SOMAIYA
VIDYAVIHAR UNIVERSITY
K J Somaiya Institute of Management

Department of Data Science & Technology

Certificate

Masters in Computer Applications
Trimester IV (2021 – 23)

This is to certify that Mr. Vaibhav Kumar Roll No. 19 of MCA, has satisfactorily completed the practical course "Project Based Learning" prescribed by the College for the Partial fulfillment of the Degree by the Somaiya Vidyavihar University, during the academic year 2021-23.

Signature of the Faculty-Incharge

Signature of the Programme Coordinator

Date of Examination

Signature of the External Examiner/s

ABSTRACT

A person who regularly buys airline tickets will be able to predict the right time to buy a ticket to get the best deal. Many airlines change fares to manage revenue. Airlines can raise prices when expect demand to increase capacity. To estimate the minimum airfare, the data for a particular air route is collected, including characteristics such as departure times, arrival times, and air routes for a specific time period. Features taken from Collected data to apply machine learning (ML) models.Airfare prices are affected by many factors such as flight distance, time of ticket purchase, fuel price, etc. Each service provider has its own proprietary rules and algorithms to set prices accordingly. Recent advances in Artificial Intelligence (AI) and Machine Learning (ML) make it possible to infer such rules and model possible price variations.

3. Table Of Contents

1.Introduction

Today, airlines use complex strategies and methods to fix airfares dynamically. These strategies take into account a number of financial, marketing, commercial and social factors that are closely related to the final price of an airline ticket.

Because the pricing models applied by airlines are very complex, it is difficult for customers to get the lowest price airfare because the price changes dynamically.

For this reason, a number of techniques, capable of providing the right time for buyers to buy airline tickets by predicting the price of airfares, have been proposed recently. The majority of these methods use sophisticated predictive models from the computational intelligence research field known as machine learning (ML).

Airfare prices can be unpredictable, today we will see a price, tomorrow check the price of that flight is another matter.
To get around this, we got the airfares of different airlines for the period from March to June 2019 and between different cities we aim to build a model that predicts the prices of flights using different input characteristics.

A flight price prediction programme that estimates ticket prices for a specific date based on factors including source, destination, stops, and airline.

Machine learning is one of the hottest research topics in computer science and engineering, applicable to many disciplines. It provides a set of algorithms, methods, and tools capable of demonstrating a kind of machine intelligence.

The power of ML lies in the modeling tools provided, which can be trained, through a learning process, with a set of data that describes a given problem and response to similar data. commonly seen.

Machine learning (ML) is the study of computer algorithms that are improved through the use of experience and data. Machine learning algorithms build models based on sample data (training data) and use those models to make predictions and decisions without being programmed.

Machine learning algorithms have various applications such as fraud detection, email filtering, and more. One such machine learning application is in the “aviation industry,” predicting flight prices. There are various factors/characteristics that affect the flight price, such as distance, flight time, number of stops, etc. These factors help create patterns for pricing flights, and machine learning models are trained on these patterns to make future predictions, automate the process, and speed up the process.

2. Implementation and Design

Python:
It is a high-level, general-purpose programming language.It primarily focuses on code readability with the use of significant indentation unlike curly braces in java or c,c++.
It supports multiple programming paradigms, including structured, OOPs and Exception , file handling and can be widely used for performing Machine Learning Algorithms.

Machine Learning:
Machine Learning is a Domain of Data Science which primarily focuses on making machines capable enough to imitate Human inteligence.It provides a set of algorithms, methods, and tools capable of demonstrating a kind of machine intelligence.
The power of ML lies in the modeling tools provided, which can be trained, through a learning process, with a set of data that describes a given problem and response to similar data. commonly seen.

Anaconda Navigator:
Desktop GUI application which is a cluster of Application such as jupyter Notebook,Datalore,spyder,pycharm professional,and many more tools that help in developing machine learning algorithms or for performing all sorts of analysis.

Jupyter:
It is a Software that provides various tools and libraries to create machine learning algorithm.Usually Python is a preferred language as it provides various modules like numpy,pandas,matplotlib to perform complex algorithms faster and efficiently.

