CHAPTER 1

INTRODUCTION

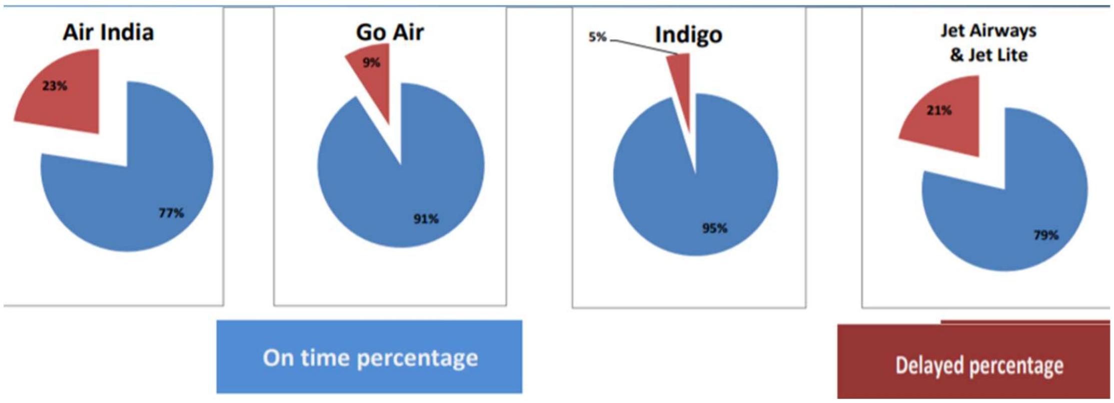
1.1 Introduction

During the most defining period of human history, where computing has moved from mainframes to PCs to cloud, and now to artificial intelligence. A fundamental sub-area of artificial intelligence has come into notice, called as Machine Learning, which enables computers to get into a mode of self-learning without being explicitly programmed. With the concept of machine learning, we have been able to apply complex mathematical computations to big data iteratively and automatically, that too with efficient speed, this phenomenon has been encompassing momentum over the last several years. On the other hand, data mining involves data discovery and sorting it among large data sets available to identify the required patterns and establish relationships with the aim of solving problems through data analysis. Simply combining, machine learning and data mining use the same type of approach and set of algorithms, except the kind of data pre-processing and end prediction varies. By combining these two core areas to predict and present the most accurate results possible. Flight delay has been the subject of several studies in recent years. With the increase in the demand for air travel, effects of flight delay have been increasing. The Federal Aviation Administration (FAA) estimates that commercial aviation delays, cost airlines more than $3 billion per year and according to BTS, the total number of arrival delay in 2016 were 860,646. Impacts of flight delay in future are likely to get worse due to an increase in the air traffic congestion, growth of commercial airlines and increase in the number of passengers per year. While flight delays are likely to persist in future due to unavoidable factors such as weather and unpredictable flight maintenance, we create a predictive algorithm to forecast flight delay.

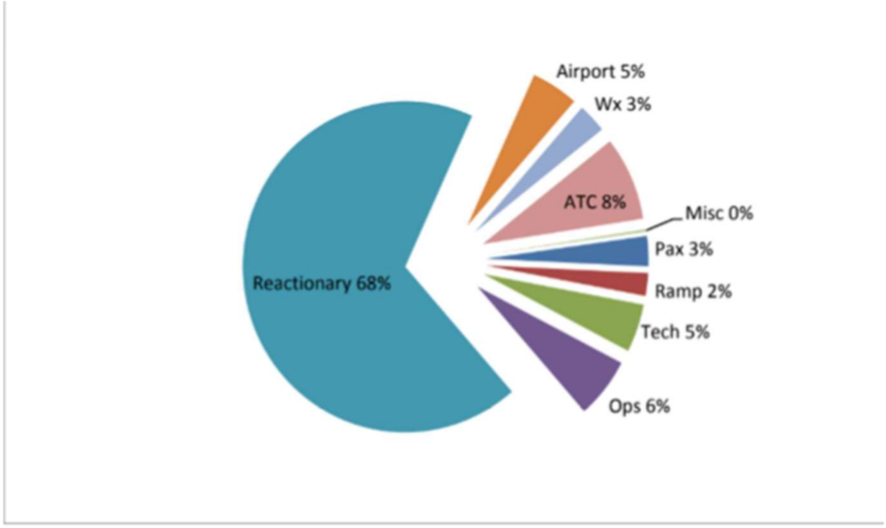
CHAPTER 2

LITERATURE SURVEY

- Airlines industry incurs an average cost of about $11,000 per delayed flight based on 61000 delayed flights per month average.



- The above figure shows the percentages of flights delayed at Delhi airport for a month of January



- Reasons for delay of flights has been analyzed, which is presented above. It has been found that majority of delays have been attributed to ‘Reactionary’.

2.1 Existing Solutions

- Providing an alternate flight to the customer if possible.
- Providing compensation to the customer.
- Delay will be known only after getting information from airlines.

2.2 Disadvantages

- By this people who travel for business conferences can miss their meetings.
- It also affects the people of who are in urgency of travelling and also it makes people to Wait for long hours in airport which makes airlines lose their reputation.

2.3 Proposed System

- We predict the delay by considering the previous delays of a flight.
- Helps organizations to improve there performance.

2.4 Advantages

- By knowing the delay in prior people who are in urgency could prefer in booking another flight for their journey especially business travellers.
- Airlines can utilize this model , as it can help them retain customers.

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATIONS

3.1 Hardware requirements

In hardware requirement we require all those components which will provide us the platform for the development of the project. The minimum hardware required for the development of this project is as follows

Ram	:minimum 2 GB
Hard disk	:minimum 50 GB
Processor	: i3

These all are the minimum hardware requirement required for our project. We want to make our project to be used in any. Type of computer therefore we have taken minimum configuration to a large extent.2 GB ram is used so that we can execute our project in a least possible RAM.50 GB hard disk is used because project sometimes can have more size. Others enhancements are according to the needs.

