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.

1.1 Block Diagram

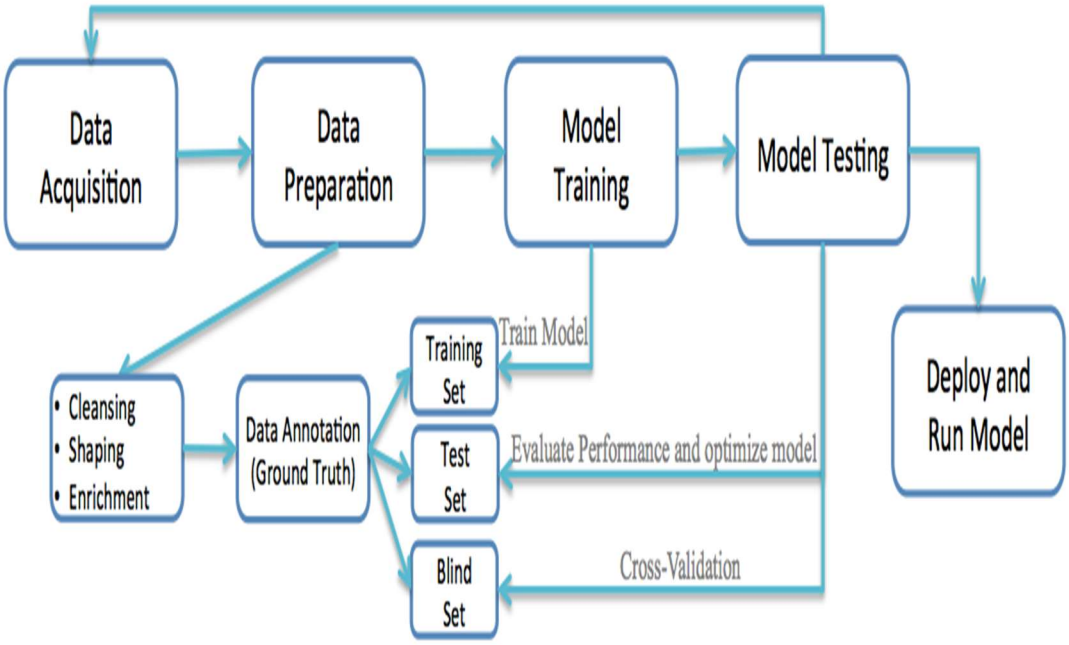

Fig.No:1.1 Block Diagram for flight delay prediction.

In the above block diagram first data acquisition is to be done and next data preparation i.e. data preprocessing which includes cleansing, shaping, enrichment then training which consists of training set, test set and then test and final step is to deploy.

LITERATURE SURVEY

[1] Suvojit Manna , Sanket Biswas , Riyanka Kundu , Somnath Rakshit , ” A statistical approach to predict flight delay using gradient boosted decision tree”, IEEE.

Supervised machine learning algorithms have been used extensively in different domains of machine learning like pattern recognition, data mining and machine translation. Similarly, there has been several attempts to apply the various supervised or unsupervised machine learning algorithms to the analysis of air traffic data. However, no attempts have been made to apply Gradient Boosted Decision Tree, one of the famous machine learning tools to analyse those air traffic data. This paper investigates the effectiveness of this successful paradigm in the air traffic delay prediction tasks. By combining this regression model based on the machine learning paradigm, an accurate and sturdy prediction model has been built which enables an elaborated analysis of the patterns in air traffic delays. Gradient Boosted Decision Tree has shown a great accuracy in modeling sequential data.

https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8264986

[2] Anish M. Kalliguddi , Aera K. Leboulluec , ”Predictive Modeling of Aircraft Flight Delay”, hrpub.

A predictive model using MLR is developed on the training data. It was observed that all the variables were significant with an r-square of 0.84. That means our developed model can explain 84% of the variation in the data. A stepwise regression was applied after the main model was developed and it ended up giving same results as the original model. Forward/backward stepwise regression is always used on the actual model to see the effect of each predictor variable on the response variable.

http://www.hrpub.org/download/20171130/UJM3-12110417.pdf

[3] Pranalli Chandraa and Prabakaran.N and Kannadasan.R , "Airline Delay Predictions using Supervised Machine Learning", International_Journal_of_pure_and_Applied_Mathematics.