Visual Studio code : Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

We have used two datasets , one is for training which includes the price column and another one is the Test dataset which does not include the price column.
The output column "price" should be predicted on this set. Below is a description of the functions available on the dataset –

1. Airline: The name of the airline.
2. Date_of_Journey: The date of the journey
3. Source: The source from which the service begins.
4. Destination: The destination where the service ends.
5. Route: The route taken by the flight to reach the destination.
6. Dep_Time: The time when the journey starts from the source.
7. Arrival_Time: Time of arrival at the destination.
8. Duration: Total duration of the flight.
9. Total_Stops: Total stops between the source and destination.
10. Additional_Info: Additional information about the flight
11. Price: The price of the ticket

Web Application Framework, or simply Web Framework, represents a collection of libraries and modules that enable web application developers to create applications without worrying about low-level details such as protocols and thread management. increase.

Flask is a web application framework written in Python. It was developed by Armin Ronacher, who led a team of international Python enthusiasts called Poocco. Flask is based on the WSGI toolkit and Jinja2 template engine. Both are Pokko projects.

It then starts a web server which is available only on your computer. In a web browser open localhost on port 5000 (the url) http://127.0.0.1:5000/ .

The first step is to install Flask. Python comes with a package manager named pip
`pip install flask.`

Flowchart :

