OPTIMIZING FLIGHT BOOKING DECISIONS THROUGH MACHINE LEARNING PRICE PREDICTIONS

Submitted in partial fulfillment of requirements for the award of the Degree

Bachelor of Computer Science

In the faculty of Computer Science of Bharathiar University, Coimbatore

Submitted by

TEAM ID: NM2023TMID19697

G. SAKTHIVEL

V. NISHANTH

M. MANOJKUMAR

J. MANIKANDAN

Under the guidance of

Dr. T. RAJENDRAN, M.SC., M.Phil., Ph.D.,

Assistant professor

Department of Computer Science

DEPARTMENT OF COMPUTER SCIENCE
GOVERNMENT ARTS AND SCIENCE, COLLEGE, KANGAYAM,
(Affiliated To Bharathiar University, Coimbatore)

APRIL-2023

GOVERNMENT ARTS AND SCIENCE COLLEGE , KANAGAYAM

NAAN MUDHALVAN PROJECT WORK

(AFFILIATED TO BHARATHIAR UNIVERSITY)

COIMBATORE

TITLE : Optimizing Flight Booking Decisions Through ML Price Predictions

This is to certify that this is a bonofide record of work done by the above students of III B.Sc (CS) Degree **NAAN MUDHALVAN PROJECT** during the year 2022-2023

Submitted for the Naan Mudhalvan project work held on…………..2023

CLASS MENTOR	**HEAD OF THE DEPARTMENT**
Dr. T. RAJENDRAN, M.SC., M.Phil., Ph.D.,	**Dr. K. JAYANTHI, M.SC., M.Phil., Ph.D.,**

TABLE OF CONTENTS

1.INTRODUCTION

OPTIMIZING FLIGHT BOOKING DECISIONS THROUGH MACHINE LEARNING PRICE PREDICTIONS

People who work frequently travel through flight will have better knowledge on best discount and right time to buy the ticket. For the business purpose many airline companies change prices according to the seasons or time duration. They will increase the price when people travel more. Estimating the highest prices of the airlines data for the route is collected with features such as Duration, Source, Destination, Arrival and Departure. Features are taken from chosen dataset and in the price wherein the airline price ticket costs vary overtime. we have implemented flight price prediction for users by using KNN, decision tree and random forest algorithms. Random Forest shows the best accuracy of 80% for predicting the flight price. also, we have done correlation tests and metrics for the statistical analysis.

The flight ticket buying system is to purchase a ticket many days prior to flight take-off so as to stay away from the effect of the most extreme charge. Mostly, aviation routes don"t agree this procedure. Plane organizations may diminish the cost at the time, they need to build the market and at the time when the tickets are less accessible. They may maximize the costs. So, the cost may rely upon different factors. To foresee the costs this venture uses AI to exhibit the ways of flight tickets after some time. All organizations have the privilege and opportunity to change its ticket costs at any time. Explorer can set aside cash by booking a ticket at the least costs. People who had travelled by flight frequently are aware of price fluctuations. The airlines use complex policies of Revenue Management for execution of distinctive evaluating systems. The evaluating system as a result changes the charge depending on time, season, and festive days to change the header or footer on successive pages. The ultimate aim of the airways is to earn profit whereas the customer searches for the minimum rate. Customers usually try to buy the ticket well in advance of departure date so as to avoid hike in airfare as date comes closer. But actually, this is not the fact. The customer may wind up by giving more than they ought to for the same seat.

Optimal timing for airline ticket purchasing from the consumer‟s perspective is challenging principally because buyers have insufficient information for reasoning about future price movements. In this project we majorly targeted to uncover underlying trends of flight prices in India using historical data and also to suggest the best time to buy a flight ticket. The project implements the validations or contradictions towards myths regarding the airline industry, a comparison study among various models in predicting the optimal time to buy the flight ticket and the amount that can be saved if done so. Remarkably, the trends of the prices are highly sensitive to the route, month of departure, day of departure, time of departure, whether the day of departure is a holiday and airline carrier.

2. PROBLEM SELECTION

Predicting flight price prediction using machine learning is an interesting and important project that can have significant implications in the field of airways. Here are some steps you can follow to select a suitable project:

- Specify the business problem
- Business Requirements
- Literature Survey
- Social Or Business Impact

Specify The Business Problem:

People who work frequently travel through flight will have better knowledge on best discount and right time to buy the ticket. For the business purpose many airline companies change prices according to the seasons or time duration. They will increase the price when people travel more. Estimating the highest prices of the airlines data for the route is collected with features such as Duration, Source, Destination, Arrival and Departure. Features are taken from chosen dataset and in the price wherein the airline price ticket costs vary overtime. we have implemented flight price prediction for users by using KNN, decision tree and random forest algorithms. Random Forest shows the best accuracy of 80% for predicting the flight price. also, we have done correlation tests and metrics for the statistical analysis.

Business Requirements:

The business requirements for a machine learning model to predict personal loan approval include the ability to accurately predict loan approval based on applicant information, Minimise the number of false positives (approved loans that default) and false negatives (rejected loans that would have been successful). Provide an explanation for the model's decision, to comply with regulations and improve transparency.

Literature Survey:

As the data is increasing daily due to digitization in the banking sector, people want to apply for loans through the internet. Machine Learning (ML), as a typical method for information investigation, has gotten more consideration increasingly. Individuals of various businesses are utilising ML calculations to take care of the issues dependent on their industry information. Banks are facing a significant problem in the approval of the loan. Daily there are so many applications that are challenging to manage by the bank employees, and also the chances of some mistakes are high.Most banks earn profit from the loan, but it is risky to choose deserving customers from the number of applications.There are various algorithms that have been used with varying levels of success. Logistic regression, decision tree, random forest, and neural networks have all been used and have been able to accurately predict loan defaults. Commonly used features in these studies include credit score, income, and employment history, sometimes also other features like age, occupation, and education level.

Social Or Business Impact:

Social Impact: - Personal loans can stimulate economic growth by providing individuals with the funds they need to make major purchases, start businesses, or invest in their education.

Business Model/Impact: - Personal loan providers may charge fees for services such as loan origination, processing, and late payments. Advertising the brand awareness and marketing to reach out to potential borrowers to generate revenue.

