FLIGHT BOOKING PRICE PREDICTION PROJECT REPORT

Team Id : NM2023TMID31598

	Name of the Student	NM ID
TEAM LEADER	P.Muthu lakshmi	B091F9A96F5D2B9B0654ECCD6B5CA3DF
TEAM MEMBERS	S.Subha	71C76C3B3ED2C58A76B449308ECB06C6
	M.Muppudathi	6C54D137EC4120985484247 55A89A5EB
	A.Santhiya	824553D6BA40A9AB40C4B21575305C3E

GOVERNMENT ARTS AND SCIENCE COLLEGE

KADAYANALLUR

INDEX

1. INTRODUCTION

1.1 Overview

- A flight price prediction project aims to predict the future prices of airline tickets using machine learning algorithms. The goal is to help travelers make informed decisions about when to book their flights to get the best possible price.
- The project involves collecting historical flight data, including departure and arrival airports, dates, airlines, and ticket prices, and using this data to train a machine learning model. The model can then be used to make predictions about future flight prices based on new input data.
- To build a flight price prediction model, several steps are typically involved:
- Data Collection: Collecting historical flight data from multiple sources, such as airlines, travel agencies, and third-party websites.
- Data Cleaning: Cleaning and preprocessing the data to remove outliers, missing values, and other errors.
