**OPTIMIZING FLIGHT BOOKING DECISIONS THROUGH MACHINE LEARNING PRICE PREDICTIONS**

**TEAM NAME: SHINCHAN**

**TEAM LEADER: PRATHEEPA.A**

**TEAM MEMBER: BRINDHA.S**

**KALIDAS.R**

**SAKTHIVEL.R**

Optimizing Flight Booking Decisions

through Machine Learning Price Predictions

# **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 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 CHOOSEN DATASET AND IN THE PRICE WHEREIN THE AIRLINE PRICTICE IS VARY OVERTIME. we have implemented flight price prediction for users by using KNN, DECISION RANDOM FOREST ALGORITHMS. RANDOM FOREST SHOWS THE BEST ACCURACY OF 80% FOR PREDICTING THE ALSO, WE HAVE DONE CORRELATION TESTS AND METRICS FOR THE STATISTICAL ANALYSIS.**

## **Overview:**

## **OPTIMIZING FLIGHT BOOKING DECISION MACHINE THROUGH PRICE PREDICTIONS**

## **PROJECT DESCRIPTION**

# An airlines ticket booking system is a software application designed to automate the process of booking and managing airline tickets. It enables customers to search for and book flights, view flight schedules, and manage their reservations online.

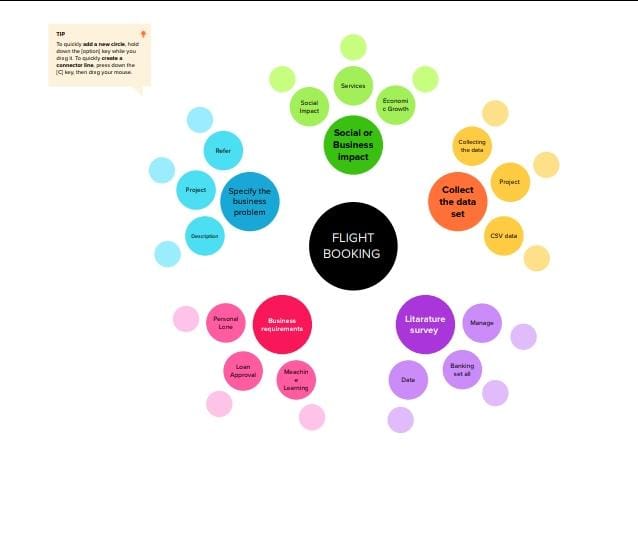
# purpose

# The web based “airline reservation system” project is an attempt to stimulate the basic concepts of airline reservation system. The system enables the customer to do the things such as search for airline flights for two travel cities on a specified date, choose a flight based on the details, reservation of flight and cancellation of reservation. The system allows the airline passenger to search for flights that are available between the two travel cities, namely the “Departure city” and “Arrival city” for a particular departure and arrival dates. The system displays all the flight’s details such as flight no, name, price and duration of journey etc.

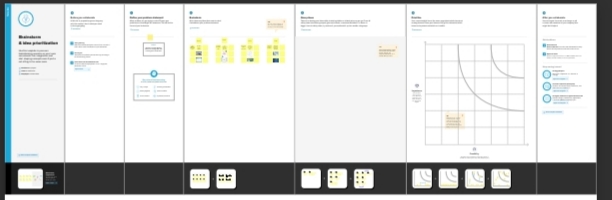
### The web based “airline reservation system” project is an attempt to stimulate the basic concepts of airline reservation system. The system enables the customer to do thethings such as search for airline flights for two travel cities on a specified date, choose a flight based on the details, reservation of flight and cancellation of reservation. The system allows the airline passenger to search for flights that are available between the two travel cities, namely the “Departure city” and “Arrival city” for a particular departure and arrival dates. The system displays all the flight’s details such as flight no, name, price and duration of journey etc.