3.IDEATION

Collect the dataset:

There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc.In this project we have used .csv data. This data is downloaded from kaggle.com. Please refer to the link given below to download the dataset.Link: https://www.kaggle.com/code/anshigupta01/flight-price-prediction/data As the dataset is downloaded. Let us read and understand the data properly with the help of some visualisation techniques and some analysing techniques. There are a number of techniques for understanding the data. But here we have used some of it. In an additional way, you can use multiple techniques.

Read the dataset:

Our dataset format might be in .csv, excel files,.txt,.json, etc. We can read the dataset with the help of pandas.In pandas we have a function called read_csv() to read the dataset. As a parameter we have to give the directory of csv file.

Data Preparation:

As we have understood how the data is let‟s pre-process the collected data. The download data set is not suitable for training the machine learning model as it might have so much of randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps.

- Handling missing values

- Handling categorical data

- Handling outliers

• Scaling Techniques

• Splitting dataset into training and test set

These are the general steps of pre-processing the data before using it for machine learning. Depending on the condition of your dataset, you may or may not have to go through all these steps. We have 1 missing value in Route column, and 1 missing value in Total stops column. We will meaningfully replace the missing values going further

Replacing Missing values:

We further replace „NaN" values in „City3" with „None", since rows where „City3" is missing did not have any stop, just the source and the destination. We also replace missing values in „Arrival_date" column with values in „Date" column, since the missing values are those values where the flight took off and landed on the same date. We also replace missing values in „Travel_mins" as 0, since the missing values represent that the travel time was in terms on hours only, and no additional minutes. Using the above steps, we were successfully able to treat all the missing values from our data. We again check the info in our data and find out that the dataset still has data types for multiple columns as „object", where it should be „int We then convert the „Travel_hours" column to „int" data type, and the operation happens successfully. We now have a treated dataset with 10682 rows and 17 columns (16 independent and 1 dependent variable). We create separate lists of categorical columns and numerical columns for plotting and analyzing the data.

Label Encoding:

Label encoding converts the data in machine readable form, but it assigns a unique number (starting from to each class of data. it performs the conversion of categorical data into numericformat.Airline','Source','Destination','Total_Stops','City1','City2','City3','Additional_Info'into number format. So that it helps the model in better understanding of the dataset and enables the model to learn more complex structures.

Output Columns:

- Initially in our dataset we have 19 features. So, in that some features are not more important to get output (Price)
- So i removed some unrelated features and I selected important features. So, it makes easy to understand. Now we have only 12 Output Columns

4. REQUIREMENT ANALYSIS

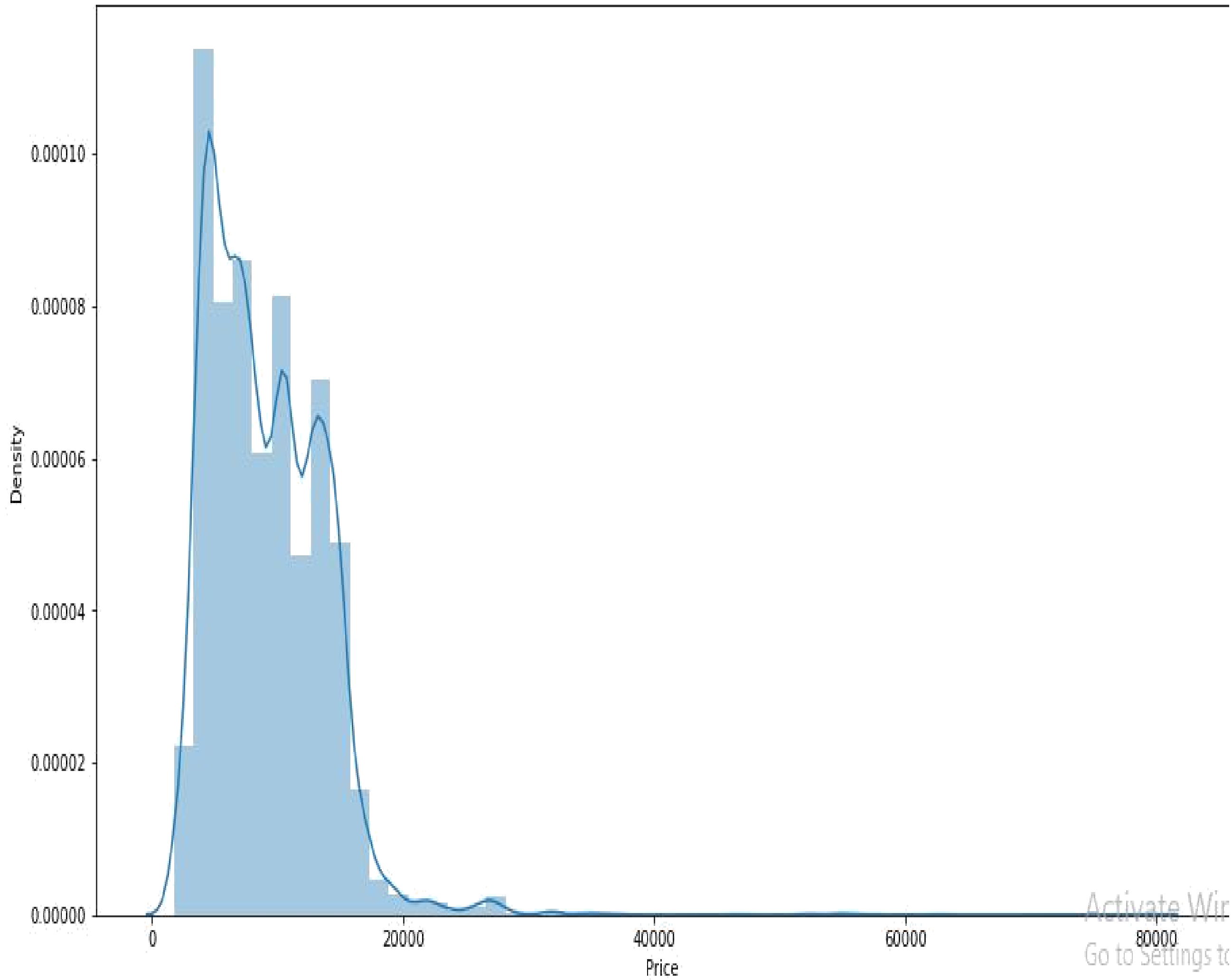
DESCRIPTIVE ANALYSIS:

Descriptive statistical Descriptive analysis is to study the basic features of data with the statistical process. Here pandas has a worthy function called describe. With this describe function we can understand the unique, top and frequent values of categorical features. And we can find mean, std, min, max and percentile values of continuous features.

