Flight Price Prediction

Submitted by,

M.Maneesh rao

1.INTRODUCTION

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.

1.1 OVERVIEW

The average price for an airline ticket in the united states in November 2022 was $280,about 35% higher than November 2021,according to statistics from the U.S.Bureau of Labor statistics .November's average price was down from may 2022 when the average price for a domestic flight hit an all-time high of $336.

A flighty price prediction application which predicts fares of flight for a particular date based on various parameters like source,destination,stops & airline.

1.2 PURPOSE

The main objective of the project is , 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.

2.LITERATURE SURVEY

It is very difficult for the customer to purchase a flight ticket at the minimum price. For this several techniques are used to

obtain the day at which the price of air ticket will be minimum. Most of these techniques are using sophisticated artificial intelligence(AI) research is known as Machine Learning.

Utilizing AI models, [2] connected PLSR(Partial Least Square Regression) model to acquire the greatest presentation to get the least cost of aircraft ticket buying, having 75.3% precision. Janssen [3] presented a direct quantile blended relapse model to anticipate air ticket costs for cheap tickets numerous prior days takeoff. Ren, Yuan, and Yang [4], contemplated the exhibition of Linear Regression (77.06% precision), Naive Bayes (73.06% exactness, Softmax Regression (76.84% precision) and SVM (80.6% exactness) models in anticipating air ticket costs. Papadakis [5] anticipated that the cost of the ticket drop later on, by accepting the issue as a grouping issue with the assistance of Ripple Down Rule Learner (74.5 % exactness.), Logistic Regression with 69.9% precision and Linear SVM with the

(69.4% exactness) Machine Learning models.

Gini and Groves[2] took the Partial Least Square Regression(PLSR) for developing a model of predicting the best purchase time for flight tickets. The data was collected from major travel journey booking websites from 22 February 2011 to 23 June 2011. Additional data were also collected and are used to check the comparisons of the performances of the final model.

Janssen [3] built up an expectation model utilizing the Linear Quantile Blended Regression strategy for SanFrancisco to NewYork course with existing every day airfares given by www.infare.com. The model utilized two highlights including the number of days left until the takeoff date and whether the flight date is at the end of the week or weekday. The model predicts airfare well for the days that are a long way from the takeoff date, anyway for a considerable length of time close the takeoff date, the expectation isnt compelling.

Wohlfarth [15] proposed a ticket buying time enhancement model dependent on an extraordinary pre-preparing step known as macked point processors and information mining systems (arrangement and bunching) and measurable investigation strategy. This system is proposed to change over heterogeneous value arrangement information into added value arrangement direction that can be bolstered to unsupervised grouping calculation. The value direction is bunched into gathering dependent on comparative estimating conduct. Advancement model gauge the value change designs. A treebased order calculation used to choose the best coordinating group and afterward comparing the advancement model.

A study by Dominguez-Menchero [16] recommends the ideal buying time dependent on nonparametric isotonic relapse method for a particular course, carriers, and timeframe. The model gives the most extreme number of days before buying a flight ticket. two sorts of the variable are considered for the expectation. One is the passage and date of procurement.

3.THEORITICAL ANALYSIS

This project is to use machine learning techniques to model the behaviour of flight prices over the time and predict the price of the flight-ticket.

To study how airline ticket prices changes over time ,extract the factors that influence these functions and describe how they’re correlated.