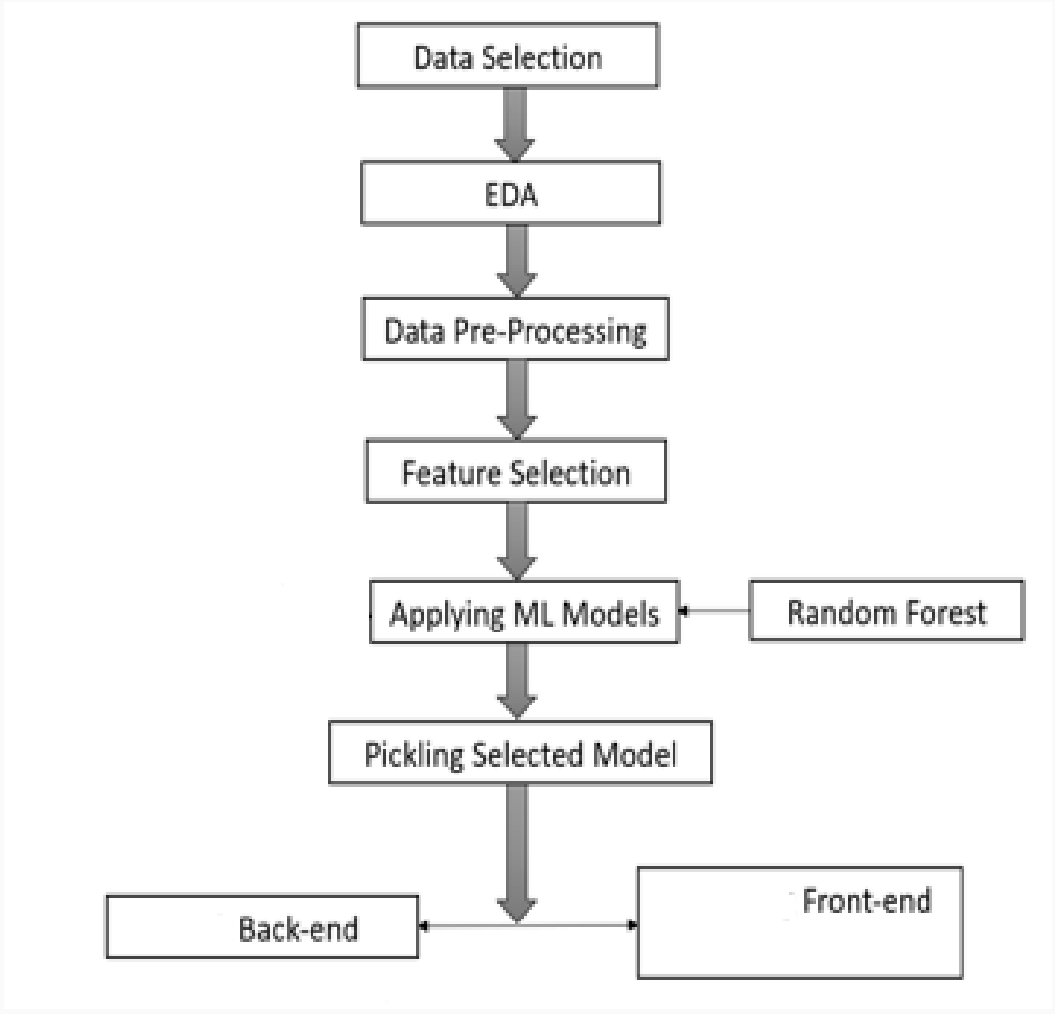


Fig 1 : Flow Chart

CODE SNIPPETS :

app.py | home.html | styles.css

app.py > predict

```python
from flask import Flask, request, render_template
from flask_cors import cross_origin
import sklearn
import pickle
import pandas as pd

app = Flask(__name__)
model = pickle.load(open("flight_rf.pkl", "rb"))



@app.route("/")
@cross_origin()
def home():
    return render_template("home.html")



@app.route("/predict", methods = ["GET", "POST"])
@cross_origin()
```

```python
def predict():
    if request.method == "POST":

        # Date_of_Journey
        date_dep = request.form["Dep_Time"]
        Journey_day = int(pd.to_datetime(date_dep, format="%Y-%m-%dT%H:%M").day)
        Journey_month = int(pd.to_datetime(date_dep, format ="%Y-%m-%dT%H:%M").month)


        # Departure
        Dep_hour = int(pd.to_datetime(date_dep, format ="%Y-%m-%dT%H:%M").hour)
        Dep_min = int(pd.to_datetime(date_dep, format ="%Y-%m-%dT%H:%M").minute)


        # Arrival
        date_arr = request.form["Arrival_Time"]
        Arrival_hour = int(pd.to_datetime(date_arr, format ="%Y-%m-%dT%H:%M").hour)
        Arrival_min = int(pd.to_datetime(date_arr, format ="%Y-%m-%dT%H:%M").minute)
        # print("Arrival : ", Arrival_hour, Arrival_min)

        # Duration
        dur_hour = abs(Arrival_hour - Dep_hour)
        dur_min = abs(Arrival_min - Dep_min)
        # print("Duration : ", dur_hour, dur_min)
```

```
    # Total Stops
    Total_stops = int(request.form["stops"])
    # print(Total_stops)

    # Airline
    # AIR ASIA = 0
    airline=request.form['airline']
    if(airline=='Jet Airways'):
        Jet_Airways = 1
        IndiGo = 0
        Air_India = 0
        Multiple_carriers = 0
        SpiceJet = 0
        Vistara = 0
        GoAir = 0
        Multiple_carriers_Premium_economy = 0
        Jet_Airways_Business = 0
        Vistara_Premium_economy = 0
        Trujet = 0

    elif (airline=='IndiGo'):
        Jet_Airways = 0
        IndiGo = 1
        Air_India = 0
        Multiple_carriers = 0
        SpiceJet = 0
```

```
            Trujet,
            Vistara,
            Vistara_Premium_economy,
            s_Chennai,
            s_Delhi,
            s_Kolkata,
            s_Mumbai,
            d_Cochin,
            d_Delhi,
            d_Hyderabad,
            d_Kolkata,
            d_New_Delhi
        ]])

        output=round(prediction[0],2)

        return render_template('home.html',prediction_text="Estimated Flight price is Rs. {}".format(output))


    return render_template("home.html")
```

Jupyter Notebook :