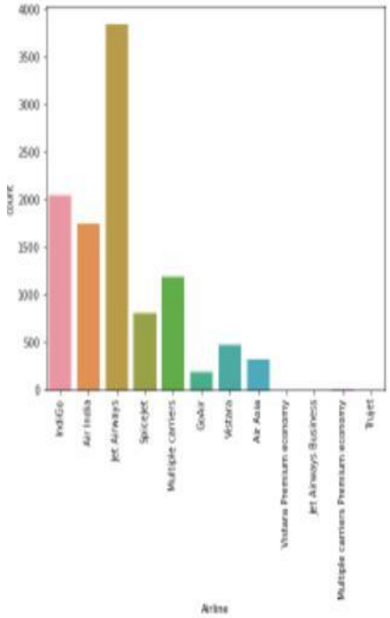
We now plot distribution plots to check the distribution in numerical data (Distribution of 'Price' Column). The seaborn.displot() function is used to plot the displot. The displot represents the variable univariate distribution of data variable as an argument and returns the plot with the density distribution. Here, I used distribution(displot) on 'Price' column. It estimates the probability of distribution of continous across various data.

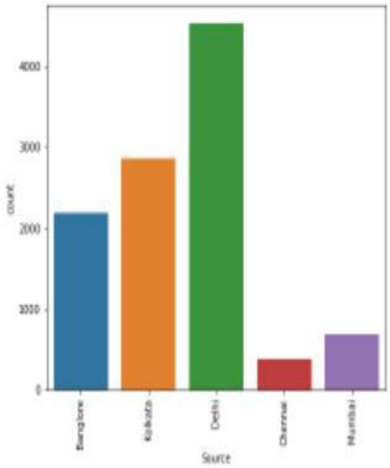
We Now Plot Distribution Plots To Check The Distribution In Numerical Data (Distribution Of 'Price' Column):

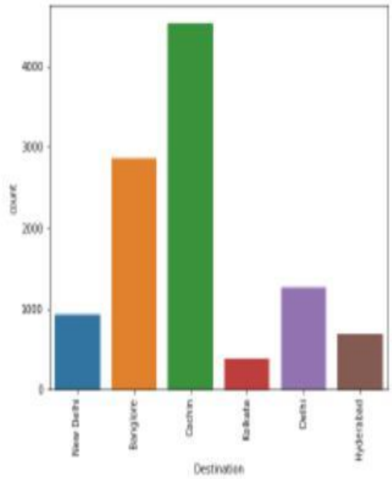
- The seaborn.displot() function is used to plot the displot. The displot represents the univariate distribution of data variable as an argument and returns the plot with the density distribution. Here, I used distribution(displot) on 'Price' column.
- It estimates the probability of distribution of continous variable across various data.

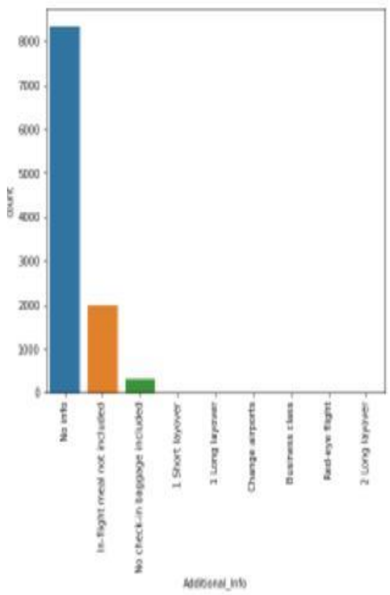

Density
0.00010
0.00008
0.00006
0.00004
0.00002
0.00000
0
20000
40000
60000
80000
Price
Activate Wi
Go to Settings t

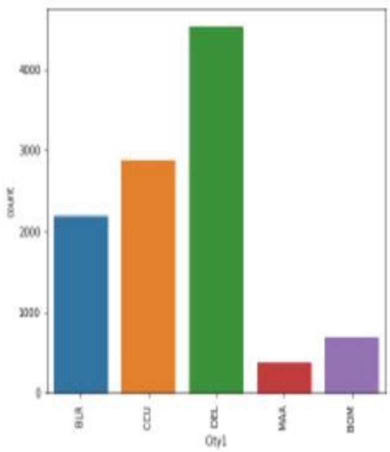
VISUAL ANALYSIS:





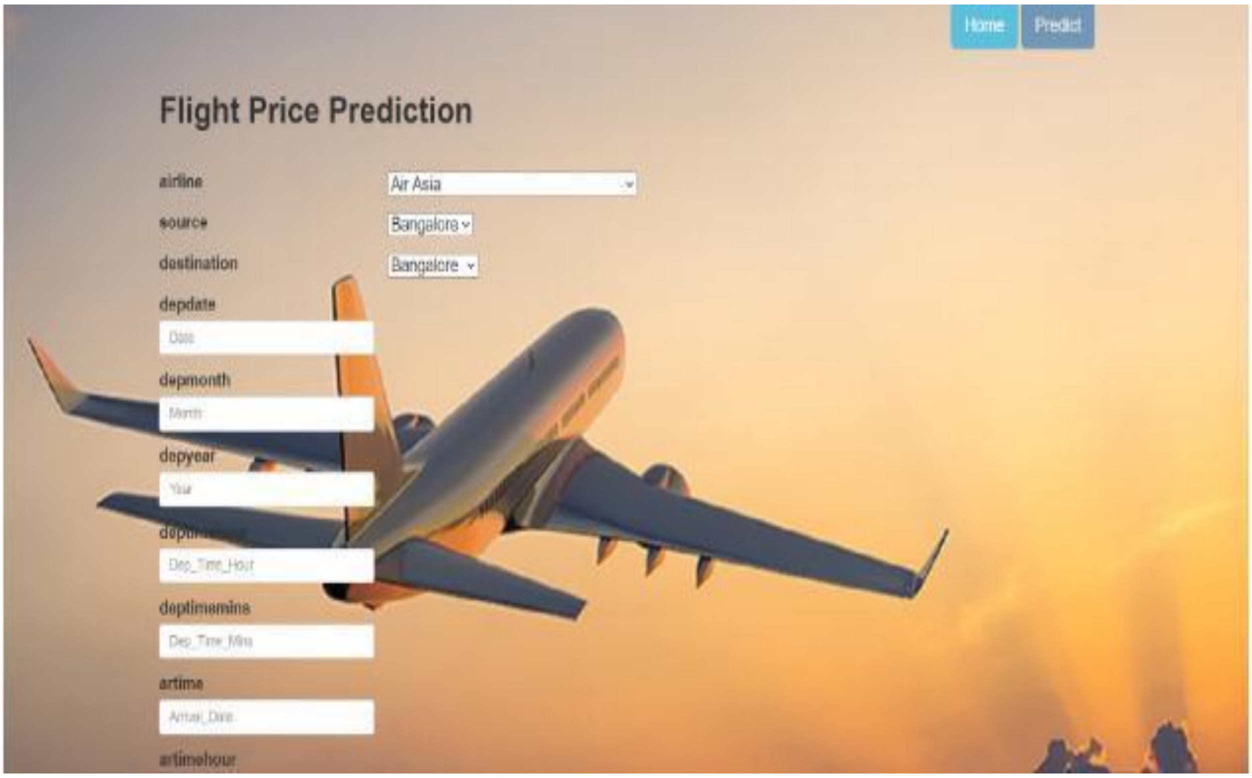


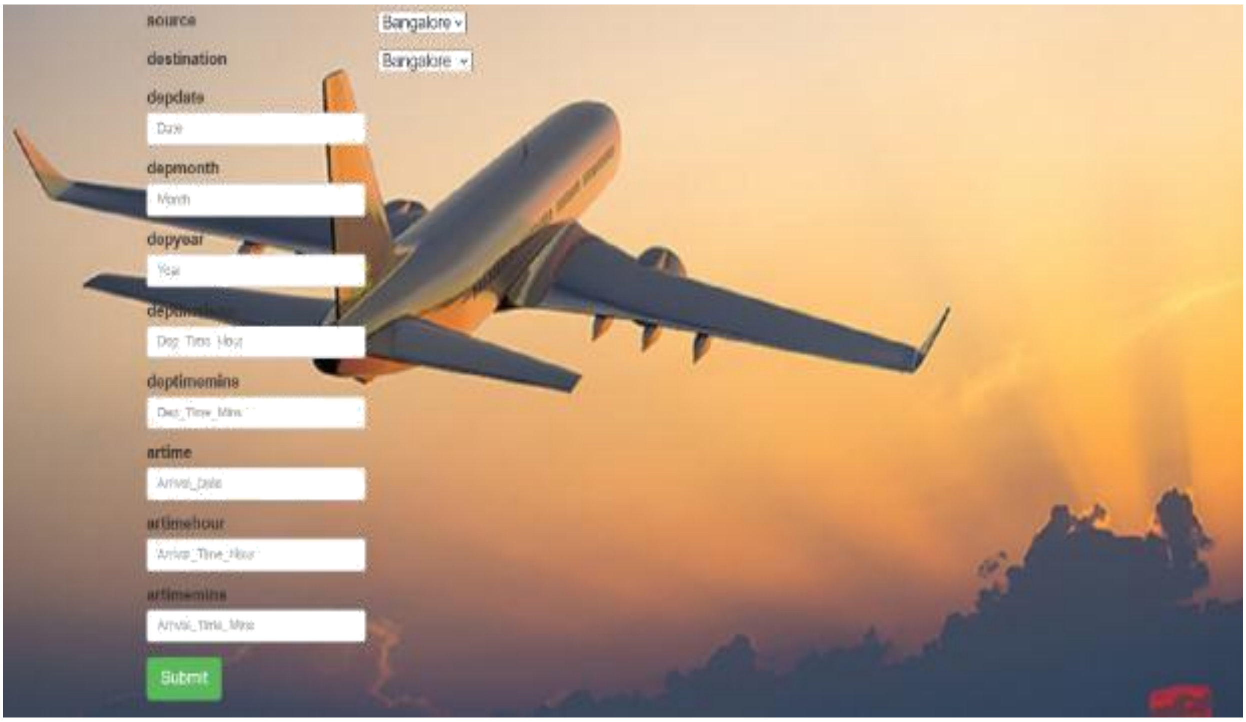


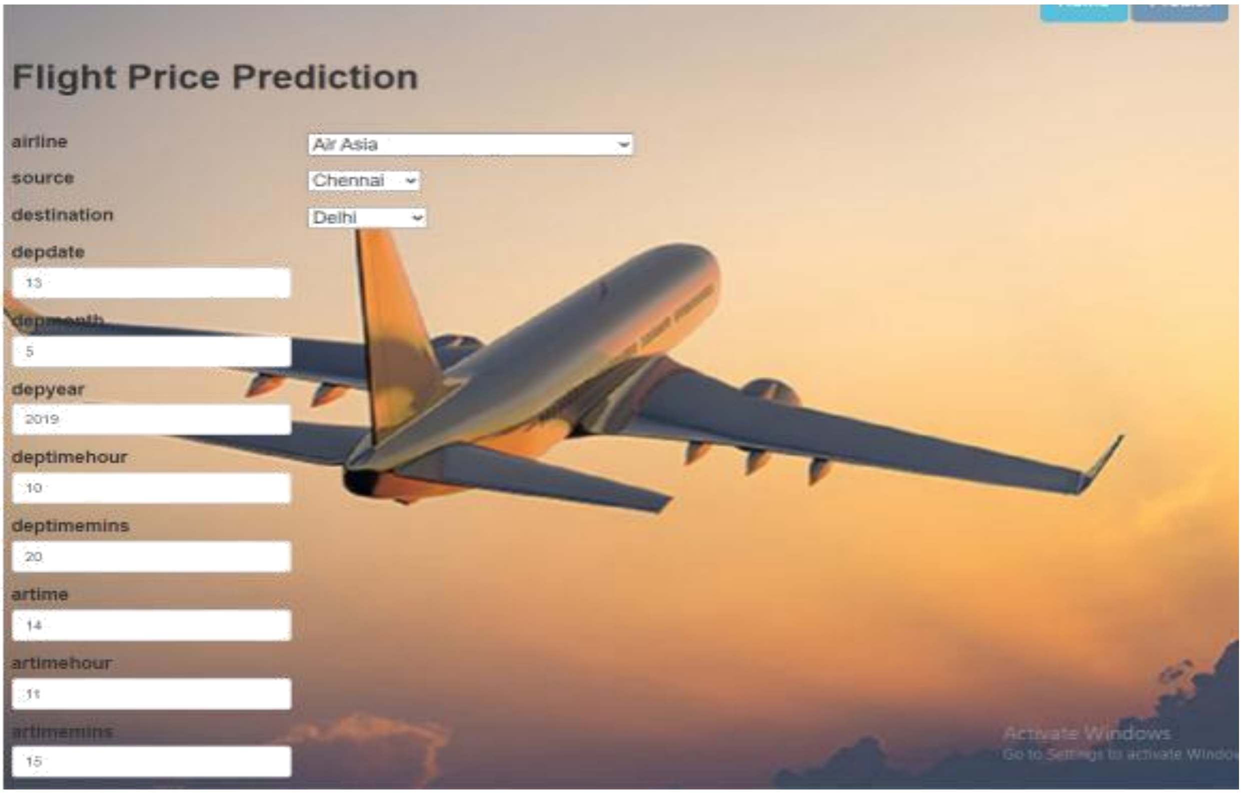


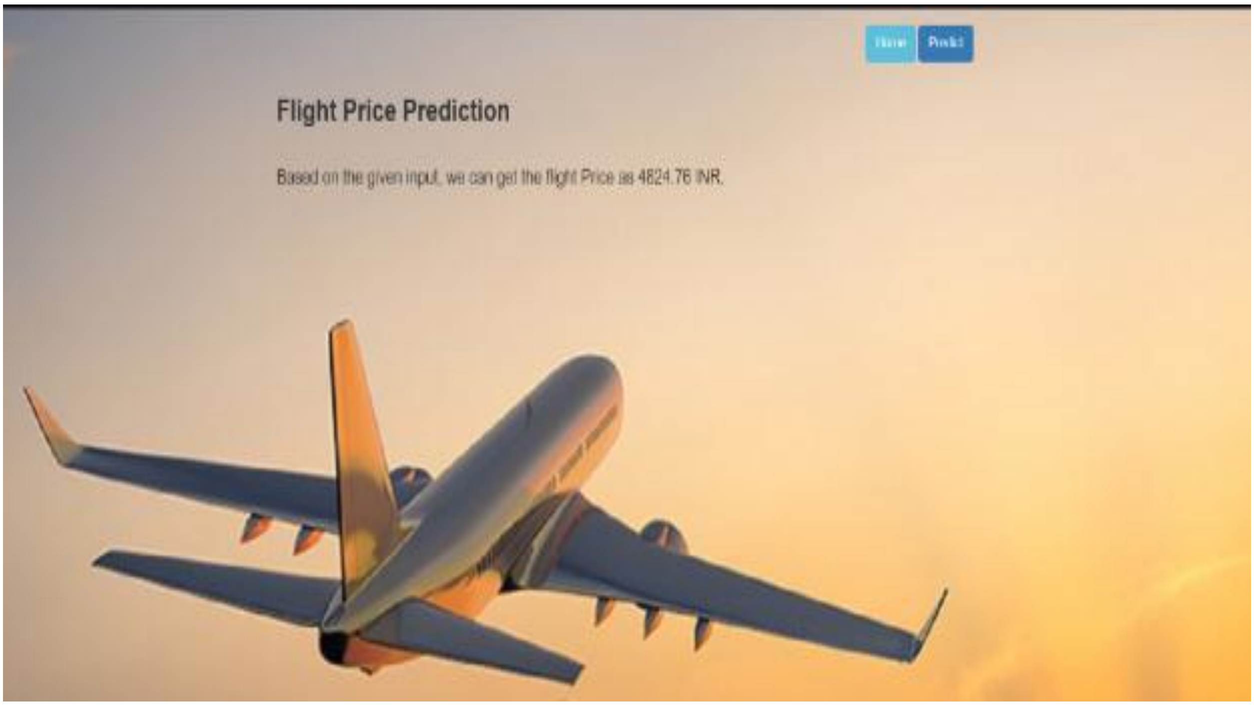
Activate Windows
Go to Settings to activate Windo

INPUT AND OUTPUT DESIGNS




source
Bangalore
destination
Bangalore
depdate
Date
depmonth
Month
depyear
Year
Dep_Time_Hour
deptimemins
Dep_Time_Mins
artime
Arrival_Date
artimehour
Arrival_Time_Hour
artimemins
Arrival_Time_Mins
Submit


Flight Price Prediction
airline
Air Asia
source
Chennai
destination
Delhi
depdate
13
5
depyear
2019
deptimehour
10
deptimemins
20
artime
14
artimehour
11
artimemins
15
Activate Windows


Home
Predict
Flight Price Prediction
Based on the given input, we can get the flight Price as 4824.76 INR.

9. PROJECT DEVELOPMENT PHASE

Saving the best model after comparing its performance using different evaluation metrics means selecting the model with the highest performance and saving its weights and configuration. This can be useful in avoiding the need to retrain the model every time it is needed and also to be able to use it in the future.

Integrate with Web Framework:

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI. This section has the following tasks

● Building HTML Pages

● Building server side script

● Run the web application

Building Html Pages:

For this project create two HTML files namely home.html

● predict.html

● submit.html

and save them in the templates folder. Build Python code: Import the libraries Load the saved model. Importing the flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (__name__) as argument.

Render HTML page:

Here we will be using a declared constructor to route to the HTML page which we have created earlier. In the above example, „/" URL is bound with the home.html function. Hence, when the home page of the web server is opened in the browser, the html page will be rendered. Whenever you enter the values from the html page the values can be retrieved using POST Method. Retrieves the value from UI: Here we are routing our app to predict() function. This function retrieves all the values from the HTML page using Post request. That is stored in an array. This array is passed to the model.predict() function. This function returns the prediction. And this prediction value will be rendered to the text that we have mentioned in the submit.html page earlier. Main Function:

Run the web application:

● Open anaconda prompt from the start menu

● Navigate to the folder where your python script is.

● Now type "python app.py" command

● Navigate to the localhost where you can view your web page.

● Click on the predict button from the top left corner, enter the inputs, click on the submit button, and see the result/prediction on the web.