3.2 Software requirements

Software's can be defined as programs which run on our computer .it act as petrol in the vehicle. It provides the relationship between the human and a computer. It is very important to run software to function the computer. Various software's are needed in this project for its development.

Operating system	: windows
Languages	: HTML,CSS,JAVASCRIPT,PYTHON

Web Framework	: Flask
Editor	:Jupyter Notebook.

3.3 User Interfaces

Notebook Dashboard

When you launch jupyter notebook the first page that you encounter is the Notebook Dashboard.

Notebook Editor

Once you've selected a Notebook to edit, the Notebook will open in the Notebook Editor.

Interactive User Interface Tour of the Notebook

If you would like to learn more about the specific elements within the Notebook Editor, you can go through the user interface tour by selecting Help in the menubar then selecting User Interface Tour.

Edit Mode and Notebook Editor

When a cell is in edit mode, the Cell Mode Indicator will change to reflect the cell's state. This state is indicated by a small pencil icon on the top right of the interface. When the cell is in command mode, there is no icon in that location.

File Editor

Now let's say that you've chosen to open a Markdown file instead of a Notebook file whilst in the Notebook Dashboard. If so, the file will be opened in the File Editor.

3.4 Functional Requirements

The functional requirements describe the interaction between the system and its environment, independent of its implementation and services provided by the system.

- Add records as admin

  To add records to increase the accuracy
- Fill the required details

  To fill required details in the form to get flight delay details
- View results

  To get information about the flight delays

3.5 Non-functional Requirements

3.5.1 Maintainability

Software development team will be the maintenance team for any error or defect.

3.5.2 Predictability

The system can never crash and must produce predictable result.

3.5.3 Reliability

The system will be available 100% of the time.

3.5.4 Understandability

All users can learn to operate major use cases without outside assistance.

3.5.5 Testability

All major use cases must be tested. 100% of the quality requirements must be measurable.

3.5.6 Security

Changing data is only allowed to admins and forbidden to any user. Program run without web, that mean protected from hacker.

3.6 Modules

The modules can be divided into different types.

3.6.1 UI Module Form:

In these module user enter the journey of his details that is date, origin, destination, carrier.

3.6.2 UI module Result:

In these module user can see the prediction result and can make decision on it.

3.7 Usecase Diagram

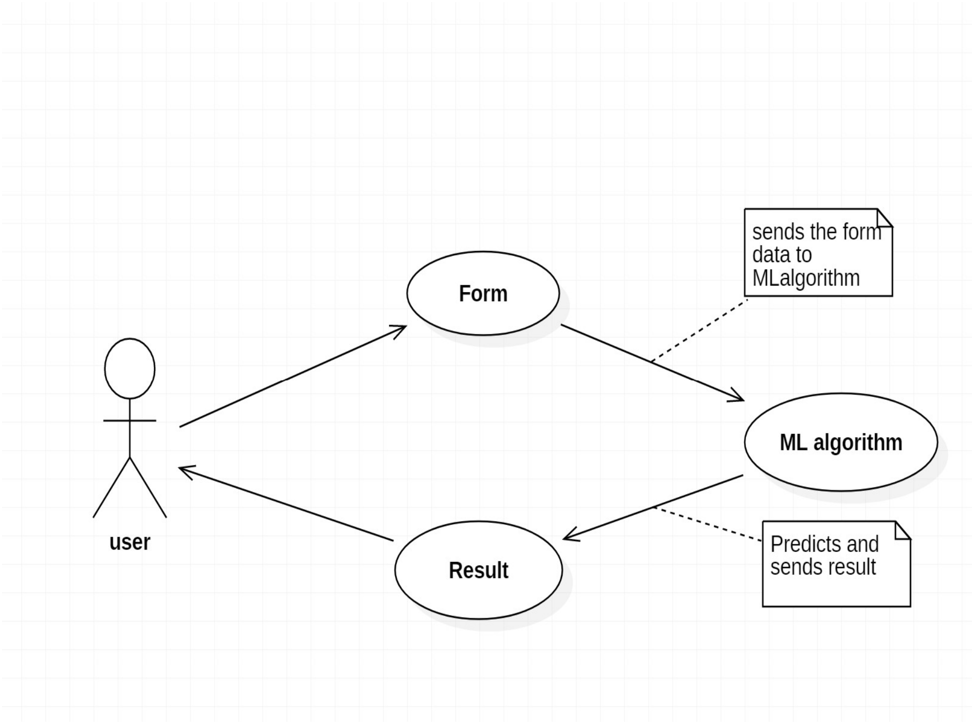


Figure:3.7.1 USECASE

USECASE NAME:	FORM	USECASE ID	1
ACTORS:	USER		
DESCRIPTION:	In these user need to fill the details regarding flight carrier,origin,destination and date.		
Main Flow :	Step 1:Fill the form with flight journey details		
	Step 2:click submit button		
	Step 3:sends form data to ml algorithm.		
POST-CONDITION(s):	Checks main flow step 1 whether flight journey details are filled after clicking main flow step 2		

Table No:3.1 form usecase

USECASE NAME:	ML ALGORITHM	USECASE ID	2
ACTORS:			
DESCRIPTION:	As we have already trained algorithm now when it receives user form it predicts based on user data.		
PRE-CONDITION:			
Main Flow :	Step 1:Receives user input form data		
	Step 2:predicts based on user data		
	Step 3:sends result output to user.		