Supervised Learning algorithm here will model relationships and dependencies between the aimed prediction output and the input features, such that I'll be predicting the output values for new data based on the relationships which are learned from the previous data set. Supervised Learning problems can be further categorized into following problems • Classification – It is a type problem in which the output variable is an entire category itself, such as "Win" or "Lose", the entire input data is classified into the category variables; it is generally used largely for recommendation problems • Regression – It is a type of problem is which the output variable is a real value, such as few raw data values related to something. This is the problem type massively used for prediction analysis, and hence will be used in this project.
https://www.researchgate.net/journal/1314.3395_International_Journal_of_Pure_and_Applied_Mathematics

[4] Yi Ding, "Predicting flight delay based on multiple linear regression", bioscience

In statistics, linear regression is an approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X. The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression. In linear regression, the relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models.
https://iopscience.iop.org/article/10.1088/1755-1315/81/1/012198

SOFTWARE REQUIREMENTS SPECIFICATIONS

3.1 INTRODUCTION

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of Use Cases that describe user interactions that the software must provide. Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules

3.2 PURPOSE

A software requirements specification (SRS) is a detailed description of a software system to be developed with its functional and non-functional requirements. The SRS is developed based the agreement between customer and contractors. It may include the Use Cases of how user is going to interact with software system.

3.3 SCOPE

One of the most important items in the requirements specification is the precise scope definition of the project. The Software Requirements Specification is a communication tool between users and software designers. The specific goals of the SRS are Facilitating reviews and Describing the scope of work. Accuracy of this is important since SRS is also used for estimation and costing. If the project is for the development of a product, product vision defines the scope and the target user base of the product. The software requirements specification document lists sufficient and necessary requirements for the project development. To derive the requirements, the developer needs to have clear and thorough

understanding of the products under development. This is achieved through detailed and continuous communications with the project team and customer throughout the software development process.

3.4 FUNCTIONAL REQUIREMENTS

- **Form**

  User need to enter the details of flight ID, origin point and destination point in the user interface and submit the details to the model.

- **View results**

  Based on the details given by the user result is obtained in the form of.

3.5 USE CASE DIAGRAM

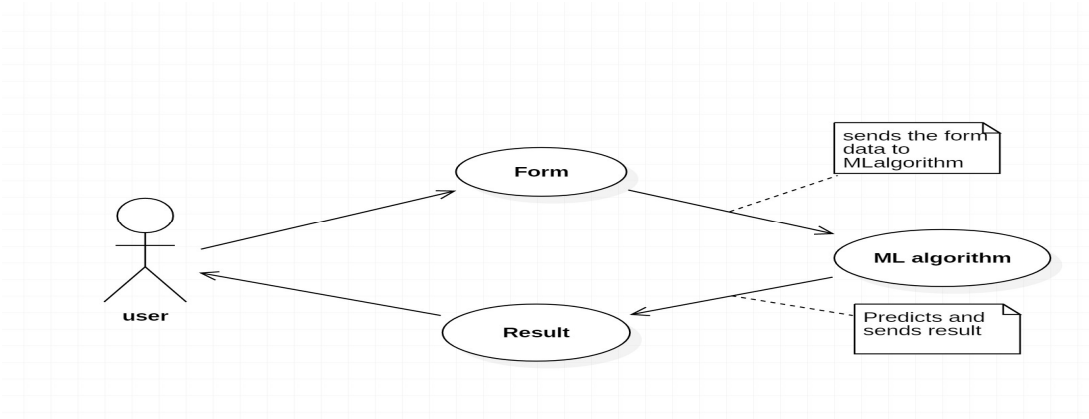

Fig No:3.5 Use Case Diagram for predicting flight delays.

