OPTIMIZING FLIGHT BOOKING DECISIONS THROUGH MACHINE LEARNING PRICE PREDICTIONS

S.Hariharan
S.Mytheswaran
M.Monisha
B.Durga

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.

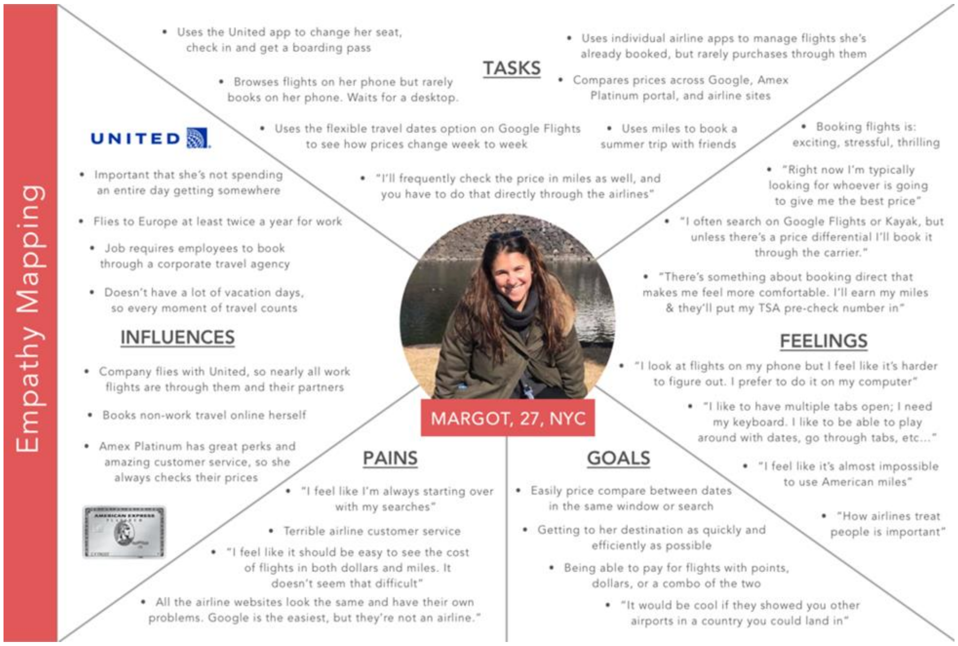
A flighty price prediction application which predicts fares of flight for a particular date based on various parameters like source,destination,stops & airline.

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.PROBLEM DEFINITION AND DESIGN THINKING

2.1 EMPATHY MAP



Empathy Mapping

UNITED

TASKS

- Uses the United app to change her seat, check in and get a boarding pass
- Browses flights on her phone but rarely books on her phone. Waits for a desktop.
- Uses the flexible travel dates option on Google Flights to see how prices change week to week
- "I'll frequently check the price in miles as well, and you have to do that directly through the airlines"
- Uses individual airline apps to manage flights she's already booked, but rarely purchases through them
- Compares prices across Google, Amex Platinum portal, and airline sites
- Uses miles to book a summer trip with friends

INFLUENCES

- Important that she's not spending an entire day getting somewhere
- Flies to Europe at least twice a year for work
- Job requires employees to book through a corporate travel agency
- Doesn't have a lot of vacation days, so every moment of travel counts
- Company flies with United, so nearly all work flights are through them and their partners
- Books non-work travel online herself
- Amex Platinum has great perks and amazing customer service, so she always checks their prices

FEELINGS

- Booking flights is: exciting, stressful, thrilling
- "Right now I'm typically looking for whoever is going to give me the best price"
- "I often search on Google Flights or Kayak, but unless there's a price differential I'll book it through the carrier."
- "There's something about booking direct that makes me feel more comfortable. I'll earn my miles & they'll put my TSA pre-check number in"
- "I look at flights on my phone but I feel like it's harder to figure out. I prefer to do it on my computer"
- "I like to have multiple tabs open; I need my keyboard. I like to be able to play around with dates, go through tabs, etc…"
- "I feel like it's almost impossible to use American miles"
- "How airlines treat people is important"

