INTRODUCTION

# Overview

# Flight booking is a vital part of the modern travel experience, enabling individuals and businesses to reserve seats on planes for their desired journeys. It is a process that has evolved significantly over the years, with technological advancements making it more convenient and accessible. With the rise of online booking platforms, travelers can now search for and book flights from anywhere in the world, expanding their reach and options. This has transformed the tourism and hospitality industry, providing people with the freedom to plan their trips more efficiently and effectively. In this article, we will provide an overview of the flight booking process, including the different types of flights available, the various factors to consider when booking a flight, and the impact of technology on the flight booking process. We will also discuss the challenges and opportunities associated with flight booking and how it has contributed to the growth of the travel industry.

# Purpose

# The primary purpose of flight booking is to reserve a seat on a plane for travel to a particular destination. It allows individuals and businesses to plan their trips efficiently and effectively, saving time and effort. Flight booking provides travelers with the flexibility to choose from a wide range of airlines, schedules, and pricing options, making it easier to find the best flight that suits their needs and budget.

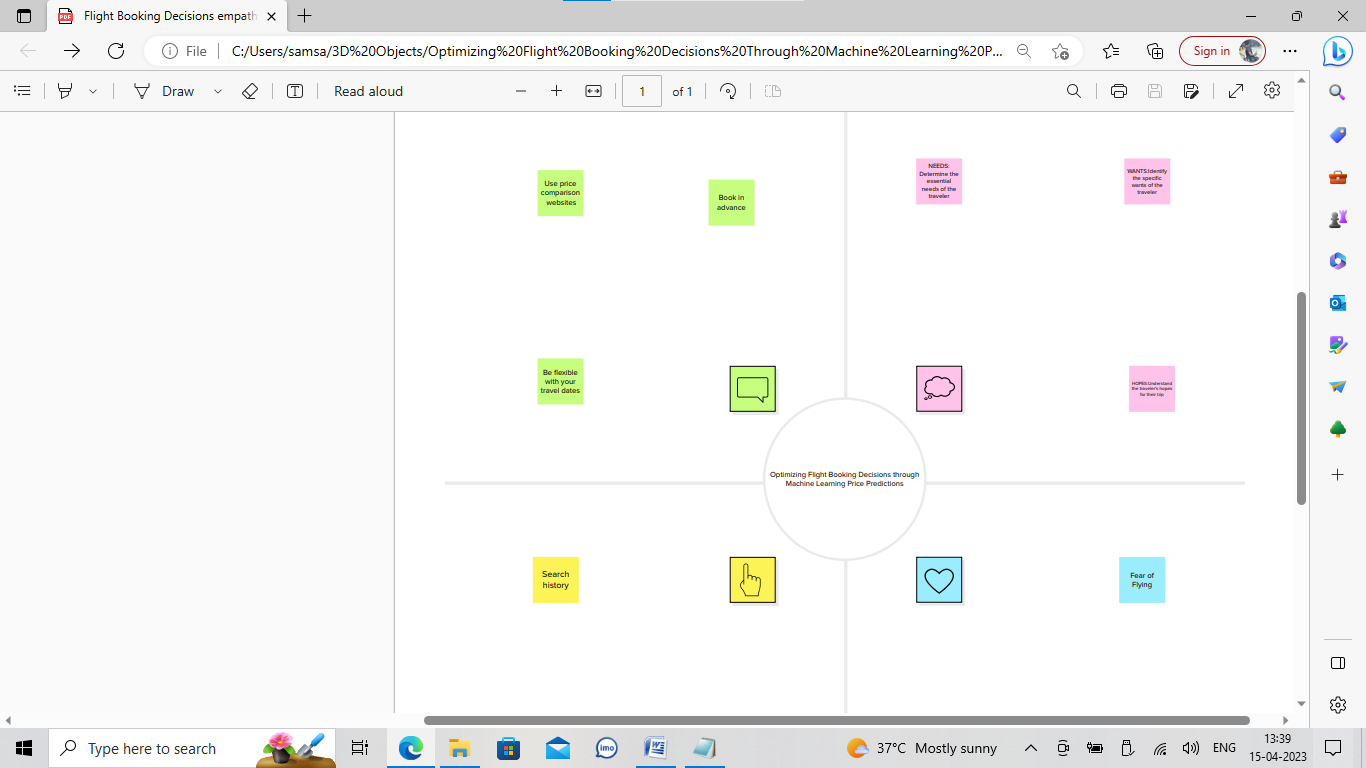
# Additionally, flight booking plays a crucial role in the tourism and hospitality industry, enabling airlines, airports, and other travel-related businesses to operate effectively. It generates revenue for these businesses and creates job opportunities, contributing to the growth of the travel industry.

