GOVERNMENT OF TAMILNADU

Naan Muthalvan - Project-Based Experiential Learning

Optimizing Flight Booking Decisions Through Machine Learning Price Prediction

Submitted by

Team ID: NM2023TMID22898

T. SUBIKSHAA - (20326ER069)

M. SURIYA - (20326ER070)

B. SWATHI - (20326ER071)

S. SWATHI - (20326ER072)

Under the guidance of

Mrs. P.SANGEETHA, M.Sc., M.Phil.,

Guest Lecturer

PG and Research Department of Computer Science

M.V.MUTHIAH GOVERNMENT ARTS COLLEGE FOR WOMEN

(Affiliated To Mother Teresa Women's University, Kodaikanal)

Reaccredited with "A" Grade by NAAC

DINDIGUL-624001.

APRIL - 2023

M.V.MUTHIAH GOVERNMENT ARTS COLLEG FOR WOMEN
(Affiliated to Mother Teresa Women's University, Kodaikanal)
Reaccredited with "A" Grade by NAAC
Dindigul - 624 001

PG & RESEARCH DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this is a bonafide record of the project entitled, **OPTIMIZING FLIGHT BOOKING THROUGH MACHINE LEARNING PRICE PREDICTION**"doneby**Ms.T.SUBIKSHAA-(20326ER069),Ms.M.SURYA(20326ER070), Ms.B.SWATHI-(20326ER071) and Ms.S.SWATHI-(20326ER072)**. This is submitted in partial fulfillment for the award of the degree of **Bachelor of Science in Computer Science in M.V.MUTHIAH GOVERNMENT ARTS COLLEGE FOR WOMEN,DINDIGUL** during the period of December 2022 to April 2023.

P. Sangeetha

Project Mentor(s) **Head of the Department**

Submitted for viva-voce Examination held on ___11.04.2023____________

TABLE OF CONTENTS

1. INTRODUCTION

1.1. OVERVIEW:

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.

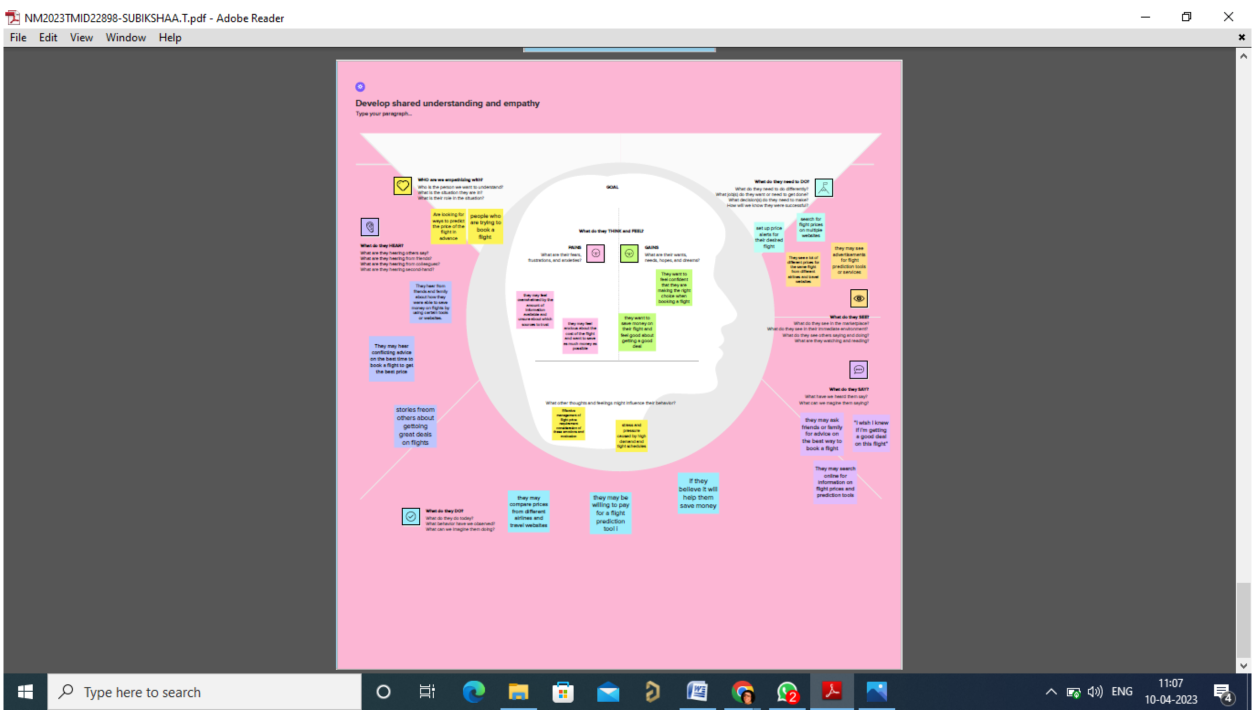
1.2 PURPOSE:

The purpose of flight price prediction is to help travelers make informed decisions about when to purchase flight tickets. Flight prices are known to fluctuate frequently, and predicting future price changes can be challenging for consumers. By using machine learning algorithms and historical flight data, flight price prediction models can provide travelers with estimated ticket prices for their desired travel dates. This can help travelers decide whether to book their flight immediately or wait for a better deal. Additionally, flight price prediction can also benefit airlines by optimizing their pricing strategies and increasing revenue.

