsYu6
", "url":"https://eu-gb.ml.cloud.ibm.com"

}

wml_clients=APIClient(wml_cred)wml_clients.spaces.list()

 space_id="6d7c1218-3aca-4256-be3d-d610732530b1" wml_clients.set.default_space(space_id)
 wml_clients.software_specifications.list(500) MODEL_NAME="randomforest"
 DEPLOYMENT_NAME="rf_deployment"

 DEMO_MODEL=rf

 soft_sepc_id=wml_clients.software_specifications.get_id_by_name("runtime-22.1-py3.9")


model_props={ wml_clients.repository.ModelMetaNames.NAME:MODEL_NAME,
   wml_clients.repository.ModelMetaNames.TYPE:"scikit-learn_1.0",
   wml_clients.repository.ModelMetaNames.SOFTWARE_SPEC_UID: soft_sepc_id


 }
```

NOTE: APIKey must need to create to deploy and connect API

After successful of deployment, deployed is appeared in Deployment section as follow,



Deployments

2 spaces

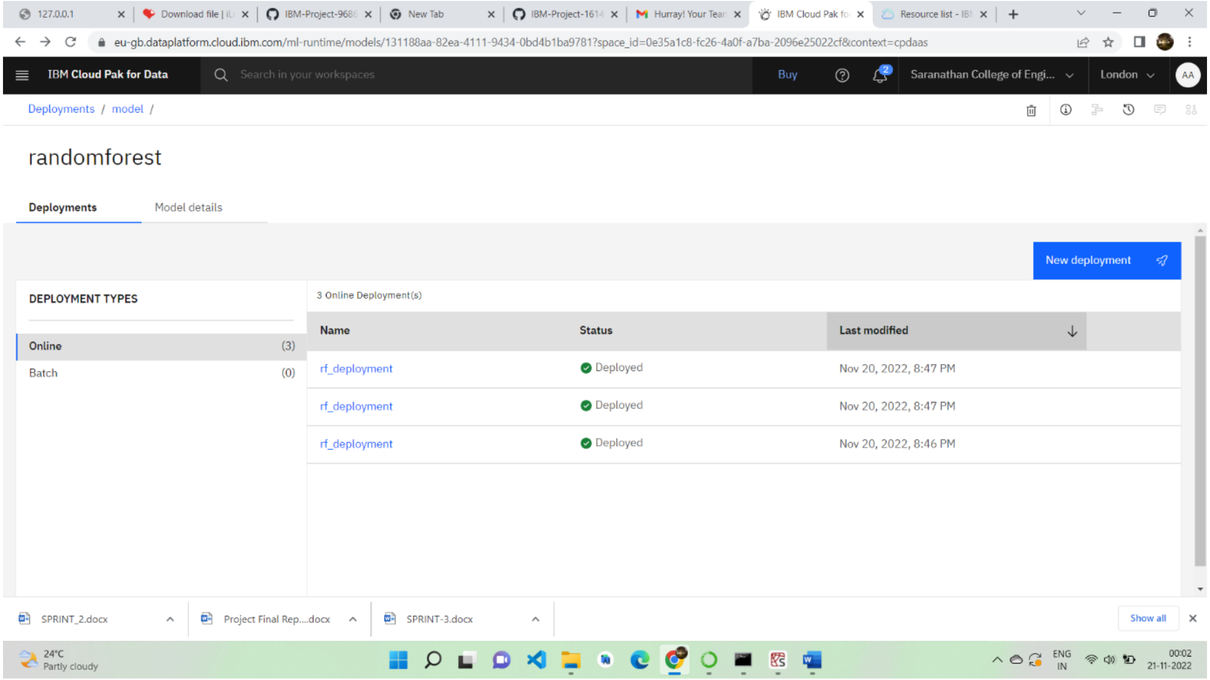
New deployment space +

Activity Spaces

Filter by: All spaces

Which deployment space are you looking for?

| Name | Last modified | Your role | Collaborators | Tags | Online deployments | Jobs |
|---|---|---|---|---|---|---|
| model | Nov 20, 2022, 8:25 PM | Admin | AA | | 5 | 0 |
| deployment | Nov 20, 2022, 4:56 PM | Admin | AA | | 0 | 0 |


127.0.0.1
Download file | iL
IBM-Project-968
New Tab
IBM-Project-1614
Hurray! Your Tear
IBM Cloud Pak fo
Resource list - IBI
eu-gb.dataplatform.cloud.ibm.com/ml-runtime/models/131188aa-82ea-4111-9434-0bd4b1ba9781?space_id=0e35a1c8-fc26-4a0f-a7ba-2096e25022cf&context=cpdaas
IBM Cloud Pak for Data
Search in your workspaces
Buy
Saranathan College of Engi...
London
AA
SPRINT_2.docx
Project Final Rep....docx
SPRINT-3.docx
Show all
24°C
Partly cloudy
ENG
IN
00:02
21-11-2022

Deployments / model /

randomforest

Deployments Model details

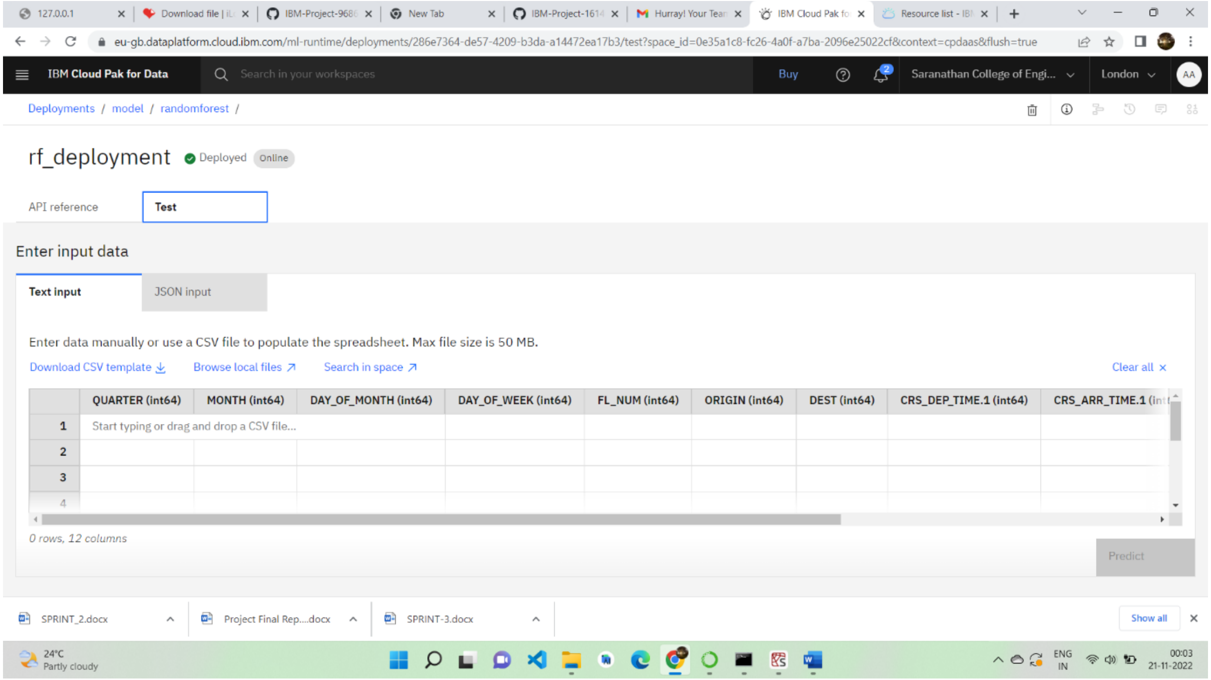
New deployment

DEPLOYMENT TYPES

| | |
|---|---|
| Online | (3) |
| Batch | (0) |

3 Online Deployment(s)

| Name | Status | Last modified |
|---|---|---|
| rf_deployment | ✓ Deployed | Nov 20, 2022, 8:47 PM |
| rf_deployment | ✓ Deployed | Nov 20, 2022, 8:47 PM |
| rf_deployment | ✓ Deployed | Nov 20, 2022, 8:46 PM |



Deployments / model / randomforest /

rf_deployment ● Deployed Online

API reference | Test

Enter input data

Text input | JSON input

Enter data manually or use a CSV file to populate the spreadsheet. Max file size is 50 MB.

Download CSV template | Browse local files | Search in space | Clear all

| | QUARTER (int64) | MONTH (int64) | DAY_OF_MONTH (int64) | DAY_OF_WEEK (int64) | FL_NUM (int64) | ORIGIN (int64) | DEST (int64) | CRS_DEP_TIME.1 (int64) | CRS_ARR_TIME.1 (int |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Start typing or drag and drop a CSV file... | | | | | | | | |
| 2 | | | | | | | | | |
| 3 | | | | | | | | | |
| 4 | | | | | | | | | |

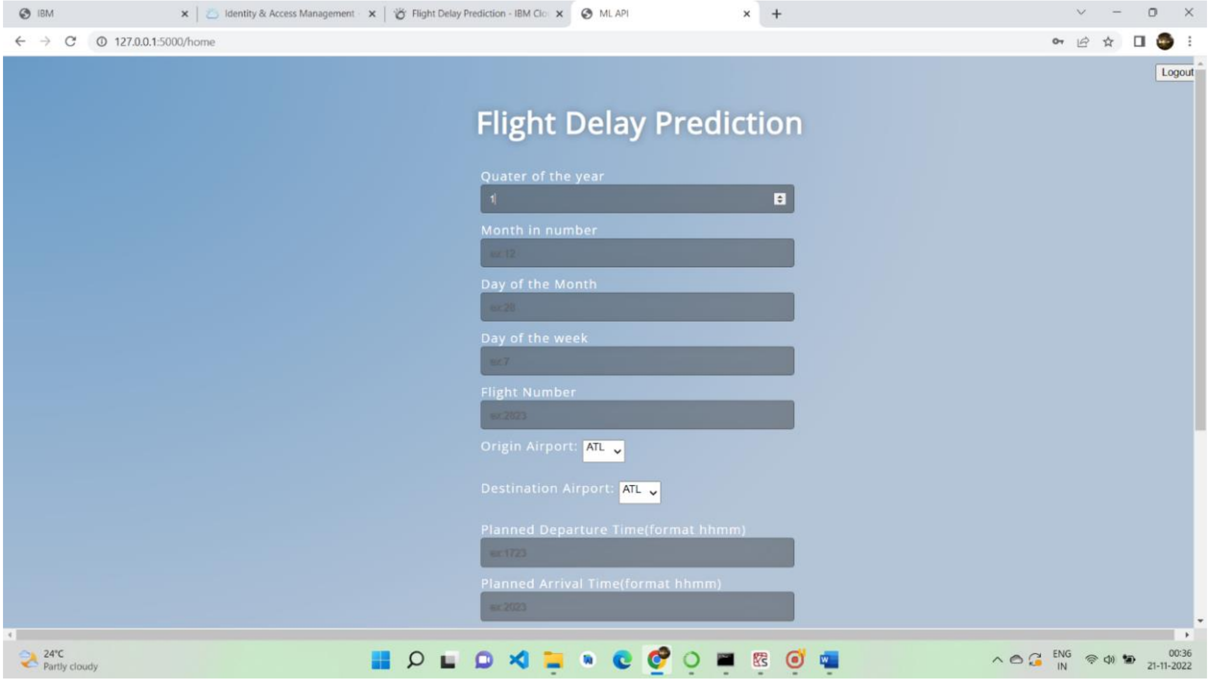
0 rows, 12 columns

Predict

After these, need to copy API requesting codes on required language(python).

## Creating Dashboard using HTML/CSS:

Frontend Dashboard is created using HTML/CSS,Result as web page like,



127.0.0.1:5000/home

Logout

Flight Delay Prediction

Quater of the year: 1

Month in number

Day of the Month

Day of the week

Flight Number

Origin Airport: ATL

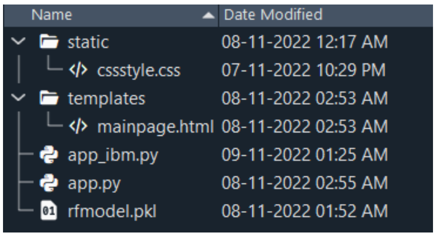
Destination Airport: ATL

Planned Departure Time(format hhmm)

Planned Arrival Time(format hhmm)

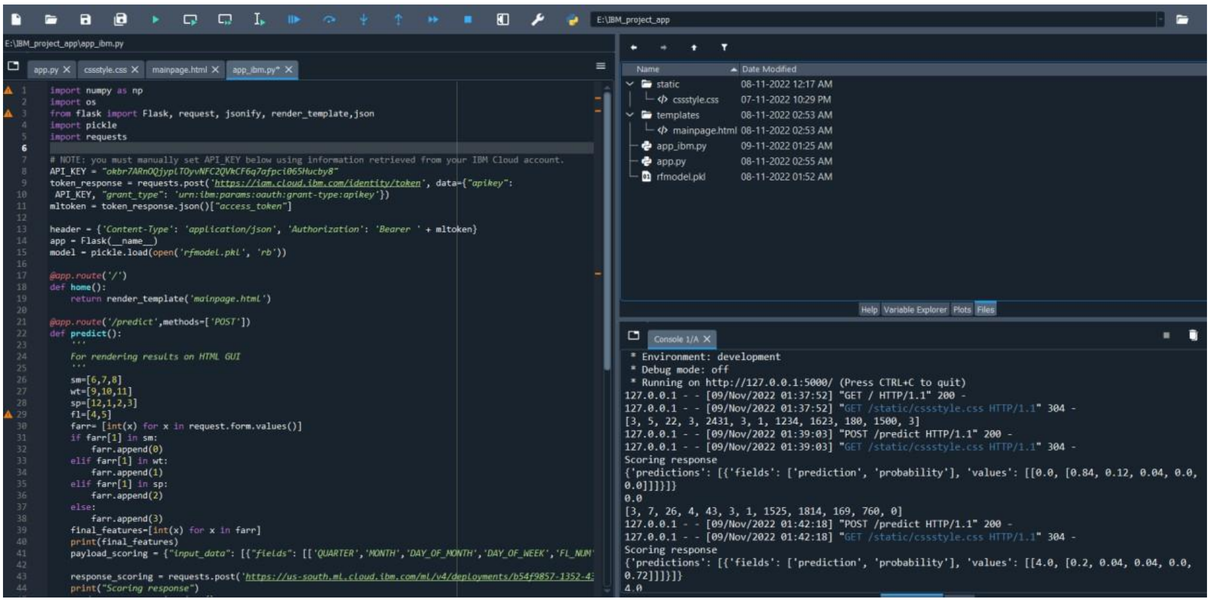
## Create web app and Hosting in falsk:

First thing, need to create directory as follow,



| Name | Date Modified |
|---|---|
| static | 08-11-2022 12:17 AM |
| cssstyle.css | 07-11-2022 10:29 PM |
| templates | 08-11-2022 02:53 AM |
| mainpage.html | 08-11-2022 02:53 AM |
| app_ibm.py | 09-11-2022 01:25 AM |
| app.py | 08-11-2022 02:55 AM |
| rfmodel.pkl | 08-11-2022 01:52 AM |

Then, code the required logic in app.py file with API connection , request and response code.Spyder IDE looks like,