Table No:3.2 ML Algorithm

USECASE NAME:	RESULT	USECASE ID	3
ACTORS:	USER		
DESCRIPTION:	After receiveing from ml algorithm it display the output in 1 or 0 that is delayed or not delayed.		
Main Flow :	Step 1:Receives output from ml algorithm		
	Step 2:Display output		

Table No:3.3 form result

CHAPTER 4

DESIGN

4.1 Logical View

4.1.1 Class Diagram

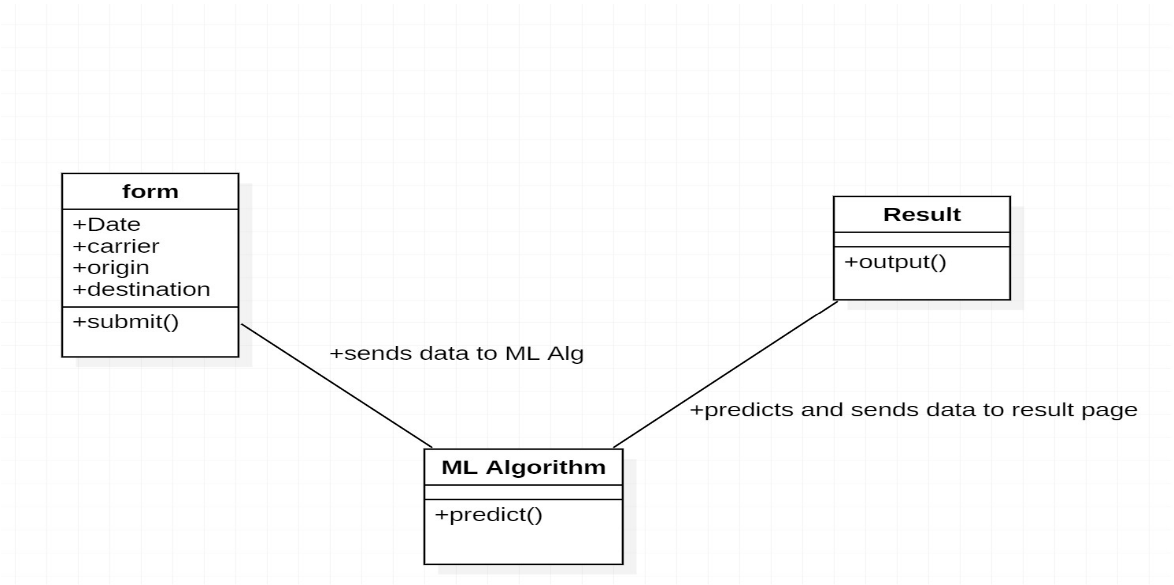


Figure:4.1.1

4.1.2 Sequence Diagram

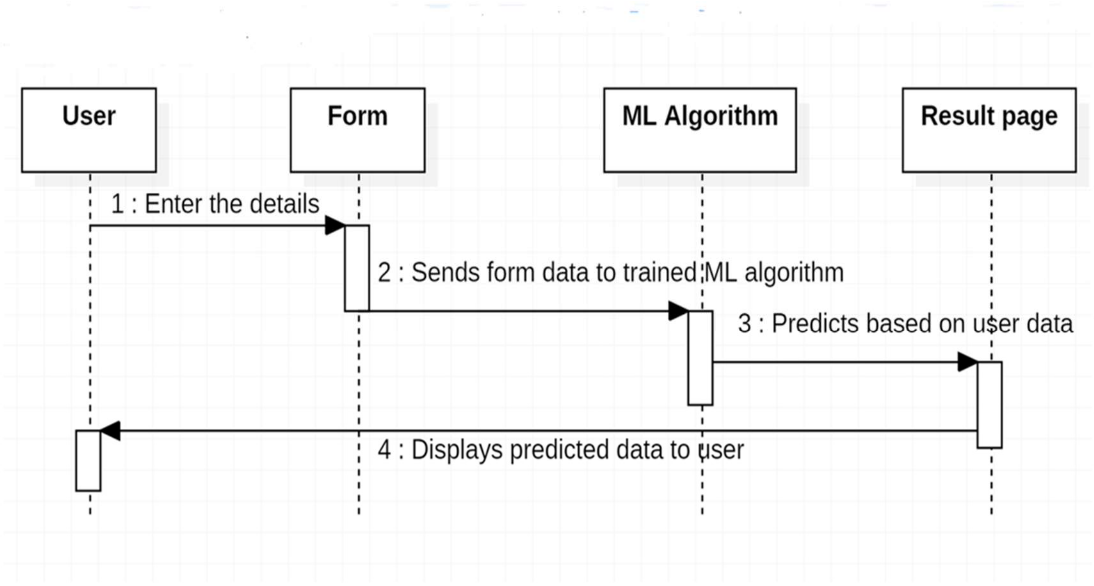

User
Form
ML Algorithm
Result page
1 : Enter the details
2 : Sends form data to trained ML algorithm
3 : Predicts based on user data
4 : Displays predicted data to user

Figure:4.1.2

4.1.3 Collaboration Diagram

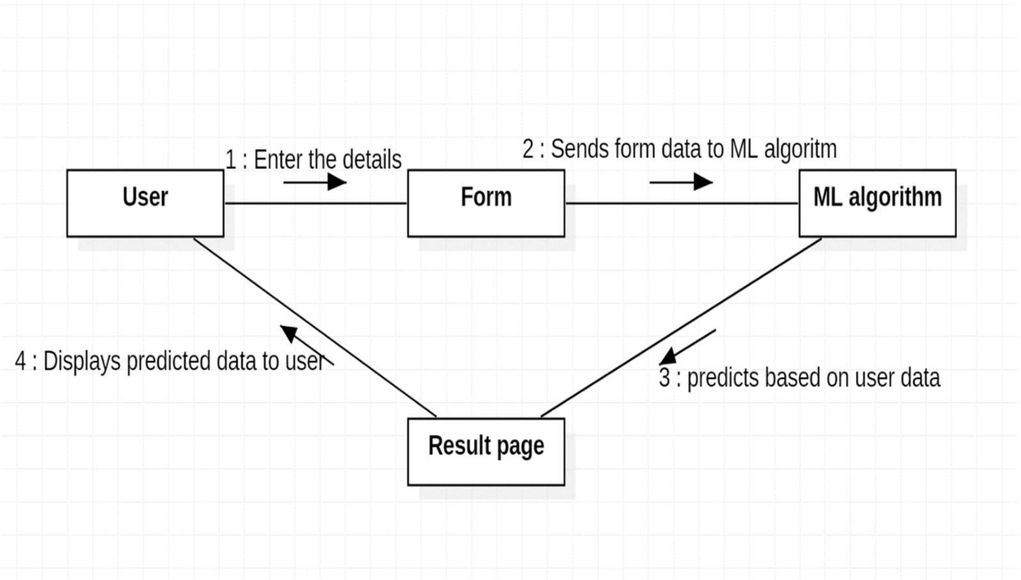

1 : Enter the details
2 : Sends form data to ML algoritm
User
Form
ML algorithm
4 : Displays predicted data to user
3 : predicts based on user data
Result page