2. Problem Definition and Design Thinking

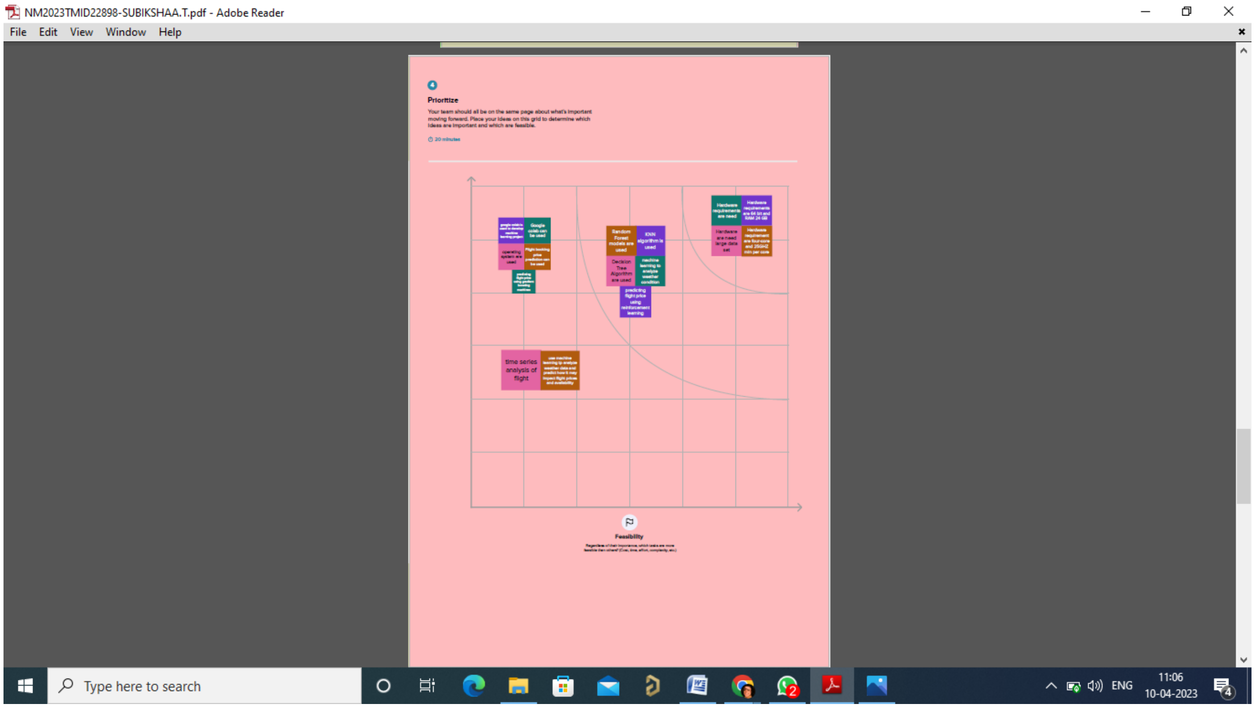
2.1 Empathy Map:

An Empathy map is a collaborative tool teams can use to gain a deeper insight intotheir customers. Much like a user person, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community.



2.2 Ideation & Brainstorm Map:

Brainstorming is a group problem-solving method that involves the spontneous contribution of creative ideas and solutions. This technique requires intensive,freewheeling discussion in which every member of the group is encouraged to think aloud and suggest as many ideas as possible based on their diverse knowledge