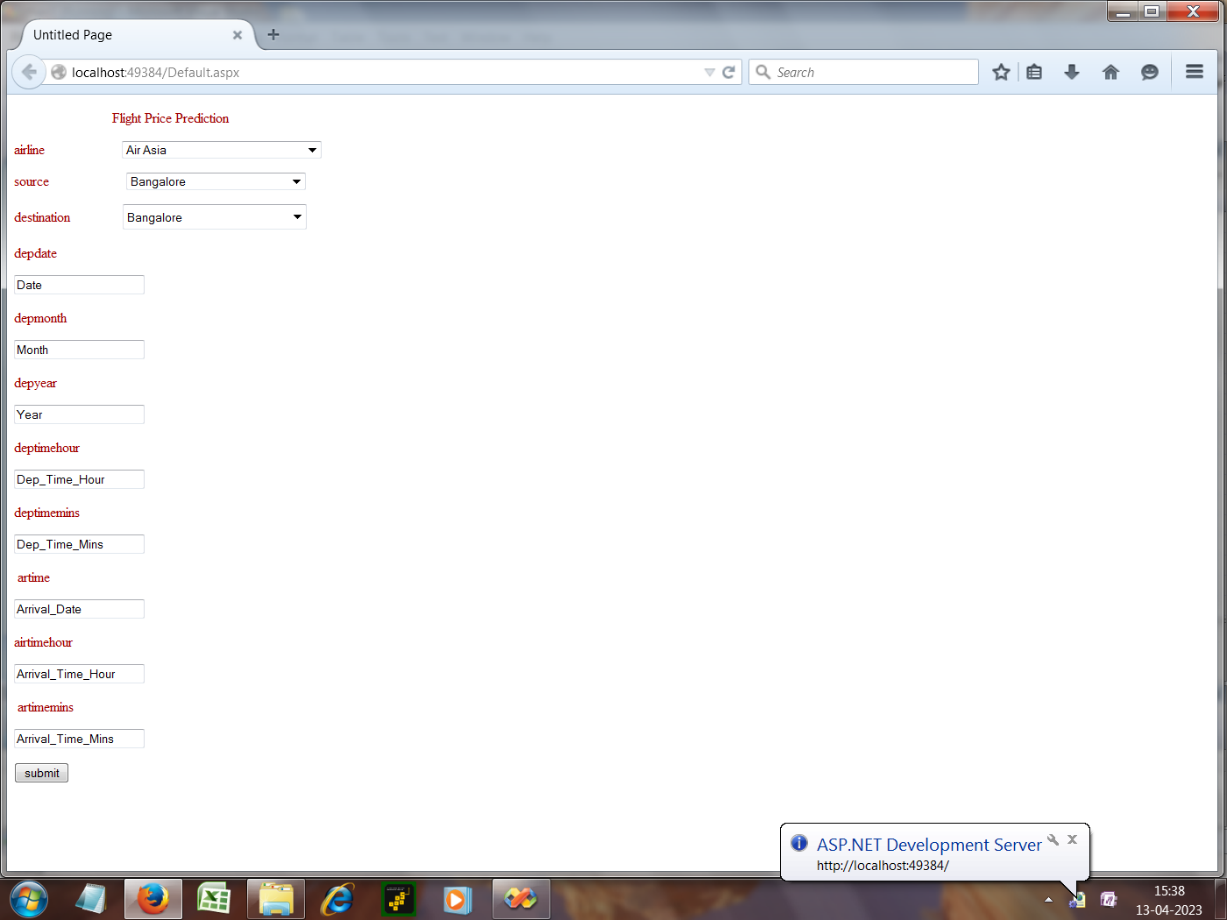
# PROBLEM definition & desing THINKING

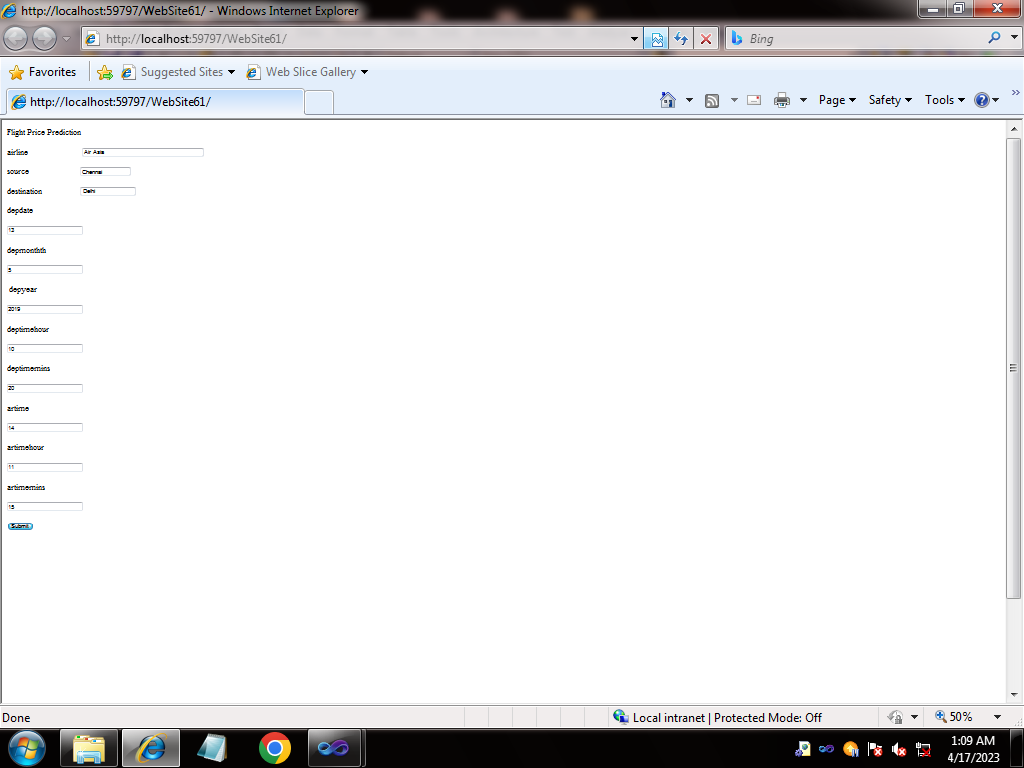
# 2.1 EMPATHY MAP

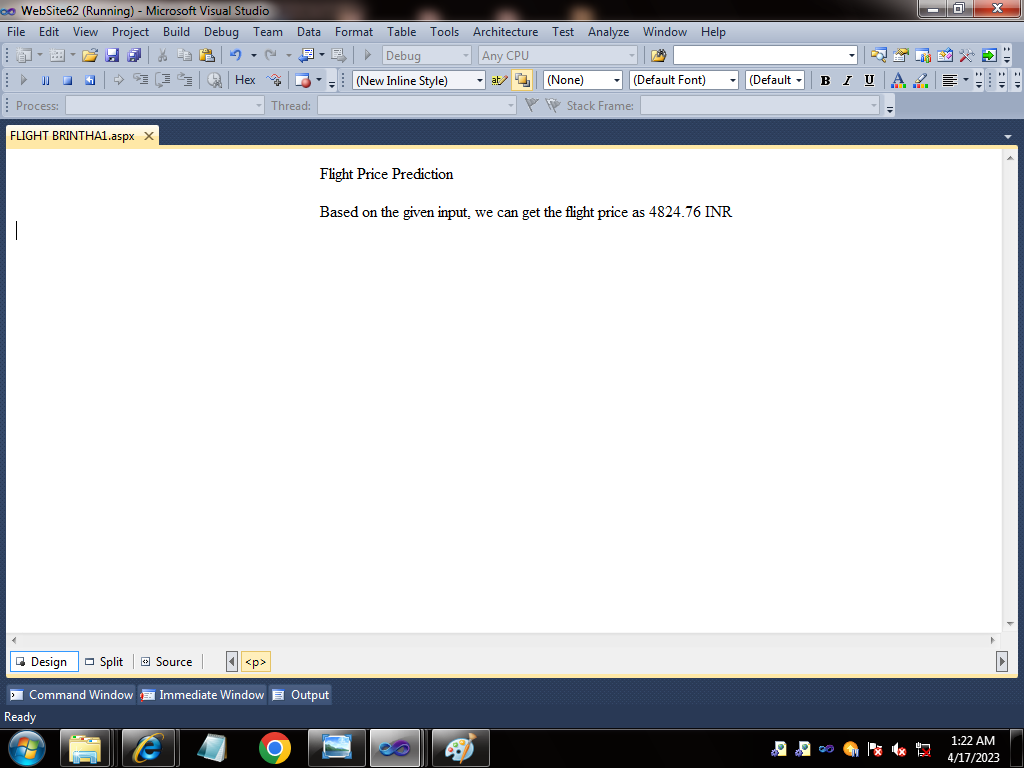


**2.2. Brainstorming :**

**2.3VISUAL SUDIO:-**







**3.RESULT:**

* **READ THE DATASET:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **Handbook** | **Unnamed: 1** | **SmartBridge Educational Services Pvt. Ltd.** |
| **NaN** | | | **NaN** | **NaN** | **Optimizing** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **Flight** | **NaN** | **NaN** | **NaN** | **Booking** | **NaN** | **NaN** | **NaN** | **NaN** | **Decisions** | **NaN** | **NaN** | **NaN** | NaN | NaN | NaN |
