GOVERNMENT OF TAMILNADU

Naan Muthalvan - Project-Based Experiential Learning

Flight Price Prediction Using Machine Learning

Submitted by

TEAM ID : NM2023TMID22225

P.RANJITHA (20326ER028)

S.PRIYAREKA(20326ER026)

S.RAJAGOWRI (20326ER027)

M.RENUGA DEVI (20326ER029)

Under the guidance of

Mrs. J. SUKANYA, MCA., M.Phil.,

Assistant Professor

PG and Research Department of Computer Science

M.V.MUTHIAH GOVERNMENT ARTS COLLEGE FOR WOMEN

(Affiliated To Mother Teresa Womens University, Kodaikanal)

Reaccredited with ”A” Grade by NAAC

DINDIGUL-624001.

APRIL - 2023

M.V.MUTHIAH GOVERNMENT ARTS COLLEG FOR WOMEN

(Affiliated to Mother Teresa Womens 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," FLIGHT PRICE PREDICTION" done **by Ms.S.PRIYAREKA (20326ER026) and Ms.S.RAJAGOWARI (20326ER027) and Ms.P.RANJITHA (20326ER028) and Ms.M.RENUGA DEVI (20326ER029)**. 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.

J.Sukany-

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

Submitted for viva-voce Examination held on 11. 04. 2023

TABLE OF CONTENTS

ABSTRACT

Flight ticket fare is the most fluctuating data which varies every day. Depending on the various factors that affect it directly or indirectly. we cannot say that the price of flight ticket fare remains the same or not. It is quite a tough task to predict the flight ticket fare. It may change throughout the week, month or some days, but it can be predicted nearly accurate to the actual flight ticket fare.The prime objective of our project "Improved Flight Prediction System" is to make aprediction of the flight ticket fare for the future flights. The proposed approach is using machine learning algorithm and we are using supervised learning .The regression model which we have selected for our prediction is "Extreme Gradient Boosting". In this approach we have developed this project in python language for backend. For the GUI Bootstrap, HTML,CSS using diango framework and python using tkinter.

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

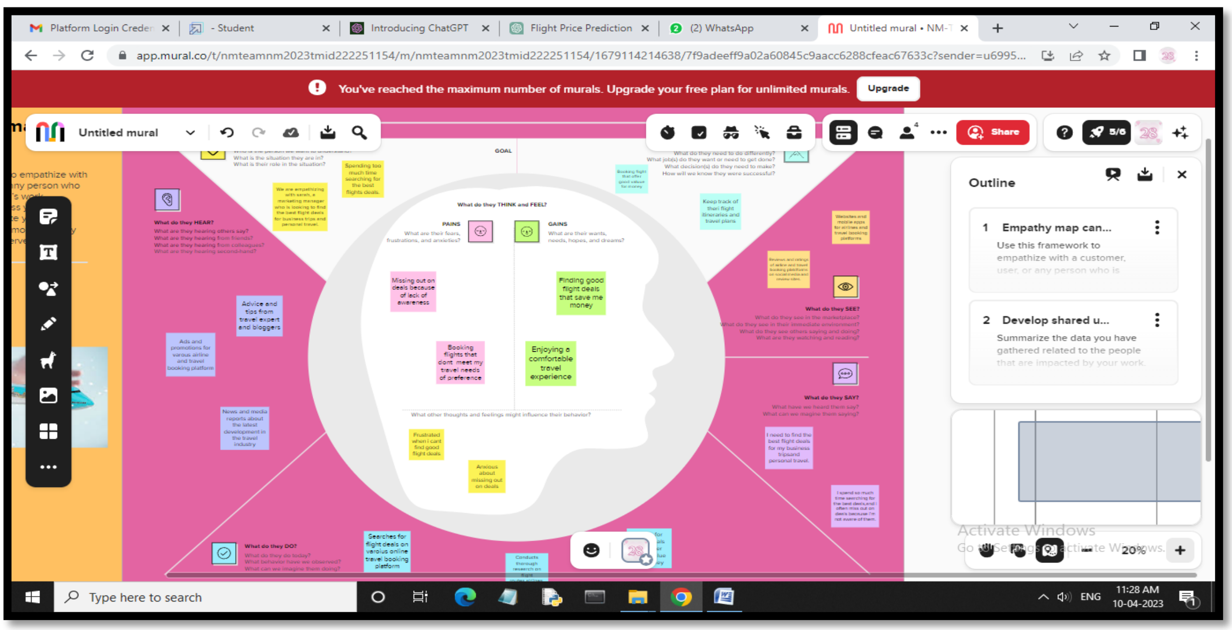
The purpose of flight price prediction is to help travelers make informed decisions about when to purchase flight tickets. Flight prices are known to fluctuate frequently, and predicting future price changes can be challenging for consumers. By using machine learning algorithms and historical flight 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 & DESIGN THINKING