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

sns.set()
```

In [2]:
```
train_data = pd.read_excel(r"F:\SELF STUDY\PROJECT\Flight-Price-Prediction\Data_Train.xlsx")
```

In [3]:
```
pd.set_option('display.max_columns', None)
```

In [4]:
```
train_data.head()
```

Out[4]:

	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 [9]:
```
train_data["Journey_day"] = pd.to_datetime(train_data.Date_of_Journey, format="%d/%m/%Y").dt.day
```

In [10]:
```
train_data["Journey_month"] = pd.to_datetime(train_data["Date_of_Journey"], format = "%d/%m/%Y").dt.month
```

In [11]:
```
train_data.head()
```

Out[11]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price	Journey_day	Journey_month
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897	24	3
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662	1	5
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882	9	6
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218	12	5
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302	1	3

In [23]:
```
# As Airline is Nominal Categorical data we will perform OneHotEncoding

Airline = train_data[["Airline"]]

Airline = pd.get_dummies(Airline, drop_first= True)

Airline.head()
```

Out[23]:

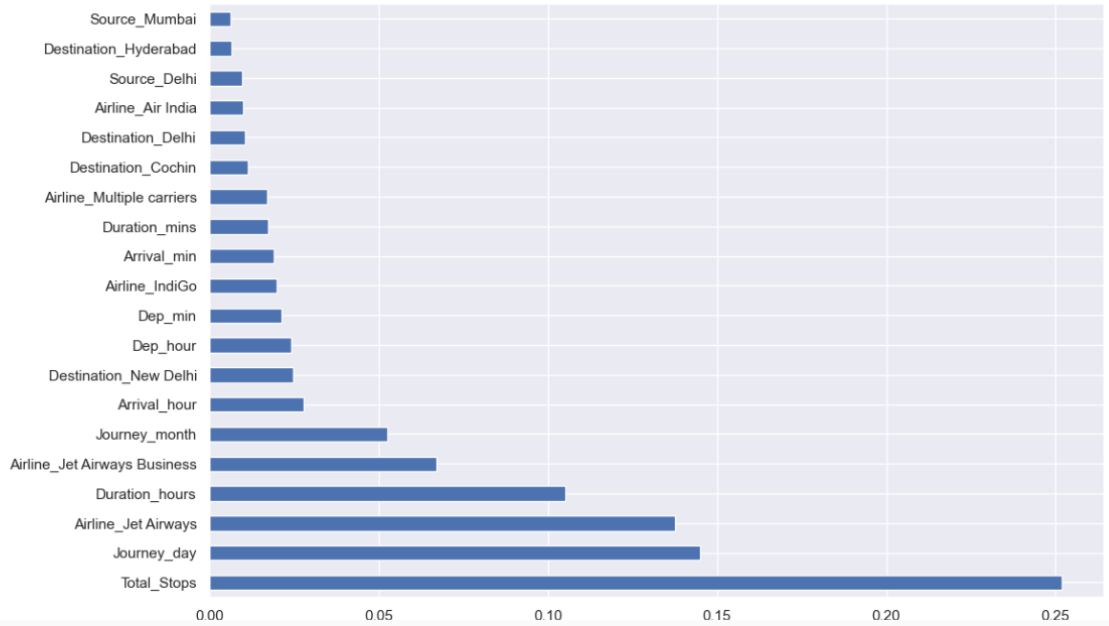
	Airline_Air India	Airline_GoAir	Airline_IndiGo	Airline_Jet Airways	Airline_Jet Airways Business	Airline_Multiple carriers	Airline_Multiple carriers Premium economy	Airline_SpiceJet	Airline_Trujet	Airline_Vistara	Airline_Vist Premiu econor
0	0	0	1	0	0	0	0	0	0	0	
1	1	0	0	0	0	0	0	0	0	0	
2	0	0	0	1	0	0	0	0	0	0	
3	0	0	1	0	0	0	0	0	0	0	
4	0	0	1	0	0	0	0	0	0	0	

In [28]:
```
# As Destination is Nominal Categorical data we will perform OneHotEncoding

Destination = train_data[["Destination"]]

Destination = pd.get_dummies(Destination, drop_first = True)