|  |  |  | **NaN** | **through** | **NaN** | **NaN** | **Machine** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **Learning** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **Price** | NaN | NaN | NaN |
|  |  |  |  | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **Predictions** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | NaN | NaN | NaN |
|  |  | **People who work frequently travel through flight will have better knowledge on best discount and right time** | | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | NaN | NaN | NaN |
|  |  | **to buy the ticket. For the business purpose many airline companies change prices according to the** | | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | **NaN** | NaN | NaN | NaN |

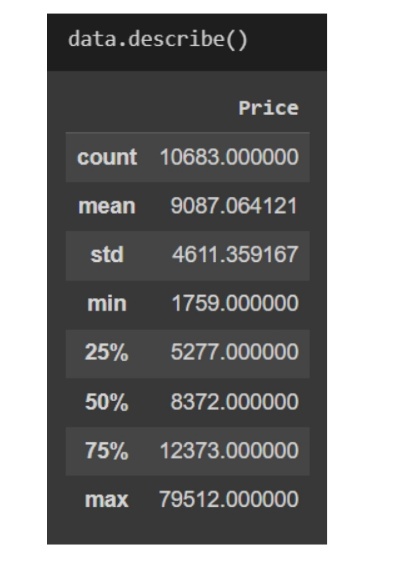
* **DATA PREPAATION;**