The above figure represents use case diagram of proposed system, where user inputs flight details, train and test. The algorithm works to identify delay of flight. The actor and use case is represented. An eclipse shape represents the use case namely input form, ML algorithm, Result.

3.6 USE CASE DESCRIPTION TABLES

USECASE NAME	FORM	USECASE ID	1
ACTORS	USER		
Description	As we have already trained the algorithm now when it receives user form it predicts based on the user data.		
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	Checks main flow step 1 whether flight journey details are filled after clicking main flow step 2		

Table No:3.6.1 Use Case table for form.

USECASE NAME	ML ALGORITHM	USECASE ID	2
ACTORS	Algorithm		
DESCRIPTION	As we have already trained algorithm now when it receives user form it predicts based on user data.		
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:3.6.2 Use Case table for ML Algorithm.

USECASE NAME	RESULT	USECASE ID	3
ACTORS	USER		
DESCRIPTION	After receiving 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:3.6.3 Use Case table for result page.

3.7 NON-FUNCTIONAL REQUIREMENTS

3.7.1 MAINTAINABILITY

It is easy to maintain the system by adding new records to the dataset and loading the dataset and getting the results based on the given input easily.

3.7.2 SCALABILITY

The software must easily be transferred to another environment, including install ability. It is easily portable as it is implied on a regular computer. The user can access the computer from the place where the software was installed.

3.7.3 PERFORMANCE

Less time for detection of delay of flight once the input is arrived from the user. Similarly, the training time also less as we are giving entire data in a single dataset.

3.7.4 ACCURACY

The accuracy generated by our work is outperformed than any other existing models. We can predict the delays accurately through our proposed system.

3.7.5 SECURITY

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

3.8 SOFTWARE REQUIREMENTS

Operating system : Windows

Languages : HTML,CSS,JAVASCRIPT,PYTHON

Web Framework : Flask

Editor :Jupyter Notebook.

3.9 HARDWARE REQUIREMENTS

RAM : 2 GB

Hard disk : 50 GB

Pocessor :Intel core i3

DESIGN

4.1 Class Diagram

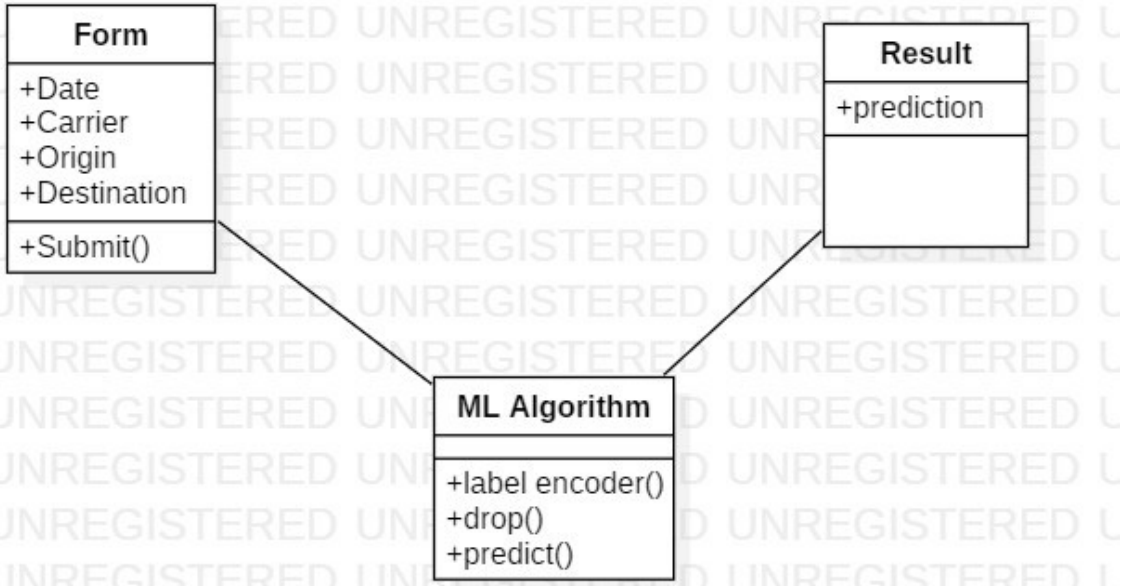

Fig 4.1 Class Diagram for predicting flight delays.