Destination.head()
```

Out[28]:

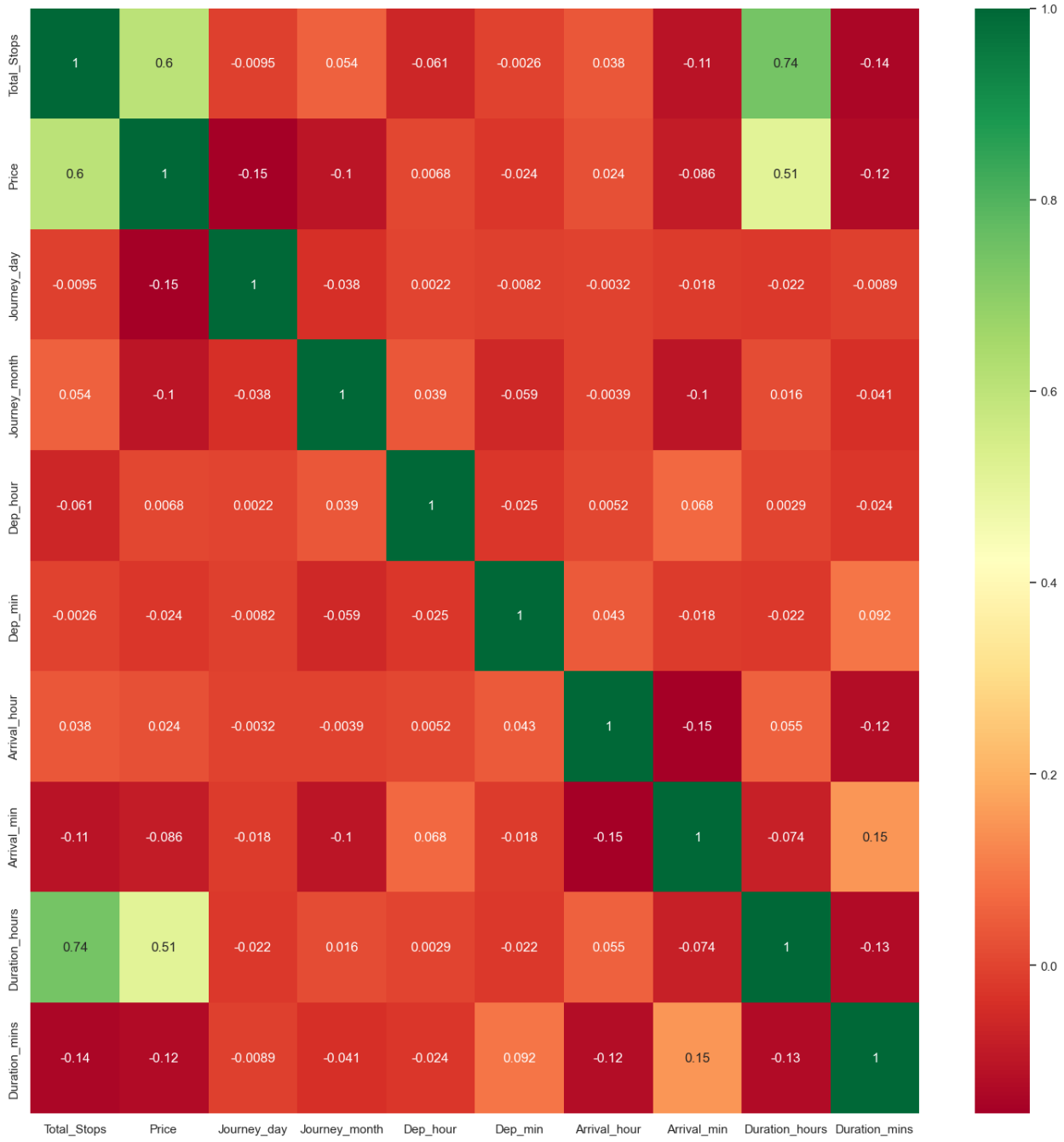
	Destination_Cochin	Destination_Delhi	Destination_Hyderabad	Destination_Kolkata	Destination_New Delhi
0	0	0	0	0	1
1	0	0	0	0	0
2	1	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	1

In [50]:
```
#plot graph of feature importances for better visualization