3.2 HARDWARE / SOFTWARE DESIGNING

The hardware required for the development of this project is:

Processor : Intel CoreTM i5-9300H
Processor speed : 2.4GHz
RAM Size : 8 GB DDR
System Type : X64-based processor

SOFTWARE DESIGNING:

The software required for the development of this project is:

Desktop GUI : Anaconda Navigator
Operating system : Windows 10
Front end : HTML, CSS,JAVASCRIPT
Programming : PYTHON

4.EXPERIMENTAL INVESTIGATION

IMPORTING AND READING THE DATASET

Importing the Libraries

First step is usually importing the libraries that will be needed in the program.
Pandas: It is a python library mainly used for data manipulation.
NumPy: This python library is used for numerical analysis.
Matplotlib and Seaborn: Both are the data visualization library used for plotting graph which will help us for understanding the data. **csr_matrix() :**A dense matrix stored in a NumPy array can be converted into a sparse matrix using the CSR representation by calling the csr_matrix() function.
Train_test_split: used for splitting data arrays into training data and for testing data.
Pickle: to serialize your machine learning algorithms and save the serialized format to a file.

Reading the Dataset

For this project, we make use of three different datasets (Books_Ratings, Books, Users). We will be

selecting the important features from these datasets that will help us in recommending the best results.

The next step is to read the dataset into a data structure that's compatible with pandas. Let's load a .csv data file into pandas. There is a function for it, called **read_csv().**We will need to locate the directory of the CSV file at first (it's more efficient to keep the dataset in the same directory as your program).If the dataset in same directory of your program, you can directly read it, without any path. After the next Steps we made following bellow:

1.Data visualization

2.Collabrative and filtering

3.Creating the Model

4.Test and save the model

5.Buil Python Code

6.Build HTML Code

7.Run the Application

We are the following above sections we did and investigate it.

5.FLOWCHART

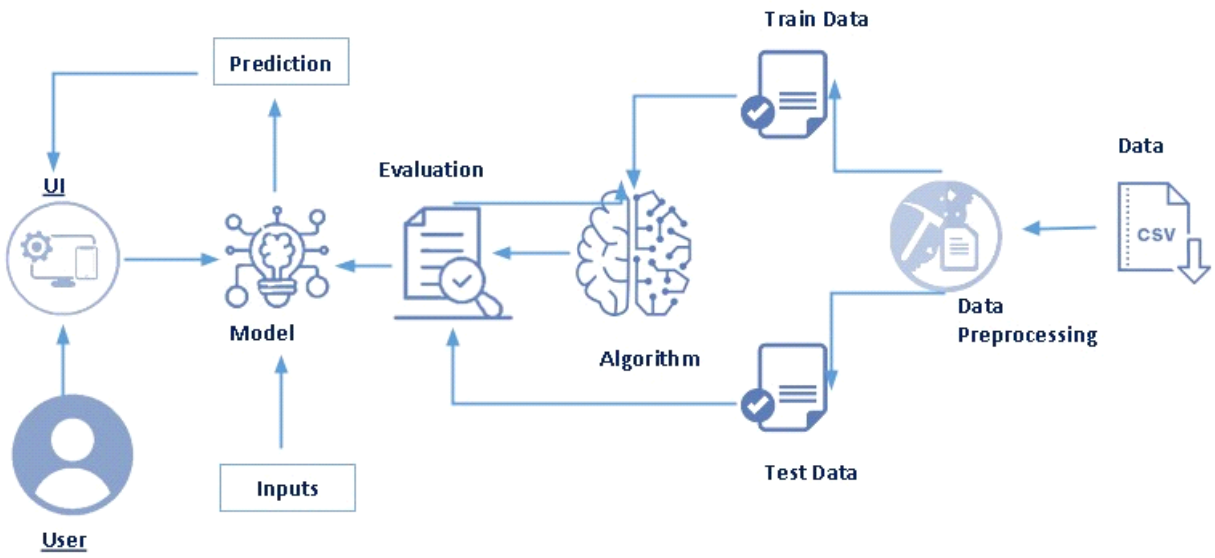

Fig 5.1 Flowchart of the project

Project Flow:

1. User interacts with the UI (User Interface) to upload the input features.
2. Uploaded features/input is analysed by the model which is integrated.

   Once a model analyses the uploaded inputs, the prediction is showcased on the UI.