The above class diagram consists of different classes named form class to collect the input details like flight ID, origin and destination from the user and ML Algorithm class to predict the result and result class to show the output.

4.2 Sequence Diagram

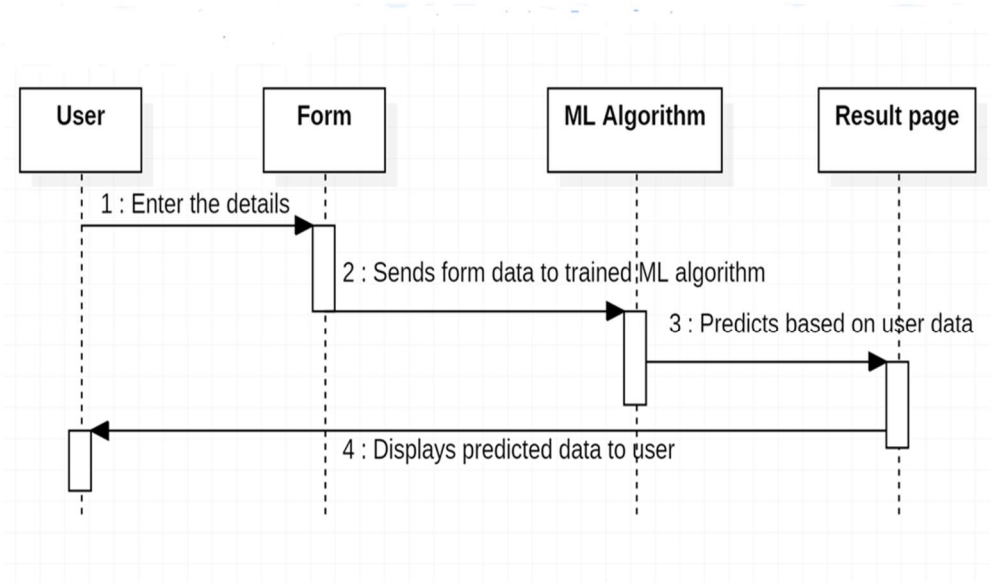

Fig 4.2 Sequence Diagram for predicting flight delays.

The above sequence diagram depicts the sequence diagram for Predicting flight delays. Firstly, the user enters the details of the flight in the user interface and these input details are fetched to the ML Algorithm for prediction purpose and the result is obtained.

4.2.1 Sequence Diagram for user

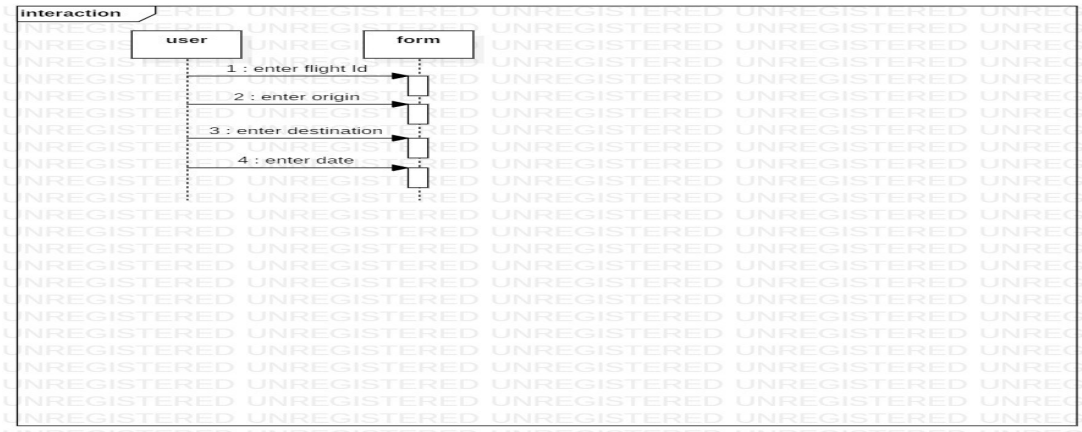

Fig.No:4.2.1 Sequence Diagram for user.

