FLIGHT DELAY PREDICTION SYSTEM USING MACHINE LEARNING

Bonafede record of work done by

TARUN VISVA R	(19Z358)
KRISHNA TEJA B	(19Z326)
HARISH J	(19Z318)
DHANUSH REDDY N	(19Z333)

Professional Readiness for Innovation, Employability, and Entrepreneurship

GUIDE: VANI K

BACHELOR OF ENGINEERING

BRANCH: COMPUTER SCIENCE AND ENGINEERING

NOVEMBER 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PSG COLLEGE OF TECHNOLOGY
(Autonomous Institution)

COIMBATORE – 641 004

1. INTRODUCTION

1.1 Project Overview

The airline industry has been the backbone of transportation ever since the 1950s. It is important to ensure that the aviation industry has safety and punctuality at its peak, as many VIPS, business personalities, and sports teams use aircraft as their primary transport for important events. Although the industry's safety and customer service record is close to the best it has ever been, there are still rare occasions where passengers are inconvenienced by delays or even cancellations which end up costing the industry and the economy a lot. According to the estimation by the Total Delay Impact Study, the total cost of air transportation delays to air travelers and the airline industry in 2007 was $32.9 billion in the US, resulting in a $4 billion reduction in GDP. Therefore, predicting flight delays can improve airline operations and passenger satisfaction, which will positively impact the economy. Thus, building the right system using a suitable model is of great importance.

1.2 Purpose

The project aims to develop a flight delay prediction system that can predict the flight delays and improve the airline operations and passenger satisfaction.

2. LITERATURE SURVEY

2.1 Existing problem

Flight delay prediction using Isolation Forest algorithm[1]:

In this paper, an analysis has been conducted to discover the main causes of flight delays and to explore a few machine learning algorithms to find the most suitable one. According to statistics from the Bureau of Transportation, more than 69% of flight delays are caused due to unexpected weather conditions. This paper mainly focuses on considering weather as the main factor for flight delay anomalies. The flight dataset has been collected from the Bureau of Transportation(BTS) website, which has over 20 variables for each flight. The weather data has been collected from the National Oceanic and Atmospheric Administration (NOAA), with 4 or 5 determining factors(such as temperature, precipitation, etc..) consisting of values for each factor reported at different points in a day. The mean of the values at different points of the day are taken and used for integration with the Flight data. In the initial phase of implementation, algorithms such as random forest and KNN have been used, but their accuracy was lower (up to 62%). This is because of anomalies (in our case, flight cancellation/delay is the anomaly) in the dataset. As the main goal is to capture those deviations, the next suitable method to model this dataset is the Isolation Forest method. This is an unsupervised learning algorithm that is mainly used if the dataset contains anomalies and gives high accuracy in those cases as its main purpose is to capture the anomalies. This model gives an accuracy of 76%. Certain improvements can be done to improve the accuracy but it involves additional overhead. For example, Instead of calculating the mean of weather values at different points in time, real-time airport weather for each flight could be collected, which turns out to be a tedious task. Also, accuracy may rise a little if factors other than the weather are also considered.

Analysis of classification models on flight delay prediction[2]:

This paper explores the parameters affecting flight delays and analyzes various machine learning models that can be used for the prediction of flight delays. Different studies conducted in different places show that flights are delayed due to different parameters. For example, the main parameters that affect the airline network in the US are visibility, wind, and departure time, and in Iran are fleet age and aircraft type. Different classification models such as Logistic Regression, K-Nearest Neighbor (KNN), Gaussian Naïve Bayes, Decision Tree, Support Vector Machine (SVM), Random Forest, and Gradient Boosted Trees have been used for prediction. The main objective of this study is to predict flight delays based on labeled data. Therefore, a supervised learning classification algorithm was selected as the appropriate one. They calculated the values accuracy, precision, recall, and f1 score to conclude the better model among the chosen models. The result shows that the highest values of accuracy, precision, recall, and f1-score are generated by the Decision Tree model (accuracy: 0.9778; precision: 0.9777; recall: 0.9778; f1-score: 0.9778). Other tree-based ensemble classifiers also show good performance. Random Forest and Gradient Boosted Tree have an accuracy of 0.9240 and 0.9334, significantly higher than the rest of the models. The other four base classifiers Logistic Regression, KNN, Gaussian Naïve Bayes, and SVM, are not tree-based and therefore do not show a good performance in classification. The KNN model has the least performance since its precision and f1-score are the lowest among the seven models.