- Feature Engineering: Creating new features that can help the model better predict flight prices, such as time of day, day of the week, and seasonality.
- Model Selection: Selecting the appropriate machine learning algorithm based on the type of data and the problem being solved.
- Model Training: Training the model on historical data to learn patterns and relationships between features and flight prices.

1.2Purpose

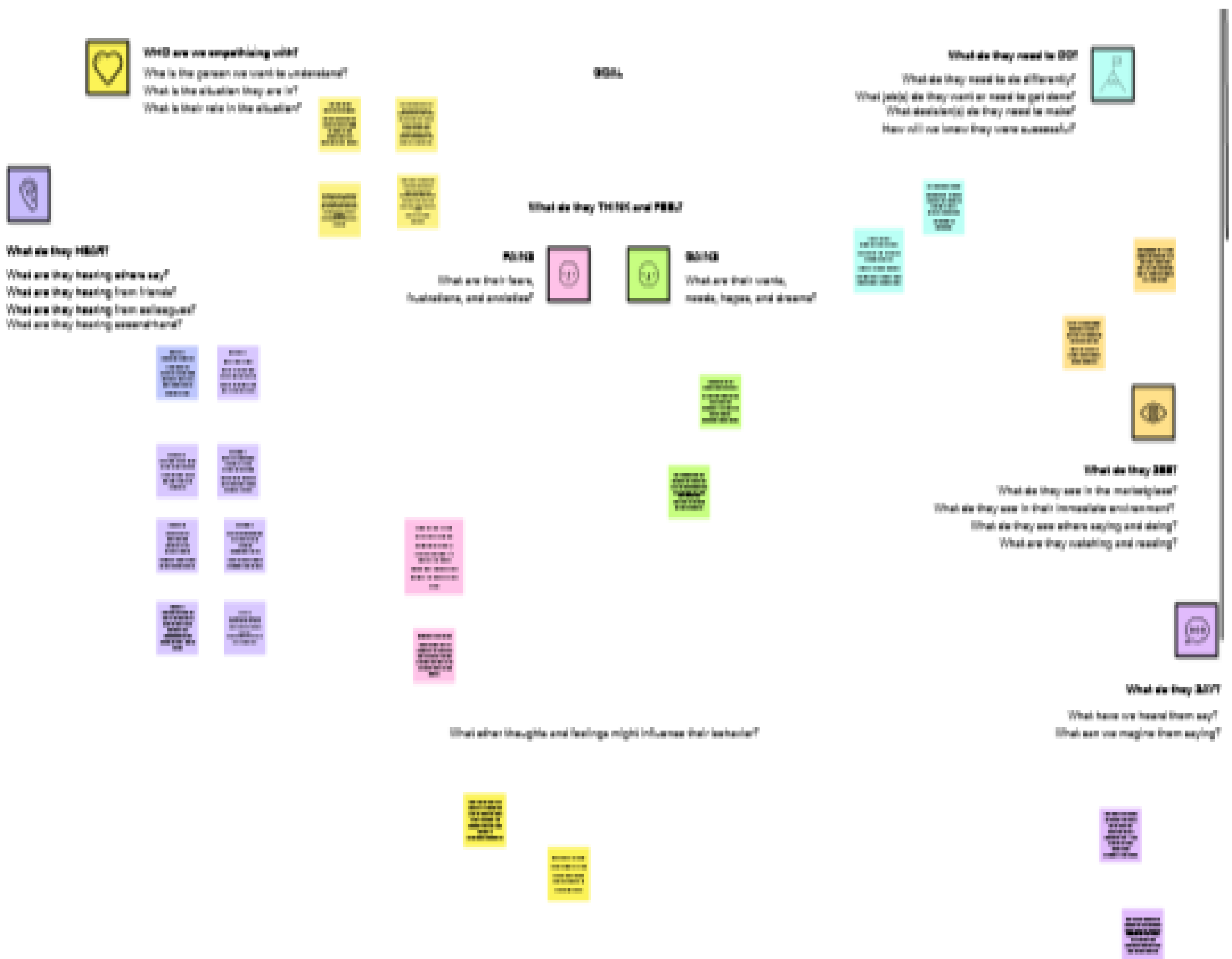
- The purpose of a flight price prediction project is to provide travelers with accurate and timely information about future flight prices, allowing them to make informed decisions about when to book their flights. By predicting flight prices in advance, travelers can take advantage of lower prices and avoid overpaying for their tickets.
- Flight price prediction models can also be useful for airlines and travel agencies to optimize their pricing strategies and revenue management. By accurately predicting future demand and adjusting prices accordingly, airlines can maximize their revenue and reduce the number of unsold seats.
- Additionally, flight price prediction models can be used to improve customer satisfaction by providing personalized recommendations and alerts based on a user's travel preferences and past booking history.