MARGOT, 27, NYC

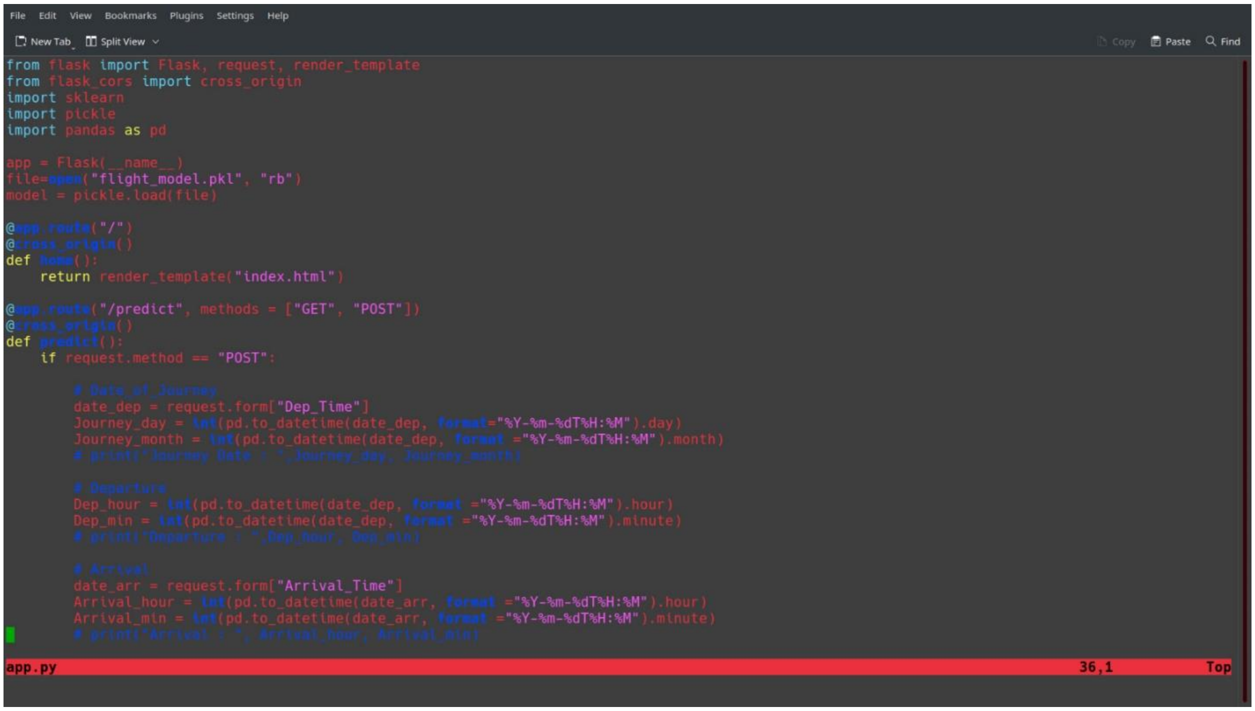
PAINS

- "I feel like I'm always starting over with my searches"
- Terrible airline customer service
- "I feel like it should be easy to see the cost of flights in both dollars and miles. It doesn't seem that difficult"
- All the airline websites look the same and have their own problems. Google is the easiest, but they're not an airline."

GOALS

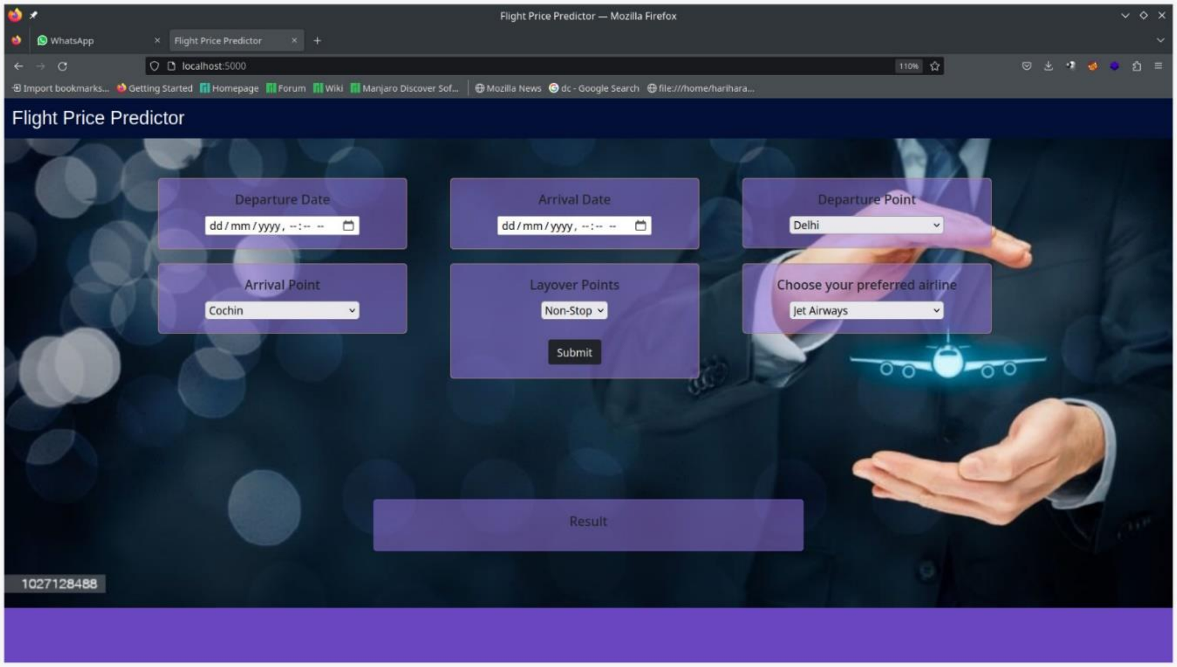
- Easily price compare between dates in the same window or search
- Getting to her destination as quickly and efficiently as possible
- Being able to pay for flights with points, dollars, or a combo of the two
- "It would be cool if they showed you other airports in a country you could land in"