Flight Delay Prediction Using XGBoost[3]:

In this paper, an analysis has been performed using XGBoost which is one of the most popular machine learning algorithms regardless of the type of prediction task at hand - regression or classification. It has become the state-of-the-art machine learning algorithm to deal with structured data. A publicly available Kaggle dataset collected from United States domestic air traffic has been used for training. The dataset consists of over 3 million samples with 19 features. XGBoost is software that can be installed on our machine and accessed from a variety of interfaces. The library is focused on computational speed and model and the performance of the model. In this implementation, Mean absolute error(MAE) has been used for delay prediction. In statistics, MAE is the average vertical distance between each point and the identity line or it is also the average horizontal distance between each point and the identity line. Based on the analysis of the results, it is evident that the integration of multidimensional heterogeneous data, combined with the application of different techniques for feature selection and regression can provide promising tools for inference in the current domain

Flight Delay Classification Prediction Using Stacking Algorithm[4]:

This paper aims to prove that the stack algorithm has advantages in airport flight delay prediction, especially for the algorithm selection problem of machine learning technology. The authors use SMOTE to preprocess an imbalanced dataset and use Boruta Algorithm for feature Selection. The author uses five supervised ML algorithms for first-level learners(KNN, Gaussian Naive Bayes, Random forest, Decision Tree, and Logistic regression) and Logistic regression for second-level learners. The data set is from Logan International Airport in Boston, Massachusetts, the United States, which contains 298914 flight datasets and 67822 delayed flights. The SMOTE algorithm is used to balance the dataset. The features are selected using the Boruta algorithm that is applied to nine features(weather is not included in feature selection) and 4 features were marked as critical (arrival time, day of month, month, and departure time), and stack-based learning is used with k-fold cross validation. Comparing the level one learners, the Random forest and KNN had good prediction results with accuracy exceeding 0.8 and 0.7 respectively. Whereas Gaussian Naive Bayes and Logistic regression performed poorly. The stacking algorithm also yields good results with an accuracy of over 0.8. The stacking algorithm is also stable as the result does not vary significantly even if learners are removed from the stacking algorithm. The authors conclude that stacking algorithms are good for algorithm selection and it is also stable even when learners are removed from the stacking algorithm.

2.2 References

1. Miloš Vereš, Flight delay/cancellation prediction using machine learning Adapting new ways to help stranded passengers

2. Yuemin Tang, Airline Flight Delay Prediction Using Machine Learning Models

3. K.P. Surya Teja, Vigneswara Reddy, Dr. Shaik Subhani. Flight Delay Prediction Using Machine Learning Algorithm XGBoost

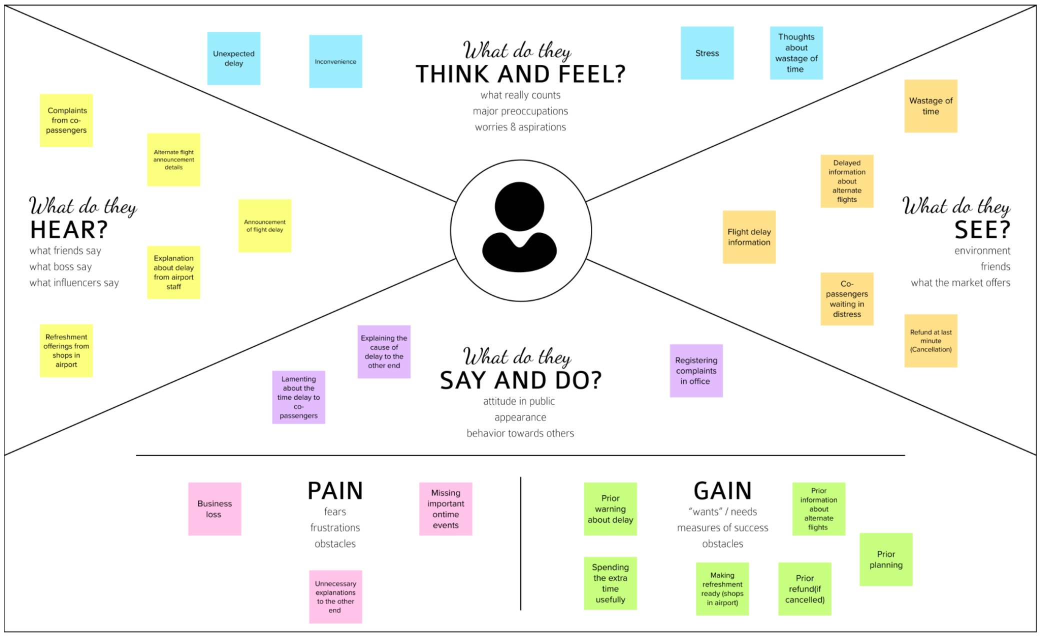
4. Jia Yi, Honghai Zhang, Hao Liu, et al. Flight Delay Classification Prediction Based on Stacking Algorithm