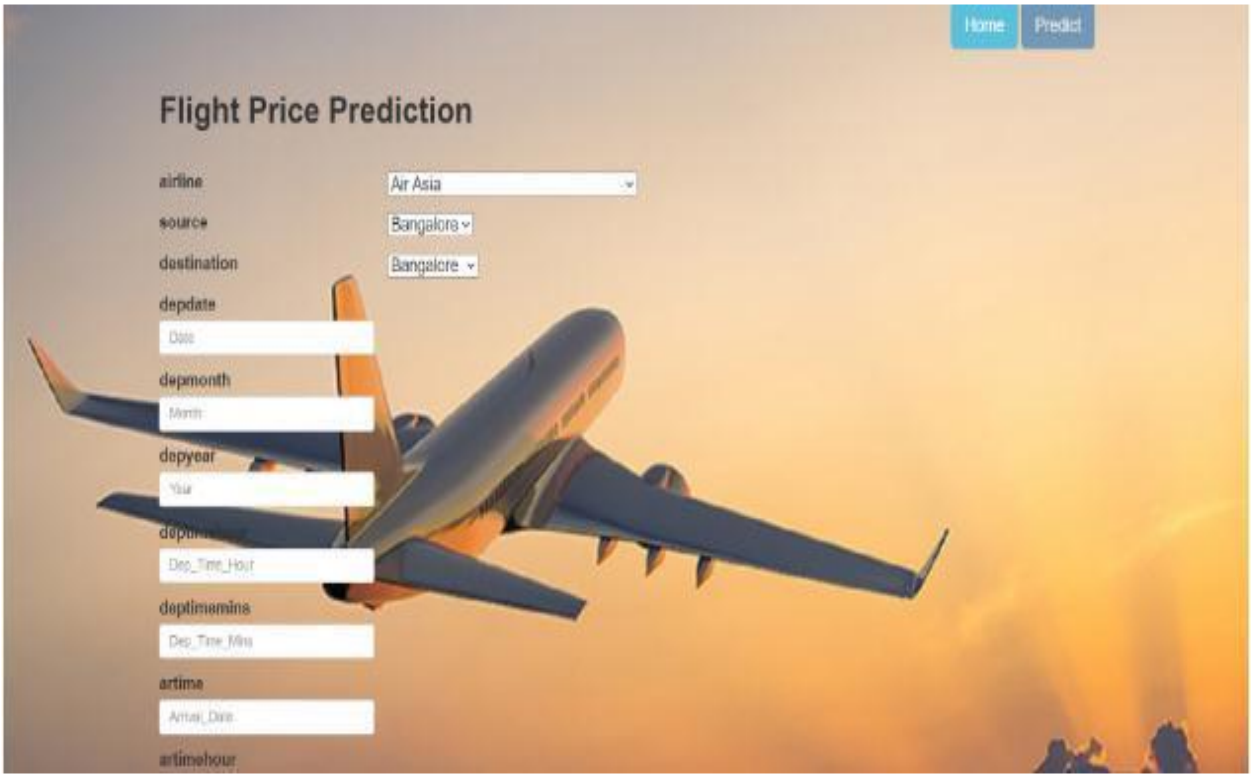
10. CONCLUSION

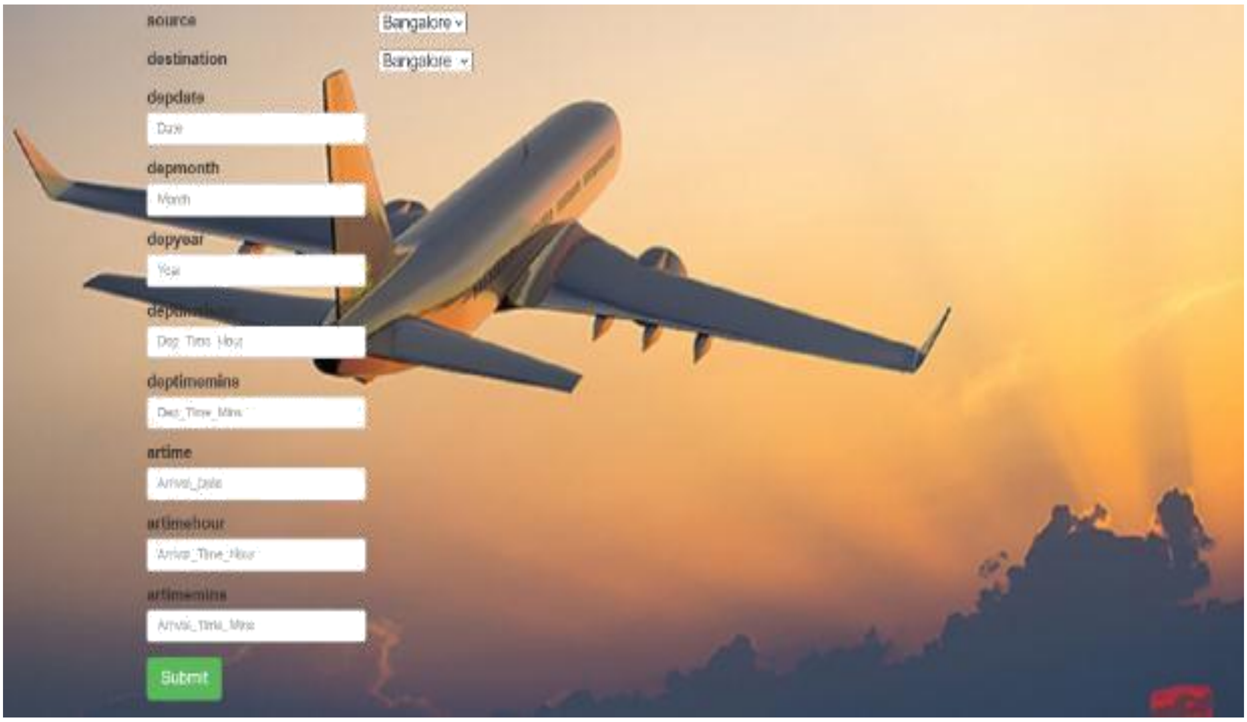
Machine Learning algorithms are applied on the dataset to predict the dynamic fare of flights. This gives the predicted values of flight fare to get a flight ticket at minimum cost. Data is collected from the websites which sell the flight tickets so only limited information can be accessed. The values of R-squared obtained from the algorithm give the accuracy of the model. In the future, if more data could be accessed such as the current availability of seats, the predicted results will be more accurate. Finally, we have created the entire process of predicting an airline ticket and given a proof of our predictions based on the previous trends with our prediction.