Figure:4.1.3

4.2 Process view

Activity Diagram

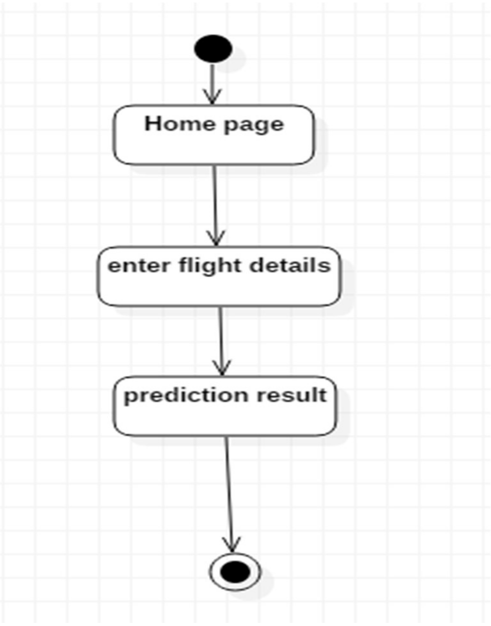


Figure:4.2

4.3 Deployment view

Component Diagram

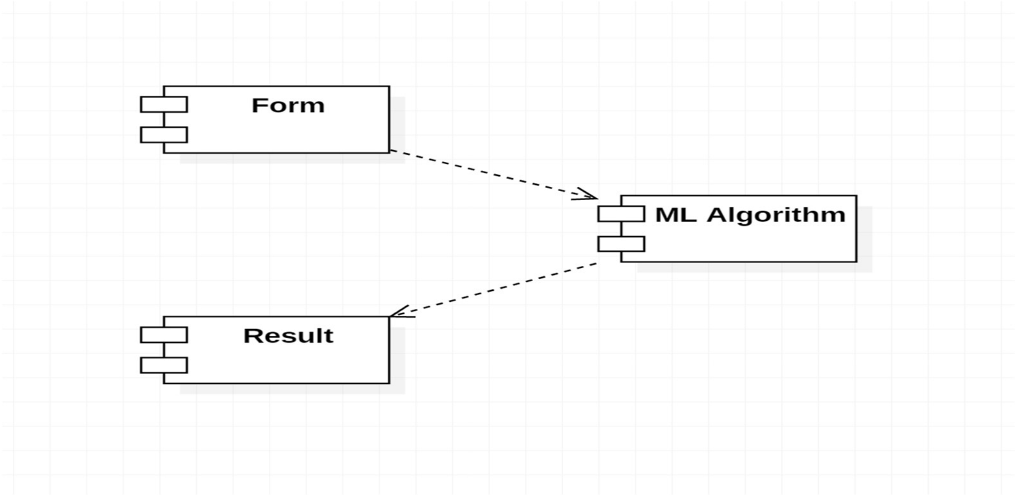


Figure:4.3

4.4 Physical view

Deployment Diagram

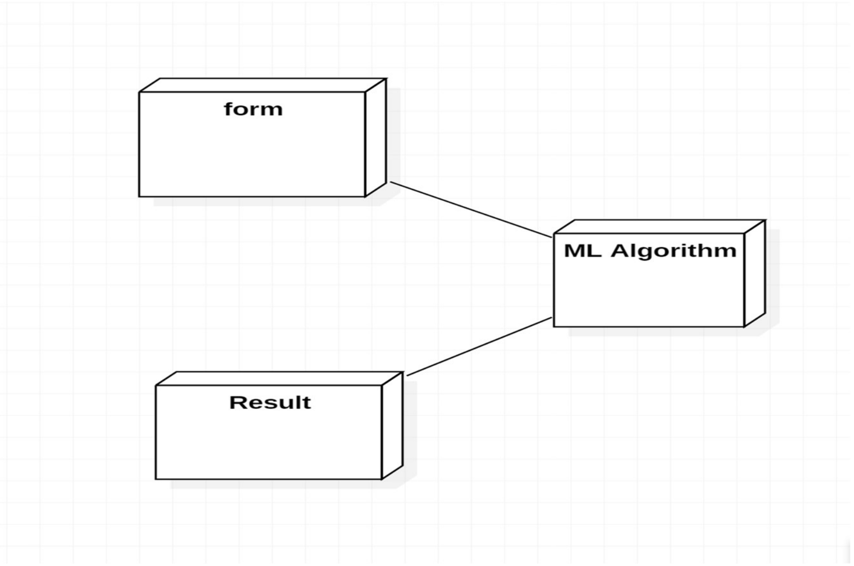


Figure:4.4

CHAPTER 5

CODING

5.1. Language Description

In this project, we have used 4 different languages. They are:

HTML

CSS

JAVA SCRIPT

PYTHON

HTML

HYPER TEXT MARKUP LANGUAGE (HTML) is the standard markup language used for the creation of the web pages. It acts as a skeleton for any web pages. With the help of CSS (Cascading Style Sheet) and JAVASCRIPT, a trio of technologies will be formed, this will be used for designing the outlook of WORLD WIDE WEB (WWW). HTML talks about the structure of the webpage semantically which originally includes the appearance of the document. The building blocks of the HTML pages are the HTML elements. HTML provides a means of creating a structured and formatted document by giving off different tags such as <img>, <h1-h6> (heading tags), <p>, <li>, <a href> and other tags too. The browsers do not display the tags instead they display the content of the page. HTML can execute the programs written in other scripting languages like JAVASCRIPT, which is used to navigate from one webpage to other webpage. The WORLD WIDE WEB CINSORTIUM (W3C) maintains HTML, CSS and many other languages and has encouraged the use of CSS in the wide presentation of HTML since 1997. We have been seeing how HTML has moved in the direction w0ith the help of