```python
import numpy as np
import os
from flask import Flask, request, jsonify, render_template,json
import pickle
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "okbr7ARnOQjypLTOyvNFC2QVkCF6q7afpci065Hucby8"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
app = Flask(__name__)
model = pickle.load(open('rfmodel.pkl', 'rb'))

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

@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    sm=[6,7,8]
    wt=[9,10,11]
    sp=[12,1,2,3]
    fl=[4,5]
    farr= [int(x) for x in request.form.values()]
    if farr[1] in sm:
        farr.append(0)
    elif farr[1] in wt:
        farr.append(1)
    elif farr[1] in sp:
        farr.append(2)
    else:
        farr.append(3)
    final_features=[int(x) for x in farr]
    print(final_features)
    payload_scoring = {"input_data": [{"fields": [['QUARTER','MONTH','DAY_OF_MONTH','DAY_OF_WEEK','FL_NUM'

    response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/b54f9857-1352-4:
    print("Scoring response")
```

```
Name                  Date Modified
static                08-11-2022 12:17 AM
  cssstyle.css        07-11-2022 10:29 PM
templates             08-11-2022 02:53 AM
  mainpage.html       08-11-2022 02:53 AM
app_ibm.py            09-11-2022 01:25 AM
app.py                08-11-2022 02:55 AM
rfmodel.pkl           08-11-2022 01:52 AM
```