2.1 EMPATHY MAP

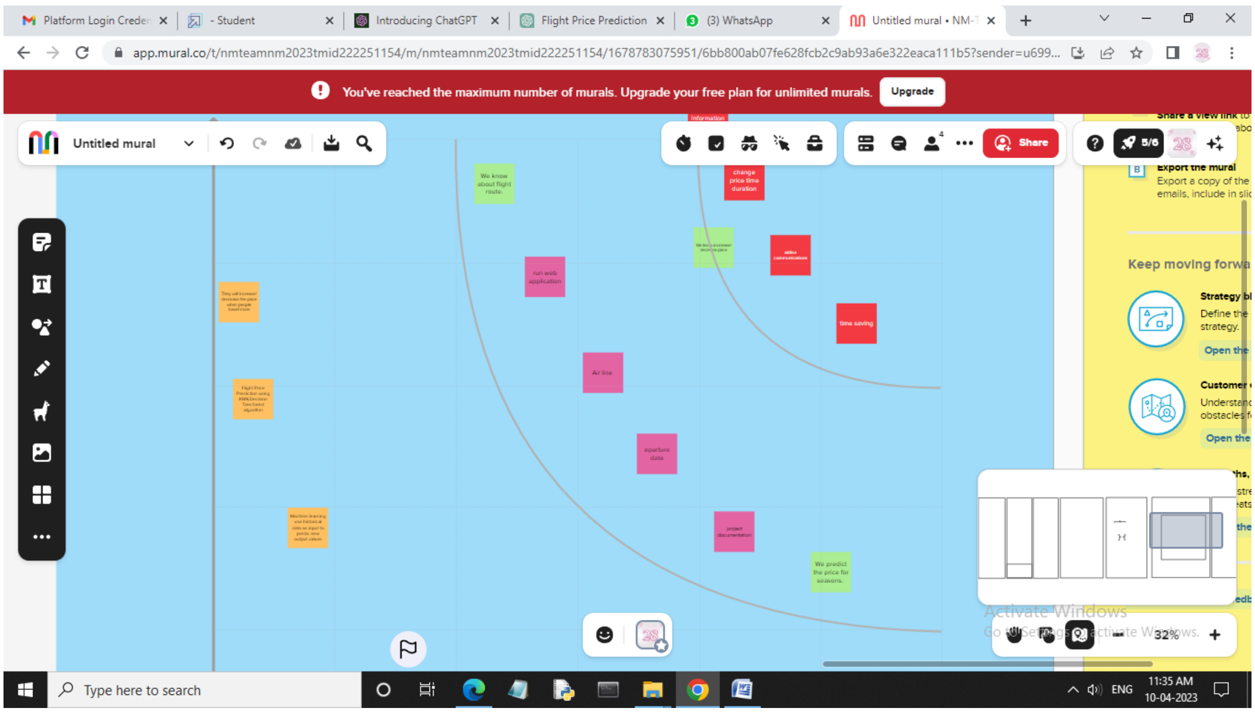
Empathy for Identifying Patterns and Trends in Campus Placement Data using Machine Learning

Empathy in this case can refer to the ability of the machine learning algorithm to understand the context and nuances of the data it is analyzing. This includes understanding the factors that may impact the placement of students, such as their academic performance, background, and the current job market.