2.3 Problem Statement Definition

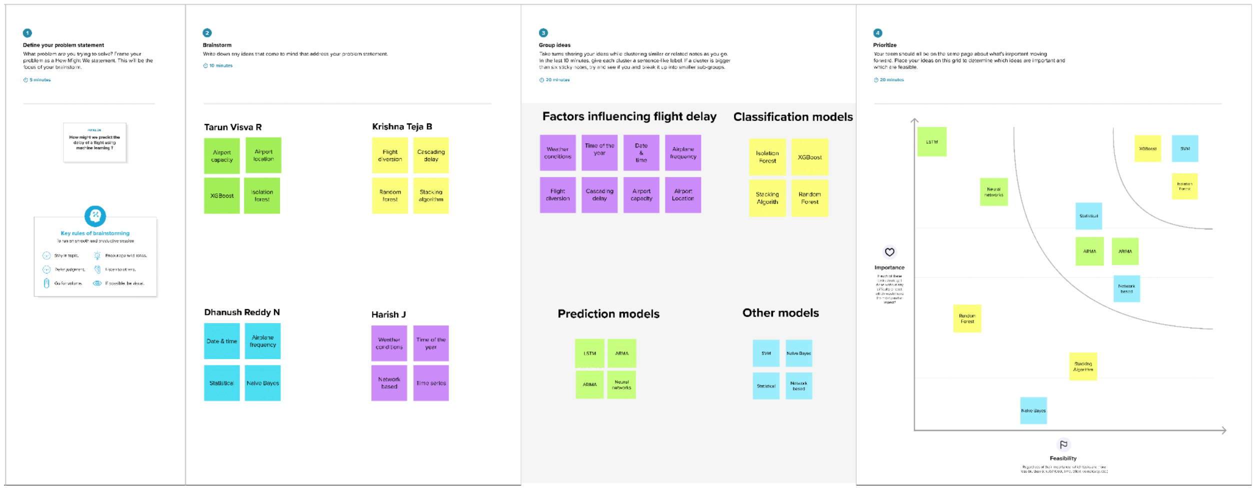
To develop a flight delay prediction model using Machine Learning models such an s decision tree to classify whether a flight is delayed or not based on the origin, destination, and date of journey

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas


What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Unexpected delay
Inconvenience
Stress
Thoughts about wastage of time
What do they
HEAR?
what friends say
what boss say
what influencers say
Complaints from co-passengers
Alternate flight announcement details
Announcement of flight delay
Explanation about delay from airport staff
Refreshment offerings from shops in airport
What do they
SEE?
environment
friends
what the market offers
Wastage of time
Delayed information about alternate flights
Flight delay information
Co-passengers waiting in distress
Refund at last minute (Cancellation)
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
Explaining the cause of delay to the other end
Lamenting about the time delay to co-passengers
Registering complaints in office
PAIN
fears
frustrations
obstacles
Business loss
Missing important ontime events
Unnecessary explanations to the other end
GAIN
"wants" / needs
measures of success
obstacles
Prior warning about delay
Prior information about alternate flights
Prior planning
Spending the extra time usefully
Making refreshment ready (shops in airport)
Prior refund(if cancelled)

3.2 Ideation & Brainstorming



1 **Define your problem statement**

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

5 minutes

PROBLEM

How might we predict the delay of a flight using machine learning ?

Key rules of brainstorming

To run an smooth and productive session

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

2 **Brainstorm**

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

10 minutes

3 **Group ideas**

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, 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

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.