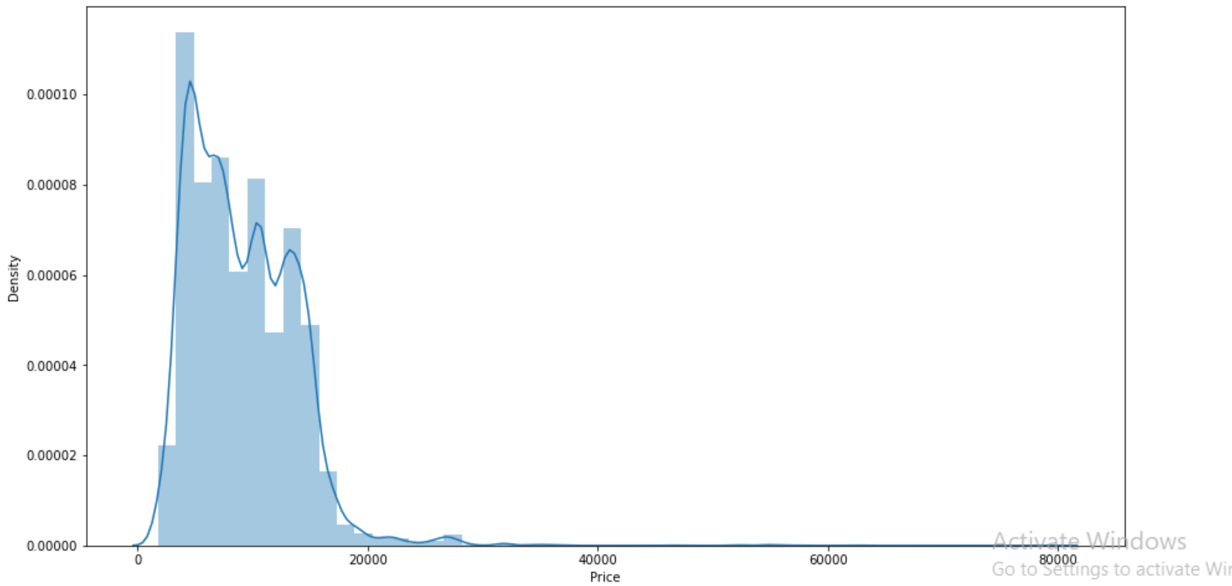

NM2023TMID22898-SUBIKSHAA.T.pdf - Adobe Reader
File Edit View Window Help
4
Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
20 minutes
Google colab can be used
operating system are used
Random Forest models are used
KNN algorithm is used
Decision Tree Algorithm are used
machine learning to analyse weather condition
predicting flight price using reinforcement learning
Hardware requirements are need
Hardware requirements are 64 bit and RAM 24 GB
Hardware are need large data set
Hardware requirement are four-core and 25GHZ min per core
time series analysis of flight
Feasibility
Type here to search
ENG
11:06
10-04-2023