The above sequence diagram depicts the sequence diagram for Predicting flight delays. Firstly, the user enters the details of the flight in the user interface as input details.

4.2.2 Sequence Diagram for result

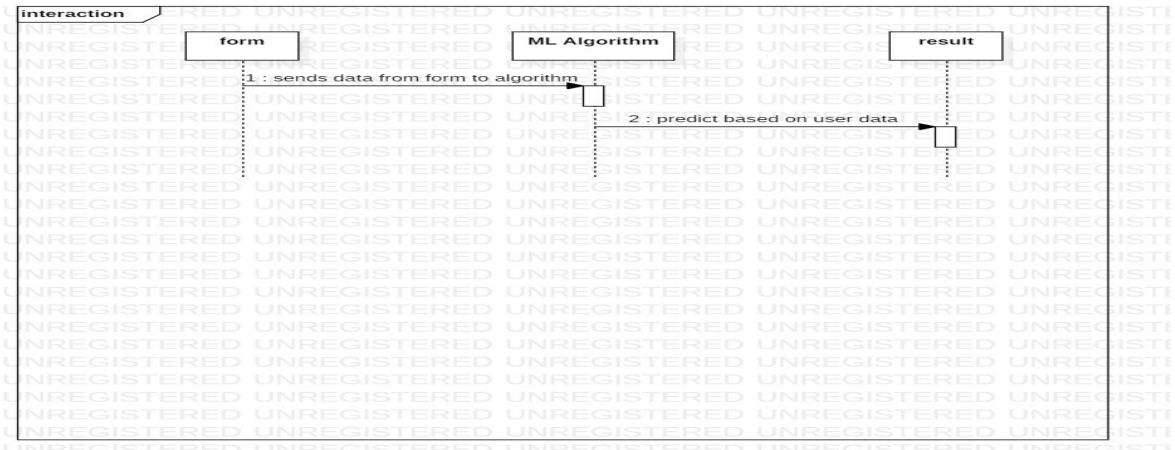

Fig.No:4.2.2 Sequence Diagram for result

The above sequence diagram depicts the sequence diagram for Predicting flight delays. Firstly, the user enters the details of the flight in the user interface as input details and these inputs are fetched to the algorithm and result is obtained.

4.3 Activity Diagram

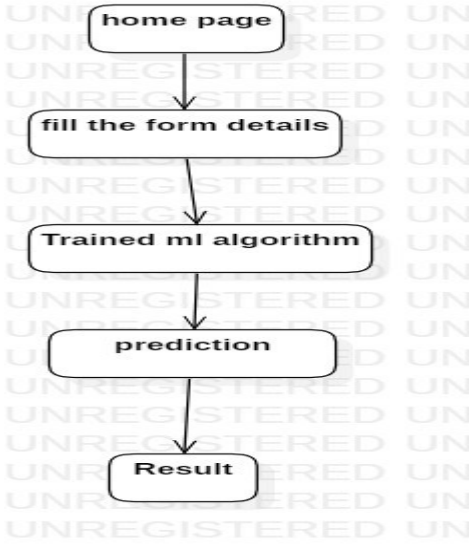

Fig.No:4.3 Activity Diagram for predicting flight delays.

The above Activity diagram depicts the Activity diagram for Predicting flight delays, it represents the flow of control from user to training model, testing and getting the output.