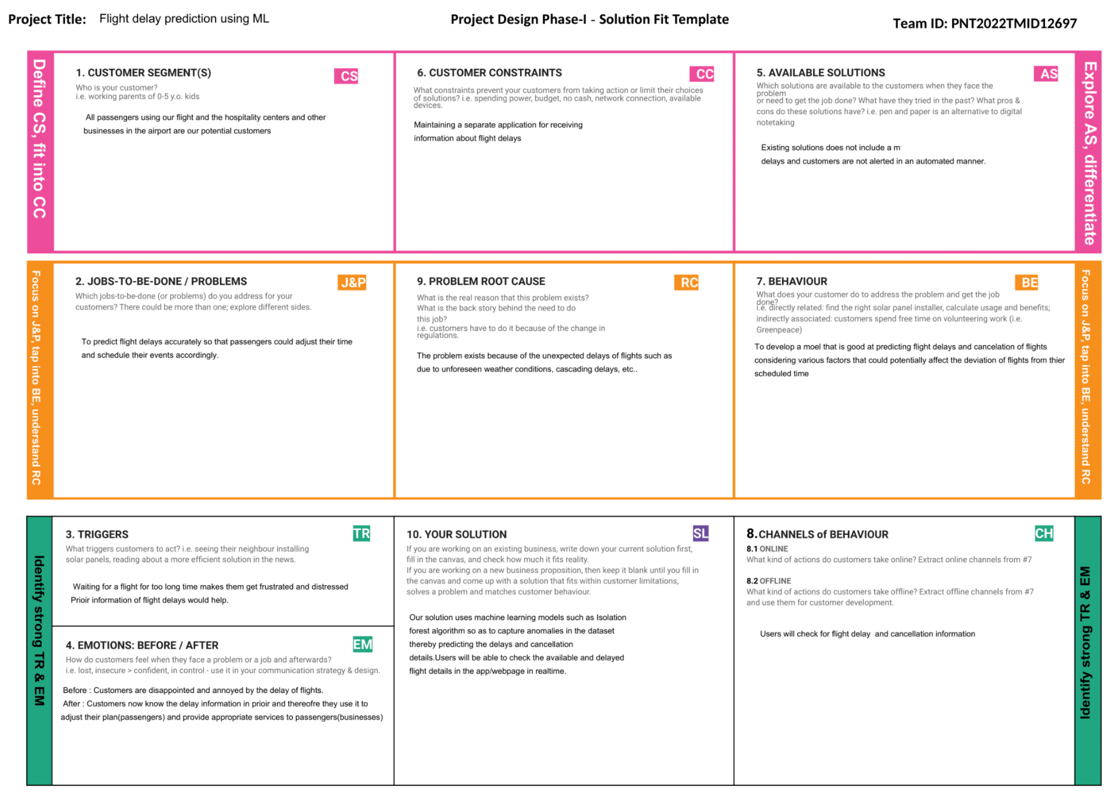
20 minutes

3.3 *Proposed Solution*

S.No.	Parameter	Description
	Problem Statement (Problem to be solved)	Over the last twenty years, air travel has been increasingly preferred among travelers, mainly because of its speed and in some cases comfort. This has led to phenomenal growth in air traffic. An increase in air traffic growth has also resulted in massive levels of aircraft delays on the ground and in the air. These delays are responsible for large economic and environmental losses. The main objective of the model is to predict flight delays accurately in order to optimize flight operations and minimize delays.
	Idea / Solution description	Using a machine learning model, we can predict flight arrival delays. The input to our algorithm is rows of feature vectors like departure date, departure delay, distance between the two airports, scheduled arrival time etc. We then use a decision tree classifier to predict if the flight arrival will be delayed or not. A flight is considered to be delayed when the difference between scheduled and actual arrival times is greater than 15 minutes. Furthermore, we compare decision tree classifiers with logistic regression and a simple neural network for various figures of merit.
	Novelty / Uniqueness	Along with the information whether the flight is being delayed or not, the approximate time of arrival (after delay) is also predicted so that passengers can adjust their schedule accordingly. Other flights whose departures are delayed due to the arrival delay of one flight are also identified and notifications are sent to the respective passengers.

	Social Impact / Customer Satisfaction	This notifies people about the delay in flights well in advance so that the passengers need not get frustrated knowing the slow down at the last minute. Normal passengers can arrive patiently at the airport. Important people (VIPs, Doctors, etc..) can reschedule events based on the delay information.
	Business Model (Revenue Model)	Hospitality centers and businesses could use this solution to prepare refreshment, recreation and other appropriate services to the customers waiting in the lobby.
	Scalability of the Solution	Since this application is hosted as a web page in a cloud platform, anyone can sign in and obtain the information on any device from their browser.

3.4 Problem Solution fit



Project Title: Flight delay prediction using ML

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID12697

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

All passengers using our flight and the hospitality centers and other businesses in the airport are our potential customers

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Maintaining a separate application for receiving information about flight delays

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

Existing solutions does not include a m
delays and customers are not alerted in an automated manner.

Explore AS, differentiate

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

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

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

To predict flight delays accurately so that passengers could adjust their time and schedule their events accordingly.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

The problem exists because of the unexpected delays of flights such as due to unforeseen weather conditions, cascading delays, etc..

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

To develop a moel that is good at predicting flight delays and cancelation of flights considering various factors that could potentially affect the deviation of flights from thier scheduled time

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

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Waiting for a flight for too long time makes them get frustrated and distressed
Prioir information of flight delays would help.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Before : Customers are disappointed and annoyed by the delay of flights.
After : Customers now know the delay information in prioir and thereofre they use it to adjust their plan(passengers) and provide appropriate services to passengers(businesses)

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

Our solution uses machine learning models such as Isolation forest algorithm so as to capture anomalies in the dataset thereby predicting the delays and cancellation details.Users will be able to check the available and delayed flight details in the app/webpage in realtime.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

Users will check for flight delay and cancellation information

Identify strong TR & EM

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Admin authentication	Confirmation via OTP
FR-4	Admin dashboard	Admin uploads dataset to train the model
FR-4	Display result	The flight delay information will be displayed in the web page