3. RESULT

```
#Distribution of 'PRICE' Column
plt.figure(figsize=(15,8))
sns.distplot(data.Price)
```

```
C:\Users\SmartBridge-PC\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated fu
nction and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with si
milar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

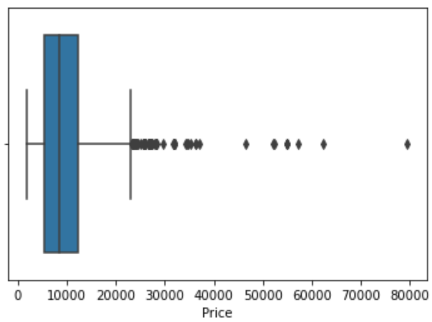
```
<AxesSubplot:xlabel='Price', ylabel='Density'>
```

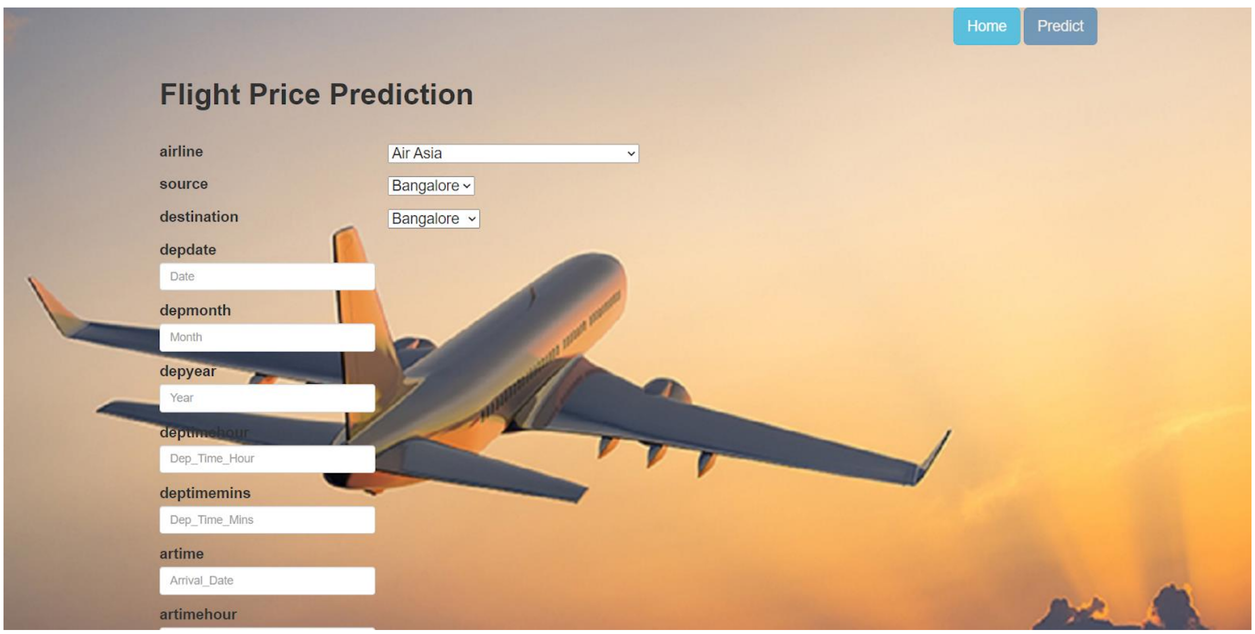


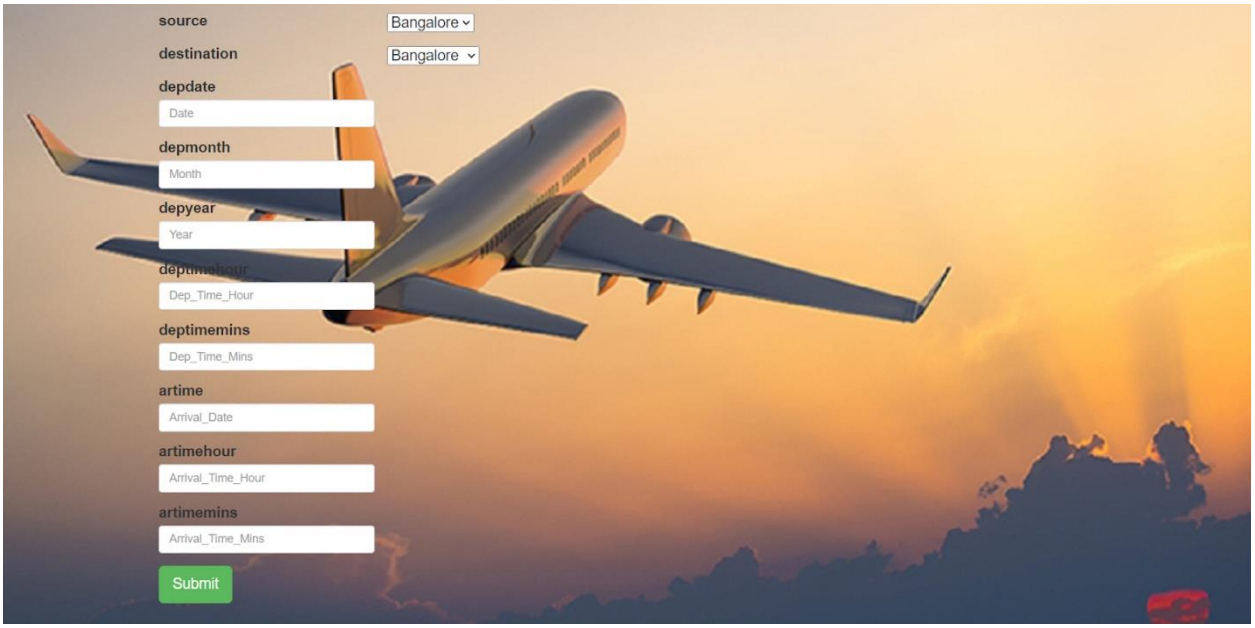
```python
# Detecting the Outliers
import seaborn as sns
sns.boxplot(data['Price'])
```

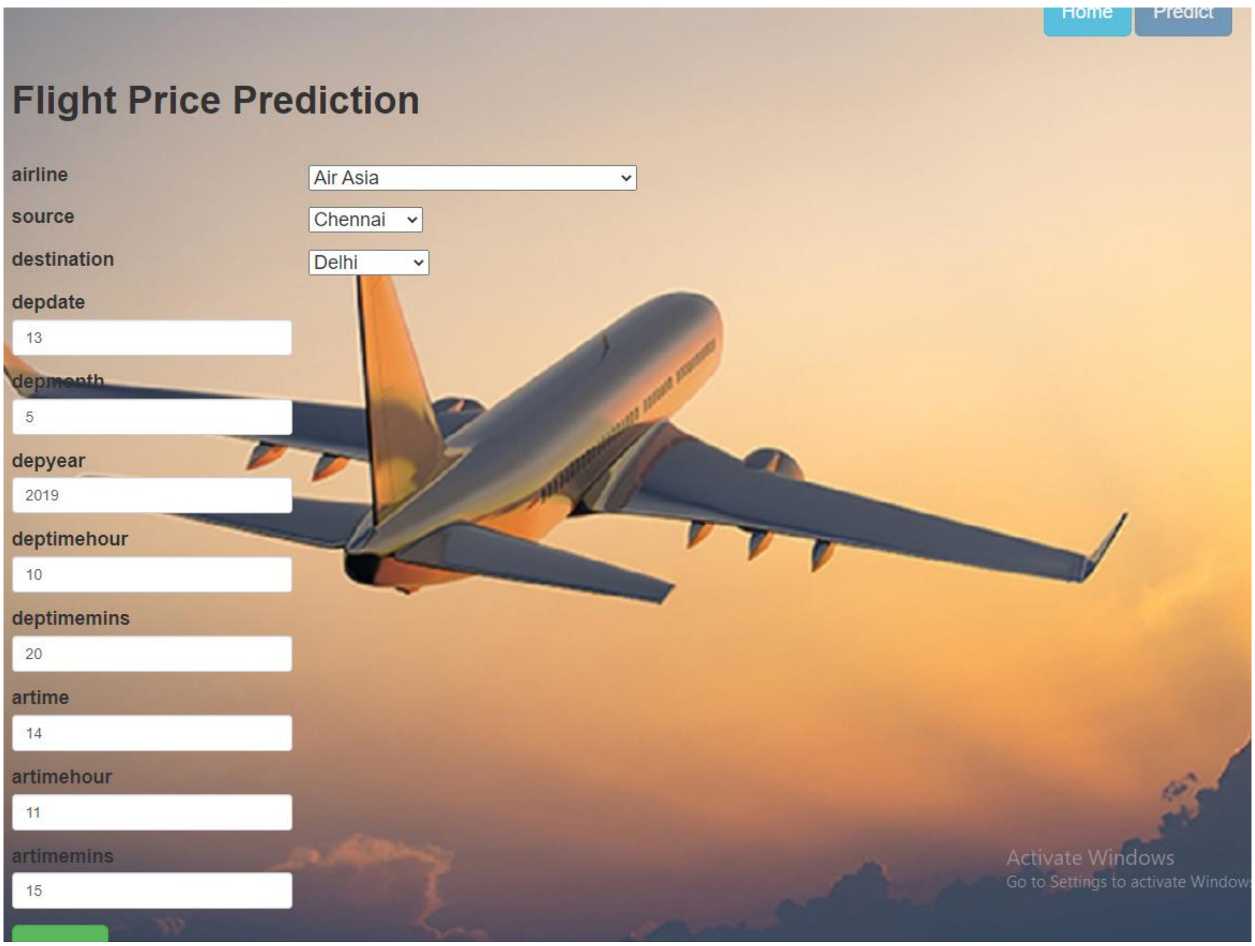
```
C:\Users\SmartBridge-PC\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