# Moreover, flight booking has become more accessible and convenient with the advancement of technology, making it possible for people to book flights from anywhere in the world at any time. This has made travel more accessible to a wider range of people, providing more opportunities for exploration, education, and business.

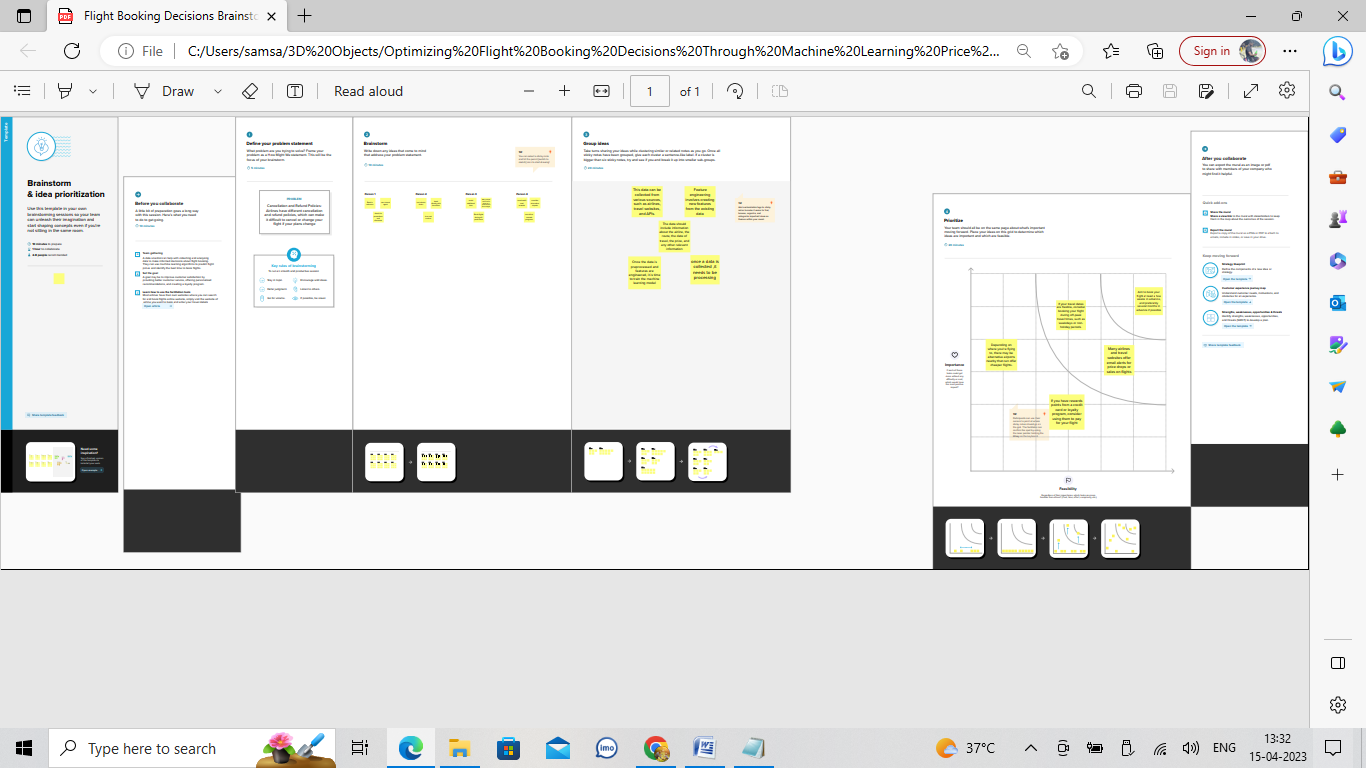
# In summary, the purpose of flight booking is to provide individuals and businesses with a convenient and efficient means of reserving seats on a plane for travel, while also contributing to the growth of the travel industry.