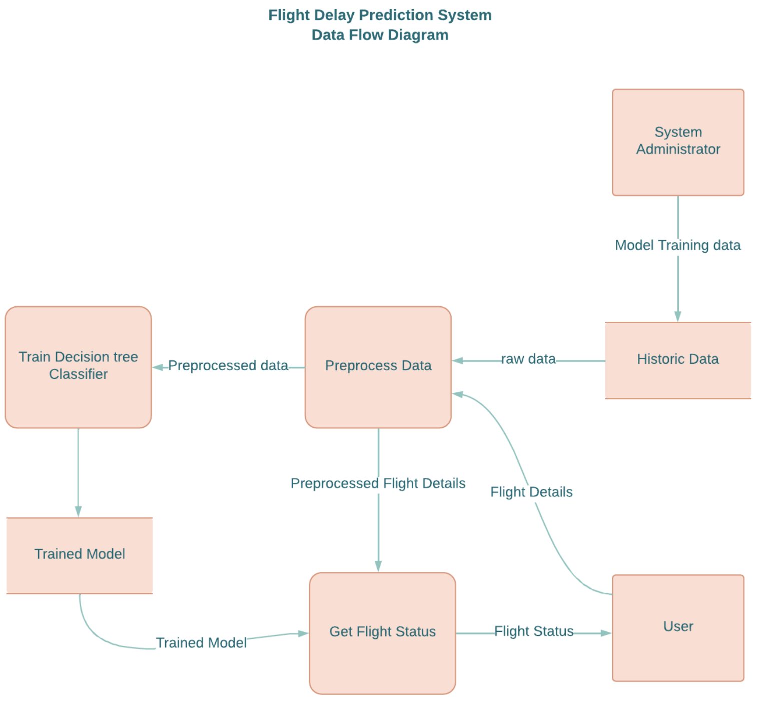
4.2 Non-Functional requirements

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

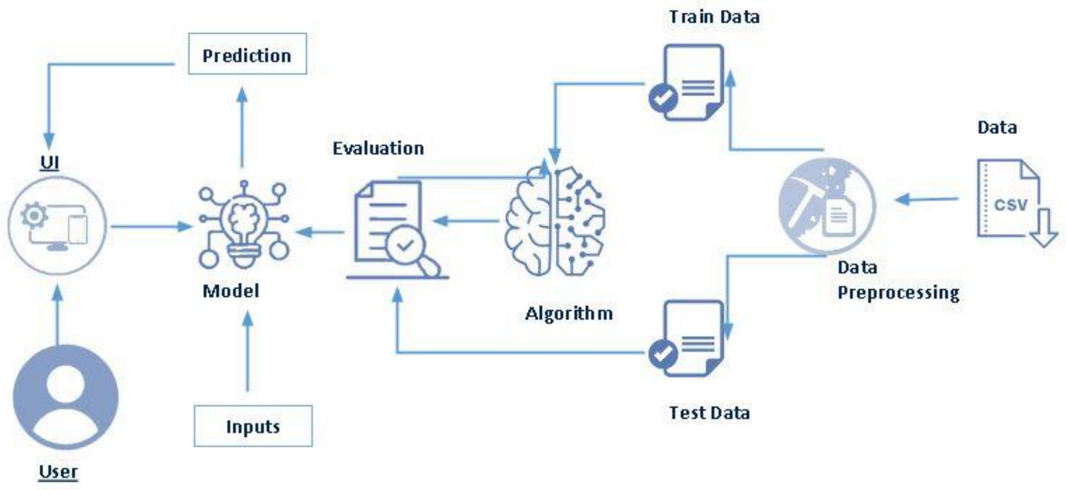
FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	Easily accessible as the application is hosted as a web page
NFR-2	**Security**	User and admin require login to access the web page
NFR-3	**Reliability**	The system focuses on preventing lifetime failures
NFR-4	**Performance**	The system performance is elevated as the cloud server it is hosted on is usually a powerful computational resource
NFR-5	**Availability**	The system runs on a cloud server. So, it is always available
NFR-6	**Scalability**	The application can be easily scaled to meet user demands

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 *User Stories*

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria
Customer (Mobile user and web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard
		USN-2	As a user, I will receive a confirmation email once I have registered for the application	I can receive a confirmation email & click confirm
		USN-4	As a user, I can register for the application through Gmail	I can access my account / dashboard
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my account / dashboard
	Dashboard	USN-6	As a user i can enter the flight details to get the flight status	I can view flight status for particular flight
Administrator	Authentication	USN-7	As an administrator, I can give my login credentials to view the admin dash board	I can view the admin dashboard
	Dashboard	USN-8	As an administrator, I can upload a dataset csv file to train the model	I can upload the dataset and train the model

6. PROJECT PLANNING & SCHEDULING

6.1 *Sprint Planning & Estimation*

User Type	Functional Requirement (Epic)	User Story Number	Priority	Release
Customer (Mobile user and web user)	Registration	USN-1	High	Sprint - 1
		USN-2	High	Sprint - 1
		USN-4	Medium	Sprint - 1
	Login	USN-5	High	Sprint - 2
	Dashboard	USN-6	High	Sprint - 2
Administrator	Authentication	USN-7	High	Sprint - 3

	Dashboard	USN-8	High	Sprint - 4

7. *CODING & SOLUTIONING*

7.1 *Viewing if flight is delayed given the Source, Destination and Journey date*

The Solution first deals with the Machine learning model that predicts if the flight is delayed based on the given input. The model requires the Source, destination(in the form of three-letter codes), journey date, and the distance between the source and destination. This input is then preprocessed to the following format:

a. Source: label encoded
b. Destination: label encoded
c. Distance: integer
d. quarter
e. month
f. day of month
g. day of week