plt.figure(figsize = (12,8))
feat_importances = pd.Series(selection.feature_importances_, index=X.columns)
feat_importances.nlargest(20).plot(kind='barh')
plt.show()
```



We can see that Total_stops is the feature with the highest feature importance in deciding Price.

Correlation between all the Features

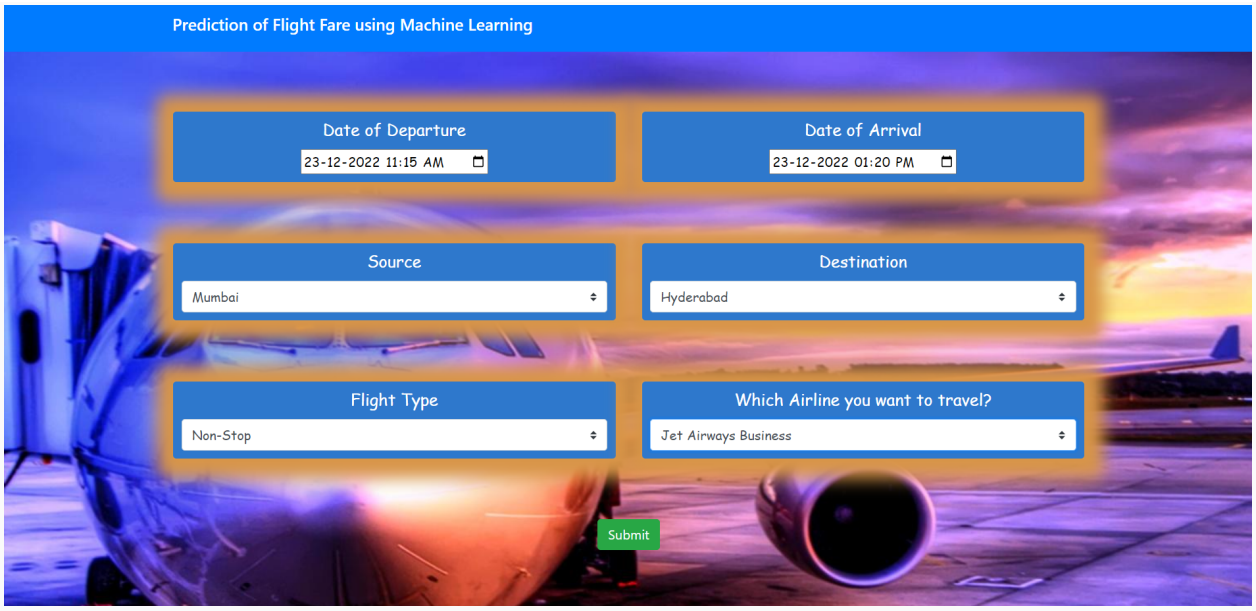


```
In [51]: from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)

In [52]: from sklearn.ensemble import RandomForestRegressor
         reg_rf = RandomForestRegressor()
         reg_rf.fit(X_train, y_train)

Out[52]: RandomForestRegressor()

In [53]: y_pred = reg_rf.predict(X_test)

In [54]: reg_rf.score(X_train, y_train)

Out[54]: 0.9524938973503172

In [55]: reg_rf.score(X_test, y_test)

Out[55]: 0.7960086189295617
```

Evaluating the model accuracy is an essential part of the process of creating machine learning models to describe how well the model is performing in its predictions. The MSE, MAE, and RMSE metrics are mainly used to evaluate the prediction error rates and model performance in regression analysis.