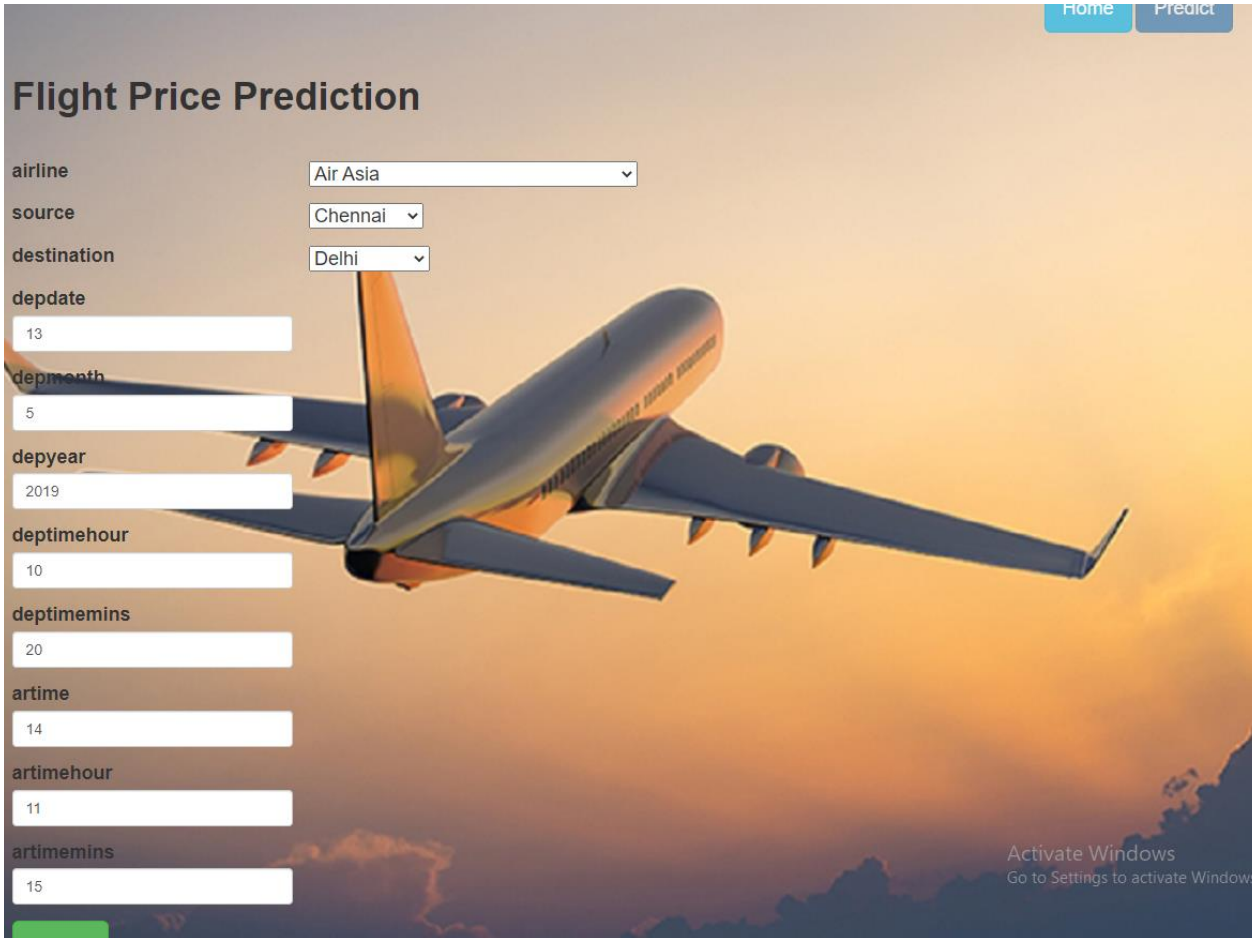
```
<AxesSubplot:xlabel='Price'>
```

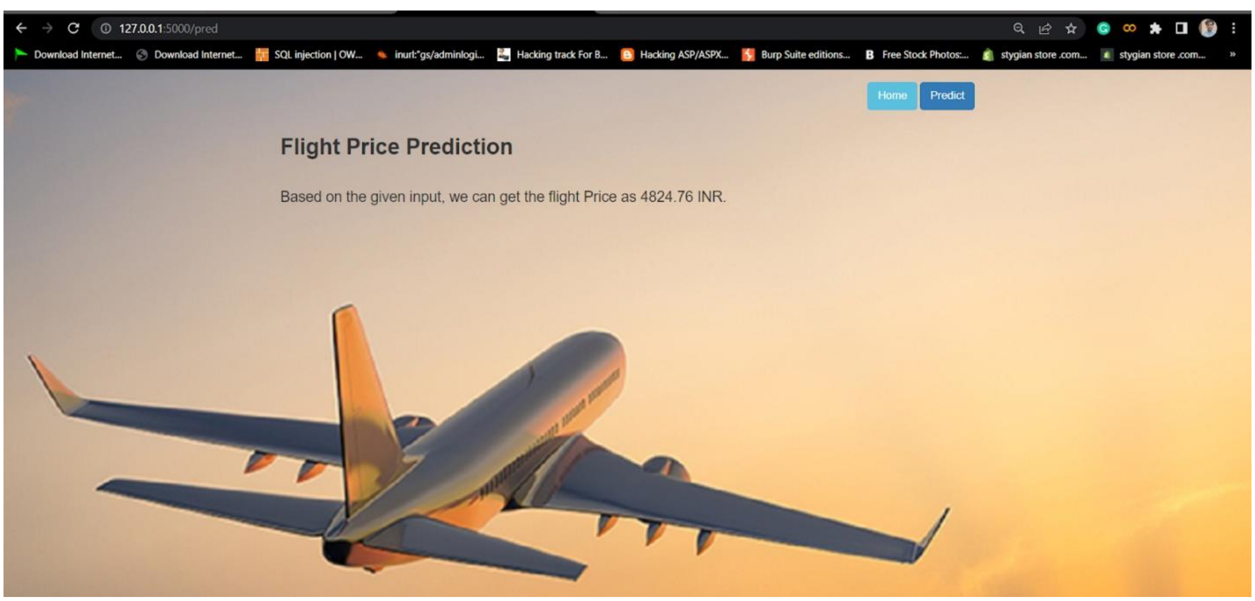





source
Bangalore
destination
Bangalore
depdate
Date
depmonth
Month
depyear
Year
deptimehour
Dep_Time_Hour
deptimemins
Dep_Time_Mins
artime
Arrival_Date
artimehour
Arrival_Time_Hour
artimemins
Arrival_Time_Mins
Submit


Home
Predict
Flight Price Prediction
airline
Air Asia
source
Chennai
destination
Delhi
depdate
13
depmonth
5
depyear
2019
deptimehour
10
deptimemins
20
artime
14
artimehour
11
artimemins
15
Activate Windows
Go to Settings to activate Windows


Home
Predict
Flight Price Prediction
airline
Air Asia
source
Chennai
destination
Delhi
depdate
13
depmonth
5
depyear
2019
deptimehour
10
deptimemins
20
artime
14
artimehour
11
artimemins
15
Activate Windows
Go to Settings to activate Windows


127.0.0.1:5000/pred
Home
Predict
Flight Price Prediction
Based on the given input, we can get the flight Price as 4824.76 INR.

4. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- Flexibility: Flight price prediction can provide travelers with insights on the best times to book flights, including recommendations on whether to book early or wait for prices to drop. This flexibility allows travelers to make informed decisions based on their travel plans, preferences, and budget, giving them more control over their travel arrangements.
- Customization: Flight price prediction can provide personalized recommendations to travelers based on their travel preferences and past booking history. This allows travelers to receive tailored suggestions on the best times to book flights that align with their travel plans and preferences, making the booking process more personalized and convenient.
- Time Savings: Monitoring flight prices can be time-consuming, especially when travelers are trying to find the best deals. Flight price prediction can save travelers time by automating the process of monitoring prices and providing them with timely information on when to book their flights. This eliminates the need for travelers to constantly check prices and allows them to focus on other aspects of their trip planning.