Handbook [nan]  
Unnamed: 1 [nan]  
SmartBridge Educational Services Pvt. Ltd. [nan]

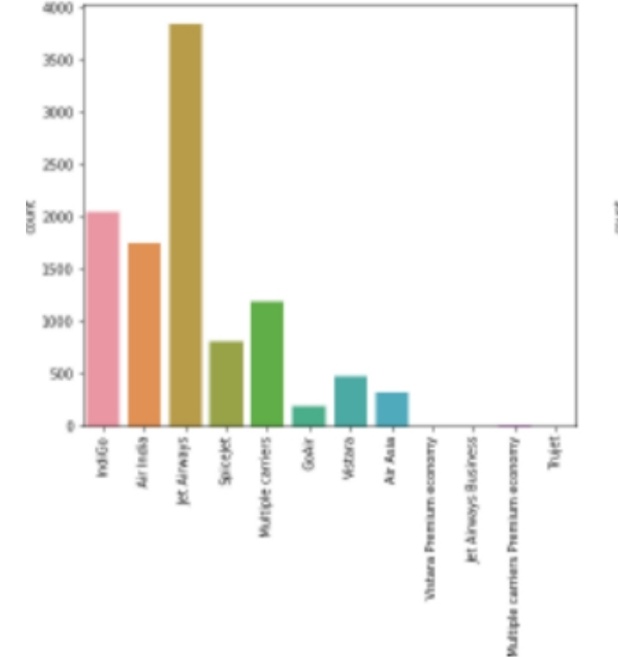
* DATA:

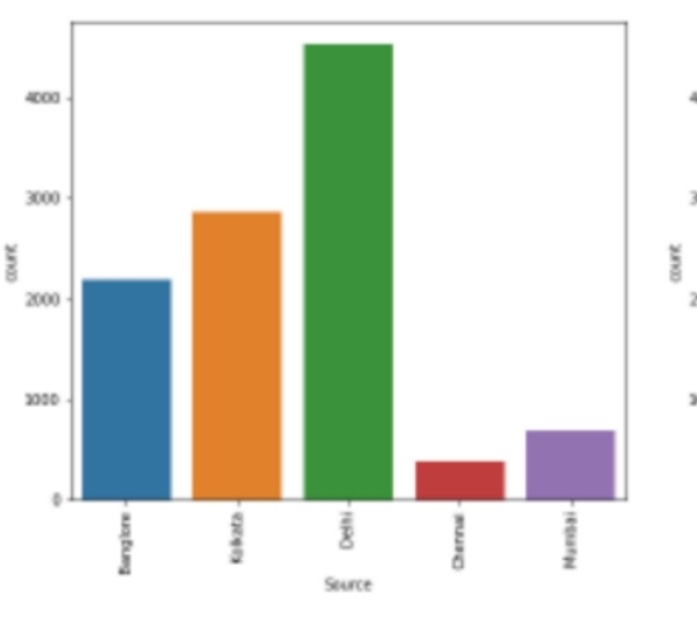
<class 'pandas.core.frame.DataFrame'> MultiIndex: 250 entries, (nan, nan, nan, 'Optimizing', nan, nan, nan, nan, nan, 'Flight', nan, nan, nan, 'Booking', nan, nan, nan, nan, 'Decisions', nan, nan, nan) to ('Create document as per the template provided', nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan) Data columns (total 3 columns): # Column Non-Null Count Dtype --- ------ -------------- ----- 0 Handbook 0 non-null float64 1 Unnamed: 1 0 non-null float64 2 SmartBridge Educational Services Pvt. Ltd. 0 non-null float64 dtypes: float64(3) memory usage: 21.8+ KB