2. problem Definition & DesignThinking

- The problem that a flight price prediction project seeks to solve is the unpredictability and volatility of airline ticket prices. Airline ticket prices are subject to various factors such as demand, seasonality, competition, and fuel prices. As a result, ticket prices can fluctuate frequently, making it challenging for travelers to determine the best time to book their flights.

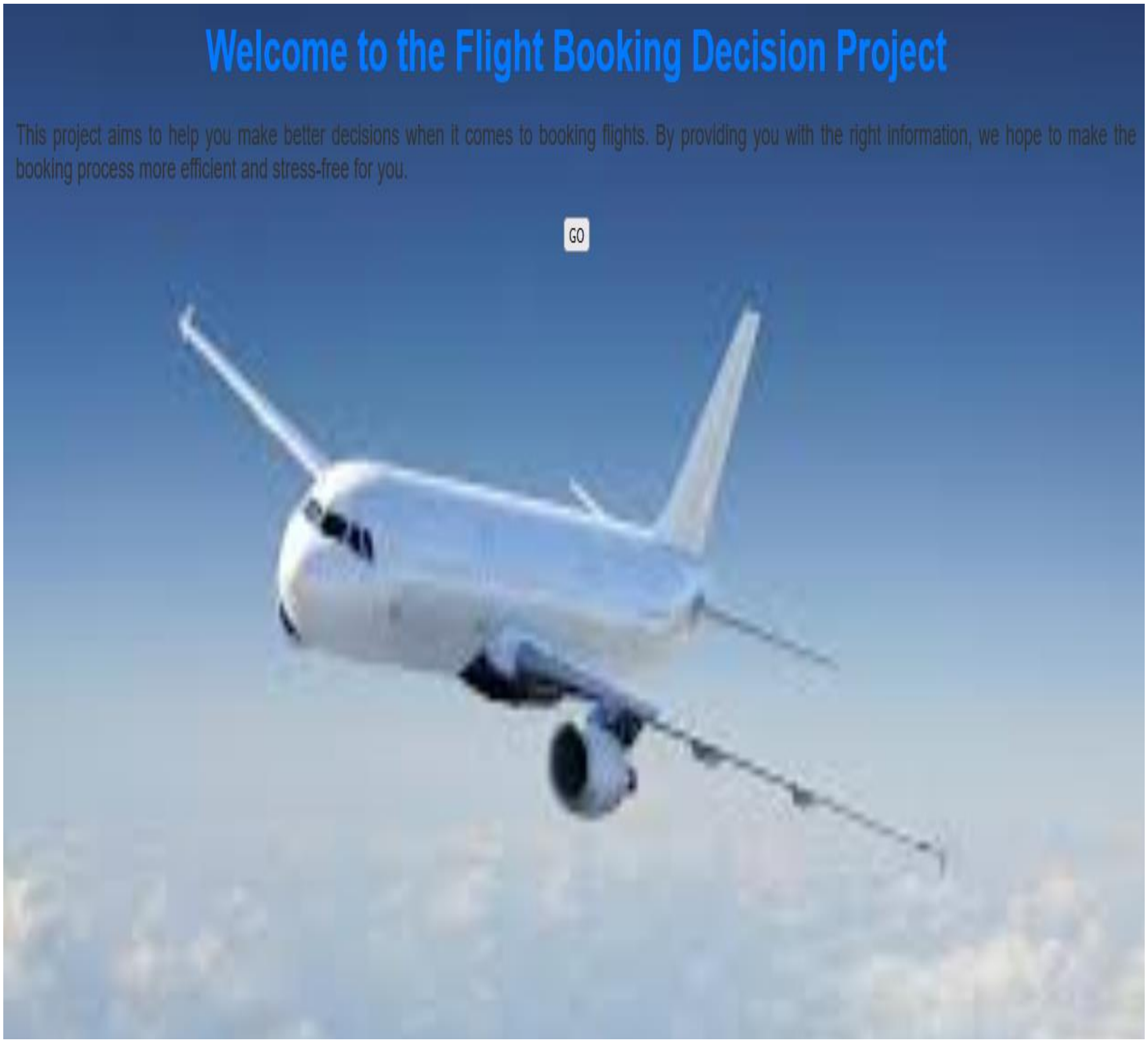
- The problem is compounded by the fact that many airlines use dynamic pricing strategies, where ticket prices can change multiple times a day based on demand and other factors. This can lead to confusion and frustration for travelers who may feel like they are being charged unfair or arbitrary prices.
- A flight price prediction project aims to solve this problem by providing travelers with accurate and timely predictions of future flight prices. By using machine learning algorithms to analyze historical flight data and other relevant factors, the project can predict future ticket prices and provide recommendations on when to book flights to get the best price.