- Planning and Budgeting: Flight price prediction can help travelers plan and budget their trips more effectively. By having an estimate of future flight prices, travelers can better plan their travel expenses, allocate their budget accordingly, and make informed decisions about their travel plans. This can be particularly useful for budget-conscious travelers or those with limited travel budgets.

DISADVANTAGES:

- Cost savings: One of the primary benefits of flight price prediction is that it can help travelers save money by allowing them to find the best deals on flights. By predicting future price changes, travelers can choose to book their flights when prices are expected to be lower.
- Convenience: Flight price prediction can make it easier for travelers to plan their trips, as they can get a sense of how much they can expect to pay for their flights in advance. This can help them make informed decisions about when to book their flights and how much to budget for their trip.
- Time-saving: Flight price prediction can also save time for travelers who would otherwise need to spend hours monitoring prices and comparing different flights manually. By using a flight price prediction tool, travelers can quickly and easily find the best deals without having to do all the legwork themselves.
- Increased revenue for airlines: Flight price prediction can also benefit airlines by helping them optimize their pricing strategies and increase revenue. By accurately predicting demand and setting prices accordingly, airlines can ensure that their flights are fully booked and generate maximum revenue.

5. APPLICATIONS

- Travel planning: Flight price prediction can be used to plan trips ahead of time, allowing travelers to make informed decisions about when to book their flights. By predicting flight prices, travelers can plan their trips around the most affordable times to travel.
- Cost savings: Flight price prediction can help travelers save money by identifying the most affordable times to travel. Travelers can use this information to book their flights during off-peak times, when prices are likely to be lower.
- Competitive pricing: Airlines can also use flight price prediction to stay competitive by offering lower prices during periods of low demand. By using predictive models, airlines can adjust their pricing strategies to match demand and stay competitive in the market.