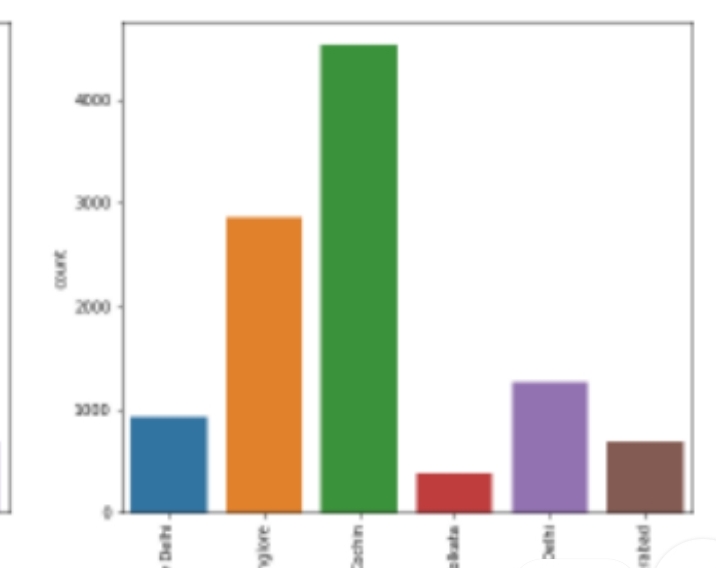
EXPLORATORY DATA ANALYSIS:

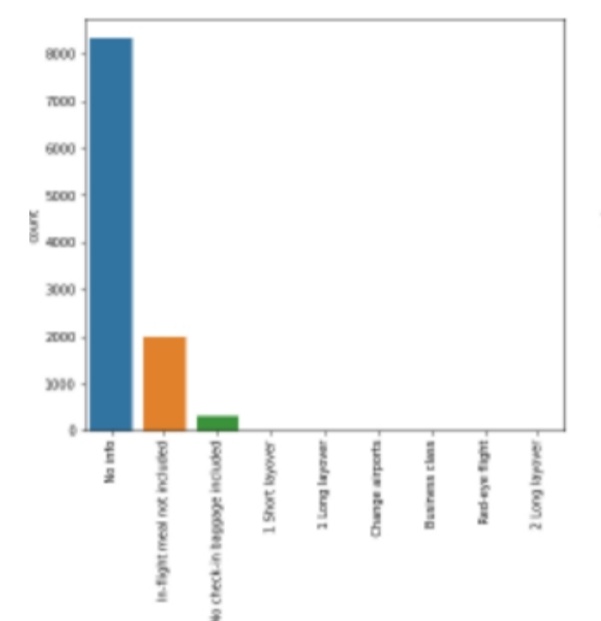


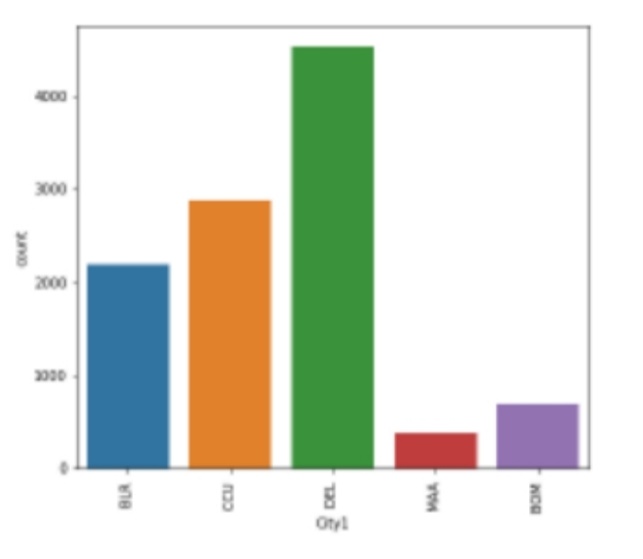
VISUAL ANALYSIS:

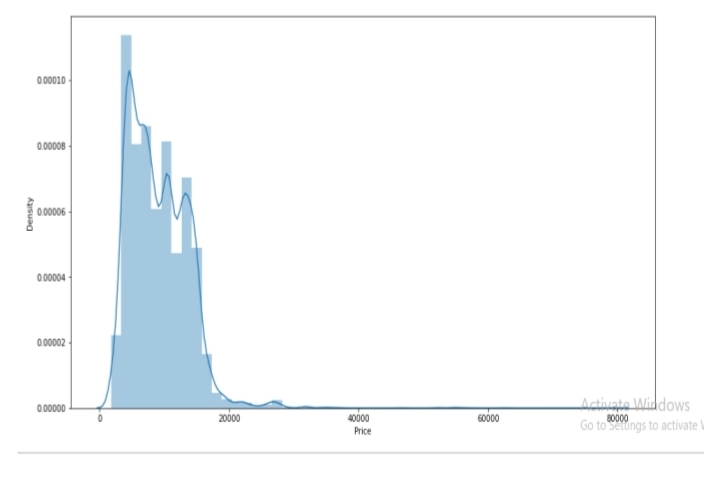




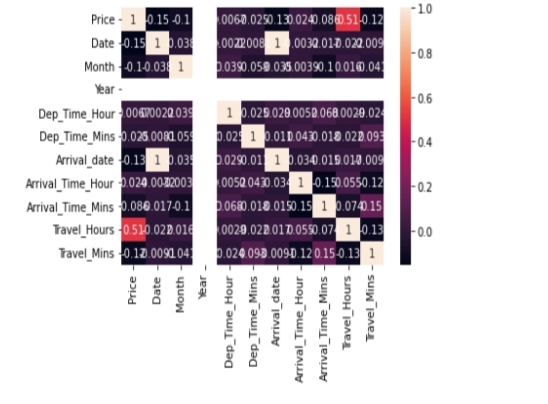




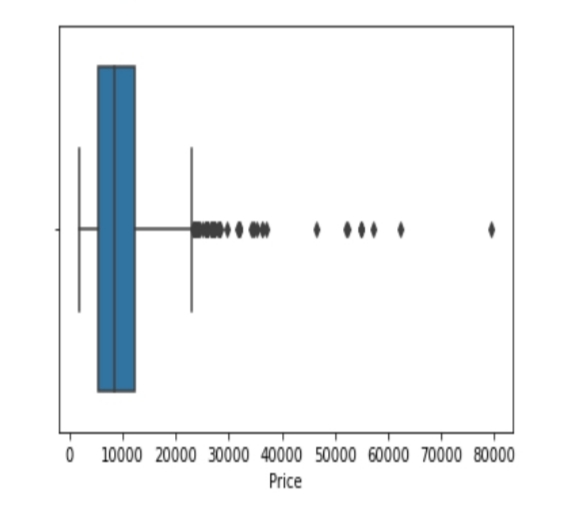




HEAT MAP:



OUTLIER:



2.ADVANTAGES & DISADVANTAGES

# ADVANTAGES :

# Easy to book tickets

# Saves time and money

# Provides evEN information about flight

# 24/7 customer support through chat and calls

# Mobile AVAILABILITY:

# Send automated tickets to the customer by mail

# Easy Refund Policies

# Available for both Domestic and International Airlines

# Disadvantages :

#### YOU NEED INTERNET ACCESS:

# If you run tours and activities in remote areas where you aren’t able to get on the Internet, online booking might not be for you. You’ll NEED RELIABLE INTERNET ACCESS TO CHECK YOUR RESERVATIONS AND TO ADD BOOKINGS THAT ARE MADE ONLINE. THE GOOD NEWS FOR PEEK PRO USERS IS THAT OUR MOBILE APPS WORK OFFLINE. THIS MEANS THAT YOU CAN CONNECT TO THE INTERNET, sync your bookings, and go to areas with bad reception. In those areas, it doesn’t even matter if your internet cuts out – the Peek Pro apps will work.