```
 * Environment: development
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [09/Nov/2022 01:37:52] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Nov/2022 01:37:52] "GET /static/cssstyle.css HTTP/1.1" 304 -
[3, 5, 22, 3, 2431, 3, 1, 1234, 1623, 180, 1500, 3]
127.0.0.1 - - [09/Nov/2022 01:39:03] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [09/Nov/2022 01:39:03] "GET /static/cssstyle.css HTTP/1.1" 304 -
Scoring response
{'predictions': [{'fields': ['prediction', 'probability'], 'values': [[0.0, [0.84, 0.12, 0.04, 0.0,
0.0]]]}]}
0.0
[3, 7, 26, 4, 43, 3, 1, 1525, 1814, 169, 760, 0]
127.0.0.1 - - [09/Nov/2022 01:42:18] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [09/Nov/2022 01:42:18] "GET /static/cssstyle.css HTTP/1.1" 304 -
Scoring response
{'predictions': [{'fields': ['prediction', 'probability'], 'values': [[4.0, [0.2, 0.04, 0.04, 0.0,
0.72]]]}]}
4.0
```

Run the app.py file.

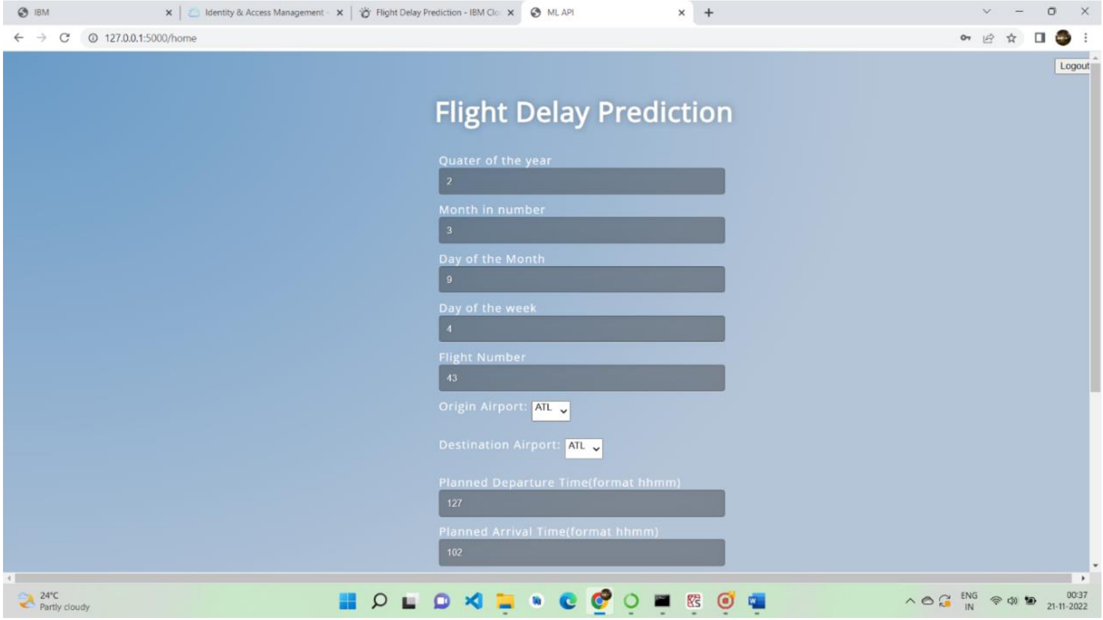
Localhost url is displayed in console, copy and paste in browser then search it , frond end HTML?CSS page is displayed. Successfully created and hosted web app in flask.

If any error caused as flask in production mode, thenSet FLASK_ENV=Development,

Then run the app

## VIII.Testing web app

Enter the data on the required fields,



Flight Delay Prediction

Quater of the year: 2

Month in number: 3

Day of the Month: 9

Day of the week: 4

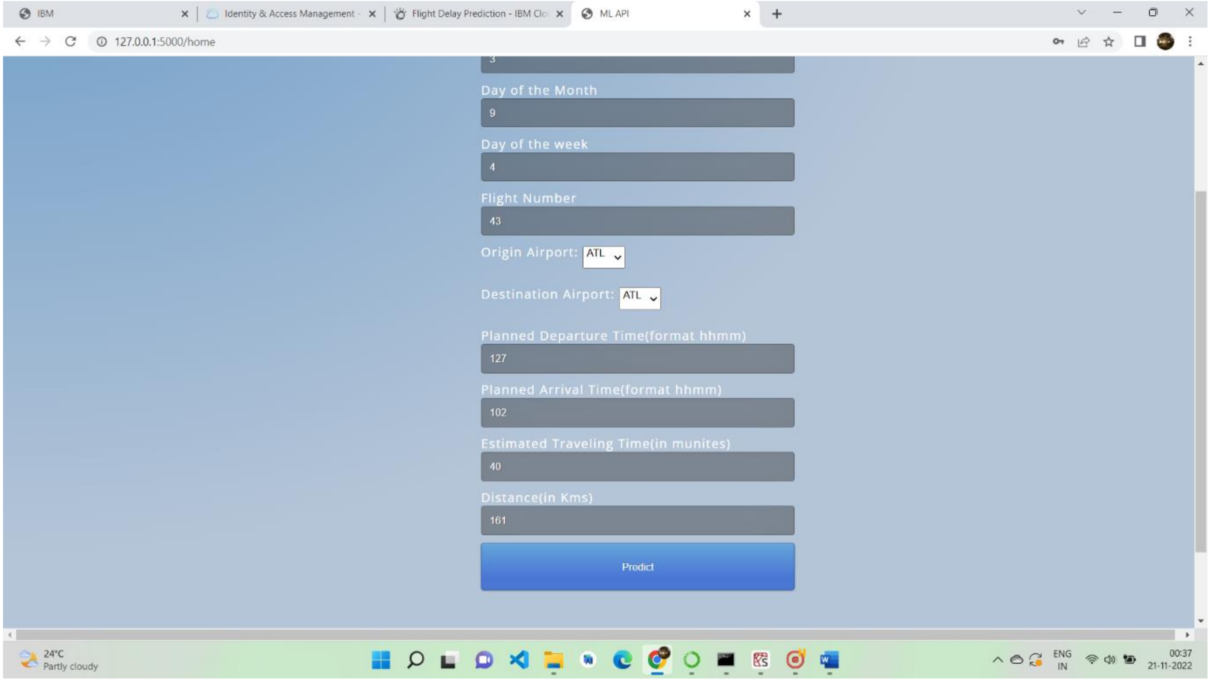
Flight Number: 43

Origin Airport: ATL

Destination Airport: ATL

Planned Departure Time(format hhmm): 127

Planned Arrival Time(format hhmm): 102



3

Day of the Month
9

Day of the week
4

Flight Number
43

Origin Airport: ATL

Destination Airport: ATL

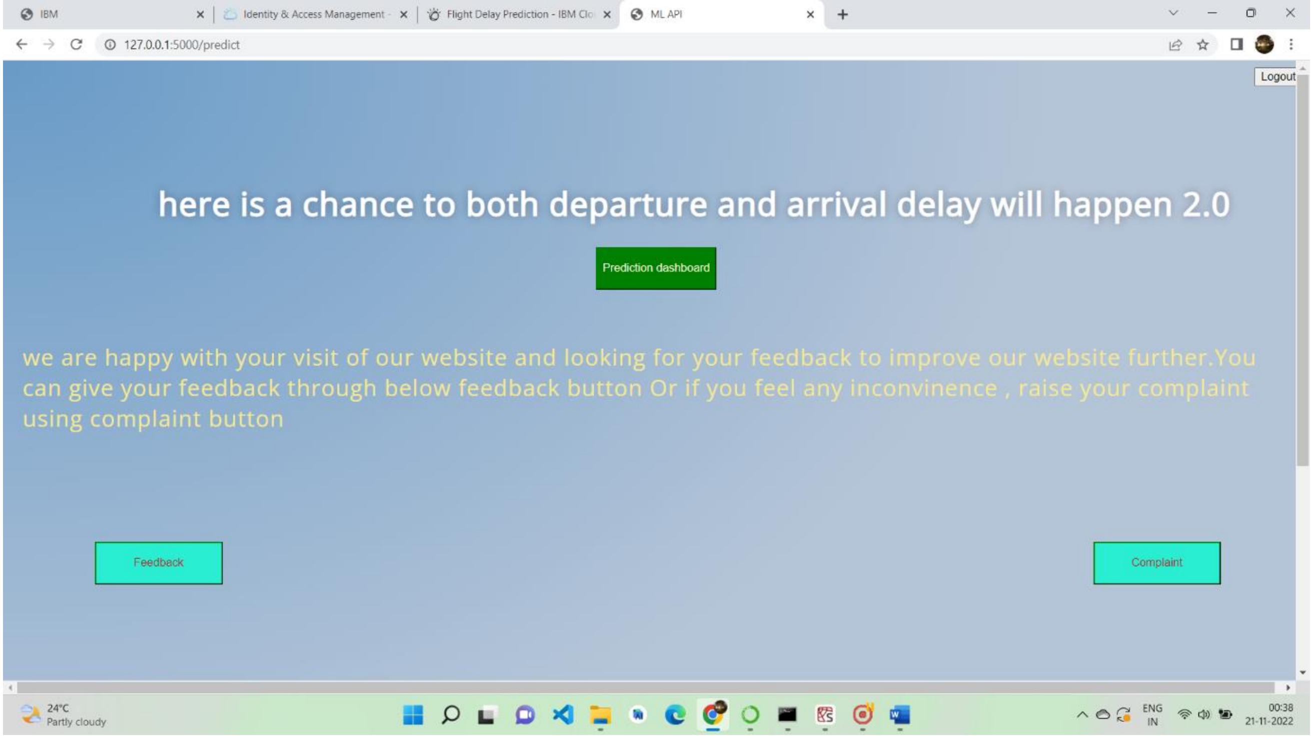
Planned Departure Time(format hhmm)
127

Planned Arrival Time(format hhmm)
102

Estimated Traveling Time(in munites)
40

Distance(in Kms)
161

Output is predicted by ML model successfully.



here is a chance to both departure and arrival delay will happen 2.0

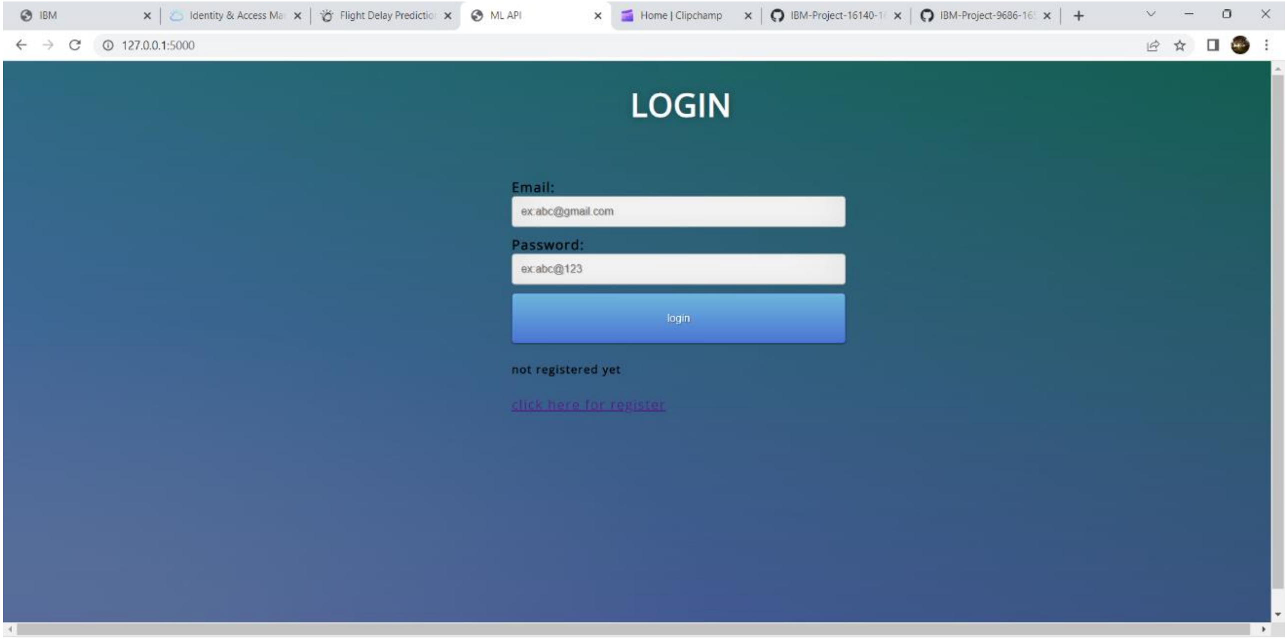
we are happy with your visit of our website and looking for your feedback to improve our website further.You can give your feedback through below feedback button Or if you feel any inconvinence , raise your complaint using complaint button

## SPRINT-3:

- **Build templates of login and register pages**
- **Connect templates and login, register operation in flaskTest run the app**

## Build templates of login and register pages:

Using, HTML/CSS , Login and register templates are build



LOGIN

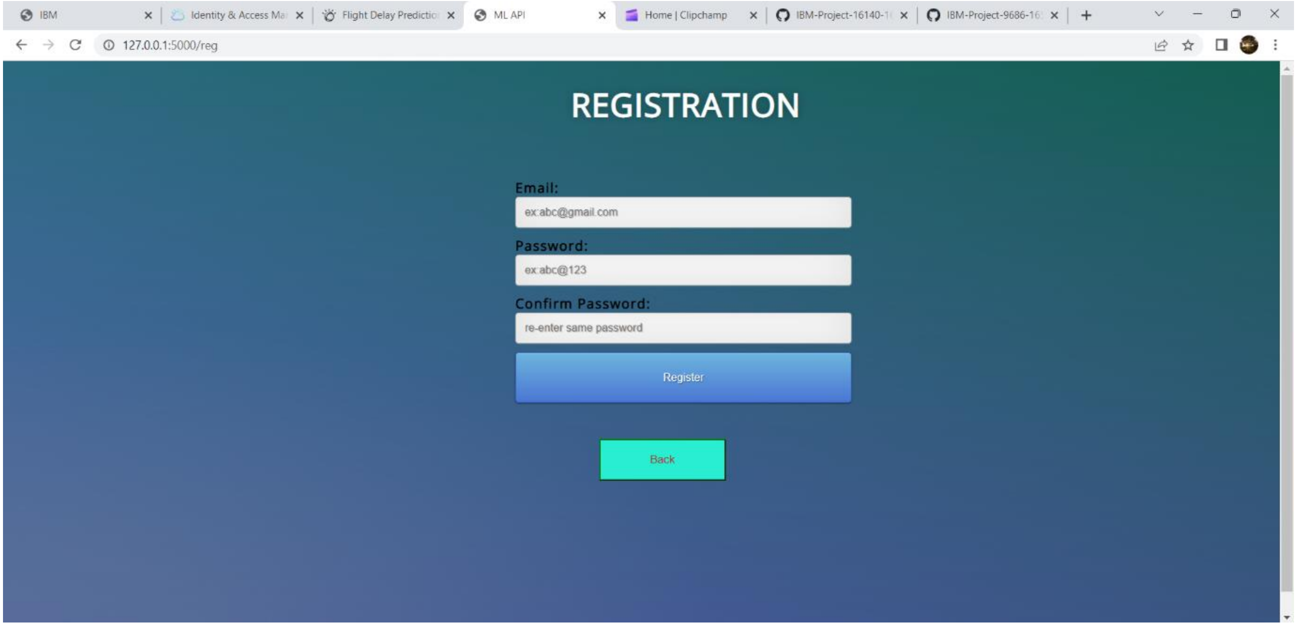
Email: ex:abc@gmail.com

Password: ex:abc@123

not registered yet

click here for register

Above picture shows the login page of this app.



REGISTRATION

Email: ex:abc@gmail.com

Password: ex:abc@123

Confirm Password: re-enter same password

Above picture is register page of web app.

## Connect templates and login, register operation in flask:

There is a need to connect database for store and retrieve user details. In this projectSQlite3 database is used to store and retrieve user details.

Login code as follow:

```
@app.route('/')def main():
return render_template('login.html')
@app.route('/login',methods=['POST','GET'])def login():
if request.method=='POST':try:
fv=[x for x in request.form.values()] email=request.form["email"] pswd=request.form["pswd"]
conn=sqlite3.connect("database1.db")cur=conn.cursor()
cur.execute("SELECT password FROM login WHERE email=?;",(str(email),))result=cur.fetchone()
cur.execute("SELECT * FROM login")if result:
if result[0]==pswd:
flash("login successfully",'success')return redirect('/home')
else:
return render_template("login.html", error="please enter correct password")
else:
print("register")
flash("please Register",'danger')
return redirect('/reg')
except Exception as e:return "hello error"
```

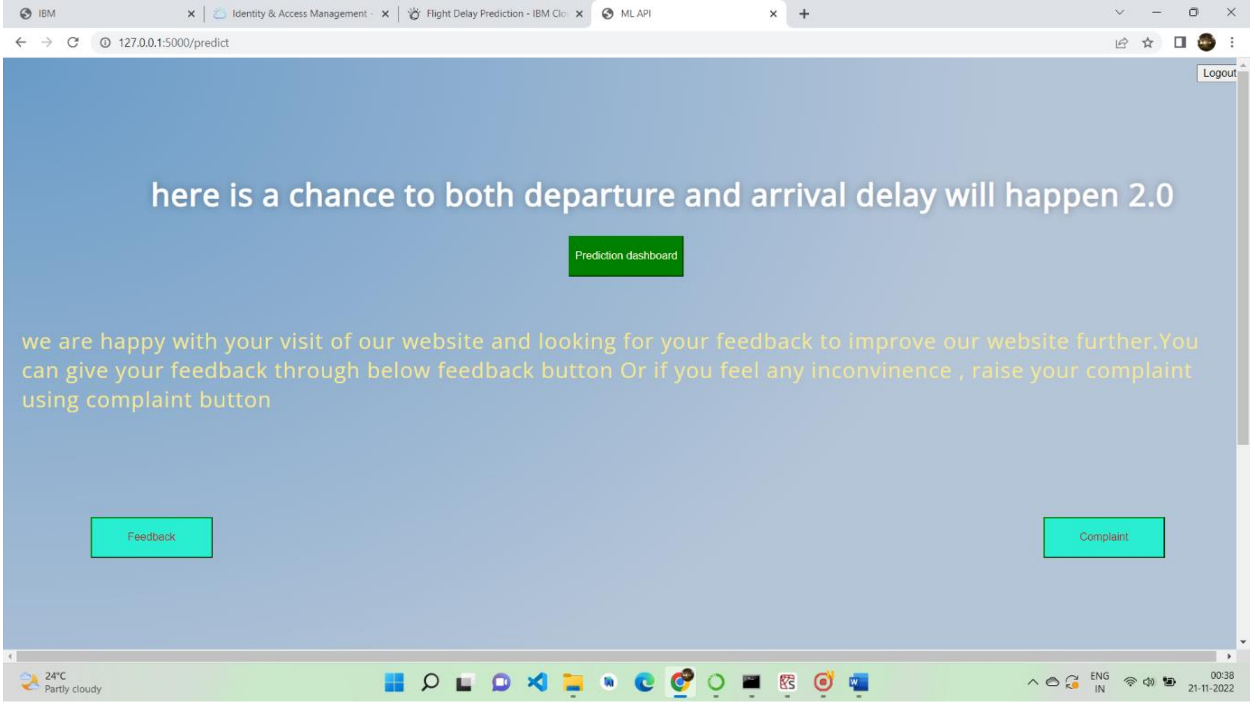
Register code as follow:

```
.route('/reg')def reg():
return render_template("register.html")
@app.route('/register',methods=['POST','GET']) def register():
if request.method=='POST':try:
fv=[x for x in request.form.values()] email=request.form["email"] pswd=request.form["pswd"]
conn=sqlite3.connect("database1.db")cur=conn.cursor()
cur.execute("SELECT * FROM login WHERE email=?;",(str(email),))result=cur.fetchone()
if result:
flash("user already exist,please login",'danger')return redirect('/')
else:
cur.execute("INSERT INTO  login(email,password)values(?,?)",(str(email),str(pswd)))
conn.commit()
cur.execute("SELECT * FROM login") flash("Registered successfully",'success')return
render_template('login.html')
except Exception as e: print(e) #flash(e,'danger') return "hello error1"
```

By using above code, successfully connect login and register pages to app successfully.

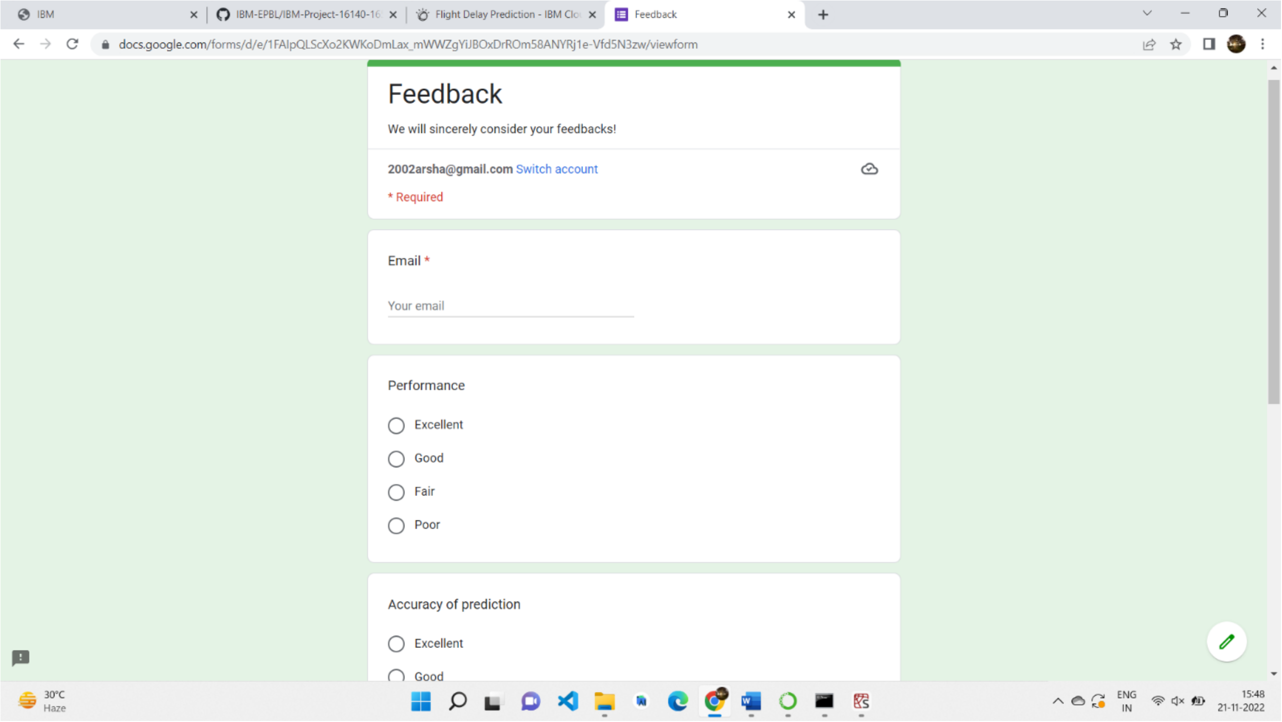
# SPRINT-4

Result page added,



here is a chance to both departure and arrival delay will happen 2.0

we are happy with your visit of our website and looking for your feedback to improve our website further.You can give your feedback through below feedback button Or if you feel any inconvinence , raise your complaint using complaint button


IBM
IBM-EPBL/IBM-Project-16140-16
Flight Delay Prediction - IBM Clo
Feedback
docs.google.com/forms/d/e/1FAlpQLScXo2KWKoDmLax_mWWZgYiJBOxDrROm58ANYRj1e-Vfd5N3zw/viewform
30°C
Haze
ENG
IN
15:48
21-11-2022

## Feedback

We will sincerely consider your feedbacks!

2002arsha@gmail.com Switch account

* Required

Email *

Your email

Performance

○ Excellent

○ Good

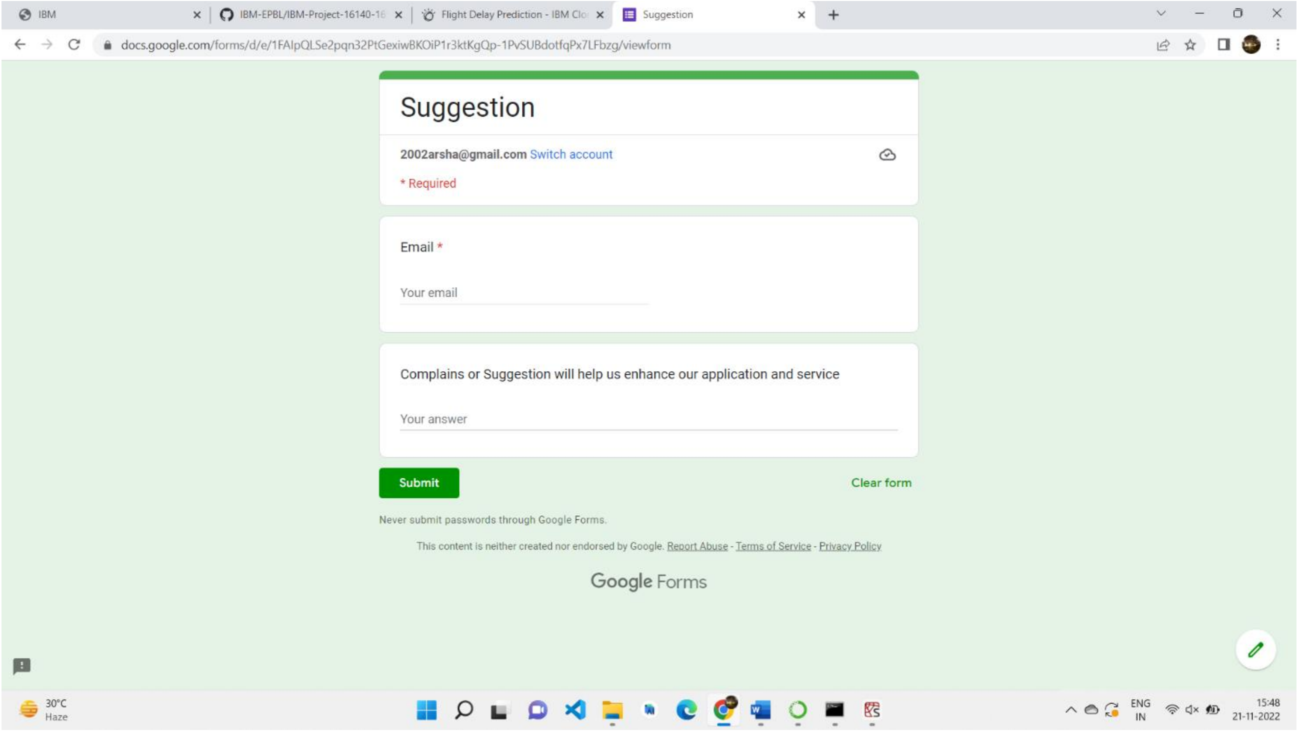
○ Fair

○ Poor

Accuracy of prediction

○ Excellent

○ Good



IBM
IBM-EPBL/IBM-Project-16140-16
Flight Delay Prediction - IBM Clo
Suggestion
docs.google.com/forms/d/e/1FAIpQLSe2pqn32PtGexiwBKOiP1r3ktKgQp-1PvSUBdotfqPx7LFbzg/viewform

Suggestion

2002arsha@gmail.com Switch account

* Required

Email *

Your email

Complains or Suggestion will help us enhance our application and service

Your answer

Submit

Clear form

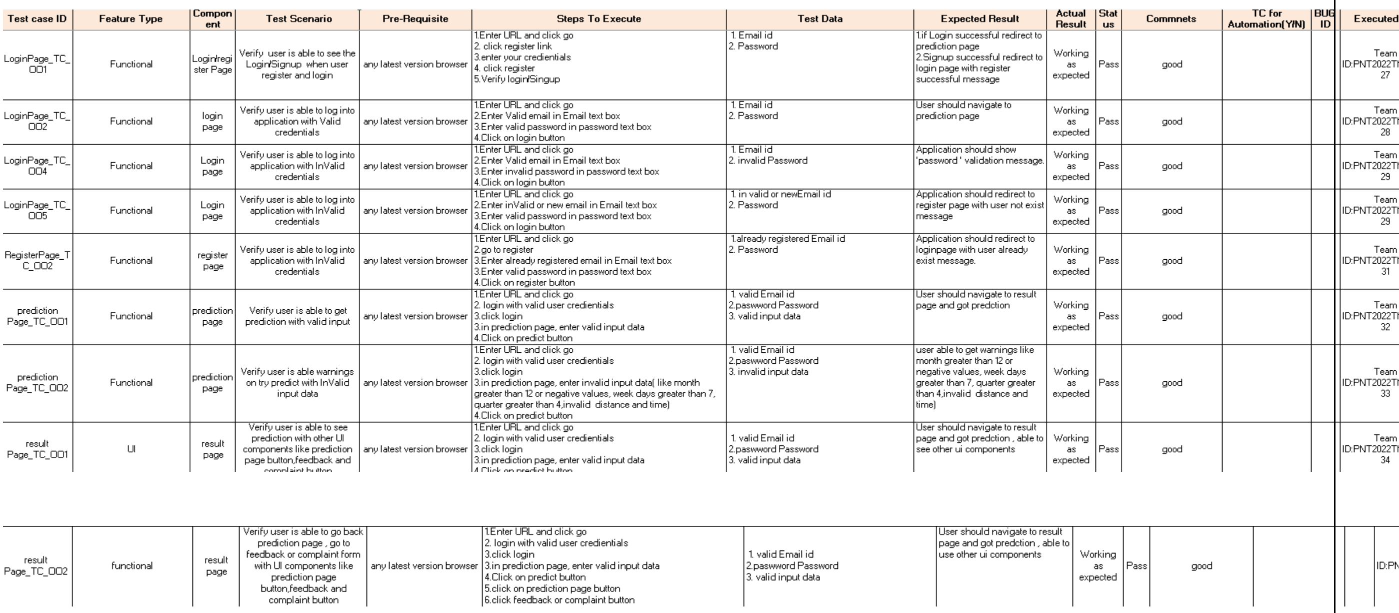
Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Privacy Policy

Google Forms

30°C
Haze
ENG
IN
15:48
21-11-2022

# 1.Test Cases



| Test case ID | Feature Type | Component | Test Scenario | Pre-Requisite | Steps To Execute | Test Data | Expected Result | Actual Result | Status | Commnets | TC for Automation(Y/N) | BUG ID | Executed |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LoginPage_TC_001 | Functional | Login/register Page | Verify user is able to see the Login/Signup when user register and login | any latest version browser | 1.Enter URL and click go<br>2. click register link<br>3.enter your credientials<br>4. click register<br>5.Verify login/Singup | 1. Email id<br>2. Password | 1.if Login successful redirect to prediction page<br>2.Signup successful redirect to login page with register successful message | Working as expected | Pass | good | | | Team ID:PNT2022TI 27 |
| LoginPage_TC_002 | Functional | login page | Verify user is able to log into application with Valid credentials | any latest version browser | 1.Enter URL and click go<br>2.Enter Valid email in Email text box<br>3.Enter valid password in password text box<br>4.Click on login button | 1. Email id<br>2. Password | User should navigate to prediction page | Working as expected | Pass | good | | | Team ID:PNT2022TI 28 |
| LoginPage_TC_004 | Functional | Login page | Verify user is able to log into application with InValid credentials | any latest version browser | 1.Enter URL and click go<br>2.Enter Valid email in Email text box<br>3.Enter invalid password in password text box<br>4.Click on login button | 1. Email id<br>2. invalid Password | Application should show 'password ' validation message. | Working as expected | Pass | good | | | Team ID:PNT2022TI 29 |
| LoginPage_TC_005 | Functional | Login page | Verify user is able to log into application with InValid credentials | any latest version browser | 1.Enter URL and click go<br>2.Enter inValid or new email in Email text box<br>3.Enter valid password in password text box<br>4.Click on login button | 1. in valid or newEmail id<br>2. Password | Application should redirect to register page with user not exist message | Working as expected | Pass | good | | | Team ID:PNT2022TI 29 |
| RegisterPage_TC_002 | Functional | register page | Verify user is able to log into application with InValid credentials | any latest version browser | 1.Enter URL and click go<br>2.go to register<br>3.Enter already registered email in Email text box<br>3.Enter valid password in password text box<br>4.Click on register button | 1.already registered Email id<br>2. Password | Application should redirect to loginpage with user already exist message. | Working as expected | Pass | good | | | Team ID:PNT2022TI 31 |
| prediction Page_TC_001 | Functional | prediction page | Verify user is able to get prediction with valid input | any latest version browser | 1.Enter URL and click go<br>2. login with valid user credientials<br>3.click login<br>3.in prediction page, enter valid input data<br>4.Click on predict button | 1. valid Email id<br>2.paswword Password<br>3. valid input data | User should navigate to result page and got predction | Working as expected | Pass | good | | | Team ID:PNT2022TI 32 |
| prediction Page_TC_002 | Functional | prediction page | Verify user is able warnings on try predict with InValid input data | any latest version browser | 1.Enter URL and click go<br>2. login with valid user credientials<br>3.click login<br>3.in prediction page, enter invalid input data( like month greater than 12 or negative values, week days greater than 7, quarter greater than 4,invalid distance and time)<br>4.Click on predict button | 1. valid Email id<br>2.paswword Password<br>3. invalid input data | user able to get warnings like month greater than 12 or negative values, week days greater than 7, quarter greater than 4,invalid distance and time) | Working as expected | Pass | good | | | Team ID:PNT2022TI 33 |
| result Page_TC_001 | UI | result page | Verify user is able to see prediction with other UI components like prediction page button,feedback and complaint button | any latest version browser | 1.Enter URL and click go<br>2. login with valid user credientials<br>3.click login<br>3.in prediction page, enter valid input data<br>4.Click on predict button | 1. valid Email id<br>2.paswword Password<br>3. valid input data | User should navigate to result page and got predction , able to see other ui components | Working as expected | Pass | good | | | Team ID:PNT2022TI 34 |
| result Page_TC_002 | functional | result page | Verify user is able to go back prediction page , go to feedback or complaint form with UI components like prediction page button,feedback and complaint button | any latest version browser | 1.Enter URL and click go<br>2. login with valid user credientials<br>3.click login<br>3.in prediction page, enter valid input data<br>4.Click on predict button<br>5.click on prediction page button<br>6.click feedback or complaint button | 1. valid Email id<br>2.paswword Password<br>3. valid input data | User should navigate to result page and got predction , able to use other ui components | Working as expected | Pass | good | | | ID:PN |

## 2.User Acceptance Test

### 1.Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Developing a flight delay prediction model using machine learning project at the time of the release to User Acceptance Testing (UAT).

### 2.Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and howthey were resolved

| Resolution | Severity 1 | Severity 2 | Severity 3 | Severity 4 | Subtotal |
|---|---|---|---|---|---|
| By Design | 8 | 4 | 2 | 3 | 15 |
| Duplicate | 0 | 0 | 0 | 0 | 0 |
| External | 2 | 1 | 0 | 1 | 4 |
| Fixed | 15 | 10 | 4 | 5 | 34 |
| Not Reproduced | 0 | 0 | 0 | 0 | 0 |
| Skipped | 0 | 0 | 0 | 0 | 0 |
| Won't Fix | 0 | 0 | 0 | 0 | 0 |
| Totals | 23 | 15 | 6 | 9 | 53 |

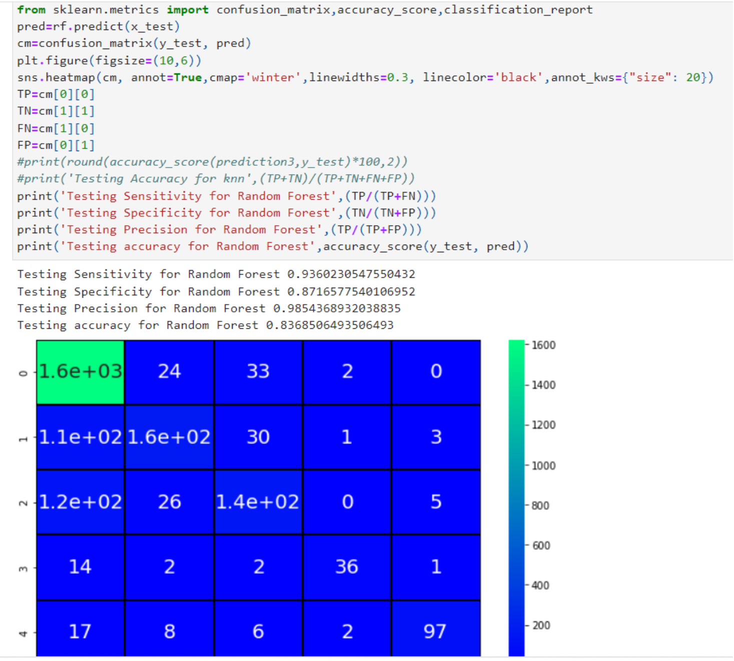
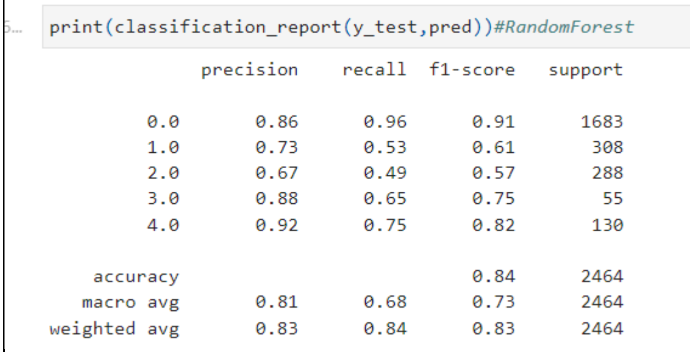
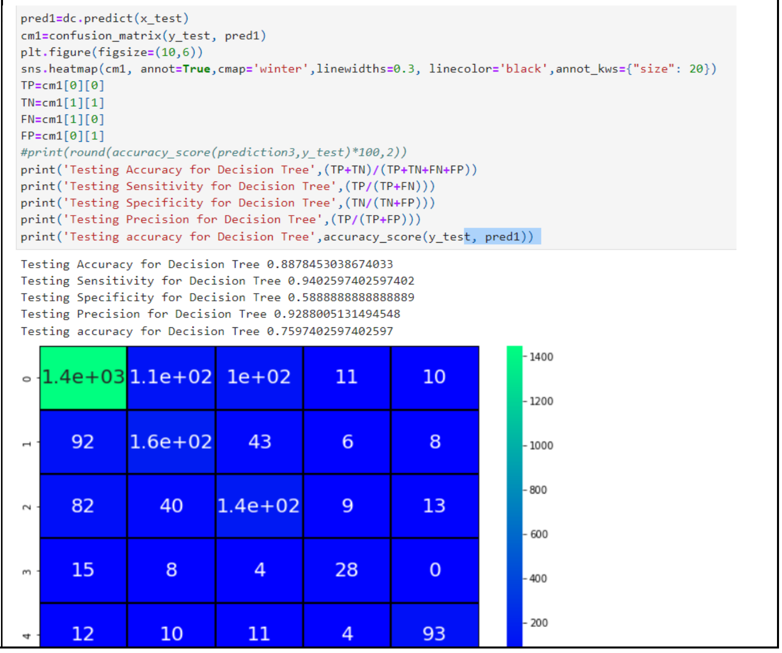
### 3.Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

| Section | Total Cases | Not Tested | Fail | Pass |
|---|---|---|---|---|
| Client Application | 12 | 0 | 0 | 12 |
| Security | 2 | 0 | 0 | 2 |
| Exception Reporting | 6 | 0 | 0 | 6 |
| Final Report Output | 4 | 0 | 0 | 4 |

# IX. Results

## 1.Performance Metrics

| | | | |
|---|---|---|---|
| 1 | **Metrics** | **Classification Model: Confusion Matrix –**<br>**Accuray Score-**<br>**& Classification Repor** | <br><br> |