Using this formatted input the decision tree outputs a 0 or 1. 0 indicating that the flight was not delayed and 1 indicating that the flight was delayed. The following code is used to preprocess the dataset. The encoders used during training are also used when prediction is required. The encoders are saved locally as a pickle file in the folder encoders.

```
import os
import pickle
import pandas as pd
from sklearn.preprocessing import LabelEncoder

train_columns = ["QUARTER", "MONTH", "DAY_OF_MONTH", "DAY_OF_WEEK",
"ORIGIN", "DEST", "DISTANCE", "DEP_DEL15"]

class Preprocess:
    def __init__(self) -> None:
        self.base_path = os.path.join(os.getcwd(), 'encoders')
        self.origin_encoder = LabelEncoder()
        self.dest_encoder = LabelEncoder()

        if not os.path.exists(self.base_path):
            print('Encoders not found, Creating Files')
            os.mkdir(self.base_path)

        else:
            with open(os.path.join(self.base_path, 'origin_encoder.pkl'),
'rb') as file:
                classes = pickle.load(file)
                self.origin_encoder.classes_ = classes
            with open(os.path.join(self.base_path, 'dest_encoder.pkl'),
'rb') as file:
                classes = pickle.load(file)
                self.dest_encoder.classes_ = classes
    def __check_cloumns(self) -> list:
        cols = []
        for val in train_columns:
            if val not in self.data.columns:
                cols.append(val)
        if len(cols) == 0:
            return None
        return cols

    def train_preprocess(self, data_path) -> pd.DataFrame:
        self.data = pd.read_csv(data_path)

        if self.__check_cloumns() is not None:
            print("CSV does not contain the required columns",
self.__check_cloumns())
            return False

        del_columns = []
        for col in self.data.columns:
            if col not in train_columns:
                del_columns.append(col)

        for col, val in zip(self.data.columns,
self.data.columns.str.match('Unamed')):
            if val:
                del_columns.append(col)

        self.data = self.data.drop(del_columns, axis  = 1)
        self.data = self.data.dropna()

        self.data['ORIGIN'] =
self.origin_encoder.fit_transform(self.data['ORIGIN'])
        self.data['DEST'] =
self.dest_encoder.fit_transform(self.data['DEST'])

        with open(os.path.join(self.base_path, 'origin_encoder.pkl'),
'wb') as file:
            pickle.dump(self.origin_encoder.classes_, file)
        with open(os.path.join(self.base_path, 'dest_encoder.pkl'), 'wb')
as file:
            pickle.dump(self.dest_encoder.classes_, file)

        y = self.data['DEP_DEL15']
        x = self.data.drop(columns='DEP_DEL15')

        return x.to_numpy(),y.to_numpy()

    def test_preprocess(self, x: list)-> list:
        x[4] = self.origin_encoder.transform([x[4]])[0]
        x[5] = self.dest_encoder.transform([x[5]])[0]
        return x
```

The following code is used to train the model on the dataset and save the model locally as a pickle file. The predict function is used to predict the output which returns list containing one value which is 0 or 1.

```
import os
import pickle
import format_input
from sklearn.tree import DecisionTreeClassifier as dt
from preprocess import Preprocess

def load_model() -> dt:
    try:
        with open('model/decisionTree.pkl', 'rb') as file:
            model = pickle.load(file)
    except FileNotFoundError:
        x, y = 
Preprocess().train_preprocess('./dataset/flightdata.csv')
        model = dt()
        model.fit(x,y)
        if not os.path.exists('model'):
            os.mkdir('model')
        with open('model/decisionTree.pkl', 'wb') as file:
            pickle.dump(model, file)
    return model

def predict(origin: str, dest: str, date: str) -> int:
    model = load_model()
    x = format_input.extract_date(date)
    x.append(origin)
    x.append(dest)
    try:
        x.append(format_input.extract_distance(origin, dest))
    except IndexError:
        return [2]
    x = Preprocess().test_preprocess(x)
    return model.predict([x])
```

To format the input during the prediction phase the following functions are used. One function extracts the date to the specific format. The extract distance function obtains the distance between source and destination.

```
import pandas as pd
from datetime import datetime
import calendar

days = {
    "Monday":1,
    "Tuesday":2,
    "Wednesday":3,
    "Thursday":4,
    "Friday":5,
    "Saturday":6,
            "Sunday":7
        }

        def extract_distance(src , dest):
            df = pd.read_csv('dataset/flightdata.csv')
            return df.loc[(df['ORIGIN']==src) &
(df['DEST']==dest)]['DISTANCE'].to_numpy()[0]

        def get_quater(month: int) -> int:
            res = 1
            while(month>3):
                res = res+1
                month-=3
            return res

        def extract_date(date: str):
            x = []
            date = date.split("-")
            x.append(get_quater(int(date[1])))
            x.append(int(date[1]))
            x.append(int(date[2]))
            x.append(days[calendar.day_name[datetime(int(date[0]),
int(date[1]), int(date[2])).weekday()]])
            return x
```