#### YOU NEED TO BE READY FOR AN INFLUX OF NEW CUSTOMERS:

# Online booking software is a great way to attract new customers, many of whom prefer booking online from their computers and mobile devices. However, if you’re running a small operation without enough staff members or resources to expand your activities, growing too quickly may pose a challenge.

# 3.Not all online booking systems are created equal.

# If you choose a provider that offers poor customer service or only a few features, an online solution might be frustrating — especially when you’re ready to grow your business. It’s important to do your homework upfront and choose an online booking system that is committed to supporting you and your business for the long haul.

# 4.Avoid booking systems that don’t bring you new QUALITY CUSTOMERS.

# If you’re going to invest in a new technology platform and take the time to train your staff, we recommend you choose a booking system that fits allof your business needs — including bringing you a stream of new, high-quality customers. Some booking systems include distribution channels, but they may not attract quality customers who will give you great reviews and refer your business to friends. Make sure you do your research to find a booking system that guarantees a consistent stream of quality CUSTOMERS.

# 5.APPLICATION:

# Skyscanner

# Hopper

# Kiwi

# Momondo

# Kayak

# Google flights

# Priceline

# Skiplagged

# App in the air

# Expedia

# Travala

# Tripadvisor flights

# 6.CONCLUSION

# Recommendations and Conclusion Online ticket booking system is an application where the customer can book a ticket online and 24\*7 hours a day from anyplace in the world. Customers can also interact with theticket booking website to know any other details they want. Online ticket booking system has been developed successfully. System performance is also found to be satisfactory. This is a user-friendly application. Through this application, the cost can be reduced and efficiency is increased. There are several procedures that can be selected by customers. With the help of this application customers can book tickets, can know the status of a flight, bus or trains, a Source station and destination can be chosen according to their choice, can select seats, can choose The flight.

# 7.FUTURE SCOPe:

# 1. The research report includes specific segments by region (country), company, Type, and Application. This study provides information about the sales and revenue during the historical and forecasted period. Understanding the segments helps identify the factors that aid the market growth. The Online Airline Reservation System research report provides information about the market area, which is further subdivided into sub-regions and countries/regions. In addition to the market share in each country and sub-region, this report chapter also contains information on profit opportunities. 2. The use of online airline reservation systems has seen significant growth in recent years as more consumers prefer to book their flights online. According to a study by the International Air Transport Association, over 80% of air tickets were purchased through digital channels in 2020. Additionally, the study found that mobile bookings accounted for over 50% of all digital ticket sales. The trend towards digital booking is expected to continue, as travelers seek convenient and efficient ways to plan their trips.

# 3. In response, airlines and travel companies have invested in the development of online airline reservation systems that provide a seamless booking experience for their customers. With the growing demand for online booking, the online airline reservation system market is poised for continued growth in the coming years.

# 8.APPENDIX:

# 8.1.SOURCE CODE:

# OPTIMIZING FLIGHT BOOKING DECISION CODING:

# IMPORTING THE LIBRARIES

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, GradientBoost?

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')

READ THE DATASET:

data=pd.read\_csv('/Optimizing\_Flight\_Booking\_Decisions\_through\_Machine\_Learning\_Price\_Predictions.csv')

data.head()

DATA PREPARATION:

for i in data:

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

DATA JOURNEY:

data.Date\_of\_Jounrney=date.Date\_of\_Journey.str.split(‘/’)

data.Date\_of\_Journey

TREATING THE DATA COLUMNS:

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()

ROUTE:

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]

REPLACING MISSING VALUES:

data[‘City3’].fillna(‘None’,inplace=true)

ARRIVAL:

data[‘Arrival\_date’].fillna(data[‘Date’],inplace=true)

data[‘Travel\_Mins’].fillna(0,inplace=true)

INFO:

data.info()

LABEL ENCODING:

From sklearn.preprocessing import LabelEncoder

Le=LabelEncoder()

data.Airline=le.fit\_transform(data.Airline0

data.Source=le.fit\_transform(data.Source)

data.Destination=le.fit\_transform(data.Destinatiom)