```
from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
pred=rf.predict(x_test)
cm=confusion_matrix(y_test, pred)
plt.figure(figsize=(10,6))
sns.heatmap(cm, annot=True,cmap='winter',linewidths=0.3, linecolor='black',annot_kws={"size": 20})
TP=cm[0][0]
TN=cm[1][1]
FN=cm[1][0]
FP=cm[0][1]
#print(round(accuracy_score(prediction3,y_test)*100,2))
#print('Testing Accuracy for knn',(TP+TN)/(TP+TN+FN+FP))
print('Testing Sensitivity for Random Forest',(TP/(TP+FN)))
print('Testing Specificity for Random Forest',(TN/(TN+FP)))
print('Testing Precision for Random Forest',(TP/(TP+FP)))
print('Testing accuracy for Random Forest',accuracy_score(y_test, pred))
```

```
Testing Sensitivity for Random Forest 0.9360230547550432
Testing Specificity for Random Forest 0.8716577540106952
Testing Precision for Random Forest 0.9854368932038835
Testing accuracy for Random Forest 0.8368506493506493
```

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1.6e+03 | 24 | 33 | 2 | 0 |
| 1 | 1.1e+02 | 1.6e+02 | 30 | 1 | 3 |
| 2 | 1.2e+02 | 26 | 1.4e+02 | 0 | 5 |
| 3 | 14 | 2 | 2 | 36 | 1 |
| 4 | 17 | 8 | 6 | 2 | 97 |

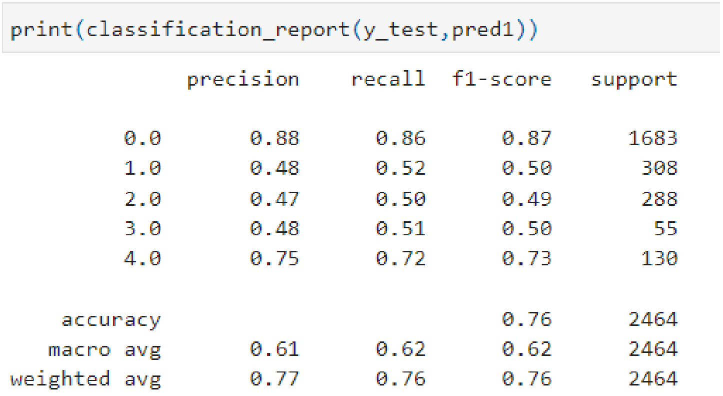
```
print(classification_report(y_test,pred))#RandomForest
```

```
              precision    recall  f1-score   support

         0.0       0.86      0.96      0.91      1683
         1.0       0.73      0.53      0.61       308
         2.0       0.67      0.49      0.57       288
         3.0       0.88      0.65      0.75        55
         4.0       0.92      0.75      0.82       130

    accuracy                           0.84      2464
   macro avg       0.81      0.68      0.73      2464
weighted avg       0.83      0.84      0.83      2464
```