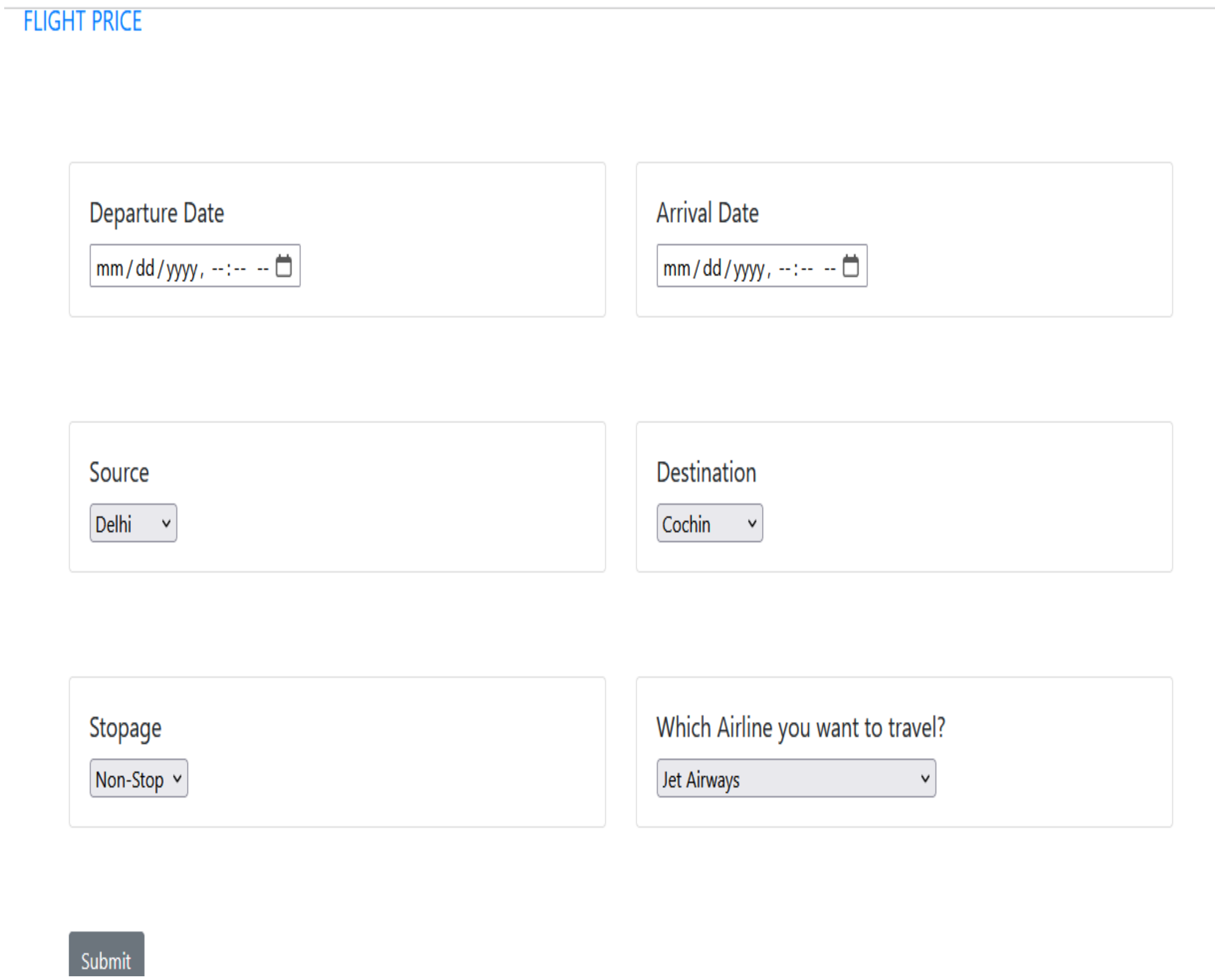
2.1 Empathy Map



2.2 Ideation & Brainstorming

3. RESULT


Welcome to the Flight Booking Decision Project
This project aims to help you make better decisions when it comes to booking flights. By providing you with the right information, we hope to make the booking process more efficient and stress-free for you.
GO


FLIGHT PRICE
Departure Date
mm/dd/yyyy, --:-- --
Arrival Date
mm/dd/yyyy, --:-- --
Source
Delhi
Destination
Cochin
Stopage
Non-Stop
Which Airline you want to travel?
Jet Airways
Submit

The value is Your Flight price is Rs. 5786.5

4. ADVANTAGES AND DISADVANTAGES

Advantages of a flight price prediction project

- Cost savings: Travelers can save money by booking flights at the optimal time based on predicted prices, resulting in significant cost savings.
- Convenience: Flight price prediction models can provide travelers with real-time updates on flight prices and recommend the best time to book, making it more convenient for them to plan their trips.
- Personalization: By analyzing user data and preferences, flight price prediction models can provide personalized recommendations and alerts based on a user's travel history and preferences.
- Revenue optimization: Airlines and travel agencies can optimize their revenue management strategies by using flight price prediction models to accurately predict demand and adjust prices accordingly.
- Competitive advantage: Businesses that can accurately predict flight prices have a competitive advantage over their competitors, as they can offer better pricing and more value to their customers.