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\_Info0

data.head()

OUTPUT COLUMNS:

data.head()

data=data[[‘Airline’,’Source’,’Destination’,’Date’,’Month’,’Year’,’Dep\_Time\_Hour’,’Dep\_Time\_Mins’,’Arrival\_date’,’Arrival\_Time

data.head()

EXPLORATORY DATA ANALYSIS:

DESCRIPTIVE STATISTICAL;

data.describe()

PLOTTING COUNTPLOTS:

import seaborn as sns

C=1

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

for I in categorical:

plt.subplot(6,3,c)

sns.countplot9data[I])

plx.xticks(rotation=90)

plt.tight\_layout(pad=3.0)

C=c+1

plt.show()

HEAT MAP:

sns.heatmap(data.corr(),annot=True)

OUTLIER:

import seabron as sns

sns.boxplot(fata[‘price’])

SCALING THE DATA:

y=data[‘price’]

x=data.drop(columns=['Price’],axis=1)

from sklearn.proprocessing import StandardScaler

ss=StandardScaler()

x\_scaled=ss.fit\_transform(x)

x\_scaled=pd.DataFrame(x\_scaled,columns=x.columns)

x\_scaled.head()

SPLITTING THE DATA:

from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test=test\_test\_split(x,y,test\_size=0.2,random\_state=42)

x\_train.head()

MODEL BIULDING:

from sklearn.ensemble import RandomForestRegressor,GradientBoostingRegressor,AdaBoostRegressor

rfr=RandomForestRegressor

gb=GradientBoostingRegressor

ad=AdaBoostRegressor

from sklearn.metrics import r2\_score,mean\_absolute\_\_error,mean\_squatred\_error

for i in [rfr,gd,ad]:

i.fit(x\_train,y\_train)

y\_pred=I.predict(x\_test)

test\_score=r2\_score(y\_test,y\_pred)

train\_score=r2\_score)9y\_train,I.predict(x\_train))

if abs(train\_score-test\_score)<=0.2:

print(I)

print(“R2score 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”,(ean\_squared\_error(y\_pred,y\_test,squared=False)))

REGRESSION MODEL:

from sklearn.neighbors import KNeighborsRegressor

from sklearn.svm import SVR

from Sklearn.tree import DecisionTreeRegressor

from sklearn.metricsimport 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\_train0

y\_pred=I.predict(x\_test)

test\_score=r2\_score(y\_train,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)))

CHECKING CROSS VALIDATION:

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())

HYPERTUNING:

from sklearn.model:\_selection import randomizedSearchCV

param\_grid={‘n\_estimators’:[10,30,50,70,100],’max-depth’:[None,1,2,3],’max\_features’:['auto’,’sprt’]}

rfr=RandomForestRegressor()

rf-res=RandomoizedSearchCV(estimator=rfr,param\_distributions=param\_grid,cv=3,verbose=2,n\_jobs=-1)

rf\_res.fit(x\_train,y\_train)

gd=gradientsBoostRegressor()

gb\_res=RandomizedSearchCv(estimator=gb,param\_distributions=param\_grid,cv=3,verbose=2,n\_jobs=-1)

gb\_res.fit(x\_train,y\_train)

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))

EVALUATION MODEL;

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

MODEL DEPLOYMENT:

import pickle

pickle.dump(rfr,open(‘model1.pk1’,’eb’))

# BUILD PYTHON CODE:

# from flask import Flask, render-template, request

# import numpy as np

# import pickle

# RENDER HTML PAGE:

# @app.route(“/home”)

# def home()

# return render\_template(‘home.html’)

# UI:

# @app.route(“/predict”)

# def home()

# return render\_template(‘predict.html’)

# @app.route(“/pred”,methods=['POST’,’GET’])

# def predict();

# x=[[int(x0 for x in request.form.values()]]

# print(x)

# x=np.array(x)

# print(x.shape)

# print(x)

# pred=model.predict(x)

# print(pred)

# return render\_template(‘submit.html’,prediction\_text=pred)

# AN LANGUAGE MODEL, I CANNOT provide a specific source code without additional information or clarification. However, as a general approach, Optimising Flight Booking Decisiion Learning with Price Predictions the given above STEPS.

# 