Overall, flight price prediction can be a useful tool for both travelers and airlines, helping to improve travel planning, budgeting, and cost savings while also supporting competitive pricing in the industry.

6 .CONCLUSION

In conclusion, flight price prediction is a valuable tool for both travelers and airlines, providing insights into future pricing trends and helping to optimize travel planning and budgeting. By using predictive models, travelers can make informed decisions about when to book their flights, while airlines can adjust their pricing strategies to match demand and stay competitive in the market. Additionally, flight price prediction can support cost savings for travelers, making travel more affordable and accessible. As technology continues to advance, we can expect flight price prediction to become even more accurate and reliable, further improving the travel experience for everyone involved.

7. FUTURE SCOPE

The field of flight price prediction is continuously evolving, with advancements in technology and data science driving new opportunities and applications. Here are some potential future scopes in flight price prediction:

- Improved accuracy: With the use of machine learning algorithms and big data, flight price prediction models can become more accurate and reliable. By incorporating more variables such as weather conditions, geopolitical events, and economic indicators, models can provide more precise predictions.
- Personalized pricing: Airlines can use flight price prediction models to offer personalized pricing based on a traveler's preferences and behavior. This can improve customer loyalty and increase revenue for airlines.
- Real-time pricing: The use of real-time data and analytics can enable airlines to adjust prices dynamically in response to changes in demand and supply. This can help airlines optimize revenue and provide better value to customers.
- Integration with other travel-related services: Flight price prediction can be integrated with other travel-related services such as hotel and car rental bookings to offer more comprehensive and personalized travel packages.

Overall, the future of flight price prediction looks promising with the potential to improve the travel experience for both travelers and airlines. As technology continues to evolve, we can expect to see even more innovative applications and advancements in this field.