Disadvantages of a flight price prediction project

- Data availability: A significant amount of historical flight data is required to build an accurate flight price prediction model, which may not be available in some regions or for some airlines.
- Data accuracy: The accuracy of the model is highly dependent on the quality and accuracy of the historical data used to train the model.
- Market unpredictability: Unforeseeable market events such as natural disasters, political instability, and pandemics can affect airline ticket prices in ways that are difficult to predict, making it challenging for the model to provide accurate predictions.
- Overreliance on the model: Travelers who rely too heavily on flight price prediction models may miss out on last-minute deals or other discounts that are not reflected in the model's predictions.
- Limited scope: Flight price prediction models are limited to predicting ticket prices and do not take into account other factors such as travel time, layovers, and airline amenities, which may be important considerations for some travelers.

5. APPLICATION

- Travel planning: Travelers can use flight price prediction models to plan their trips more effectively, by finding the best time to book flights and saving money on airline tickets.
- Revenue management: Airlines and travel agencies can use flight price prediction models to optimize their pricing strategies and revenue management, by accurately predicting demand and adjusting prices accordingly.
- Marketing: Travel companies can use flight price prediction models to offer personalized recommendations and promotions to their customers based on their travel history and preferences.
- Competitive analysis: Airlines and travel agencies can use flight price prediction models to analyze their competitors' pricing strategies and gain insights into market trends and demand.
- Customer retention: By providing personalized recommendations and alerts based on a user's travel history and preferences, flight price prediction models can help travel companies retain customers and increase loyalty.
- Forecasting and trend analysis: Flight price prediction models can also be used to analyze historical trends and make predictions about future market developments and changes in demand.

6. CONCLUTION

- In conclusion, a flight price prediction project is a valuable tool for both travelers and businesses in the travel industry. By leveraging machine learning algorithms to analyze historical flight data and other relevant factors, flight price prediction models can accurately predict future flight prices and provide recommendations on when to book flights to get the best price.
- For travelers, flight price prediction models can help save money and make travel planning more convenient by providing real-time updates on flight prices and personalized recommendations based on their travel history and preferences. For businesses, flight price prediction models can optimize pricing strategies and revenue management, provide insights into market trends and competitor pricing, and improve customer satisfaction and retention.
- However, it is essential to note that flight price prediction models have their limitations and challenges, such as data availability and accuracy, market unpredictability, and the potential for overreliance on the model. Therefore, it is crucial to use flight price prediction models as one of many tools in travel planning and pricing strategies and to stay up-to-date on market trends and events that may affect ticket prices.

7. FUTURE SCOPE

- The future scope of flight price prediction projects is promising, as advancements in technology and data analysis continue to improve the accuracy and effectiveness of these models. Some potential areas of future development include:
- Integration with travel booking platforms: Flight price prediction models could be integrated with travel booking platforms, such as Expedia or Kayak, to provide real-time pricing updates and recommendations to travelers as they browse for flights.
- Expansion to other travel services: Flight price prediction models could be expanded to include other travel services, such as hotel accommodations or rental cars, to provide a more comprehensive view of travel costs and opportunities for savings.
- Use of alternative data sources: Flight price prediction models could incorporate alternative data sources, such as social media or weather data, to improve their predictions and account for unforeseen events that may affect flight prices.
- Predictive analytics for air cargo: Air cargo companies could use flight price prediction models to predict the cost of air freight and optimize their logistics operations.

8. APPENDIX

A.source code

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        from pandas import DataFrame
        from sklearn.model_selection import train_test_split
        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')

        sns.set()
```

```
In [2]: df=pd.read_excel("D:\Data_Train.xlsx")
```

In [3]:
```
df.head()
```

Out[3]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302

In [4]:
```
category=['Airline','Source','Destination','Additional_Info']
for i in category:
    print(i,df[i].unique())
    print('----------------------------S-------------------')
```