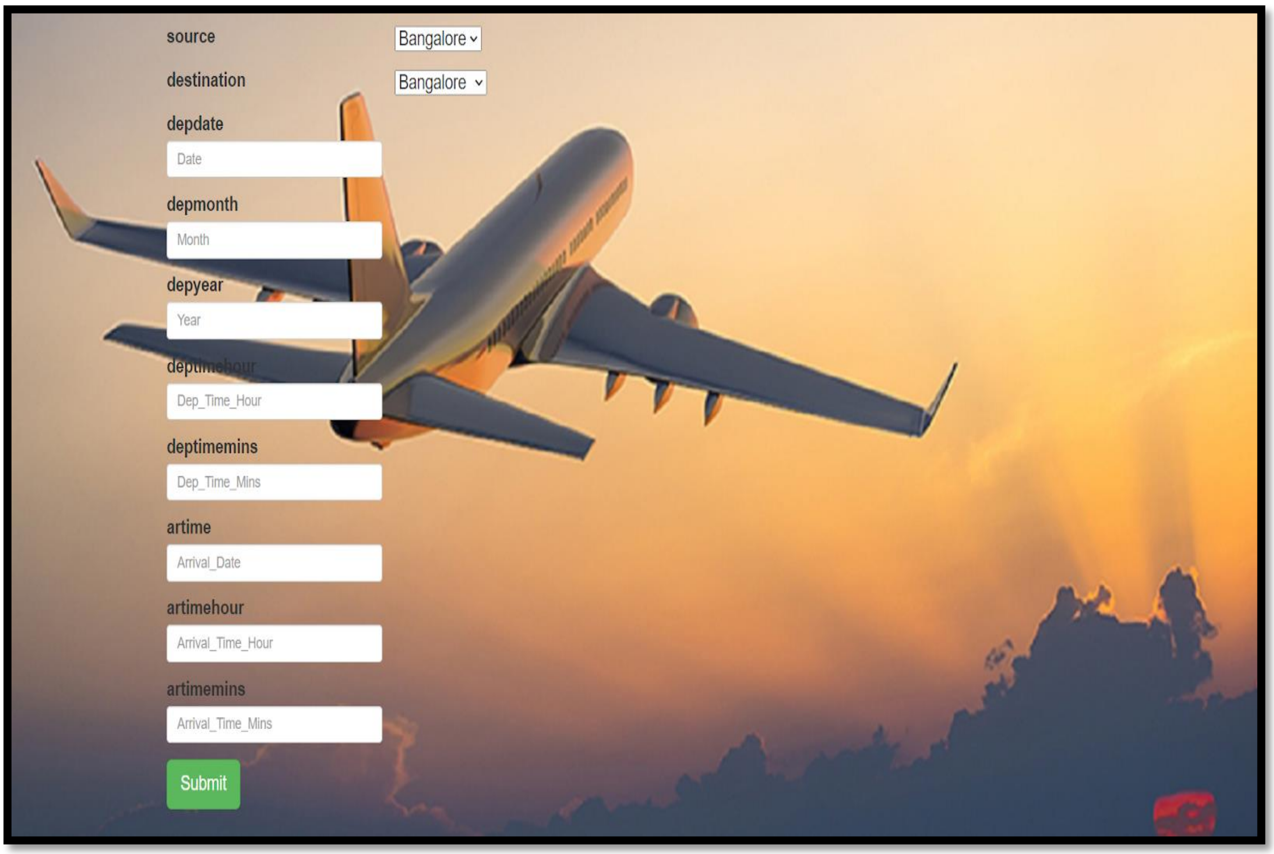


2.2 IDEATION & BRAINSTROMING MAP

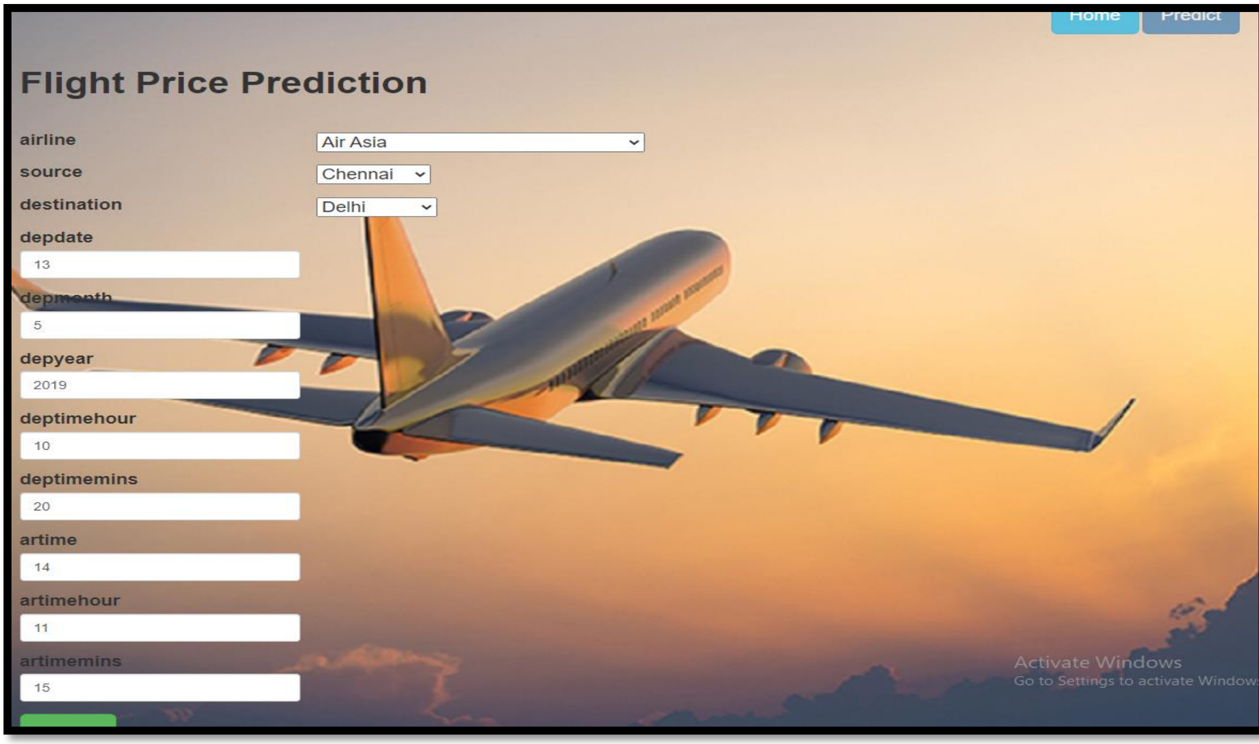
Brainstorm Map for Identifying Patterns and Trends in Campus Placement Data using Machine Learning.