1. Data collection

1. Collect the dataset or create the dataset
2. Visualizing and analyzing data
3. Importing Libraries
4. Read the DataSet

2. Data pre-processing

1. Checking for null values
2. Handling outlier
3. Handling categorical data
4. Splitting data into train and test

3. Model building

1. Import the model building libraries
2. Initializing the model
3. Training and testing the model
4. Evaluating performance of model
5. Save the model

4. Application Building

1. Create an HTML file
2. Build python code

6.RESULT

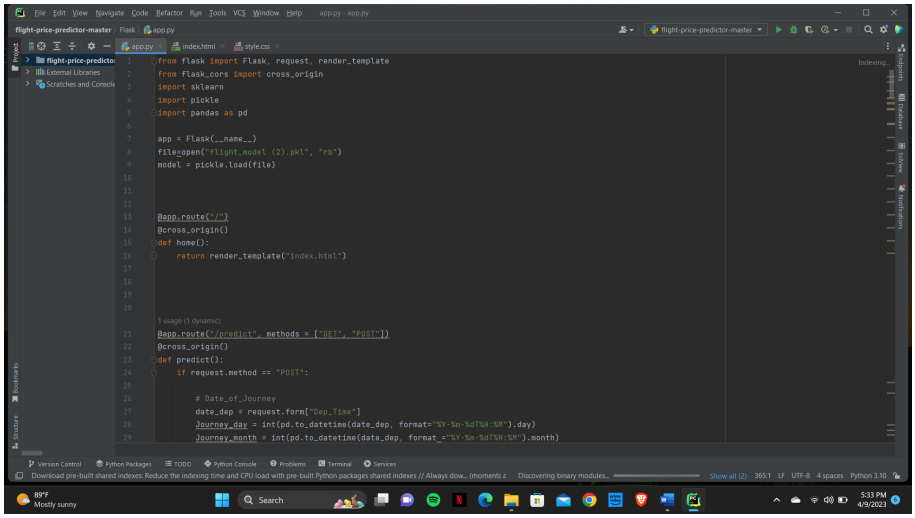

Fig 6.1 Flask code

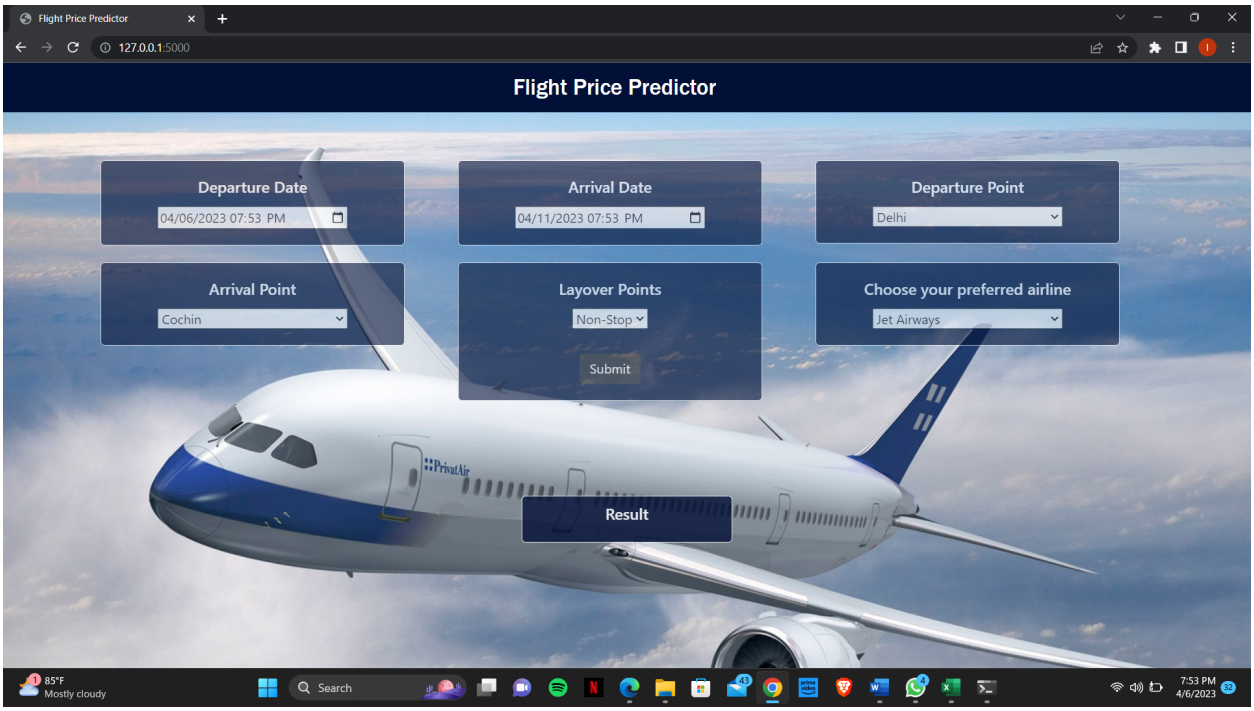

Fig 6.2 Home page for Flight Price Prediction

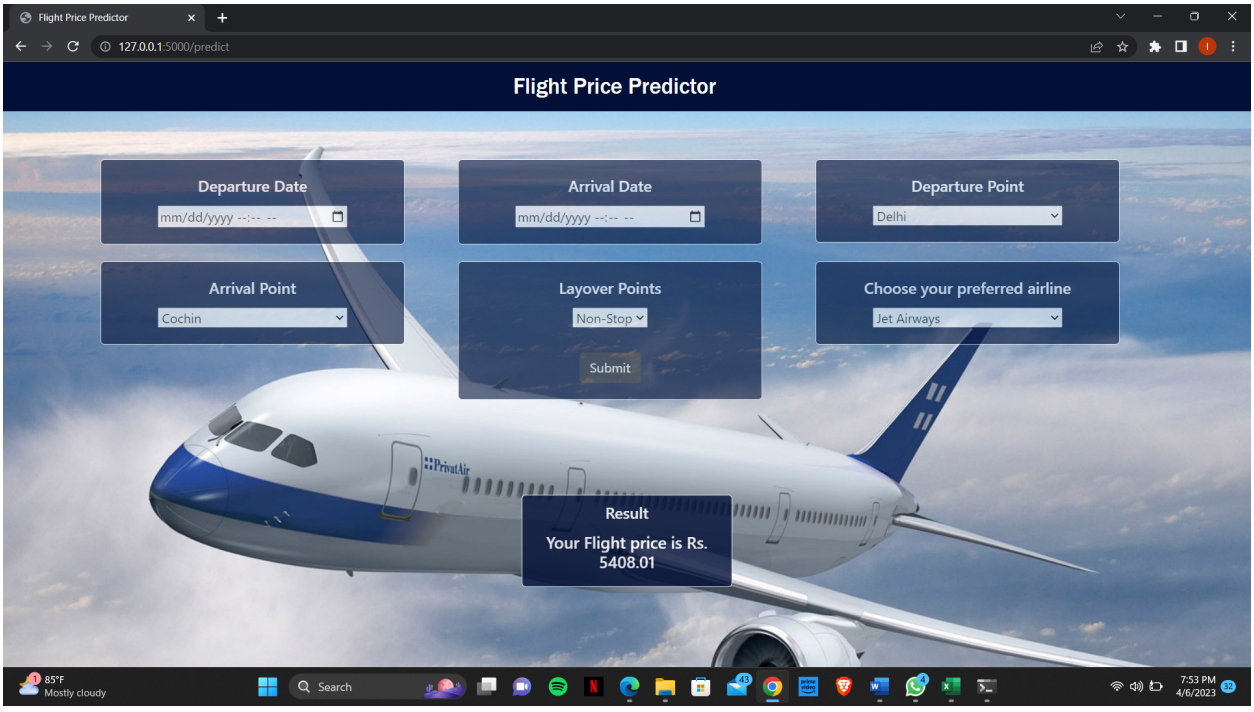

Fig 6.3 Predicting page of Flight Price Prediction

7.ADVANTAGES AND DISADVANTAGES

ADVANTAGES

1. Traveler get the fare prediction handy using which it's easy to decide the airlines.
2. Saves time in searching / deciding for airlines.