4.4 Component Diagram

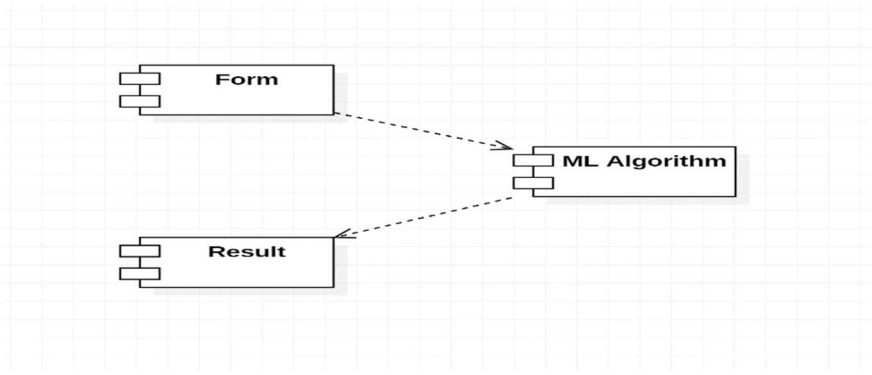

Fig.No:4.4 Component Diagram for predicting flight delays.

A component diagram depicts the components in the model. above component diagram consists of three components form, result and ML Algorithm.

4.5 Deployment Diagram

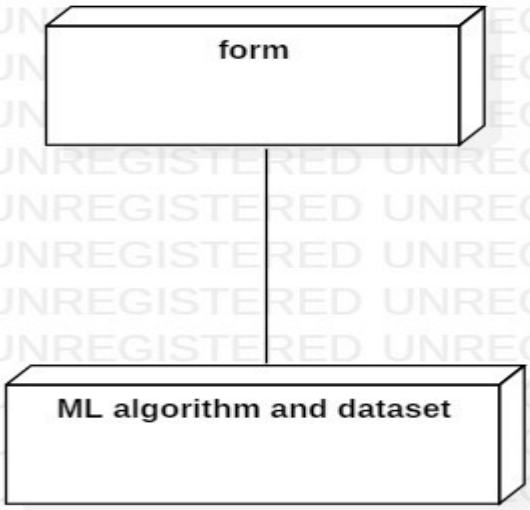

Fig.No:4.5 Deployment Diagram for predicting flight delays.

The above Deployment diagram depicts the deployment diagram for model it consists of two different components form, ML algorithm and dataset

CODING

Form

```
<!DOCTYPE html>
<html>
<HEAD>
<title>Airline-delay-Prediction App</title>
<style>
label{
display:inline-block;
width:200px;
margin-right:30px;
text-align:right;
}
input{
}
fieldset{
border:none;
width:500px;
margin:0px auto;
}
</style>
</HEAD>
<h1 align="flight">FLIGHT DELAY PREDICTION </h1>
<body class="parallax">
  <p align="left">  CARRIER :10-UNITED AIRLINES <br>
                        ORIGIN  :210-ATLANTA AIRPORT <br>
                       :219-CHICAGO O'HARE  AIRPORT<br>
```

:36-DENVER AIRPORT

DEST :101-GEORGE BUSH AIRPORT

:12-ALBANY AIRPORT

```
<form action = "{{ url_for('page') }}" method="POST" id="flight_predict" align='center'>
<fieldset>
<p><label for="txtselectyear">Select year:<input name="year" type ="text"  /> </p>
<p><label for="txtmonth">Select month:<input name="month" type ="text"  /> </p>
<p><label for="txtday">Select day:<input name="day" type ="text"  /> </p>
<p><label for="txtdate">Select Date:<input name="date" type ="date"  /> </p>
<p><label for="carrier">Select carrier:<input name="carrier"  type="text"  >
<p><label for="txtorigin">Select Origin:<input name="origin" type ="text"  /> </p>
<p><label for="txtdest">Select Destination:<input name="dest" type ="text"  /> </p>
</fieldset>
</form>
<p align="center"> <input type ="submit" value="submit" form="flight_predict"
value="Submit" align="left"  /> </p>
<!--<p> Select arrival time <input name="arrtime" type ="time" value="22:30" /> </p>
<p> Select departure time <input name="depttime" type ="time" value="13:30" /> </p>-->
</body>
</html>
```

ML algorithm

```
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']
```

```
fdata = pd.read_csv('564220792_T_ONTIME.csv')
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")
return render_template("test.html",prediction=fdata4)
if __name__ == '__main__':
  app.run(debug = True).
```

Result

```
<!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>
```

TESTING

6.1 INTRODUCTION

We had done testing using white box testing in which testing is done on the following conditions

- The given input in the form is according to its type
- The training on input data is finished.
- After training whether prediction of delay has been done or not.