3.RESULT



```
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.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)
app.py                                                          36,1          Top
```

Fig 3.1 Flask code


Flight Price Predictor — Mozilla Firefox
WhatsApp
Flight Price Predictor
localhost:5000
110%
Import bookmarks... Getting Started Homepage Forum Wiki Manjaro Discover Sof... Mozilla News dc - Google Search file:///home/harihara...

Flight Price Predictor

Departure Date: dd / mm / yyyy, --:-- --

Arrival Date: dd / mm / yyyy, --:-- --

Departure Point: Delhi

Arrival Point: Cochin

Layover Points: Non-Stop

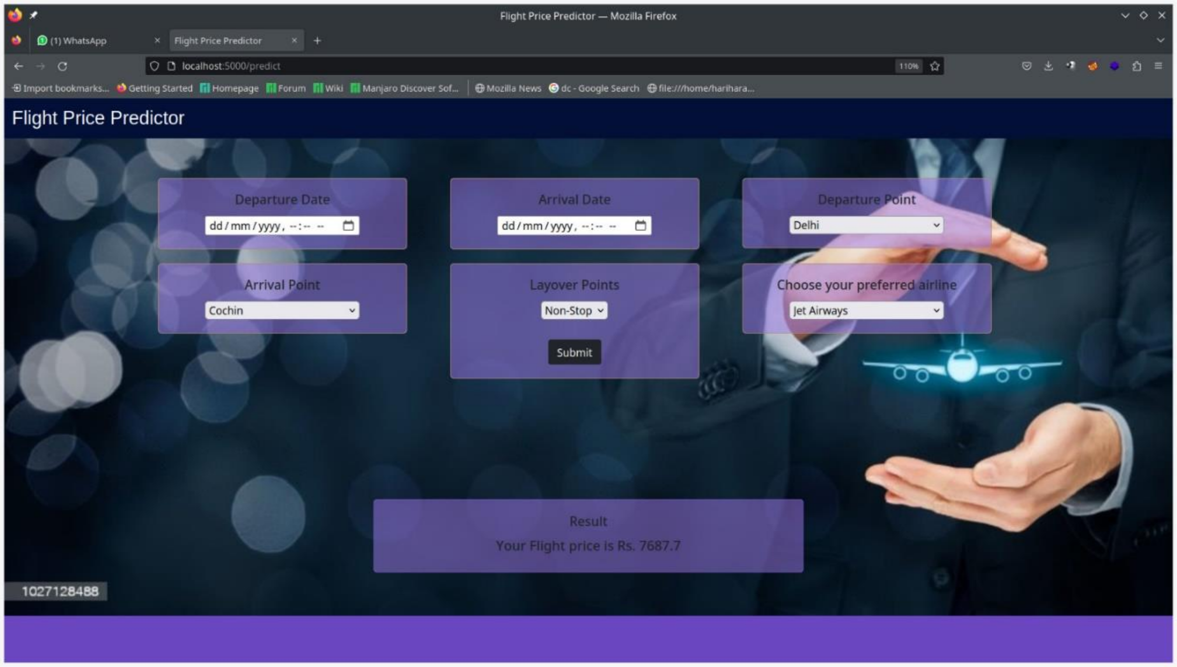
Choose your preferred airline: Jet Airways

Submit

Result

1027128488

Fig 3.2 Home page for Flight Price Prediction


Flight Price Predictor — Mozilla Firefox
(1) WhatsApp
Flight Price Predictor
localhost:5000/predict
110%
Import bookmarks... Getting Started Homepage Forum Wiki Manjaro Discover Sof... Mozilla News dc - Google Search file:///home/harihara...

Flight Price Predictor

Departure Date: dd / mm / yyyy, --:-- --

Arrival Date: dd / mm / yyyy, --:-- --

Departure Point: Delhi

Arrival Point: Cochin

Layover Points: Non-Stop

Choose your preferred airline: Jet Airways

Submit

Result

Your Flight price is Rs. 7687.7

1027128488

Fig 3.3 Predicting page of Flight Price Prediction

4.Trailhead Profile Public URL

Team Lead- https://trailblazer.me/id/mmonisha21
Team Member 1- https://trailblazer.me/id/dbabu100
Team Member 2- https://trailblazer.me/id/mytheeswaran
Team Member 3- https://trailblazer.me/id/Harikaran99

5.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.

6.APPLICATIONS

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

7.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.

8.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.

9.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)
```