DISADVANTAGES

3. Improper data will result in incorrect fare predictions.

8.APPLICATIONS

1. make traveling easier
2. Airfare tracking
3. flight search and airfare prediction
4. Airfare tracking and hotel booking.

9.CONCLUSION

In this project is to forecast the average flight price at the business segment level. We used training data to train the training data and test data to test it. These records were used to extract a number of characteristics. Our suggested model can estimate the quarterly average flight price using attribute selection strategies. To the highest possible standard, much prior studies into flight price prediction using the large dataset depended on standard statistical approaches, which have their own limitations in

terms of underlying issue estimates and hypotheses. To our knowledge, no other research has included statistics from holidays, celebrations, stock market price fluctuations, depression, fuel price, and socioeconomic information to estimate the air transport market sector; nonetheless, there are numerous restrictions. As example, neither of the databases provide precise information about ticket revenue, including such departing and arrival times and days of the week. This framework may be expanded in the future to also include airline tickets payment details, that can offer more detail about each area, such as timestamp of entry and exit, seat placement, covered auxiliary items, and so on. By merging such data, it is feasible to create a more robust and complete daily and even daily flight price forecast model. Furthermore, a huge surge of big commuters triggered by some unique events might alter flight costs in a market sector. Thus, incident data will be gathered from a variety of sources, including social media sites and media organizations, to supplement our forecasting models. We will also examine specific technological Models, such as Deeper Learning methods, meanwhile striving to enhance existing models by modifying their hyper-parameters to get the optimum design for airline price prediction.

10.FUTURESCOPE

1. More routes can be added and the same analysis can be expanded to major airports and travel routes in india.
2. The analysis can be done by increasing the data points and increasing the historical data used. That will train the model better giving better accuracies and more savings.
3. More rules can be added in the rule-based learning based on our understanding of the industry, also incorporating the offer periods given by the airlines .
4. Developing a more user-friendly interface for various routes giving more flexibility to the users.

11.BIBILOGRAPHY

1. K. Tziridis, T. Kalampokas, G. A. Papakostas and K. I. Diamantaras, Airfare prices prediction using machine learning techniques, 2017 25th European Signal Processing Conference (EUSIPCO), Kos, Greece, 2017, pp. 1036-1039, https://doi.org/10.23919/EUSIPCO.2017.8081365 .
2. Juhar Ahmed Abdella, Nazar Zaki, Khaled Shuaib, Fahad Khan, Airline ticket price and demand prediction: A survey, Journal of King Saud University - Computer and Information

Sciences, 2019, ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2019.02.001

3. Martijn Brons, Eric Pels, Peter Nijkamp, Piet Rietveld, Price elasticities of demand for passenger air travel: a meta-analysis, Journal of Air Transport Management, Volume 8, Issue 3, 2002, Pages 165-175, ISSN 0969- 6997, https://doi.org/10.1016/S0969-6997(01)00050-3 .
4. Silke J. Forbes, The effect of air traffic delays on airline prices, International Journal of Industrial Organization, Volume 26, Issue 5, 2008, Pages 1218-1232, ISSN 0167-7187, https://doi.org/10.1016/j.ijindorg.2007.12.004 .