CSS. It also includes writing data, links, pictures and even sounds and videos. Here is the sample code for writing HTML code:

CASCADING STYLE SHEET

CSS stands for Cascading Style Sheet. It is a style sheet language used for describing how the document can be presented in a markup language like HTML. CSS is designed in such a way that which enables the presentation and content separately, including layouts, fonts, colors etc. The main advantage of separation is accessibility will be improved, and have control on the specification as per the users wish, enable multiple web pages to share the formatting by specifying the related CSS in a separate .css file.

For example, under pre-CCS HKTML, heading element can be defined with any colour text has shown below::<h1><font color=”yellow”h1>Hello World</font><h1>Using CSS, the same element can be coded using different properties of style instead of HTML attributes: <h1 style=”color; yellow;”>Hello World</h1>. The advantage of CSS Might not be clear, but the power of CSS becomes more visible when the properties of Style are placed in the internal or external CSS file. CSS style information can be in a Separate document or it can be included into an HTML document. Different styles can Be imported and can be applied depending on the device output being used. The main Goal of CSS is to allow its uses to have greater control over its presentation. Depending On the website or the browser, an individual may choose different style sheets that the Designers have provided. An individual has every right to remove or add styles and View the sight using the browsers default styling

JS (JAVA SCRIPT)

JavaScript is a cross-platform, object-oriented scripting language used to make webpages interactive (e.x. having complex animations, clickable buttons, popup menus, etc.). There are also more advanced server side versions of javascript such as Node.Js which allow you to add more functionality to a website than simply downloading files (such as realtime collaboration between multiple computers). Inside a host environment (for example, a web browser), JavaScript can be connected to the objects of its environment to provide programmatic control over them.

JavaScript contains a standard library of objects, such as Array, Date, and Math, and a core set of language elements such as operators, control structures, and statements. Core JavaScript can be extended for a variety of purposes by supplementing it with additional objects; for example:

- Client-side JavaScript extends the core language by supplying objects to control a browser and its Document Object Model (DOM). For example, client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks, form input, and page navigation.
- Server-side JavaScript extends the core language by supplying objects relevant to running JavaScript on a server. For example, server-side extensions allow an application to communicate with a database, provide continuity of information from one invocation to another of the application, or perform file manipulations on a server.

This means that in the browser, JavaScript can change the way the webpage (DOM) looks. And, likewise, Node.js JavaScript on the server can respond to custom requests from code written in the browser.

PYTHON

Python is an interpreted high-level programming language for general purpose pogra-Miming. Python can be used in such a way that it emphasis code readability using significant Whitespace. It provides clear and understandable programming on both small and large CPython, the reference implementation of python, is an open source software and has a com-Munity – based development model. Python and C Python are organized by non-profit PSF (Python Software Foundation). It is an easily understandable language. It often uses English Keywords where as other languages use punctuation. Like many other languages, python do Not use curly brackets and semicolons are optional. The decrease in the indentation signifies the end of present block. Some of the advantages of learning python are:

- Easy – to – learn
- Easy –to- read
- Easy –to-maintain
- Interactive mode
- Databases

- GUI Programming
- Scalable

Some of the better features of python are listed below:

- Its supports functional and structural programming methods as well as OOP.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It can be coincided with C, C++, CORBA, JAVA etc.

The most widely used version of Python is Python 3. Although it cannot be updated anything

Other than security updates, it is widely been used programming language. Applications of

Python are: Web Scraping, Testing, Web Development and Data Analysis.

SAMPLE CODE:

HTML,CSS

```
<!DOCTYPE html>
<html>
<head>
<title>Airline-delay-Prediction App</title>
<link rel="stylesheet" href="style.css">
<link href="https://fonts.googleapis.com/css?family=Raleway" rel="stylesheet">
</head>
<h1 align="center">FLIGHT DELAY PREDICTION </h1>
<body >
  <p align="left">  CARRIER :10-UNITED AIRLINES <br>
                ORIGIN  :210-ATLANTA AIRPORT <br>
```

```
                    :219-CHICAGO O'HARE  AIRPORT<br>
                    :36-DENVER AIRPORT<br>
             DEST :101-GEORGE BUSH AIRPORT<br>
                      :12-ALBANY AIRPORT<br>
<div class="container">
  <div class="content" align="center">
        <form action = "{{ url_for('page') }}" method="POST" id="flight_predict"
align='center'>
                                                  Select Year:&nbsp;&nbsp;<input
name="year" type ="text"    /><br><br>
                          Select month:&nbsp;&nbsp;<input name="month" type
="text"   /><br><br>
                          Select day:&nbsp;&nbsp;<input name="day" type ="text"
/><br><br>
                          Select Date:&nbsp;&nbsp;<input name="date" type ="date"
/><br><br>
                          Select carrier:&nbsp;&nbsp;<input name="carrier"
type="text"   ><br><br>
                          Select Origin:&nbsp;&nbsp;<input name="origin" type
="text"   /><br><br>
                          Select Destination:&nbsp;&nbsp;<input name="dest" type
="text"   /><br><br>
                  <p align="center"> <input type ="submit" value="submit"
form="flight_predict" /> </p>
        </form>
  </div>
</div>
</body>
</html>
```

HTML

```
<!DOCTYPE html>
<html>
<title>Airline-delay-Prediction App</title>
<body>
<h3> Delay Prediction: {{ prediction }} </h3>
<p><h4> [0] = No Delay </h4></p>
<p><h4> [1] = Delay  </h4></p>

</body>
</html>
```