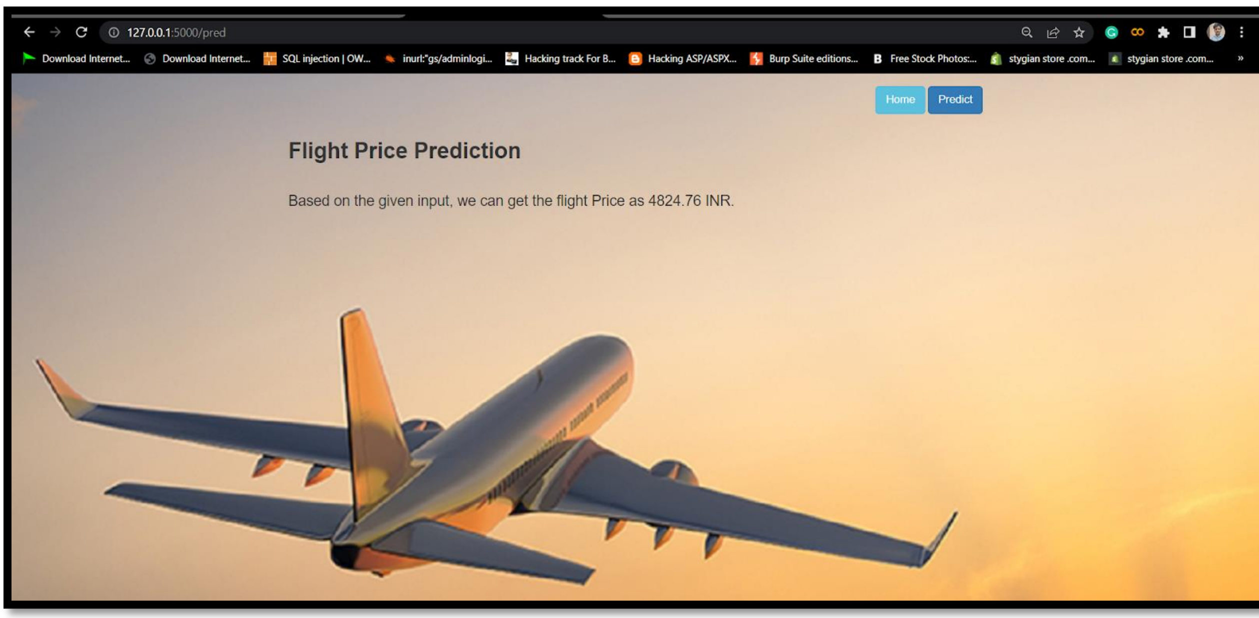


3. RESULT



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


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:

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.

DISADVANTAGES:

Reliance on historical data: Flight price prediction models are typically based on historical data, which may not always accurately reflect current market conditions or future trends. Unforeseen events such as natural disasters, geopolitical events, or changes in airline policies can significantly impact flight prices and may not be fully captured in historical data.

Overreliance on predictions: Travelers who heavily rely on flight price predictions may be inclined to delay booking their flights in the hope of getting a better deal. However, prices can also increase unexpectedly, resulting in travelers paying higher prices or missing out on desired flights. Overreliance on predictions can also result in missed opportunities for promotions, discounts, or other favorable booking conditions.

Lack of human touch: Flight price prediction models are typically automated and may not take into account subjective factors such as traveler preferences, special needs, or unique travel circumstances. Personalized advice or assistance that a human travel agent can provide may be lacking in flight price prediction tools.

5. APPLICATION

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.

Budgeting:

- By knowing the estimated cost of a flight, travelers can budget accordingly and avoid being surprised by unexpected expenses.

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

8.1 SOURCE CODE

```
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','Duration'],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','City2','City3']
numerical=['Total_Stops','Date','Month','Year','Dep_Time_Hour','Dep_Time_Mins','Arrival_date'
,'arrival_Time_Hour','Arrival_Time_Mins','Travel_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_state=42)

x_train.head()
```

```
from sklearn.ensemble import RandomForestRegressor,GradientBoostingRegressor,AdaBoostRe
gressor
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','sq
rt']}
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'))
```