APPENDIX

A Source Code of Flask:

```
from flask import Flask, request, render_template
from flask_cors import cross_origin
import sklearn
import pickle
import pandas as pd
app = Flask(__name__)
file=open("flight_model (2).pkl", "rb")
model = pickle.load(file)
@app.route("/")
@cross_origin()
def home():
return render_template("index.html")
@app.route("/predict", methods = ["GET", "POST"])
@cross_origin()
def predict():
if request.method == "POST":
# Date_of_Journey
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)
```

```
# print("Journey Date : ",Journey_day, Journey_month)
# Departure
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)
# print("Departure : ",Dep_hour, Dep_min)
# Arrival
date_arr = request.form["Arrival_Time"]
Arrival_hour = int(pd.to_datetime(date_arr, format ="%Y-%m-%dT%H:%M").hour)
Arrival_min = int(pd.to_datetime(date_arr, format ="%Y-%m-%dT%H:%M").minute)
# print("Arrival : ", Arrival_hour, Arrival_min)
# Duration
dur_hour = abs(Arrival_hour - Dep_hour)
dur_min = abs(Arrival_min - Dep_min)
# print("Duration : ", dur_hour, dur_min)
# Total Stops
Total_stops = int(request.form["stops"])
# print(Total_stops)
# Airline
# AIR ASIA = 0 (not in column)
airline=request.form['airline']
if(airline=='Jet Airways'):
Jet_Airways = 1
IndiGo = 0
Air_India = 0
Multiple_carriers = 0
SpiceJet = 0
Vistara = 0
GoAir = 0
Multiple_carriers_Premium_economy = 0
Jet_Airways_Business = 0
Vistara_Premium_economy = 0
Trujet = 0
elif (airline=='IndiGo'):
Jet_Airways = 0
IndiGo = 1
Air_India = 0
Multiple_carriers = 0
SpiceJet = 0
Vistara = 0
GoAir = 0
Multiple_carriers_Premium_economy = 0
```

```
Jet_Airways_Business = 0
Vistara_Premium_economy = 0
Trujet = 0
elif (airline=='Air India'):
Jet_Airways = 0
IndiGo = 0
Air_India = 1
Multiple_carriers = 0
SpiceJet = 0
Vistara = 0
GoAir = 0
Multiple_carriers_Premium_economy = 0
Jet_Airways_Business = 0
Vistara_Premium_economy = 0
Trujet = 0
elif (airline=='Multiple carriers'):
Jet_Airways = 0
IndiGo = 0
Air_India = 0
Multiple_carriers = 1
SpiceJet = 0
Vistara = 0
GoAir = 0
Multiple_carriers_Premium_economy = 0
Jet_Airways_Business = 0
Vistara_Premium_economy = 0
Trujet = 0
elif (airline=='SpiceJet'):
Jet_Airways = 0
IndiGo = 0
Air_India = 0
Multiple_carriers = 0
SpiceJet = 1
Vistara = 0
GoAir = 0
Multiple_carriers_Premium_economy = 0
Jet_Airways_Business = 0
Vistara_Premium_economy = 0
Trujet = 0
elif (airline=='Vistara'):
Jet_Airways = 0
```

```
IndiGo = 0
Air_India = 0
Multiple_carriers = 0
SpiceJet = 0
Vistara = 1
GoAir = 0
Multiple_carriers_Premium_economy = 0
Jet_Airways_Business = 0
Vistara_Premium_economy = 0
Trujet = 0
elif (airline=='GoAir'):
Jet_Airways = 0
IndiGo = 0
Air_India = 0
Multiple_carriers = 0
SpiceJet = 0
Vistara = 0
GoAir = 1
Multiple_carriers_Premium_economy = 0
Jet_Airways_Business = 0
Vistara_Premium_economy = 0
Trujet = 0