Problem Definition & Design Thinking

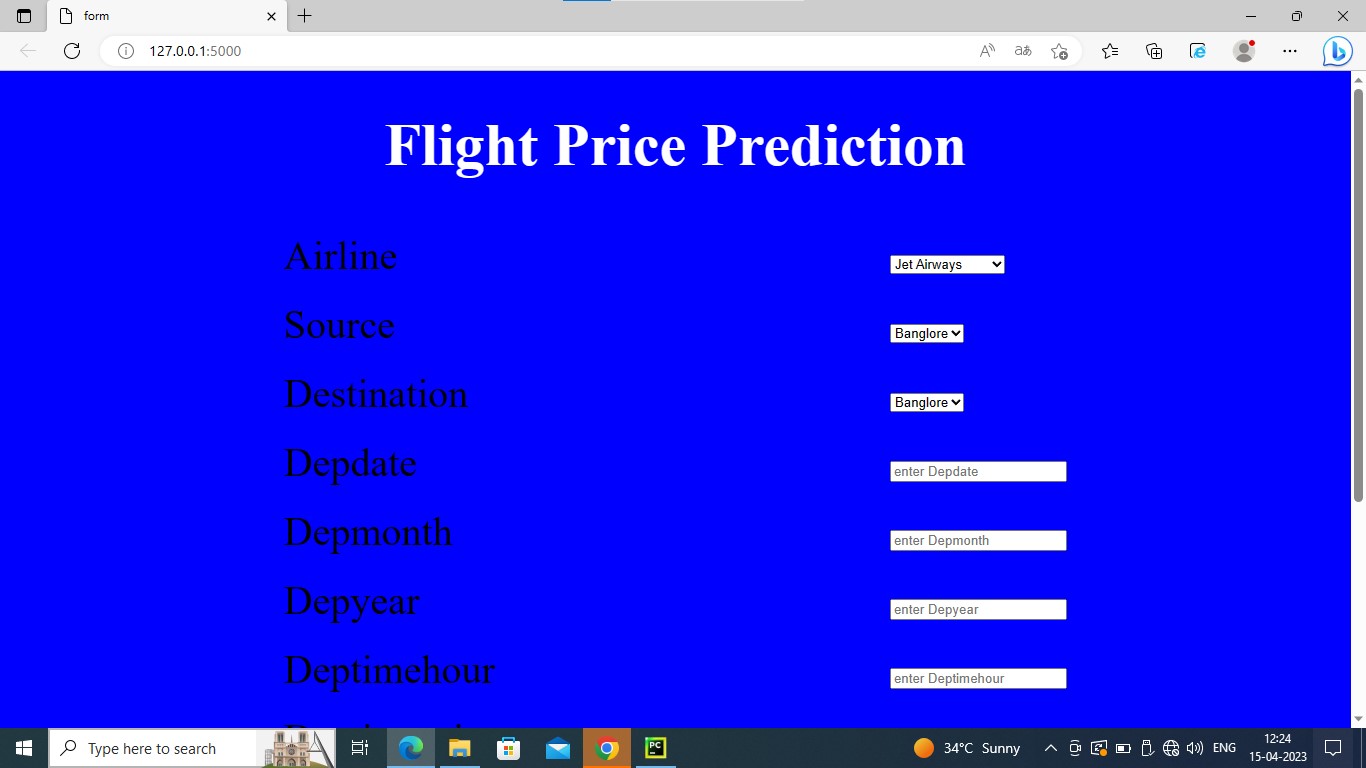
# Empathy Map:



# Ideation & Brainstorming Map:



RESULT



ADVANTAGES & DISADVANTAGES

# the specific proposed solution for flight booking that you are referring to. However, in general, here are some advantages and disadvantages of flight booking solutions:

# Advantages:

# Convenience: Online flight booking solutions provide a convenient way for customers to book flights from anywhere at any time.