PYTHON

```
from flask import Flask,render_template,request
import numpy as np
import pandas as pd
import csv
import os
import datetime
from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn import cross_validation
from sklearn.metrics import confusion_matrix, roc_curve
from sklearn.preprocessing import LabelEncoder
import matplotlib
from matplotlib import pyplot as plt
import matplotlib.pyplot as plt
from sklearn.grid_search import GridSearchCV
```

```
app = Flask(__name__)

@app.route('/')
def home():
  return render_template('frntproject1.html')
@app.route('/page',methods=['GET','POST'])
def page():
  data = {}
  if request.form:

            form_data = request.form
            data['form'] = form_data
            year = form_data['year']
            month = form_data['month']
            day = form_data['day']
            datobj = datetime.datetime.strptime(form_data['date'], '%Y-%m-%d')
            date = datetime.datetime.strftime(datobj,'%Y%m%d')
            carrier = form_data['carrier']
            origin = form_data['origin']
            dest = form_data['dest']
            print(date)

            #datee=datetime.date(2002, 12,4)
            #datee.strftime('%m%d%Y')
            #date = datetime.date(2002, 12,4).strftime("%Y%m%d")
            print(data)
            fdata = pd.read_csv('564220792_T_ONTIME.csv')
  fdata1=pd.DataFrame(columns=['YEAR','MONTH','DAY_OF_MONTH','FL_DATE
','CARRIER','ORIGIN','DEST','ARR_DEL15'])
            fdata1['YEAR']=fdata['YEAR']
            fdata1['MONTH']=fdata['MONTH']
            fdata1['DAY_OF_MONTH']=fdata['DAY_OF_MONTH']
            fdata1['FL_DATE']=fdata['FL_DATE']
            fdata1['CARRIER']=fdata['CARRIER']
            fdata1['ORIGIN']=fdata['ORIGIN']
            fdata1['DEST']=fdata['DEST']
            fdata1['ARR_DEL15']=fdata['ARR_DEL15']
            from sklearn.preprocessing import LabelEncoder
            le = LabelEncoder()
            fdata1["Carrier_Name"] = le.fit_transform(fdata1["CARRIER"])
            Carrier = list(le.classes_)
            fdata1["Origin_Point"] = le.fit_transform(fdata1["ORIGIN"])
            Origin = list(le.classes_)
            fdata1["Destination"] = le.fit_transform(fdata1["DEST"])
            Dest = list(le.classes_)
            fdata1.drop(['CARRIER','ORIGIN','DEST'], axis=1, inplace=True
  fdata1["FL_DATE"] = fdata1["FL_DATE"].apply(lambda x: int(".join(x.split("-"))))
            np.random.seed(10)
            Delay_YesNo = fdata1['ARR_DEL15']
            fdata1.drop(['ARR_DEL15'], axis=1, inplace=True)
            data_part2 = pd.DataFrame(fdata1)
            Delay_YesNo1=np.nan_to_num(Delay_YesNo)
            from sklearn.model_selection import train_test_split, cross_val_score,
GridSearchCV
            from sklearn.ensemble import RandomForestClassifier
            from sklearn import cross_validation
            from sklearn.metrics import confusion_matrix, roc_curve
            import matplotlib
            from matplotlib import pyplot as plt
            import matplotlib.pyplot as plt
            X_train, X_test, Y_train, Y_test = train_test_split(data_part2, Delay_YesNo1, test_size=0.2, random_state=42)
            from sklearn.grid_search import GridSearchCV
            rf = RandomForestClassifier()
            rf.fit(X_train, Y_train)
            df=pd.read_csv("564220793_T_ONTIME.csv")

  fdata3=pd.DataFrame(columns=['YEAR','MONTH','DAY_OF_MONTH','FL_DATE','CARRIER','ORIGIN','DEST'])
            fdata3['YEAR']=df['YEAR']
            fdata3['MONTH']=df['MONTH']
            fdata3['DAY_OF_MONTH']=df['DAY_OF_MONTH']
            fdata3['FL_DATE']=df['FL_DATE']
            fdata3['CARRIER']=df['CARRIER']
            fdata3['ORIGIN']=df['ORIGIN']
            fdata3['DEST']=df['DEST']
            fdata3["FL_DATE"] = fdata3["FL_DATE"].apply(lambda x: int(".join(x.split("-"))))#Formatting date for convinience
            from sklearn.preprocessing import LabelEncoder
            se = LabelEncoder()
            fdata3["Carrier_Name"] = se.fit_transform(fdata3["CARRIER"])
            Carrier = list(se.classes_)
            fdata3["Origin_Point"] = se.fit_transform(fdata3["ORIGIN"])
            Origin = list(se.classes_)
            fdata3["Destination"] = se.fit_transform(fdata3["DEST"])
            Dest = list(se.classes_)
```

```
            fdata3.drop(['CARRIER','ORIGIN','DEST'], axis=1, inplace=True)
            input_data = np.array([year,month,day,date,carrier,origin,dest])
            print(input_data)
            fdata4=rf.predict(input_data.reshape(1,-1))
            print(fdata4)
            return render_template("test.html",prediction=fdata4)
if __name__ == '__main__':
  app.run(debug = True).
```

CHAPTER 6

TESTING

Glen Myers states a number of rules that can serve well as testing objectives. Testing is a process of executing a program with the intent of finding an error. We can test our project using various methods but the main objective is that when:-

The first form that displays should be filled completely.

A successful submission is done only when form is filled related datatype for every input..

6.1 TESTING OBJECTIVES:

Unit testing is the testing of the individual components (units) of the software. Unit testing is conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases. When developing a strategy for unit testing, there are three basic organizational approaches that can be taken. These are top down, bottom up and isolation. In our case we simply use top down approach. The Form , which takes input from user is checked for each and every input. And form data is passed to algorithm only when every input is correct. And in the second mode our motive is to obtain a predicted output to user.