```
Airline ['IndiGo' 'Air India' 'Jet Airways' 'SpiceJet' 'Multiple carriers' 'GoAir'
 'Vistara' 'Air Asia' 'Vistara Premium economy' 'Jet Airways Business'
 'Multiple carriers Premium economy' 'Trujet']
----------------------------S-------------------
Source ['Banglore' 'Kolkata' 'Delhi' 'Chennai' 'Mumbai']
----------------------------S-------------------
Destination ['New Delhi' 'Banglore' 'Cochin' 'Kolkata' 'Delhi' 'Hyderabad']
----------------------------S-------------------
Additional_Info ['No info' 'In-flight meal not included' 'No check-in baggage included'
 '1 Short layover' 'No Info' '1 Long layover' 'Change airports'
 'Business class' 'Red-eye flight' '2 Long layover']
----------------------------S-------------------
```

In [5]:
```
df.Date_of_Journey=df.Date_of_Journey.str.split('/')
```

In [6]:
```
df.Date_of_Journey
```

Out[6]:
```
0        [24, 03, 2019]
1         [1, 05, 2019]
2         [9, 06, 2019]
3        [12, 05, 2019]
4        [01, 03, 2019]
              ...
10678     [9, 04, 2019]
10679    [27, 04, 2019]
10680    [27, 04, 2019]
10681    [01, 03, 2019]
10682     [9, 05, 2019]
Name: Date_of_Journey, Length: 10683, dtype: object
```

In [7]:
```
df['Date']=df.Date_of_Journey.str[0]
df['Month']=df.Date_of_Journey.str[1]
df['Year']=df.Date_of_Journey.str[2]
```

In [8]:
```
df.Total_Stops.unique()
```

Out[8]:
```
array(['non-stop', '2 stops', '1 stop', '3 stops', nan, '4 stops'],
      dtype=object)
```

In [9]:
```
df.Route=df.Route.str.split('→')
df.Route
```

In [9]:
```
df.Route=df.Route.str.split('→')
df.Route
```

Out[9]:
```
0                    [BLR ,  DEL]
1       [CCU ,  IXR ,  BBI ,  BLR]
2       [DEL ,  LKO ,  BOM ,  COK]
3              [CCU ,  NAG ,  BLR]
4              [BLR ,  NAG ,  DEL]
                    ...
10678                [CCU ,  BLR]
10679                [CCU ,  BLR]
10680                [BLR ,  DEL]
10681                [BLR ,  DEL]
10682   [DEL ,  GOI ,  BOM ,  COK]
Name: Route, Length: 10683, dtype: object
```

In [10]:
```
df.Additional_Info.unique()
```

Out[10]:
```
array(['No info', 'In-flight meal not included',
       'No check-in baggage included', '1 Short layover', 'No Info',
       '1 Long layover', 'Change airports', 'Business class',
       'Red-eye flight', '2 Long layover'], dtype=object)
```

In [11]:
```
df['City1']=df.Route.str[0]
df['City2']=df.Route.str[1]
df['City3']=df.Route.str[2]
df['City4']=df.Route.str[3]
df['City5']=df.Route.str[4]
df['City6']=df.Route.str[5]
```

In [13]:
```
df['Dep_Time_Hour']=df.Dep_Time.str[0]
df['Dep_Time_Mins']=df.Dep_Time.str[1]
```

In [14]:
```
df.Arrival_Time=df.Arrival_Time.str.split('')
```

In [15]:
```
df['Arrival_date']=df.Arrival_Time.str[1]
df['Time_of_Arrival']=df.Arrival_Time.str[0]
```

In [16]:
```
df['Time_of_Arrival']=df.Time_of_Arrival.str.split(':')
```

In [17]:
```
df['Arrival_Time_Hour']=df.Time_of_Arrival.str[0]
df['Arrival_Time_Mins']=df.Time_of_Arrival.str[1]
```

In [18]:
```
df.Duration=df.Duration.str.split(' ')
```

In [19]:
```
df['Travel_Hours']=df.Duration.str[0]
df['Travel_Hours']=df['Travel_Hours'].str.split('h')
df['Travel_Hours']=df['Travel_Hours'].str[0]
df.Travel_Hours=df.Travel_Hours
df['Travel_Mins']=df.Duration.str[1]