```
pred1=dc.predict(x_test)
cm1=confusion_matrix(y_test, pred1)
plt.figure(figsize=(10,6))
sns.heatmap(cm1, annot=True,cmap='winter',linewidths=0.3, linecolor='black',annot_kws={"size": 20})
TP=cm1[0][0]
TN=cm1[1][1]
FN=cm1[1][0]
FP=cm1[0][1]
#print(round(accuracy_score(prediction3,y_test)*100,2))
print('Testing Accuracy for Decision Tree',(TP+TN)/(TP+TN+FN+FP))
print('Testing Sensitivity for Decision Tree',(TP/(TP+FN)))
print('Testing Specificity for Decision Tree',(TN/(TN+FP)))
print('Testing Precision for Decision Tree',(TP/(TP+FP)))
print('Testing accuracy for Decision Tree',accuracy_score(y_test, pred1))
```

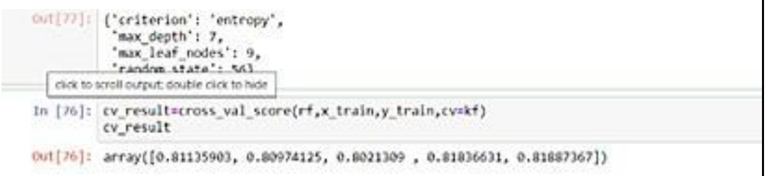
```
Testing Accuracy for Decision Tree 0.8878453038674033
Testing Sensitivity for Decision Tree 0.9402597402597402
Testing Specificity for Decision Tree 0.5888888888888889
Testing Precision for Decision Tree 0.9288005131494548
Testing accuracy for Decision Tree 0.7597402597402597
```

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1.4e+03 | 1.1e+02 | 1e+02 | 11 | 10 |
| 1 | 92 | 1.6e+02 | 43 | 6 | 8 |
| 2 | 82 | 40 | 1.4e+02 | 9 | 13 |
| 3 | 15 | 8 | 4 | 28 | 0 |
| 4 | 12 | 10 | 11 | 4 | 93 |

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>

<pre>
print(classification_report(y_test,pred1))

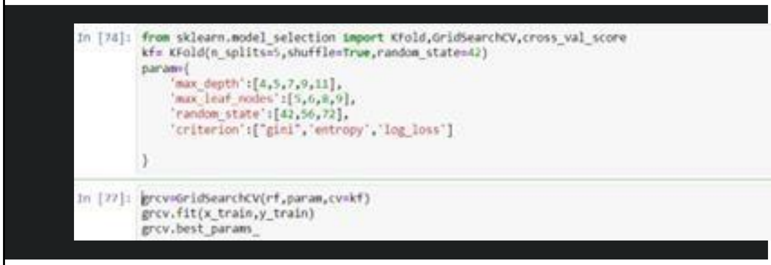
              precision    recall  f1-score   support

         0.0       0.88      0.86      0.87      1683
         1.0       0.48      0.52      0.50       308
         2.0       0.47      0.50      0.49       288
         3.0       0.48      0.51      0.50        55
         4.0       0.75      0.72      0.73       130

    accuracy                           0.76      2464
   macro avg       0.61      0.62      0.62      2464
weighted avg       0.77      0.76      0.76      2464
</pre>
</td>
</tr>
<tr>
<td><b>2</b></td>
<td>Tune the Model</td>
<td><b>Hyperparameter Tuning - Validation Method -</b></td>
<td>

<pre>
Out[77]: {'criterion': 'entropy',
          'max_depth': 7,
          'max_leaf_nodes': 9,
          'random_state': 56}

In [76]: cv_result=cross_val_score(rf,x_train,y_train,cv=kf)
         cv_result

Out[76]: array([0.81135903, 0.80974125, 0.8021309 , 0.81836631, 0.81887367])
</pre>


<pre>
In [74]: from sklearn.model_selection import Kfold,GridSearchCV,cross_val_score
         kf= KFold(n_splits=5,shuffle=True,random_state=42)
         param={
             'max_depth':[4,5,7,9,11],
             'max_leaf_nodes':[5,6,8,9],
             'random_state':[42,56,72],
             'criterion':["gini",'entropy','log_loss']

         }

In [77]: grcv=GridSearchCV(rf,param,cv=kf)
         grcv.fit(x_train,y_train)
         grcv.best_params_
</pre>
</td>
</tr>
</table>

## 2.Advantages & disadvantages

### Advantages:

- Early known of delay will reduce the tension, pressure of customer, airlinesand airport authorities.
- By early known of delay, we can reduce or avoid financial loss
- We can change our plan according to the practical scenario, if we know thedelay earlier
- Airlines and Airport Authorities make prevention techniques, find the sourceof the problem
- Airlines can inform their passengers to prepare according to that. This savesname and fame the airlines among the modern society

### Disadvantage:

- Delay due to unexpected climate change , war, natural disaster.. etc can’t predict exactly

- Need more data and analyse

## 3.Conclusion

Successfully developed the flight delay prediction model using random forest algorithm which gives best accuracy. By knowing delay earlier, many chance to avoid, so many problems and issues. This prediction gives the waring only to makeprepared us.

## 4.Future Scope

Add the feature to track the flight on real time and make prediction using realtime data by connecting to the private API which gives real time data

## 5.Appendix

## Source code

### Mainpage.html

```
<!DOCTYPE html>
<html >

<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet'
type='text/css'>
<link rel="stylesheet" href="{{url_for('static',filename='cssstyle.css')}}">

</head>

<body style = "background-color:darkgreen;">
<div class="logout" style="display : flex;
flex-direction: row-reverse";
>
<a href="{{ url_for('main')}}"><button>Logout</button></a>
</div>
 <h1 class="h1">Flight Delay Prediction</h1>

 <div class="login">


     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}" method="post">
```

```
     Quater of the year <input type="number" name="quater" placeholder="ex:3"
required="required" min='1' max='4' />
     Month in number<input type="number" name="month" placeholder="ex:12"
required="required" min='1' max='12'/>
          Day of the Month<input type="number" name="daym" placeholder="ex:28"
required="required" min='1' max='31'/>
          Day of the week<input type="number" name="dayw" placeholder="ex:7"
required="required" min='1' max='7'/>
     Flight Number<input type="number" name="fnum" placeholder="ex:2823"
required="required" max="9999"/>

          Origin Airport: <select name="airport" class="origin">
    <option value='1'>ATL</option>
    <option value='2'>DWT</option>
    <option value='3'>JFK</option>
    <option value='4'>MSP</option>
    <option value='5'>SEA</option>
         </select><br><br>
         Destination Airport: <select name="airportd" class="dest">
    <option value='1'>ATL</option>
    <option value='2'>DWT</option>
    <option value='3'>JFK</option>
    <option value='4'>MSP</option>
    <option value='5'>SEA</option>
         </select><br><br>
         Planned Departure Time(format hhmm)<input type="number" name="dtime"
placeholder="ex:1723" required="required" max="9999"/>
         Planned Arrival Time(format hhmm)<input type="number" name="atime"
placeholder="ex:2023" required="required" max="9999"/>
         Estimated Traveling Time(in munites)<input type="number" name="ttime"
placeholder="ex:180" required="required" max="9999"/>
         Distance(in Kms)<input type="number" name="distance" placeholder="ex:2500"
required="required"min='140' max="99999"/>
     <button type="submit" class="btn btn-primary btn-block">Predict</button>
   </form>

  <br>
  <br>
  <br>


 </div>


</body>
</html>
```

## Login.html

```
<!DOCTYPE html>
<html>

<head>
```

```
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet'
type='text/css'>
<link rel="stylesheet" href="{{url_for('static',filename='Lcssstyle.css')}}">


</head>
<body style = "background-color:darkgreen;">
 <h1 class="h1">LOGIN</h1>
 <div class="login">
    <form action="{{ url_for('login')}}" method="post">
       <h4 style="color:#a60c2f"> {{alert}}<h4><h4 style="color:green"> {{message}}<h4>
       Email: <input type="email" name="email" placeholder="ex:abc@gmail.com"
required="required" autocomplete />
       Password: <input type="password" name="pswd" id="pswd" placeholder="ex:abc@123"
required="required" maxlength=15/>
              {{error}}
               <span id="error" style="color:#F00;"> </span>
       <button type="submit" class="btn btn-primary btn-block" id="lbtn">login</button>
       <h5>not registered yet </h5><a href="{{url_for('reg')}}">click here for register</a>
    </form>

   <br>
   <br>
 </div>
</body>
</html>
```

## Register.html

```
<!DOCTYPE html>
<html >

<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet'
type='text/css'>
<link rel="stylesheet" href='{{url_for('static',filename='Lcssstyle.css')}}'>


</head>
```

```
<script>

function confirmPass() {
    var pass = document.getElementById("pswd").value
    var confPass = document.getElementById("cpswd").value
    if(pass != confPass) {
        //alert('Wrong confirm password !');
        document.getElementById('error').innerHTML='wrong confirm password';
                        document.getElementById('rbtn').disabled=true;
    }
    else
    {
        document.getElementById('error').innerHTML=";
                        document.getElementById('rbtn').disabled=false;
    }
}
//function hello()
//{
// console.log("Hellow world");

</script>
<body style = "background-color:darkgreen;">
 <h1 class="h1">REGISTRATION</h1>
 <div class="register">
    <h4 style="color:red"> {{alert}}<h4>
    <form action="{{ url_for('register')}}" method="post">
       Email: <input type="email" name="email" placeholder="ex:abc@gmail.com"
required="required" autocomplete />
       Password: <input type="password" name="pswd" id="pswd" placeholder="ex:abc@123"
required="required" maxlength=15/>
               Confirm Password:<input type="password" name="cpswd" id="cpswd"
placeholder="re-enter same password" required="required" oninput="confirmPass()" />
               <span id="error" style="color:#F00;"> </span>
       <button type="submit" class="btn btn-primary btn-block" id="rbtn">Register</button>
    </form>
    <br>
    <br>
<a href="{{ url_for('main')}}"><button style="color:brown; background-color:#29eed1;
border-color:green; height:50px;width:150px;position:relative;
margin-left:100px">Back</button></a>
   <br>
   <br>


 </div>


</body>
</html>
```

## Result.html

```
<html >

<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet'
type='text/css'>
<link rel="stylesheet" href="{{url_for('static',filename='cssstyle.css')}}">

</head>

<body style = "background-color:darkgreen;">
<div class="logout" style="display : flex;
flex-direction: row-reverse";
>
<a href="{{ url_for('main')}}"><button>Logout</button></a>
</div>
<br>
   <br>
   <br>
   <br>

 <h1 class="h1" style="text-align:center-left">{{ prediction_text }}</h1>

 <div class='prediction' style="display: flex; justify-content:center; height:70px;">

   <a href="{{ url_for('home')}}"><button style="color:#f5f5dc; background-color:green;border-
color:green; height:50px">Prediction dashboard</button></a>
    </div>
            <p style="color:#f7e98e; padding:15px;font-size:26px">we are happy with your visit of
our website and looking for your feedback to improve our website  further.You can give your
feedback through below
       feedback button Or if you feel any inconvinence , raise your complaint using complaint
button  </p>
   <br>
   <br>
   <br>
   <br>
   <a href="https://forms.gle/S2QXJtpeDdwXCuMk6"><button style="color:brown; background-
color:#29eed1;
   border-color:green; height:50px;width:150px;position:relative;
   margin-left:100px">Feedback</button></a>
   <a href="https://forms.gle/hiWgMKHgMVj2CnQ48"><button style="color:brown;
background-color:#29eed1;
   border-color:green; height:50px;width:150px;
   position:relative;
  left:67%">Complaint</button></a>