6.2 Integration Testing:

After the unit testing we have to perform integration testing. The goal here is to see if modules can be integrated properly, the emphasis being on testing interfaces between modules. After the modules are connected we have perform the total testing.

6.3 SYSTEM TESTING:

System testing is the process of executing software in a controlled manner, in order to answer the questions "Does the software behave as specified?" System testing is often used in association with the terms verification and validation .Verification is the checking of items, including software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Validation: Are we doing the right job?

Verification: Are we doing the job right?

The test strategies will include five different types of testing as describes below:-

1. **Logical Testing:** This will be used to test every aspect of both modes, report and query as soon as it is implemented, using valid, invalid and extreme data test data will be added to test each code module and results compared with the expected results. Sufficient data will be added to ensure that there is at least one entry in each category. Subsequent tests will often involved adding new data, which will be deleted when the test works satisfactorily.

2. **Functional Testing: -** In this menu items were tested to ensure no functions has been missed out. This is done for the smooth working of the project.

3. **System Testing: -** This is done after the completion of system; all the queries were carried out again to ensure that no errors have been introduced.

CHAPTER 7

SCREEN SHOTS

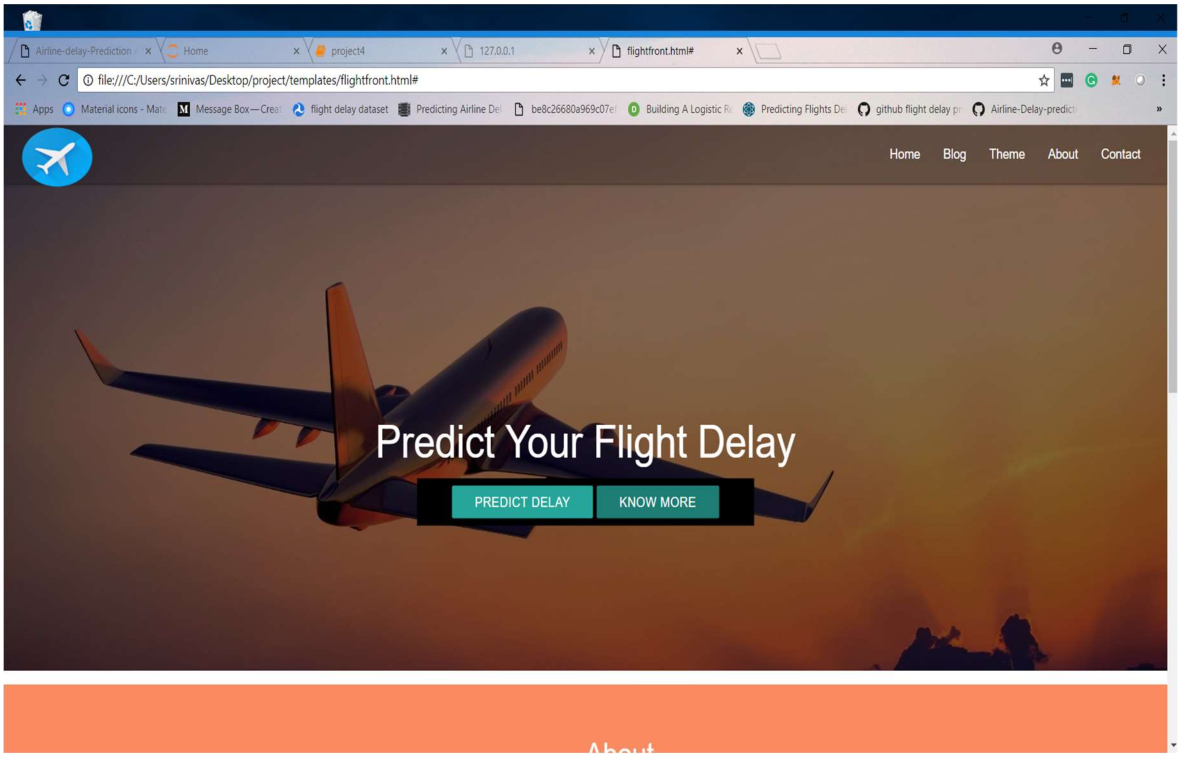
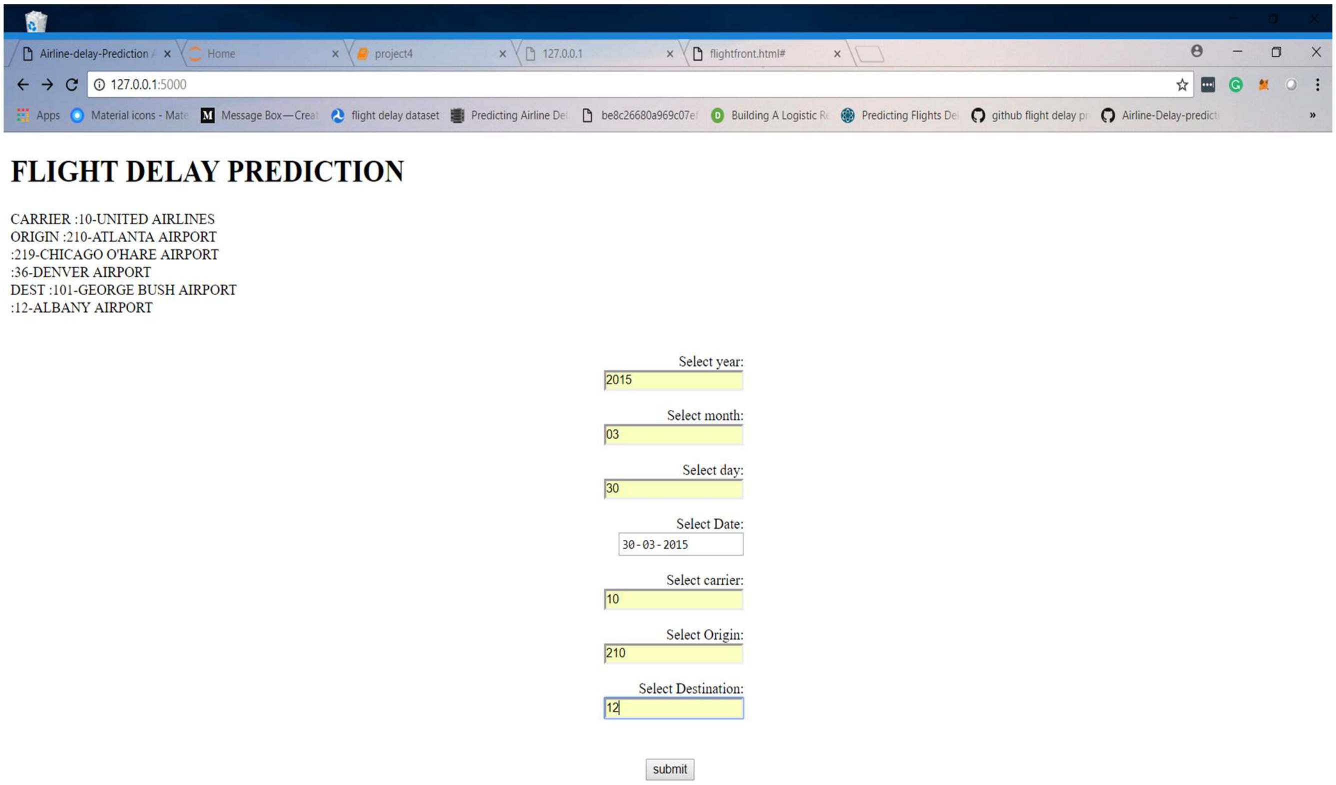