# Wide variety: Online flight booking solutions offer a wide range of flight options, making it easier for customers to find the flight that suits their need.

# Comparison shopping: Customers can easily compare prices and flight schedules among different airlines, allowing them to choose the best deal for them.

# Easy to access information: Customers can access flight information, such as flight status, departure and arrival times, and gate information easily and quickly

# 

# Disadvantages

# Technical difficulties: Online flight booking solutions may experience technical difficulties, leading to issues such as slow loading times, errors, or system crashes.

# Security concerns: Online flight booking solutions may pose security risks, such as data breaches, which could result in the theft of personal information.

# Hidden fees: Some online flight booking solutions may not include all fees upfront, leading to unexpected charges at checkout.

# Lack of personalization: Online flight booking solutions may not provide the same level of personalization as booking through a travel agent.

# Limited customer support: Online flight booking solutions may have limited customer support, making it difficult for customers to resolve any issues that may arise during or after the booking process.

APPLICATIONS

# Online flight booking solutions can be applied in various areas, including:

# Travel agencies: Travel agencies can use online flight booking solutions to offer their customers a convenient and cost-effective way to book flights.

# Airlines: Airlines can use online flight booking solutions to allow customers to book flights directly on their website.

# Aggregator websites: Aggregator websites can use online flight booking solutions to compare prices and schedules across multiple airlines and offer customers the best deals.

# Mobile applications: Mobile applications can use online flight booking solutions to allow customers to book flights directly from their smart phone or tablet.

# Corporate travel management: Companies can use online flight booking solutions to manage their employees' travel arrangements, including booking flights, hotels, and rental cars.

# Tour operators: Tour operators can use online flight booking solutions to offer customers flight packages as part of their tour packages.

CONCLUSION

# In conclusion, flight booking solutions have become an essential tool in the travel and tourism industry, providing customers with a convenient and cost-effective way to book flights from anywhere at any time. Online flight booking solutions offer a wide range of flight options, allowing customers to compare prices and schedules across multiple airlines to find the best deals.

# However, there are also some disadvantages to online flight booking solutions, such as technical difficulties, security concerns, hidden fees, and limited customer support. Despite these challenges, the benefits of online flight booking solutions outweigh the disadvantages, making it an essential tool for travel agencies, airlines, aggregator websites, mobile applications, corporate travel management, and tour operators.

# In summary, flight booking solutions have revolutionized the way people book flights, making it easier and more accessible for customers to travel to their desired destinations. With the increasing demand for travel and tourism, flight booking solutions will continue to play a vital role in the industry, providing a seamless and efficient way for customers to book flights and explore the world.

FUTURE SCOPE

# the future of flight booking to improve the customer experience and drive innovation in the travel and tourism industry. Here are some possibilities:

# 1. Personalization: Flight booking solutions can incorporate AI and machine learning algorithms to personalize the booking experience for each customer. This can include recommendations based on their past travel history, preferences, and travel behavior.

# 2. Virtual Reality: Virtual reality technology can be used to provide customers with a more immersive experience when browsing and booking flights, allowing them to virtually explore their destination and aircraft before booking.

# 3. Mobile Integration: With the increasing use of smartphones and tablets, flight booking solutions can integrate with mobile apps to offer a seamless booking experience for customers on the go.

# 4. Blockchain Technology: Blockchain technology can be used to provide a secure and transparent booking system, ensuring that customer data is protected and booking processes are secure.

# 5. Sustainability: In response to the growing concern about climate change, flight booking solutions can incorporate sustainability measures, such as offering carbon offset options, encouraging eco-friendly travel, and promoting green travel options.