</body>
</html>
```

## Cssstyle.css

```
@import url(https://fonts.googleapis.com/css?family=Open+Sans);
.btn { display: inline-block; *display: inline; *zoom: 1; padding: 4px 10px 4px; margin-bottom: 0; font-size: 13px; line-height: 18px; color: #333333; text-align: center;text-shadow: 0 1px 1px rgba(255, 255, 255, 0.75); vertical-align: middle; background-color: #f5f5f5; background-image: -moz-linear-gradient(top, #ffffff, #e6e6e6); background-image: -ms-linear-gradient(top, #ffffff, #e6e6e6); background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#ffffff), to(#e6e6e6)); background-image: -webkit-linear-gradient(top, #ffffff, #e6e6e6); background-image: -o-linear-gradient(top, #ffffff, #e6e6e6); background-image: linear-gradient(top, #ffffff, #e6e6e6); background-repeat: repeat-x; filter: progid:dximagetransform.microsoft.gradient(startColorstr=#ffffff, endColorstr=#e6e6e6, GradientType=0); border-color: #e6e6e6 #e6e6e6 #e6e6e6; border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25); border: 1px solid #e6e6e6; -webkit-border-radius: 4px; -moz-border-radius: 4px; border-radius: 4px; -webkit-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05); -moz-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05); box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05); cursor: pointer; *margin-left: .3em; }
.btn:hover, .btn:active, .btn.active, .btn.disabled, .btn[disabled] { background-color: #e6e6e6; }
.btn-large { padding: 9px 14px; font-size: 15px; line-height: normal; -webkit-border-radius: 5px; -moz-border-radius: 5px; border-radius: 5px; }
.btn:hover { color: #333333; text-decoration: none; background-color: #e6e6e6; background-position: 0 -15px; -webkit-transition: background-position 0.1s linear; -moz-transition: background-position 0.1s linear; -ms-transition: background-position 0.1s linear; -o-transition: background-position 0.1s linear; transition: background-position 0.1s linear; }
.btn-primary, .btn-primary:hover { text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25); color: #ffffff; }
.btn-primary.active { color: rgba(255, 255, 255, 0.75); }
.btn-primary { background-color: #4a77d4; background-image: -moz-linear-gradient(top, #6eb6de, #4a77d4); background-image: -ms-linear-gradient(top, #6eb6de, #4a77d4); background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#6eb6de), to(#4a77d4)); background-image: -webkit-linear-gradient(top, #6eb6de, #4a77d4); background-image: -o-linear-gradient(top, #6eb6de, #4a77d4); background-image: linear-gradient(top, #6eb6de, #4a77d4); background-repeat: repeat-x; filter: progid:dximagetransform.microsoft.gradient(startColorstr=#6eb6de, endColorstr=#4a77d4, GradientType=0);  border: 1px solid #3762bc; text-shadow: 1px 1px 1px rgba(0,0,0,0.4); box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.5); }
.btn-primary:hover, .btn-primary:active, .btn-primary.active, .btn-primary.disabled, .btn-primary[disabled] { filter: none; background-color: #4a77d4; }
.btn-block { width: 100%; display:block; }

* { -webkit-box-sizing:border-box; -moz-box-sizing:border-box; -ms-box-sizing:border-box; -o-box-sizing:border-box; box-sizing:border-box;}
```

```
html { width: 100%; height:150%;}

body {
        width: 100%;
        height:100%;
        font-family: 'Open Sans', sans-serif;
        background: #b5c5d8;
        color: #fff;
        font-size: 16px;
        text-align:center-left;
        letter-spacing:1.2px;
        background: rgb(105,155,200);
background: -moz-radial-gradient(top left, ellipse cover, rgba(105,155,200,1) 0%,
rgba(181,197,216,1) 57%);
 background: -webkit-gradient(radial, top left, 0px, top left, 100%, color-
stop(0%,rgba(105,155,200,1)), color-stop(57%,rgba(181,197,216,1)));
 background: -webkit-radial-gradient(top left, ellipse cover, rgba(105,155,200,1)
0%,rgba(181,197,216,1) 57%);
 background: -o-radial-gradient(top left, ellipse cover, rgba(105,155,200,1)
0%,rgba(181,197,216,1) 57%);
 background: -ms-radial-gradient(top left, ellipse cover, rgba(105,155,200,1)
0%,rgba(181,197,216,1) 57%);
 background: radial-gradient(ellipse at top left, rgba(105,155,200,1) 0%,rgba(181,197,216,1)
57%);
 filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#699bc8',
endColorstr='#b5c5d8',GradientType=1 );

}
.login {
        position: absolute;
        top: 40%;
        left: 50%;
        margin: -150px 0 0 -150px;
        width:400px;
        height:400px;
}

.h1 {font-size: 38px;color: #fff; text-shadow: 0 0 10px rgba(0,0,0,0.3); letter-spacing:1px; text-
align:center;margin-left: 90px; }
body .origin{padding-bottom:10px}
body .dest{padding-bottom:10px}
body .btn{padding:20px}
input {
        width: 100%;
        margin-bottom: 10px;
        background: rgba(0,0,0,0.3);
        border: none;
        outline: none;
        padding: 10px;
        font-size: 13px;
        color: #fff;
        text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
        border: 1px solid rgba(0,0,0,0.3);
        border-radius: 4px;
        box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px rgba(255,255,255,0.2);
        -webkit-transition: box-shadow .5s ease;
        -moz-transition: box-shadow .5s ease;
        -o-transition: box-shadow .5s ease;
        -ms-transition: box-shadow .5s ease;
        transition: box-shadow .5s ease;
}
input:focus { box-shadow: inset 0 -5px 45px rgba(100,100,100,0.4), 0 1px 1px
rgba(255,255,255,0.2); }
```

## **<u>Cssstyle.css</u>**

```
@import url(https://fonts.googleapis.com/css?family=Open+Sans);
.btn { display: inline-block; *display: inline; *zoom: 1; padding: 4px 10px 4px; margin-bottom:
0; font-size: 13px; line-height: 18px; color: #333333; text-align: center;text-shadow: 0 1px 1px
rgba(255, 255, 255, 0.75); vertical-align: middle; background-color: #f5f5f5; background-image:
-moz-linear-gradient(top, #ffffff, #e6e6e6); background-image: -ms-linear-gradient(top, #ffffff,
#e6e6e6); background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#ffffff), to(#e6e6e6));
background-image: -webkit-linear-gradient(top, #ffffff, #e6e6e6); background-image: -o-linear-
gradient(top, #ffffff, #e6e6e6); background-image: linear-gradient(top, #ffffff, #e6e6e6);
background-repeat: repeat-x; filter:
progid:dximagetransform.microsoft.gradient(startColorstr=#ffffff, endColorstr=#e6e6e6,
GradientType=0); border-color: #e6e6e6 #e6e6e6 #e6e6e6; border-color: rgba(0, 0, 0, 0.1)
rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25); border: 1px solid #e6e6e6; -webkit-border-radius: 4px; -
moz-border-radius: 4px; border-radius: 4px; -webkit-box-shadow: inset 0 1px 0 rgba(255, 255,
255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05); -moz-box-shadow: inset 0 1px 0 rgba(255, 255, 255,
0.2), 0 1px 2px rgba(0, 0, 0, 0.05); box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px
2px rgba(0, 0, 0, 0.05); cursor: pointer; *margin-left: .3em; }
.btn:hover, .btn:active, .btn.active, .btn.disabled, .btn[disabled] { background-color: #e6e6e6; }
.btn-large { padding: 9px 14px; font-size: 15px; line-height: normal; -webkit-border-radius: 5px;
-moz-border-radius: 5px; border-radius: 5px; }
.btn:hover { color: #333333; text-decoration: none; background-color: #e6e6e6; background-
position: 0 -15px; -webkit-transition: background-position 0.1s linear; -moz-transition:
background-position 0.1s linear; -ms-transition: background-position 0.1s linear; -o-transition:
background-position 0.1s linear; transition: background-position 0.1s linear; }
.btn-primary, .btn-primary:hover { text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25); color: #ffffff; }
.btn-primary.active { color: rgba(255, 255, 255, 0.75); }
.btn-primary { background-color: #4a77d4; background-image: -moz-linear-gradient(top,
#6eb6de, #4a77d4); background-image: -ms-linear-gradient(top, #6eb6de, #4a77d4);
background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#6eb6de), to(#4a77d4));
background-image: -webkit-linear-gradient(top, #6eb6de, #4a77d4); background-image: -o-
linear-gradient(top, #6eb6de, #4a77d4); background-image: linear-gradient(top, #6eb6de,
#4a77d4); background-repeat: repeat-x; filter:
progid:dximagetransform.microsoft.gradient(startColorstr=#6eb6de, endColorstr=#4a77d4,
GradientType=0);  border: 1px solid #3762bc; text-shadow: 1px 1px 1px rgba(0,0,0,0.4); box-
shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.5); }
```

```
.btn-primary:hover, .btn-primary:active, .btn-primary.active, .btn-primary.disabled, .btn-primary[disabled] { filter: none; background-color: #4a77d4; }
.btn-block { width: 100%; display:block; }

* { -webkit-box-sizing:border-box; -moz-box-sizing:border-box; -ms-box-sizing:border-box; -o-box-sizing:border-box; box-sizing:border-box;}

html { width: 100%; height:150%;}

body {
	width: 100%;
	height:100%;
	font-family: 'Open Sans', sans-serif;
	background: #b5c5d8;
	color: #fff;
	font-size: 16px;
	text-align:center-left;
	letter-spacing:1.2px;
	background: rgb(105,155,200);
background: -moz-radial-gradient(top left, ellipse cover, rgba(105,155,200,1) 0%, rgba(181,197,216,1) 57%);
 background: -webkit-gradient(radial, top left, 0px, top left, 100%, color-stop(0%,rgba(105,155,200,1)), color-stop(57%,rgba(181,197,216,1)));
 background: -webkit-radial-gradient(top left, ellipse cover, rgba(105,155,200,1) 0%,rgba(181,197,216,1) 57%);
 background: -o-radial-gradient(top left, ellipse cover, rgba(105,155,200,1) 0%,rgba(181,197,216,1) 57%);
 background: -ms-radial-gradient(top left, ellipse cover, rgba(105,155,200,1) 0%,rgba(181,197,216,1) 57%);
 background: radial-gradient(ellipse at top left, rgba(105,155,200,1) 0%,rgba(181,197,216,1) 57%);
 filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#699bc8', endColorstr='#b5c5d8',GradientType=1 );
}
.login {
	position: absolute;
	top: 40%;
	left: 50%;
	margin: -150px 0 0 -150px;
	width:400px;
	height:400px;
}

.h1 {font-size: 38px;color: #fff; text-shadow: 0 0 10px rgba(0,0,0,0.3); letter-spacing:1px; text-align:center;margin-left: 90px; }
body .origin{padding-bottom:10px}
body .dest{padding-bottom:10px}
body .btn{padding:20px}
input {
	width: 100%;
	margin-bottom: 10px;
```

```
        background: rgba(0,0,0,0.3);
        border: none;
        outline: none;
        padding: 10px;
        font-size: 13px;
        color: #fff;
        text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
        border: 1px solid rgba(0,0,0,0.3);
        border-radius: 4px;
        box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px rgba(255,255,255,0.2);
        -webkit-transition: box-shadow .5s ease;
        -moz-transition: box-shadow .5s ease;
        -o-transition: box-shadow .5s ease;
        -ms-transition: box-shadow .5s ease;
        transition: box-shadow .5s ease;
}
input:focus { box-shadow: inset 0 -5px 45px rgba(100,100,100,0.4), 0 1px 1px
rgba(255,255,255,0.2); }
```

## Lcssstyle.css