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

In [20]:
```
df.Total_Stops.replace('non_stop',0,inplace=True)
df.Total_Stops=df.Total_Stops.str.split(' ')
df.Total_Stops=df.Total_Stops.str[0]
```

In [21]:
```
df.Additional_Info.unique()
```

Out[21]:
```
array(['No info', 'In-flight meal not included',
       'No check-in baggage included', '1 Short layover', 'No Info',
       '1 Long layover', 'Change airports', 'Business class',
       'Red-eye flight', '2 Long layover'], dtype=object)
```

In [22]:
```
df.Additional_Info.replace('No Info','No info',inplace=True)
df = df.dropna(subset=['Total_Stops', 'City1', 'City2'])
```

In [23]:
```
df.isnull().sum()
```

Out[23]:
```
Airline                0
Date_of_Journey        0
Source                 0
Destination            0
Route                  0
Dep_Time               0
Arrival_Time           0
Duration               0
Total_Stops            0
Additional_Info        0
Price                  0
Date                   0
Month                  0
Year                   0
City1                  0
City2                  0
City3               3491
City4               9116
City5              10636
```

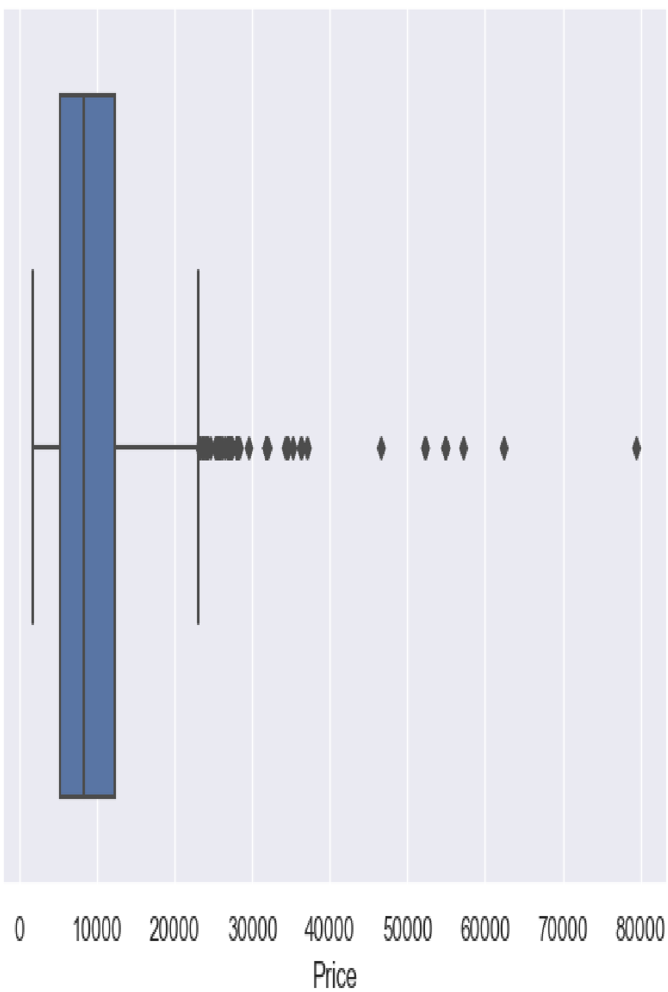
In [44]:
```
df.describe()
```

Out[44]:

	Airline	Source	Destination	Date	Month	Year	Dep_Time_Hour	Dep_Time_Mins	Arrival_date	Arrival_Time_Hour	A
count	10681.000000	10681.000000	10681.000000	10681.000000	10681.000000	10681.0	10681.000000	10681.000000	10681.000000	10681.0	
mean	3.966483	1.952064	1.436008	13.509784	4.708735	2019.0	12.490684	24.406891	0.896920	0.0	
std	2.352025	1.177165	1.474836	8.479449	1.164345	0.0	5.748989	18.767046	0.711826	0.0	
min	0.000000	0.000000	0.000000	1.000000	3.000000	2019.0	0.000000	0.000000	0.000000	0.0	
25%	3.000000	2.000000	0.000000	6.000000	3.000000	2019.0	8.000000	5.000000	0.000000	0.0	
50%	4.000000	2.000000	1.000000	12.000000	5.000000	2019.0	11.000000	25.000000	1.000000	0.0	
75%	4.000000	3.000000	2.000000	21.000000	6.000000	2019.0	18.000000	40.000000	1.000000	0.0	
max	11.000000	4.000000	5.000000	27.000000	6.000000	2019.0	23.000000	55.000000	2.000000	0.0	

In [48]:
```python
import seaborn as sns
sns.boxplot(df['Price'])
```

Out[48]: <AxesSubplot:xlabel='Price'>



In [51]:
```
###scaling the Data
```

In [52]:
```
from sklearn.preprocessing import StandardScaler
SS=StandardScaler()
```

In [53]:
```
x_scaled=SS.fit_transform(x)
```

In [54]:
```
x_scaled=pd.DataFrame(x_scaled,columns=x.columns)
x_scaled.head()
```

Out[54]:

	Airline	Source	Destination	Date	Month	Year	Dep_Time_Hour	Dep_Time_Mins	Arrival_date	Arrival_Time_Hour	Arrival_Time_Mins
0	-0.410934	-1.658354	2.416648	1.237192	-1.467619	0.0	1.654162	-0.234832	-1.260087	0.0	0.0
1	-1.261305	0.890262	-0.973718	-1.475375	0.250165	0.0	-1.303018	1.363790	0.144818	0.0	0.0
2	0.014251	0.040723	-0.295645	-0.531874	1.109057	0.0	-0.607211	0.031605	-1.260087	0.0	0.0
3	-0.410934	0.890262	-0.973718	-0.178060	0.250165	0.0	0.958355	-1.034142	1.549722	0.0	0.0
4	-0.410934	-1.658354	2.416648	-1.475375	-1.467619	0.0	0.610452	1.363790	1.549722	0.0	0.0

```
In [58]: from sklearn.ensemble import RandomForestRegressor,GradientBoostingRegressor,AdaBoostRegressor
rfr=RandomForestRegressor()
gb=GradientBoostingRegressor()
ad=AdaBoostRegressor()
```

```
In [59]: from sklearn.metrics import r2_score,mean_absolute_error,mean_squared_error
for i in [rfr,gb,ad]:
    i.fit(x_train,y_train)
    y_pred=i.predict(x_test)
    test_score=r2_score(y_test,y_pred) # Use y_test instead of y_train
    train_score=r2_score(y_train,i.predict(x_train))
    if abs(train_score-test_score)<=0.2:
        print(i)

    print("R2 score is",test_score) # Use test_score instead of r2_score(y_test, y_pred)
    print("R2 for train data",train_score)
    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 Absolute Error is",(mean_squared_error(y_test, y_pred, squared=False)))
```

```
RandomForestRegressor()
R2 score is 0.807591306712364
R2 for train data 0.9183894254749613
Mean Absolute Error is 1258.7924055321216
Mean Squared Error is 4062086.345165099
Root Mean Absolute Error is 2015.4618193270492
GradientBoostingRegressor()
R2 score is 0.7543904777503009
R2 for train data 0.7233202602774563
Mean Absolute Error is 1715.859530053641
```

In [64]:
```
#accuracy


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))
```

```
train accuracy 0.8953822817775416
test accuracy 0.7453755961372145
```

In [65]:
```
import pickle
pickle.dump(rfr,open('model1.pkl','wb'))
```

In [66]:
```
print(df.columns)
```

```
Index(['Airline', 'Source', 'Destination', 'Date', 'Month', 'Year',
       'Dep_Time_Hour', 'Dep_Time_Mins', 'Arrival_date', 'Arrival_Time_Hour',
       'Arrival_Time_Mins', 'Price', 'Additional_Info'],
      dtype='object')
```