# 6. Smart Ticketing: Smart ticketing can be used to provide customers with a more seamless and efficient way to check-in, board, and access their flight information, using technologies such as biometrics and mobile boarding passes.

# Overall, the future of flight booking solutions is full of possibilities and opportunities for innovation. By incorporating emerging technologies and customer-centric features, flight booking solutions can continue to provide a more efficient, personalized, and enjoyable experience for travelers.

APPENDIX

Main code:

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/Data\_Train.csv')

data.head()

data.isnull().sum()

data.dropna(inplace=True)

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\_Time\_Hour']=data.Arrival\_Time.str[0]

data['Arrival\_Time\_Mins']=data.Arrival\_Time.str[1]

data.Arrival\_Time\_Mins=data.Arrival\_Time\_Mins.str.split(' ')

data['Arrival\_Time\_Mins']=data.Arrival\_Time\_Mins.str[0]

data['Arrival\_Day']=data.Arrival\_Time\_Mins.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=data.Total\_Stops.str.split(' ')

data.Total\_Stops=data.Total\_Stops.str[0]

data.Total\_Stops.replace('non-stop',0,inplace=True)

data.Total\_Stops

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.isnull().sum()

data.drop(['Date\_of\_Journey','Route','Dep\_Time','Arrival\_Time','Duration'],axis=1,inplace=True)

data.isnull().sum()

data['Arrival\_Day'].fillna(data['Date'],inplace=True)

data['city3'].fillna('None',inplace=True)

data['Travel\_Mins'].fillna(0,inplace=True)

data.info()

data.head(3)

data.Total\_Stops=data.Total\_Stops.astype('int64')

data.Date=data.Date.astype('int64')

data.Month=data.Month.astype('int64')

data.Year=data.Year.astype('int64')

data.Dep\_Time\_Hour=data.Dep\_Time\_Hour.astype('int64')

data.Dep\_Time\_Mins=data.Dep\_Time\_Mins.astype('int64')

data.Arrival\_Time\_Hour=data.Arrival\_Time\_Hour.astype('int64')

data.Travel\_Mins=data.Travel\_Mins.astype('int64')

data.info()

data.Arrival\_Time\_Mins=data.Arrival\_Time\_Mins.astype('int64')

data.Arrival\_Day=data.Arrival\_Day.astype('int64')

data[data['Travel\_Hours']=='10m']

data.drop(index=6474,inplace=True,axis=0)

data.Travel\_Hours=data.Travel\_Hours.astype('int64')

column=[column for column in data.columns if data[column].dtype=='object']

column

continuous\_col =[column for column in data.columns if data[column].dtype!='objects']

continuous\_col

categorical = data[column]

categorical

numerical=data[continuous\_col]

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.Additional\_Info=le.fit\_transform(data.Additional\_Info)

data.city1=le.fit\_transform(data.city1)

data.city2=le.fit\_transform(data.city2)

data.city3=le.fit\_transform(data.city3)

data.head()

categorical

fdata=data.copy()

fdata

fdata.drop(['Additional\_Info','city1','city2','city3'],axis=1,inplace=True)

fdata

data.describe()

import seaborn as sns

c=1

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

for i in categorical:

plt.subplot(6,3,c)

sns.countplot(x = data[i])

plt.xticks(rotation=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)

sns.boxplot(data['Price'])

x=data.drop('Price',axis=1)

y=data['Price']

from sklearn.preprocessing import StandardScaler

ss=StandardScaler()

xscaled=ss.fit\_transform(x)

xscaled=pd.DataFrame(xscaled,columns=x.columns)

xscaled.head()

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

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size=0.20,random\_state=123)

x\_train.head()

from sklearn.ensemble import RandomForestClassifier,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\_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)<=0.2:

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 Sqaured Error is", (mean\_squared\_error(y\_pred,y\_test,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()

rf\_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))

knn=KNeighborsRegressor(n\_neighbors=2,algorithm='auto',metric\_params=None,n\_jobs=-1)

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

y\_train\_pred=knn.predict(x\_train)