Fig 7.1 home page



FLIGHT DELAY PREDICTION

CARRIER :10-UNITED AIRLINES
ORIGIN :210-ATLANTA AIRPORT
:219-CHICAGO O'HARE AIRPORT
:36-DENVER AIRPORT
DEST :101-GEORGE BUSH AIRPORT
:12-ALBANY AIRPORT

Select year: 2015

Select month: 03

Select day: 30

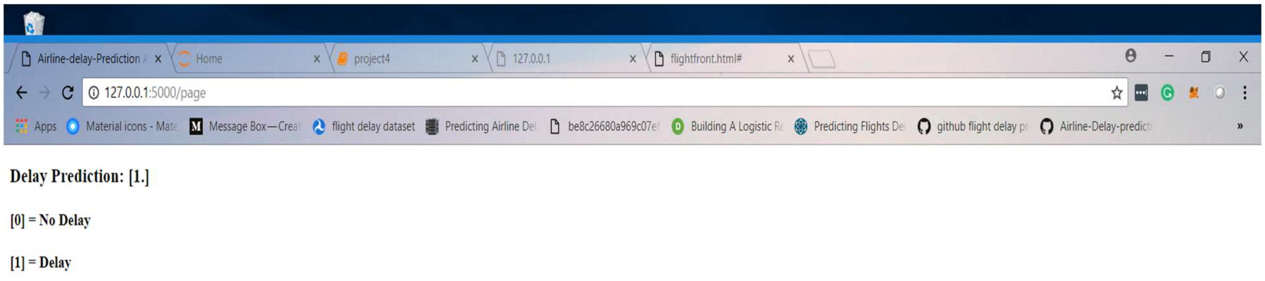
Select Date: 30-03-2015

Select carrier: 10

Select Origin: 210

Select Destination: 12

Fig 7.2 Input Window



Delay Prediction: [1.]

[0] = No Delay

[1] = Delay

Fig 7.3 Output Window

CHAPTER 8

CONCLUSIONS

This study is devoted to develop a predictive model to forecast flight delays. Data spanning for over 5 lakh observations including US domestic flights variables was used. Model based random forest algorithms are created and tested in Jupyter notebook concluding that Random forest model performs best prediction.

Overall, our models are only of limited utility since none were capable of correctly predicting flight delays with both precision and recall greater than 50%. This seemingly low performance is likely due to the many causes of flight delays being outside the scope of our data. It is unclear if it is even possible to predict whether or not a flight will be delayed so far in advance, as we have set up the problem, because so many of the causes of delays (e.g. mechanical issues and weather) cannot be known in advance. Despite this, we were successful in creating models that outperform baseline models, and perform at least about as well as prior work, even when we often use less information, and generalize to more airports. Although imperfect, this model still makes potentially useful predictions about which flights are more or less likely to be delayed. Future work may well be able to further improve this kind of flight delay prediction at time of booking, perhaps by further work on feature design and collecting other informative features about flights, and/or work on more sophisticated modeling techniques. Although the model gives very good prediction accuracy, more variables can be considered to develop a predictive model. For example, Weather data can be extracted and used to better develop a predictive model for flight delay. The future scope of this study involves various approaches that can be used to analyze the data. Principal component analysis or transformation can be done to uncover hidden relations between variables. In addition, since the data is not exactly linear, artificial neural networks or Support vector machines can be used to analyze the effect of various variables on flight delay.

REFERENCES

[1] Michael Ball, Cynthia Barnhart, Martin Dresner, Mark Hansen, Kevin Neels, Amedeo Odoni, Everett Peterson, Lance Sherry, Antonio Trani, Bo Zou (2010). 'Total Delay Impact Study', The National Center of Excellence for Aviation Operation Research (NEXTOR)

[2] (Online) Bureau of Transportation Statistics (BTS) Databases and Statistics. http://www.transtats.bts.gov/

[3] Breiman, L. Random Forests. Mach. Learn. 2001, 45, 5-32.

[4] Breiman, L.; Friedman, J. H.; Olshen, R. A.; Stone, C. J. Classification and Regression Trees; Chapman & Hall/CRC: Boca Raton, 1984

[5] A Complete Introduction to the Python Language , By Mark Summerfield