6.2 Form

In these the testing done on whether the user entered input is of correct type or not. If each entry in form is of correct type, then the details entered will be sent to algorithm else the details will not be sent to algorithm.

6.3 Result

In these after successful submission of details we get prediction based on given details and trained data. The output will be either zero or one which zero indicates no delay and one indicates delay**.**

Test case id	1
Test case description	The user should enter the details of journey
Expected result	Submits data only when entered data is correct type.
Actual result	Submits data only when entered data is correct type.
Status	Pass

Table 6.3.1 Test Case for input details.

Test case id	2
Test case description	After successful entering of details user get prediction(0 0r 1)based on given input.
Expected result	Displays zero if no delay else 1 if delay
Actual result	Displays zero if no delay else 1 if delay
Status	Pass

Table 6.3.2 Test Case for prediction.

GUI SCREENS

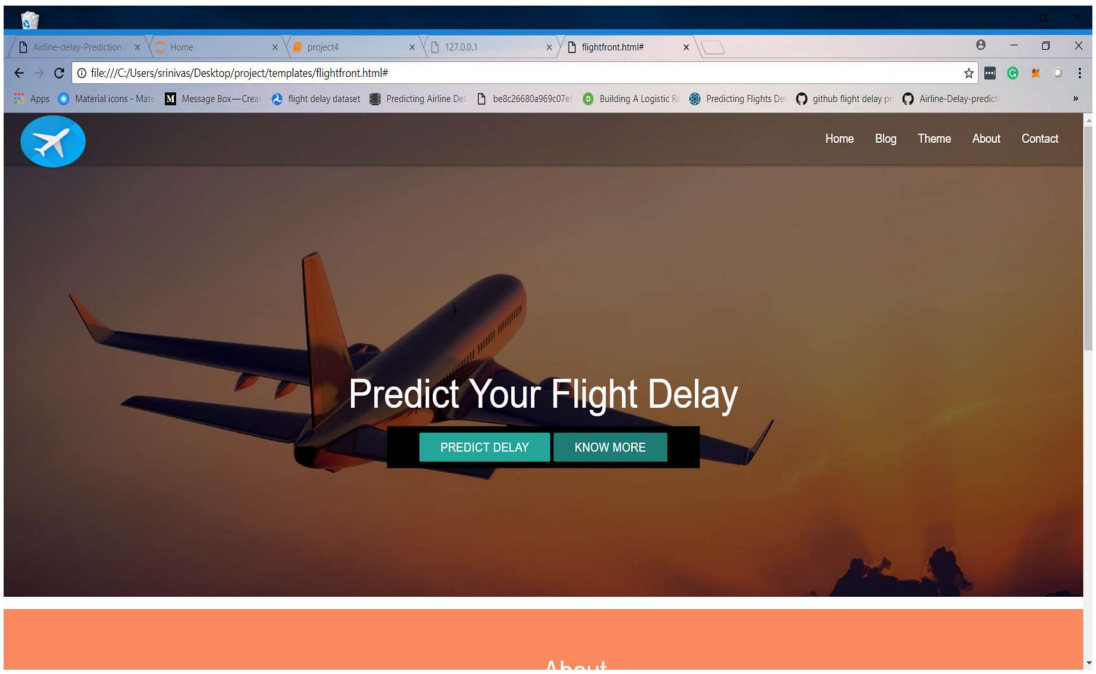

Fig 7.1 home page for predicting flight delays

The above figure represents home page of the project