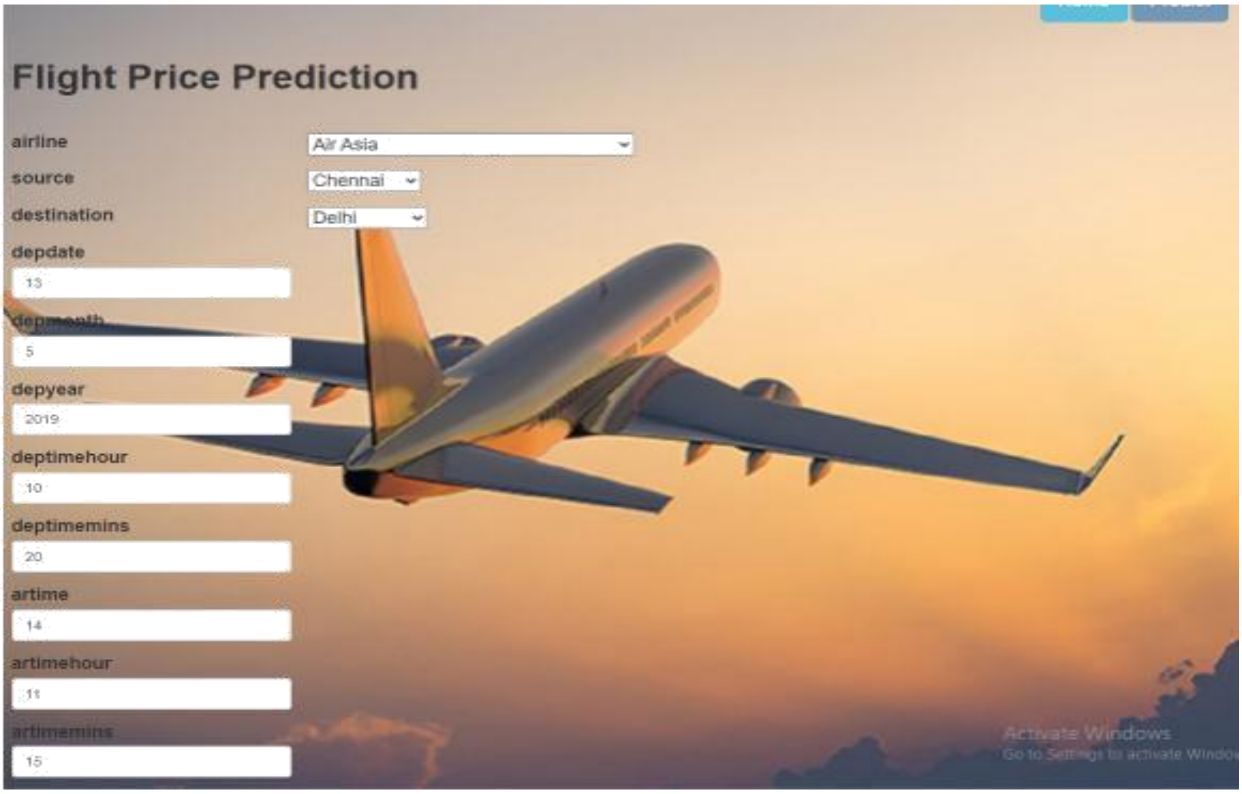
A preparatory study in “Airfare prediction”. We collected airfare data from a particular airline organization from internet and showed that it is realizable to foresee costs for flights based on recorded fare data. From the experiments we concluded which reviews impacts airfare prediction at most.

Apart from the features selected, there are other features that could improve the prediction accuracy. In the future, this work could be extended to predict the airfare prices for the entire flight map of the airline. Additional experiments on larger airfare data sets are essential, but this initial pilot study highlights the potential of machine learning models to guide consumers to make an airfare predicting purchase in the best market place

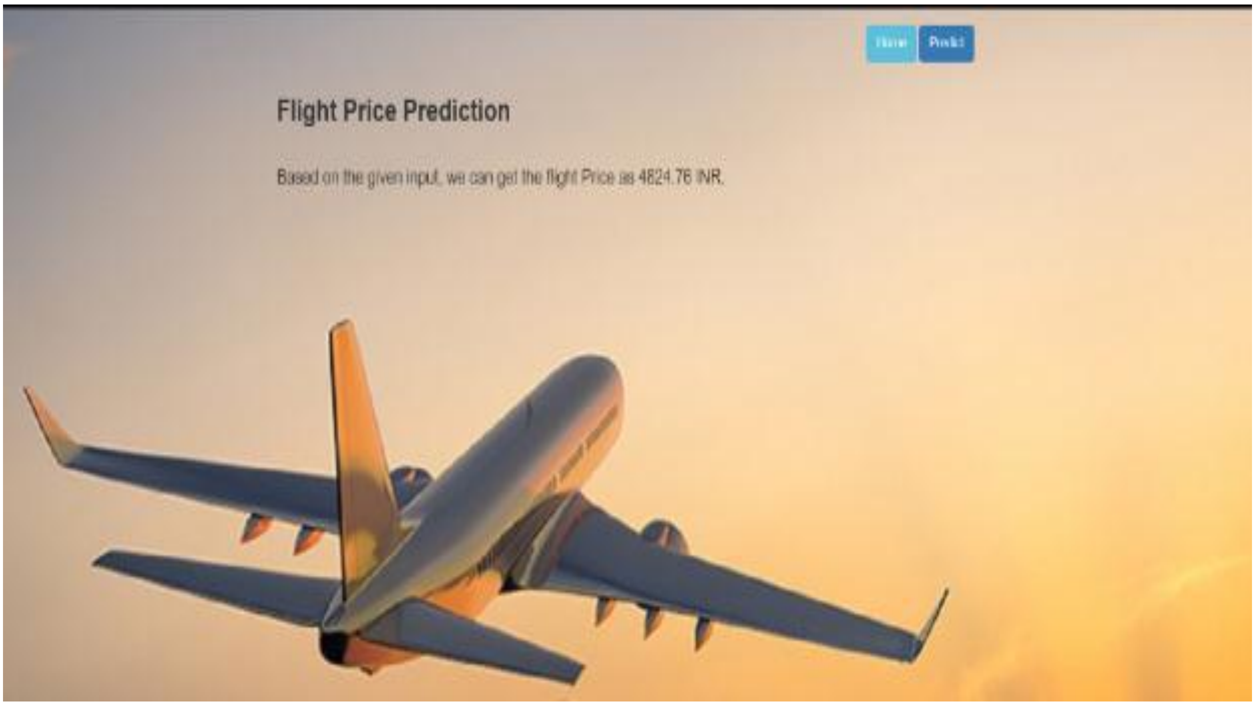
B.SAMPLE INPUT




source
Bangalore
destination
Bangalore
depdate
depmonth
depyear
deptimemins
artime
artimehour
artimemins
Submit