elif (airline=='Multiple carriers Premium economy'):
Jet_Airways = 0
IndiGo = 0
Air_India = 0
Multiple_carriers = 0
SpiceJet = 0
Vistara = 0
GoAir = 0
Multiple_carriers_Premium_economy = 1
Jet_Airways_Business = 0
Vistara_Premium_economy = 0
Trujet = 0
elif (airline=='Jet Airways Business'):
Jet_Airways = 0
IndiGo = 0
Air_India = 0
Multiple_carriers = 0
SpiceJet = 0
Vistara = 0
```

```
GoAir = 0
Multiple_carriers_Premium_economy = 0
Jet_Airways_Business = 1
Vistara_Premium_economy = 0
Trujet = 0
elif (airline=='Vistara Premium economy'):
Jet_Airways = 0
IndiGo = 0
Air_India = 0
Multiple_carriers = 0
SpiceJet = 0
Vistara = 0
GoAir = 0
Multiple_carriers_Premium_economy = 0
Jet_Airways_Business = 0
Vistara_Premium_economy = 1
Trujet = 0
elif (airline=='Trujet'):
Jet_Airways = 0
IndiGo = 0
Air_India = 0
Multiple_carriers = 0
SpiceJet = 0
Vistara = 0
GoAir = 0
Multiple_carriers_Premium_economy = 0
Jet_Airways_Business = 0
Vistara_Premium_economy = 0
Trujet = 1
else:
Jet_Airways = 0
IndiGo = 0
Air_India = 0
Multiple_carriers = 0
SpiceJet = 0
Vistara = 0
GoAir = 0
Multiple_carriers_Premium_economy = 0
Jet_Airways_Business = 0
Vistara_Premium_economy = 0
Trujet = 0
```

```
Source = request.form["Source"]
if (Source == 'Delhi'):
s_Delhi = 1
s_Kolkata = 0
s_Mumbai = 0
s_Chennai = 0
elif (Source == 'Kolkata'):
s_Delhi = 0
s_Kolkata = 1
s_Mumbai = 0
s_Chennai = 0
elif (Source == 'Mumbai'):
s_Delhi = 0
s_Kolkata = 0
s_Mumbai = 1
s_Chennai = 0
elif (Source == 'Chennai'):
s_Delhi = 0
s_Kolkata = 0
s_Mumbai = 0
s_Chennai = 1
else:
s_Delhi = 0
s_Kolkata = 0
s_Mumbai = 0
s_Chennai = 0
Source = request.form["Destination"]
if (Source == 'Cochin'):
d_Cochin = 1
d_Delhi = 0
d_New_Delhi = 0
d_Hyderabad = 0
d_Kolkata = 0
elif (Source == 'Delhi'):
d_Cochin = 0
d_Delhi = 1
d_New_Delhi = 0
d_Hyderabad = 0
d_Kolkata = 0
elif (Source == 'New_Delhi'):
d_Cochin = 0
```

```
d_Delhi = 0
d_New_Delhi = 1
d_Hyderabad = 0
d_Kolkata = 0
elif (Source == 'Hyderabad'):
d_Cochin = 0
d_Delhi = 0
d_New_Delhi = 0
d_Hyderabad = 1
d_Kolkata = 0
elif (Source == 'Kolkata'):
d_Cochin = 0
d_Delhi = 0
d_New_Delhi = 0
d_Hyderabad = 0
d_Kolkata = 1
else:
d_Cochin = 0
d_Delhi = 0
d_New_Delhi = 0
d_Hyderabad = 0
d_Kolkata = 0
prediction=model.predict([[
Total_stops,
Journey_day,
Journey_month,
Dep_hour,
Dep_min,
Arrival_hour,
Arrival_min,
dur_hour,
dur_min,
Air_India,
GoAir,
IndiGo,
Jet_Airways,
Jet_Airways_Business,
Multiple_carriers,
Multiple_carriers_Premium_economy,
SpiceJet,
Trujet,
```

```
Vistara,
Vistara_Premium_economy,
s_Chennai,
s_Delhi,
s_Kolkata,
s_Mumbai,
d_Cochin,
d_Delhi,
d_Hyderabad,
d_Kolkata,
d_New_Delhi
]])
output=round(prediction[0],2)
return render_template('index.html', prediction_result="Your Flight price is Rs. {}".format(output))
return render_template("index.html")
if __name__ == "__main__":
app.run(debug=True)
```