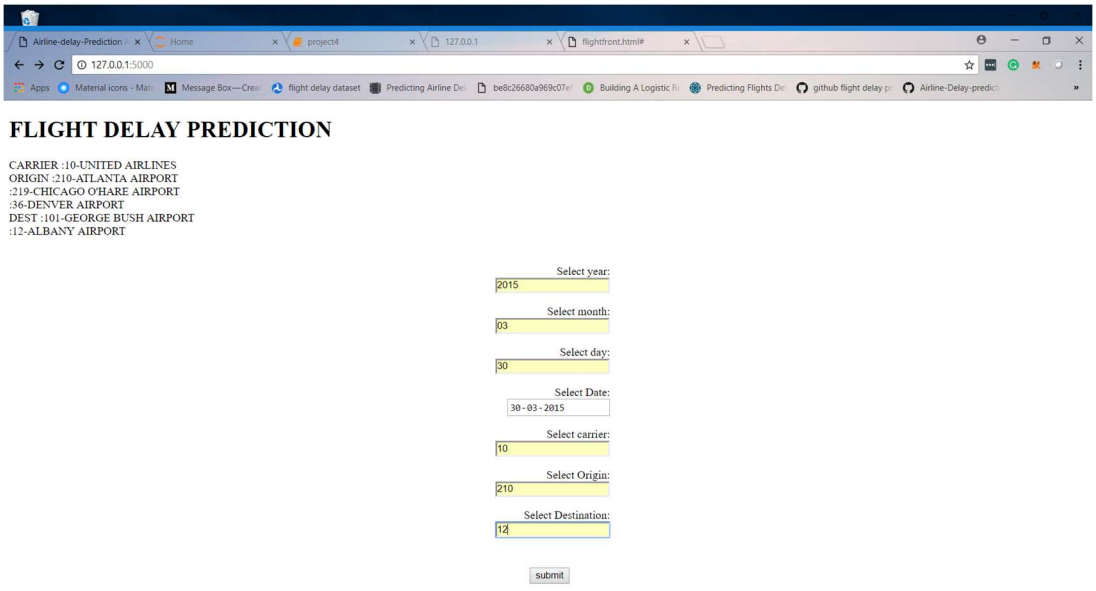

Fig 7.2 Input Window to enter details

The above figure is to fill the flight details such as date ,origin ,destination

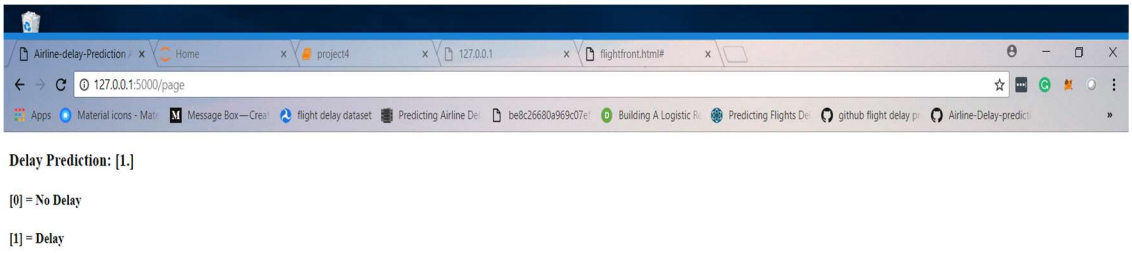

Fig 7.3 Output Window for result for predicting flight delays

The above figure displays the output i.e. prediction of delay.

CONCLUSION AND FUTURE SCOPE

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

REFERENCES / BIBLIOGRAPHY

9.1 Papers

[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.

[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

9.2 UrlS

- [1] http://www.hrpub.org/download/20171130/UJM3-12110417.pdf
- [2] https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8264986
- [3] http://www.transtats.bts.gov/
- [4] https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/
- [5]https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d

9.3 Books

1. Grady Booch, James Rumbaugh, Ivar Jacobson : The Unified Modeling Language User Guide, Pearson Education 2nd Edition.

2.Object oriented Analysis, Design and Implementation, B.Dathan,S.Ramnath,Universities Press

3. The craft of Software testing – Brian Marick, Pearson Education.