Flight Price Prediction
airline
Air Asia
source
Chennai
destination
Delhi
depdate
13
5
depyear
2019
deptimehour
10
deptimemins
20
artime
14
artimehour
11
artimemins
15
Activate Windows

C. SAMPLE OUTPUT



D. SAMPLE CODING

BIN +487 KB a1_FlightFare_Database.xlsx

BIN +41.5 KB a1_unseen_Dataset.xlsc

```
from flask import Flask,request, render_template
from flask_cors import cross_origin
import sklearn
import pickle
import pandas as pd
app = Flask(__name__)
model = pickle.load(open("c1_flight_rf.pkl","rb"))

@app.route("/predict", methods = ["GET","POST"])
@cross_origin()
def predict():
if request.method =="POST":

date_dep = request.form["Dep_Time"]
journey_day + int(pd.to_datetime(date_dep,format="%Y-%m-%dT%H:%M").day)
journey_month = int(pd.to_datetime(date_dep,format="%Y-%m-%dT%H:%M").month)
dep_hour= int(pd.to_datetime(date_dep,format ="%Y-%m-%dT%H:%M").hour)
dep_min=int(pd.to_datetime(date_dep,format="%Y-%m-%dT%H:%M").minute)
date_arr=request.form["Arrvial_Time"]
arr_hour =int(pd.to_datetime(date_arr,format="%Y-%m-%dT:%M").hour)
arr_min =int(pd.to_datetime(date_arr,format="%Y-%m-%dT:%M").minute)
Duration_hour=abs(arrvial_hour - dep_hour)
Duration_mins=abs(arrvial_min -dep_min)
Total_Stops = int(request.from["stops"])
airline=request.form['airline']
if(airline=='jet Airways'):
 Airline_AirIndia=0
 Airline_GoAir=0
 Airline_IndiGo=0
 Airline_JetAirways=1
 Airline_MultipleCarries=0
 Airline_SpiceJet=0
 Airline_Vistara=0
 Airline_other=0
 elif
```

```
(airline=='IndiGo'):
Airline_AirIndia=0
Airline_GoAir=0
Airline_IndiGo=1
Airline_JetAirways=0
Airline_MultipleCarries=0
Airline_SpiceJet=0
Airline_Vistara=0
Airline_other=0
elif
(airline=='AirIndia'):
  Airline_AirIndia=1
  Airline_GoAir=0
  Airline_IndiGo=0
  Airline_JetAirways=0
  Airline_MultipleCarries=0
  Airline_SpiceJet=0
  Airline_Vistara=0
  Airline_other=0
  elif
  (airline=='Multiple carries'):
  Airline_AirIndia=0
  Airline_GoAir=0
  Airline_IndiGo=0
  Airline_JetAirways=0
  Airline_MultipleCarries=1
  Airline_SpiceJet=0
  Airline_Vistara=0
  Airline_other=0
  elif
  (airline=='SpiceJet'):
  Airline_AirIndia=0
  Airline_GoAir=0
  Airline_IndiGo=0
  Airline_JetAirways=0
  Airline_MultipleCarries=0
  Airline_SpiceJet=1
  Airline_Vistara=0
  Airline_other=0
  elif
  (airline=='Vistara'):
  Airline_AirIndia=0
  Airline_GoAir=0
  Airline_IndiGo=0
  Airline_JetAirways=0
  Airline_MultipleCarries=0
  Airline_SpiceJet=0
  Airline_Vistara=1
  Airline_other=0
  elif
```

```
   (airline=='GoAir'):
   Airline_AirIndia=0
   Airline_GoAir=1
   Airline_IndiGo=0
   Airline_JetAirways=0
   Airline_MultipleCarries=0
   Airline_SpiceJet=0
   Airline_Vistara=0
   Airline_other==0
   else:
     Airline_AirIndia=0
     Airline_GoAir=0
     Airline_IndiGo=0
     Airline_JetAirways=0
     Airline_MultipleCarries=0
     Airline_SpiceJet=0
     Airline_Vistara=0
     Airline_other=1
Source=request.form["Source"]
if(Source=='Dehil=1'):
  Source_Dehil=1
  Source_Kollata=0
  Source_Mumbai=0
  Source_Chennai=0
  elif(Source=='Kolkata'):
       Source_Dehil=0
       Source_Mumbai=1
       Source_Mumbai=0
       Source_Chennai=0
  elif(Source=='Mumbai'):
      Source_Dehil=0
      Source_Kollata=0
      Source_Mumbai=1
      Source_Chennai=0
  elif(Source=='Chennai'):
      Source_Dehil=0
      Source_Kollata=0
      Source_Mumbai=0
      Source_Chennai=1
  else:
        Source_Dehil=0
        Source_Kollata=0
        Source_Mumbai=0
        Source_Chennai=0
        Source=request.form["Destination"]
           if(Source=='Cochin'):
             Destination_Cochin=1
             Destination_Delhi=0
             Destination_Hyderabad=0
             Destination_Kalkata=0
```

```
   elif(Source=='Hyderabad'):
        Destination_Cochin=0
        Destination_Delhi=0
        Destination_Hyderabad=1
        Destination_Kalkata=0
  elif(Source=='Kolkata'):
       Destination_Cochin=0
       Destination_Delhi=0
       Destination_Kalkata=1
       Destination_Hyderabad=0
       else:
         Destination_Delhi=0
         Destination_Hyderabad=0
         Destination_Cochin=0
         Destiantion_Kolata=0


prediction=model.predict([[
    Total_Stopes,
    journey_day,
    journey_month,
    dep_hour,
    dep_min,
    arrival_hour,
    arrivel_min,
    Duration_hour,
    Duration_mins,
    Airline_AirIndia,
    Airline_GoAir,
    Airline_IndiGo,
    Airline_JetAirways,
    Airline_MultipleCarries,
    Airline_other,
    Airline_SpiceJet,
    Airline_Vistara
    Source_Chennai,
    Source_Kolkata,
    Source_Mumbai,
    Destination_Cochin
    Destionation_Delhi,
    Destination_Hyderabad,
    Destination_Kalkata
]])
output=round(prediction[0],2)
return
render_template('home.html',prediction_text="Your Flight Price is a
Rs.{}".format(output))
return
render_template("home.html")
```

```
if __name__ == " __main__":
  app.run(debug=Ture)
```

b1_fare_prediction_model.ipynb