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

print("train accuracy",r2\_score(y\_train\_pred,y\_train))

print("test accuracy",r2\_score(y\_test\_pred,y\_test))

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))

prices=rfr.predict(x\_test)

price\_list=pd.DataFrame({'Price':prices})

price\_list

import pickle

pickle.dump(rfr,open('model1.pkl','wb'))

**App.py :**

import flask

from flask import Flask, render\_template, request

import pickle

import numpy as np

import sklearn

from flask\_ngrok import run\_with\_ngrok

import warnings

warnings.filterwarnings('ignore')

app = Flask(\_\_name\_\_)

run\_with\_ngrok(app)

model = pickle.load(open('rdf.pkl', 'rb'))

@app.route('/', methods=['GET'])

def home():

return render\_template('index.html')

@app.route('/', methods=['GET', "POST"])

def predict():

input\_values = [float(x) for x in request.form.values()]

inp\_features = [input\_values]

print(inp\_features )

prediction = model.predict(inp\_features)

if prediction == 1:

return render\_template('index.html', prediction\_text=' price predicted')

else:

return render\_template('index.html', prediction\_text='predict')

app.run()

**Idex.html :**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title> form</title>**

**<style type="text/css">**

**body{**

**background-color:blue;**

**font-size: 30px;**

**font-style: Times New Roman;**

**text-align: justify;**

**color: white;**

**}**

**table{**

**font-size: 40px;**

**font-style: Arial;**

**color: black;**

**}**

**</style>**

**</head>**

**<body>**

**<h1 align="center">Flight Price Prediction</h1>**

**<form>**

**<table align="center" cellpadding="10">**

**<tr>**

**<td>Airline</td>**

**<td>**

**<select name="Airline" id="Airline" required="required">**

**<option value="Jet Airways">Jet Airways</option>**

**<option value="IndiGo">IndiGo</option>**

**<option value="Air India">Air India</option>**

**<option value="Multiple carriers">Multiple carriers</option>**

**</select>**

**</td>**

**</tr>**

**<tr>**

**<td>Source</td>**

**<td>**

**<select name="Source" id="Source" required="required">**

**<option value="Banglore">Banglore</option>**

**<option value="Kolkata">Kolkata</option>**

**<option value="Delhi">Delhi</option>**

**<option value="Cochin">Cochin</option>**

**</select>**

**</td>**

**</tr>**

**<tr>**

**<td>Destination</td>**

**<td>**

**<select name="Destination" id="Destination" required="required">**

**<option value="Banglore">Banglore</option>**

**<option value="Kolkata">Kolkata</option>**

**<option value="Delhi">Delhi</option>**

**<option value="Cochin">Cochin</option>**

**</select>**

**</td>**

**</tr>**

**<tr>**

**<td>Depdate</td>**

**<td><input type="text" placeholder="enter Depdate" required></td>**

**</tr>**

**<tr>**

**<td>Depmonth</td>**

**<td><input type="text" placeholder="enter Depmonth" required></td>**

**</tr>**

**<tr>**

**<td>Depyear</td>**

**<td><input type="text" placeholder="enter Depyear" required></td>**

**</tr>**

**<tr>**

**<td>Deptimehour</td>**

**<td><input type="text" placeholder="enter Deptimehour" required></td>**

**</tr>**

**<tr>**

**<td>Deptimemins</td>**

**<td><input type="text" placeholder="enter Deptimemins" required></td>**

**</tr>**

**<tr>**

**<td>Artime</td>**

**<td><input type="text" placeholder="enter Artime" required></td>**

**</tr>**

**<tr>**

**<td>Artimehour</td>**

**<td><input type="text" placeholder="enter Artimehour" required></td>**

**</tr>**

**<tr>**

**<td>Artimemins</td>**

**<td><input type="text" placeholder="enter Artimemins" required></td>**

**</tr>**

**<td>**

**colspan="2" align="center">**

**<input type="submit" value="submit"/>**

**<input type="reset" value="reset" />**

**</td>**

**</tr>**

**</table>**

**</form>**

**</body>**

**</html>**