```
@import url(https://fonts.googleapis.com/css?family=Open+Sans);
.btn { display: inline-block; *display: inline; *zoom: 1; padding: 4px 10px 4px; margin-bottom:
0; font-size: 13px; line-height: 18px; color: #333333; text-align: center;text-shadow: 0 1px 1px
rgba(255, 255, 255, 0.75); vertical-align: middle; background-color: #f5f5f5; background-image:
-moz-linear-gradient(top, #ffffff, #e6e6e6); background-image: -ms-linear-gradient(top, #ffffff,
#e6e6e6); background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#ffffff), to(#e6e6e6));
background-image: -webkit-linear-gradient(top, #ffffff, #e6e6e6); background-image: -o-linear-
gradient(top, #ffffff, #e6e6e6); background-image: linear-gradient(top, #ffffff, #e6e6e6);
background-repeat: repeat-x; filter:
progid:dximagetransform.microsoft.gradient(startColorstr=#ffffff, endColorstr=#e6e6e6,
GradientType=0); border-color: #e6e6e6 #e6e6e6 #e6e6e6; border-color: rgba(0, 0, 0, 0.1)
rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25); border: 1px solid #e6e6e6; -webkit-border-radius: 4px; -
moz-border-radius: 4px; border-radius: 4px; -webkit-box-shadow: inset 0 1px 0 rgba(255, 255,
255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05); -moz-box-shadow: inset 0 1px 0 rgba(255, 255, 255,
0.2), 0 1px 2px rgba(0, 0, 0, 0.05); box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px
2px rgba(0, 0, 0, 0.05); cursor: pointer; *margin-left: .3em; }
.btn:hover, .btn:active, .btn.active, .btn.disabled, .btn[disabled] { background-color: #e6e6e6; }
.btn-large { padding: 9px 14px; font-size: 15px; line-height: normal; -webkit-border-radius: 5px;
-moz-border-radius: 5px; border-radius: 5px; }
.btn:hover { color: #333333; text-decoration: none; background-color: #e6e6e6; background-
position: 0 -15px; -webkit-transition: background-position 0.1s linear; -moz-transition:
background-position 0.1s linear; -ms-transition: background-position 0.1s linear; -o-transition:
background-position 0.1s linear; transition: background-position 0.1s linear; }
.btn-primary, .btn-primary:hover { text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25); color: #ffffff; }
.btn-primary.active { color: rgba(255, 255, 255, 0.75); }
.btn-primary { background-color: #4a77d4; background-image: -moz-linear-gradient(top,
#6eb6de, #4a77d4); background-image: -ms-linear-gradient(top, #6eb6de, #4a77d4);
background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#6eb6de), to(#4a77d4));
background-image: -webkit-linear-gradient(top, #6eb6de, #4a77d4); background-image: -o-
```

```
linear-gradient(top, #6eb6de, #4a77d4); background-image: linear-gradient(top, #6eb6de,
#4a77d4); background-repeat: repeat-x; filter:
progid:dximagetransform.microsoft.gradient(startColorstr=#6eb6de, endColorstr=#4a77d4,
GradientType=0);  border: 1px solid #3762bc; text-shadow: 1px 1px 1px rgba(0,0,0,0.4); box-
shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.5); }
.btn-primary:hover, .btn-primary:active, .btn-primary.active, .btn-primary.disabled, .btn-
primary[disabled] { filter: none; background-color: #4a77d4; }
.btn-block { width: 100%; display:block; }

* { -webkit-box-sizing:border-box; -moz-box-sizing:border-box; -ms-box-sizing:border-box; -o-
box-sizing:border-box; box-sizing:border-box;}

html { width: 100%; height:100%;}

body {
	width: 100%;
	height:100%;
	font-family: 'Open Sans', sans-serif;
	background: #b5c5d8;
	color: #fffff;
	font-size: 16px;
	text-align:center-left;
	letter-spacing:1.2px;
	background: rgb(105,155,200);
background: -moz-radial-gradient(top left, ellipse cover, rgba(105,155,200,1) 0%,
rgba(181,197,216,1) 57%);
 background: -webkit-gradient(radial, top left, 0px, top left, 100%, color-
stop(0%,rgba(105,155,200,1)), color-stop(57%,rgba(181,197,216,1)));
 background: -webkit-radial-gradient(top left, ellipse cover, rgba(105,155,200,1)
0%,rgba(181,197,216,1) 57%);
 background: -o-radial-gradient(top left, ellipse cover, rgba(105,155,200,1)
0%,rgba(181,197,216,1) 57%);
 background: -ms-radial-gradient(top left, ellipse cover, rgba(105,155,200,1)
0%,rgba(181,197,216,1) 57%);
 background: radial-gradient(ellipse at top left, rgba(105,155,200,1) 0%,rgba(181,197,216,1)
57%);
 filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#699bc8',
endColorstr='#b5c5d8',GradientType=1 );

}
.register {
	position: absolute;
	top: 40%;
	left: 50%;
	margin: -150px 0 0 -150px;
	width:400px;
	height:400px;
}
.login {
	position: absolute;
	top: 40%;
	left: 50%;
	margin: -150px 0 0 -150px;
	width:400px;
	height:400px;
}
.h1 {font-size: 38px;color: #fff; text-shadow: 0 0 10px rgba(0,0,0,0.3); letter-spacing:1px; text-align:center;margin-left: 90px; }
body .origin{padding-bottom:10px}
body .dest{padding-bottom:10px}
body .btn{padding:20px}
input {
	width: 100%;
	margin-bottom: 10px;
	background: #ffff;
	border: none;
	outline: none;
	padding: 10px;
	font-size: 13px;
	color: #fffff;
	text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
	border: 1px solid rgba(0,0,0,0.3);
	border-radius: 4px;
	box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px rgba(255,255,255,0.2);
	-webkit-transition: box-shadow .5s ease;
	-moz-transition: box-shadow .5s ease;
	-o-transition: box-shadow .5s ease;
	-ms-transition: box-shadow .5s ease;
	transition: box-shadow .5s ease;
}
input:focus { box-shadow: inset 0 -5px 45px rgba(100,100,100,0.4), 0 1px 1px rgba(255,255,255,0.2); }
```

## App_IBM.py

```
import numpy as np
import os
from flask import Flask, request, jsonify, render_template,json,redirect,url_for,flash
import pickle
import requests
import sqlite3

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "okbr7ARnOQjyplTOyvNFC2QVkCF6q7afpci065Hucby8"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
```

```
app = Flask(__name__)
app.secret_key="21433253"
model = pickle.load(open('rfmodel.pkl', 'rb'))
conn=sqlite3.connect("database1.db")
conn.execute("CREATE TABLE IF NOT EXISTS login(email TEXT PRIMARY
KEY,password TEXT)")
conn.close()

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

@app.route('/login',methods=['POST','GET'])
def login():
    if request.method=='POST':
        try:
            fv=[x for x in request.form.values()]
            email=request.form["email"]
            pswd=request.form["pswd"]
            conn=sqlite3.connect("database1.db")
            cur=conn.cursor()
            cur.execute("SELECT password FROM login WHERE email=?;",(str(email),))
            result=cur.fetchone()
            cur.execute("SELECT * FROM login")
            if result:
                if result[0]==pswd:
                    flash("login successfully",'success')
                    return redirect('/home')
                else:
                    return render_template("login.html", error="please enter correct password")

            else:
                print("register")
                flash("please Register",'danger')

                return render_template('register.html',alert="user not found,please register")

        except Exception as e:
            return "hello error"

@app.route('/reg')
def reg():
    return render_template("register.html")

@app.route('/register',methods=['POST','GET'])
def register():
    if request.method=='POST':
        try:
            fv=[x for x in request.form.values()]
            email=request.form["email"]
            pswd=request.form["pswd"]
            conn=sqlite3.connect("database1.db")
            cur=conn.cursor()
            cur.execute("SELECT * FROM login WHERE email=?;",(str(email),))
            result=cur.fetchone()
            if result:
                flash("user already exist,please login",'danger')
                return render_template('login.html',alert="user already exist,please login")
            else:
                cur.execute("INSERT INTO  login(email,password)values(?,?)",(str(email),str(pswd)))
                conn.commit()
                cur.execute("SELECT * FROM login")
                flash("Registered successfully",'success')
                return render_template('login.html',message="Registered successfully,please login")

        except Exception as e:
            print(e)
            #flash(e,'danger')
            return "hello error1"



                #return redirect('/')
   # return render_template('login.html')
@app.route('/home')
def home():
    return render_template('mainpage.html')

@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    sm=[6,7,8]
    wt=[9,10,11]
    sp=[12,1,2,3]
    fl=[4,5]
    farr= [int(x) for x in request.form.values()]
    if farr[1] in sm:
        farr.append(0)
    elif farr[1] in wt:
        farr.append(1)
    elif farr[1] in sp:
        farr.append(2)
    else:
        farr.append(3)
    final_features=[int(x) for x in farr]
    print(final_features)
    payload_scoring = {"input_data": [{"fields":
[['QUARTER','MONTH','DAY_OF_MONTH','DAY_OF_WEEK','FL_NUM','ORIGIN','DEST','
CRS_DEP_TIME.1','CRS_ARR_TIME.1','CRS_ELAPSED_TIME','DISTANCE','SEASON']],
"values": [final_features]}]}
```

```
    response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/b54f9857-1352-432a-8ab1-
144ebda20501/predictions?version=2022-11-08',
json=payload_scoring,headers={'Authorization': 'Bearer ' + mltoken})
    print("Scoring response")
    pred=response_scoring.json()
    print(pred)
    prediction=pred['predictions'][0]['values'][0][0]
    #prediction = model.predict([final_features])
    print(prediction)

    output =prediction

    if output==0:
        return render_template('result.html', prediction_text='No delay will happen
{}'.format(output))
    elif output==1:
        return render_template('result.html', prediction_text='There is a chance to departure delay
will happen {}'.format(output))
    elif output==2:
        return render_template('result.html', prediction_text='here is a chance to both departure and
arrival delay will happen {}'.format(output))
    elif output==3:
        return render_template('result.html', prediction_text='here is a chance to flight  will diverted
{}'.format(output))
    elif output==4:
        return render_template('result.html', prediction_text='here is a chance to cancel the flight
{}'.format(output))
    else:
        return render_template('result.html', prediction_text='output {}'.format(output))
'''@app.route('/predict_api',methods=['POST'])
def predict_api():

    For direct API calls trought request

    data = request.get_json(force=True)
    prediction = model.predict([np.array(list(data.values()))])
    output = prediction[0]
    return jsonify(output)'''

if __name__ == "__main__":
    os.environ.setdefault('FLASK_ENV', 'development')
    app.run(debug=False)
```

## **App.py**

```
import numpy as np
import os
from flask import Flask, request, jsonify, render_template
import pickle
```

```
app = Flask(__name__)
model = pickle.load(open('rfmodel.pkl', 'rb'))

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

@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    sm=[6,7,8]
    wt=[9,10,11]
    sp=[12,1,2,3]
    fl=[4,5]
    farr= [int(x) for x in request.form.values()]
    if farr[1] in sm:
        farr.append(0)
    elif farr[1] in wt:
        farr.append(1)
    elif farr[1] in sp:
        farr.append(2)
    else:
        farr.append(3)
    final_features=np.array(farr,dtype='int64')
    print(final_features)
    prediction = model.predict([final_features])


    output = round(prediction[0])

    if output==0:
        return render_template('mainpage.html', prediction_text='No delay will happen
{}'.format(output))
    elif output==1:
        return render_template('mainpage.html', prediction_text='There is a chance to departure
delay will happen {}'.format(output))
    elif output==2:
        return render_template('mainpage.html', prediction_text='here is a chance to both departure
and arrival delay will happen {}'.format(output))
    elif output==3:
        return render_template('mainpage.html', prediction_text='here is a chance to flight  will
diverted {}'.format(output))
    elif output==4:
        return render_template('mainpage.html', prediction_text='here is a chance to cancel the
flight {}'.format(output))
    else:
        return render_template('mainpage.html', prediction_text='output {}'.format(output))
'''@app.route('/predict_api',methods=['POST'])
```

```
def predict_api():

    For direct API calls trought request

    data = request.get_json(force=True)
    prediction = model.predict([np.array(list(data.values()))])
    output = prediction[0]
    return jsonify(output)'''

if __name__ == "__main__":
    os.environ.setdefault('FLASK_ENV', 'development')
    app.run(debug=False)
```

## GITHUB ID: https://github.com/IBM-EPBL/IBM-Project-16140-1659608167Arsha.docx

## DEMO VIDEO LINK: https://clipchamp.com/watch/hdILitv4n4S