Once the model is prepared the front-end of the project is developed using html and css. The following code is the HTML for the prediction page.

```
    <!DOCTYPE html>
    <html style="font-size: 16px;" lang="en"><head>
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
      <meta charset="utf-8">
      <meta name="keywords" content="Sign In, Sign up">
      <meta name="description" content="">
      <title>Home</title>
      <link rel="stylesheet" href="static/nicepage.css" media="screen">
      <link rel="stylesheet" href="static/summa.css" media="screen">
    <link rel="stylesheet" href="static/Home.css" media="screen">
      <script class="u-script" type="text/javascript" src="static/jquery.js" defer=""></script>
      <script class="u-script" type="text/javascript" src="static/nicepage.js" defer=""></script>
      <meta name="generator" content="Nicepage 4.21.12, nicepage.com">
      <link id="u-theme-google-font" rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Roboto:100,100i,300,300i,400,400i,500,500i,700,700i,900,900i|Open+Sans:300,300i,400,400i,500,500i,600,600i,700,700i,800,800i">

      <script type="application/ld+json">{
      "@context": "http://schema.org",
      "@type": "Organization",
      "name": ""
    }</script>
      <meta name="theme-color" content="#478ac9">
```

```
          <meta property="og:title" content="Home">
          <meta property="og:type" content="website">
        </head>
        <body data-home-page="Home.html" data-home-page-title="Home" class="u-body u-xl-mode"
data-lang="en">

          <section class="u-clearfix u-gradient u-section-2" id="sec-a1b2">
            <div class="u-clearfix u-sheet u-valign-middle-md u-valign-middle-sm u-valign-middle-xs
u-sheet-1">
              <div class="u-clearfix u-expanded-width u-gutter-10 u-layout-wrap u-layout-wrap-1">
                <div class="u-layout">
                  <div class="u-layout-row">
                    <div class="u-align-left u-container-style u-gradient u-layout-cell u-left-cell u-size-60
u-layout-cell-1" src="">
                      <div class="u-container-layout u-container-layout-1" src="">
                        <div class="u-align-center u-container-style u-group u-opacity u-opacity-90
u-radius-30 u-shape-round u-white u-group-1">
                          <div class="u-container-layout u-container-layout-2">
                            <h2 class="u-custom-font u-text u-text-default u-text-font u-text-1">Flight Delay
Predict</h2>
                            <div class="u-form u-form-1">
                              <form action="{{url_for('prediction')}}" method="POST" class="u-clearfix
u-form-spacing-10 u-inner-form"  style="padding: 10px;">
                                <div class="u-form-group u-form-partition-factor-2" style="width: 48%;float:
left;">
                                  <label for="name-9070" class="u-labeL">From</label>
                                  <input type="text" placeholder="Enter airport code (ATL ,SEA etc)"
id="name-9070" name="from" class="u-border-1 summa1 u-border-grey-30 u-input
u-input-rectangle u-white" required="required">
                                </div>
                                <div class="u-form-group u-form-partition-factor-2" style="width: 48%;float:
right;">
                                  <label for="email-9070" class="u-label">To</label>
                                  <input placeholder="Enter airport code (ATL ,SEA etc)" id="email-9070"
name="to" class="u-border-1 u-border-grey-30 u-input u-input-rectangle u-white"
required="required" type="text">
                                </div>
                                <div class="u-form-date u-form-group u-form-group-3">
                                  <label for="date-0dad" class="u-label">Date</label>
                                  <input type="date" placeholder="MM/DD/YYYY" id="date-0dad" name="date"
class="u-border-1 u-border-grey-30 u-input u-input-rectangle u-white" required="">
                                </div>
                                <div class="u-align-right u-form-group u-form-submit">
                                  <a href="#" class="u-active-palette-4-light-1 u-border-none u-btn u-btn-round
u-btn-submit u-button-style u-hover-palette-4-light-1 u-palette-4-base u-radius-10
u-btn-1">Predict</a>
                                  <input type="submit" value="submit" class="u-form-control-hidden" >
                                </div>
                                <div class="u-form-send-message u-form-send-success"> Thank you! Your
message has been sent. </div>
```

```
                                <div class="u-form-send-error u-form-send-message"> Unable to send your
message. Please fix errors then try again. </div>
                                <input type="hidden" value="" name="recaptchaResponse">
                                <input type="hidden" name="formServices"
value="4941dfb8fdd181ef2c8833948a967d09">
                              </form>
                            </div>
                            <p class="u-custom-font u-text u-text-default u-text-font
u-text-2">{{output_text}}</p>
                          </div>
                        </div>
                      </div>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </section>
       </body></html>
```

A flask app creates the routes for the application. The following code is for the flask application.

```
from flask import Flask, render_template, request
from Model import predict
app = Flask(__name__)

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

@app.route('/prediction',methods=['POST','GET'])
def prediction():
    args = [i for i in request.form.values()]
    args = tuple(args[:3])
    out = int(predict(*args)[0])
    if out == 0:
        return render_template("index.html",output_text="Flight will
arrive at the scheduled time ...")
    elif out==1:
        return render_template("index.html",output_text="Flight may be
delayed by 15 minutes or more ...")
    else:
        return render_template("index.html",output_text="Invalid source or
destination !!!")
```

The app is deployed on render (`https://render.com/`). The app dependencies are installed using the command pipevn install. The app is started using the command gunicorn main:app. The following contents show the dependencies in the Pipfile.