8. APPENDIX

A. Source Code

Importing the Libraries

```
from pandas.core.indexes import category
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier,
GradientBoostingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report, confusion_matrix
import warnings
import pickle
from scipy import stats
warnings.filterwarnings('ignore')
plt.style.use("fivethirtyeight")
data=pd.read_csv("/content/sample_data/Data_Train.csv")
data.head()

for i in category:
```

```
  print(i,data[i].unique)

data.Date_of_Journey=data.Date_of_Journey.str.split('/')
data.Date_of_Journey

data['Date']=data.Date_of_Journey.str[0]
data['Month']=data.Date_of_Journey.str[1]
data['Year']=data.Date_of_Journey.str[2]

data.Total_Stops.unique()
data.Route=data.Route.str.split('->')
data.Route

data['city1']=data.Route.str[0]
data['city2']=data.Route.str[1]
data['city3']=data.Route.str[2]
data['city4']=data.Route.str[3]
data['city5']=data.Route.str[4]
data['city6']=data.Route.str[5]

data.Dep_Time=data.Dep_Time.str.split(':')

data['Dep_Time_Hour']=data.Dep_Time.str[0]
data['Dep_Time_Mins']=data.Dep_Time.str[1]

data.Arrival_Time=data.Arrival_Time.str.split(' ')
```

```
data['Arrival_date']=data.Arrival_Time.str[1]
data['Time_of_Arrival']=data.Arrival_Time.str[0]

data['Time_of_Arrival']=data.Time_of_Arrival.str.split(':')

data['Arrival_Time_Hour']=data.Time_of_Arrival.str[0]
data['Arrival_Time_Mins']=data.Time_of_Arrival.str[1]

data.Duration=data.Duration.str.split('')

data['Travel_Hours']=data.Duration.str[0]
data['Travel_Hours']=data['Travel_Hours'].str.split('h')
data['Travel_Hours']=data['Travel_Hours'].str[0]
data.Travel_Hours=data.Travel_Hours
data['Travel_Mins']=data.Duration.str[1]

data.Travel_Mins=data.Travel_Mins.str.split('m')
data.Travel_Mins=data.Travel_Mins.str[0]

data.Total_Stops.replace('non_stop',0,inplace=True)
data.Total_Stops=data.Total_Stops.str.split(' ')
data.Total_Stops=data.Total_Stops.str[0]

data.Additional_Info.unique()

data.Additional_Info.replace('No Info','No info',inplace=True)
```

```
data.isnull().sum()
data.drop(['city4','city5','city6'],axis=1,inplace=True)

data.drop(['Date_of_Journey','Route','Dep_Time','Arrival_Time','Durati
on'],axis=1,inplace=True)
data.drop(['Time_of_Arrival'],axis=1,inplace=True)

data.isnull().sum()
data['Arrival_date'].fillna(data['Date'],inplace=True)

data['Travel_Mins'].fillna(0,inplace=True)

data.info()

data.Date=data.Date
data.Month=data.Month
data.Year=data.Year
data.Dep_Time_Hour=data.Dep_Time_Hour
data.Dep_Time_Hour=data.Dep_Time_Hour
data.Dep_Time_Mins=data.Dep_Time_Mins
data.Arrival_date=data.Arrival_date
data.Arrival_Time_Hour=data.Arrival_Time_Hour
data.Arrival_Time_Mins=data.Arrival_Time_Mins
data.Travel_Mins=data.Travel_Mins

data[data['Travel_Hours']=='5m']
```

```
data.Travel_Hours=data.Travel_Hours

categorical=['Airline','Source','Destination','Additional_Info','City1','Cit
y2','City3']
numerical=['Total_Stops','Date','Month','Year','Dep_Time_Hour','Dep_
Time_Mins','Arrival_date','arrival_Time_Hour','Arrival_Time_Mins','T
ravel_Hours','Travel_Mins']

from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()

data.Airline=le.fit_transform(data.Airline)
data.Source=le.fit_transform(data.Source)
data.Destination=le.fit_transform(data.Destination)
data.Total_Stops=le.fit_transform(data.Total_Stops)
data.City1=le.fit_transform(data.City1)
data.City2=le.fit_transform(data.City2)
data.City3=le.fit_transform(data.City3)
data.Additional_Info=le.fit_transform(data.Additional_Info)
data.head()

data.head()

data
=data[['Airline','Source','Destination','Date','Month','Year','Dep_Time_
Hour','Dep_Time_Mins','Arrival_date','Arrival_Time_Minus','Price']]
```

```
data.head()

import seaborn as sns
c=1

plt.figure(figsize=(20,45))

for i in categorical:
  plt.subplot(6,3,c)
  sns.countplot(data[i])
  plt.xticks(ritation=90)
  plt.tight_layout(pad=3.0)
  c=c+1

plt.show()
plt.figure(figsize=(15,8))
sns.displot(data.Price)
sns.heatmap(data.corr(),annot=True)

import seaborn as sns
sns.boxplot(data['Price'])
y=data['Price']
x=data.drop(columns=['Price'],axis=1)
from sklearn.preprocessing import StandardScaler
ss=StandardScaler()
x_scaled=ss.fit_transform(x)
```

```
x_scaled=pd.DataFrame(x_scaled,columns=x.columns)
x_scaled.head()
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_sta
te=42)

x_train.head()


from                    sklearn.ensemble                    import
RandomForestRegressor,GradientBoostingRegressor,AdaBoostRegressor
rfr=RandomForestRegressor
gb=GradientBoostingRegressor
ad=AdaBoostRegressor

from                    sklearn.metrics                    import
r2_score,mean_absolute_error,mean_squared_error

for i in [rfr,gb,ad]:
   i.fit(x_train,y_test)
   y_pred=i.predict(x_test)
   test_score=r2_score(y_test,y_pred)
   train_score=r2_score(y_train,i.predict(x_train))
   if abs(train_score-test_score)<=0.2:
    print(i)

   print("R2 score is",r2_score(y_test,y_pred))
```

```
    print("R2 for train data",r2_score(y_train,i.predict(x_train)))
    print("Mean Absolute Error is",mean_absolute_error(y_pred,y_test))
    print("Mean Squared Error is",mean_squared_error(y_pred,y_test))
    print("Root Mean Squared Error is",(mean_squared_error(y_pred,y_test,squared=False)))

from sklearn.neighbors import KNeighborsRegressor
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor

from sklearn.metrics import r2_score,mean_absolute_error,mean_squared_error

knn=KNeighborsRegressor()
svr=SVR()
dt=DecisionTreeRegressor()

for i in [knn,svr,dt]:
  i.fit(x_train,y_train)
  y_pred=i.predict(x_test)
  test_score=r2_score(y_test,y_pred)
  train_score=r2_score(y_train,i.predict(x_train))
  if abs(train_score-test_score)<=0.1:
    print(i)
    print('R2 Score is',r2_score(y_test,y_pred))
    print('R2 Score for train data',r2_score(y_train,i.predict(x_train)))
    print('Mean Absolute Error is',mean_absolute_error(y_test,y_pred))
```

```
    print('Mean Squared Error is',mean_squared_error(y_test,y_pred))
    print('Root Mean Squared Error is',(mean_squared_error(y_test,y_pred,squared=False)))
from sklearn.model_selection import cross_val_score
for i in range(2,5):
  cv=cross_val_score(rfr,x,y,cv=i)
  print(rfr,cv.mean())

from sklearn.model_ selection import RandomizedSearchCV

param_grid={'n_estimators':[10,30,50,70,100],'max_depth':[None,1,2,3],'max_features':['auto','sqrt']}
rfr=RandomForestRegressor()
rfr_res=RandomizedSearchCV(estimator=rfr,param_distributions=param_grid,cv=3,verbose=2,n_jobs=-1)

rf_res.fit(x_train,y_train)

gb=GradientBoostingRegressor()
gb_res=`RandomizedSearchCV(estimator=gb,param_distributions=param_grid,cv=3,verbose=2,n_jobs=-1)

gb_res.fit(x_train,y_train)
rfr=RandomForestRegressor(n_estimators=10,max_features='sqrt',max_depth=None)
rfr.fit(x_train,y_train)
y_train_pred=rfr.predict(x_train)
```

```
y_test_pred=rfr.predict(x_test)
print("train accuracy",r2_score(y_train_pred,y_train))
print("test accuracy",r2_score(y_test_pred,y_test))
price_list=pd.DataFrame({'Price':prices})

price_list
import pickle
pickle.dump(rfr.open('model1.pk1','wb'))
```