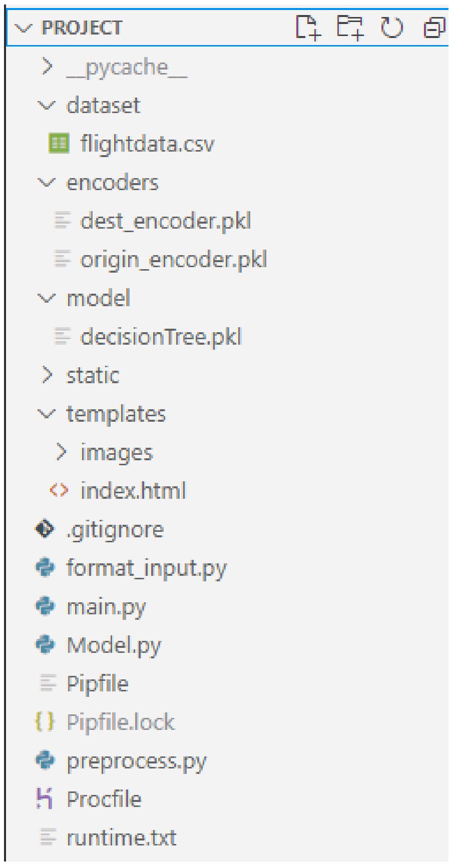
```
[[source]]
url = "https://pypi.org/simple"
verify_ssl = true
name = "pypi"

[packages]
sklearn = "*"
numpy = "*"
pandas = "*"
flask = "*"
oauthlib = "*"
scikit-learn = "*"
gunicorn = "*"

[dev-packages]

[requires]
python_version = "3.10"
```

The following figure shows the directory structure of the project.



```
PROJECT
  > __pycache__
  v dataset
      flightdata.csv
  v encoders
      dest_encoder.pkl
      origin_encoder.pkl
  v model
      decisionTree.pkl
  > static
  v templates
    > images
      index.html
  .gitignore
  format_input.py
  main.py
  Model.py
  Pipfile
  Pipfile.lock
  preprocess.py
  Procfile
  runtime.txt
```

8. TESTING

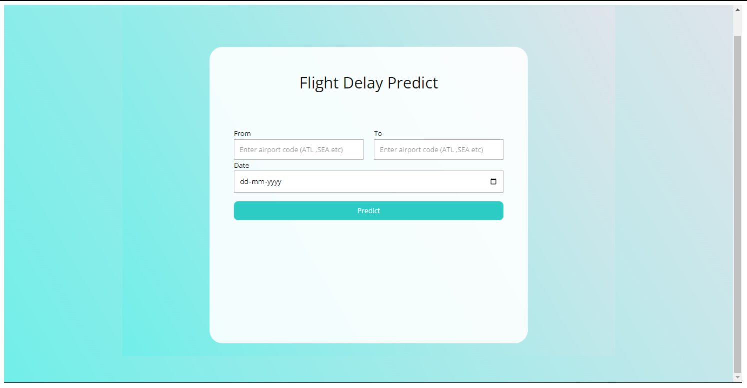
8.1 User Acceptance Testing

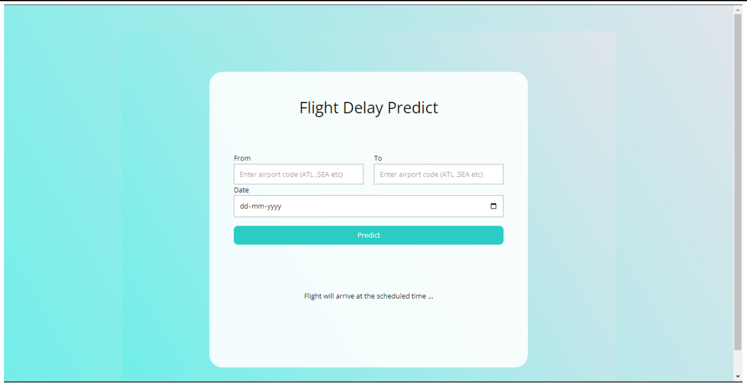
This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Model Prediction	8	0	0	8
Front end	8	0	0	8

9. RESULTS

9.1 Performance Metrics

S.No.	Parameter	Screenshot / Values
1.	Dashboard design	 Flight Delay Predict From Enter airport code (ATL ,SEA etc) To Enter airport code (ATL ,SEA etc) Date dd-mm-yyyy Predict

		 Flight Delay Predict From To Enter airport code (ATL ,SEA etc) Enter airport code (ATL ,SEA etc) Date dd-mm-yyyy Predict Flight will arrive at the scheduled time ...
2.	Data Responsiveness	Max time to load page webpage(ms): 418 Max time to show results in webpage(ms): 1005 Max time for the model to predict the result including preprocessing steps(ms): 31.25

10.ADVANTAGES & DISADVANTAGES

The model is fast in predicting and uses decision trees. The Decision tree classifier requires less training time and less data to train the model. The project's disadvantage is that the delay's extent is not predicted.

11. CONCLUSION

This report explains the implementation of the flight delay prediction system and the metrics used for testing the system. The flight delay prediction system is successfully deployed in the internet using render for viewing the webpage.

12. FUTURE SCOPE

More complex models can be used to predict the time delay for each flight. This might require more data if deep learning models are used for time series analysis.

13.APPENDIX

GitHub

https://github.com/IBM-EPBL/IBM-Project-12655-1659456758

Project Demo Link

https://drive.google.com/file/d/11RzbLAnc6lBbZuH6GsuXuHQzEyAwdme8/view