```
In [58]: from sklearn import metrics

In [59]: print('Mean absolute error :', metrics.mean_absolute_error(y_test, y_pred))
         print('Mean Squared Error :', metrics.mean_squared_error(y_test, y_pred))
         print('Root Mean Squared Error :', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

         Mean absolute error : 1176.2989813107133
         Mean Squared Error : 4373414.232344949
         Root Mean Squared Error : 2091.2709610055194

In [60]: metrics.r2_score(y_test, y_pred)

Out[60]: 0.7971708186549799
```

Then with the help of HTML template following output :



Along with Pre-processing of the data set , this project uses Random Forest ,
Random Forest is a popular machine learning algorithm that belongs to supervised learning techniques. It can be used for classification and regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and improve the performance of the model.Some advantages of Random Forest Algorithm are

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.
- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

Machine learning models often take hours or days to run, especially on large data sets with many features. If your machine shuts down, you will lose your model and have to train from scratch.Pickle is a useful Python tool that allows you to save your ML models, minimize lengthy retraining times, and share, validate, and reload pre-trained machine learning models. Most data scientists working in ML will use Pickle or Joblib to save their ML model for future use.Pickle is a generic object serialization module that can be used to serialize and deserialize objects. While it is often combined with saving and reloading trained machine learning models, it can really be used on any type of object. This is how you can use Pickle to save a trained model to a file and reload it to get predictions.

Save the model to reuse it again

In [72]:
```
import pickle
# open a file, where you ant to store the data
file = open('flight_rf.pkl', 'wb')

# dump information to that file
pickle.dump(rf_random, file)
```

In [73]:
```
model = open('flight_rf.pkl','rb')
forest = pickle.load(model)
```

In [74]:
```
y_prediction = forest.predict(X_test)
```

In [75]:
```
metrics.r2_score(y_test, y_prediction)
```

Out[75]:
```
0.8129549483358792
```

4. Results , Testing and Analysis

Testing

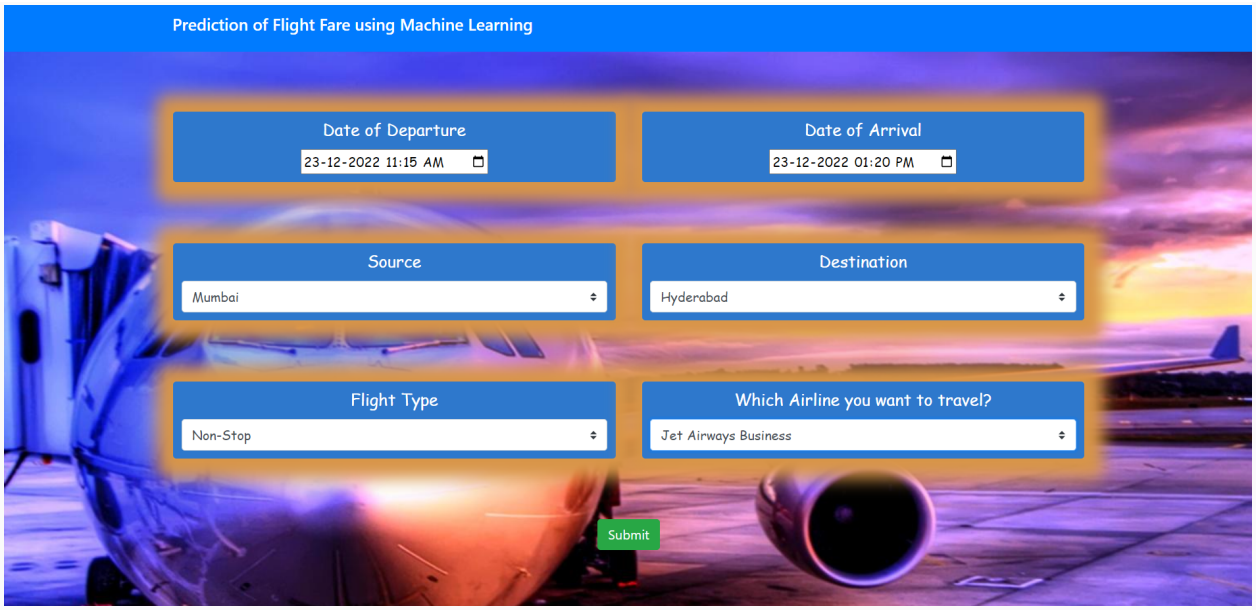
Test ID	Summary	Test Step	Test Data	Expected Result	Result
Test 1	Test the Behavior of Required fields	Send form without any parameters	no data	Form should not be submit	PASS
		Send form with some parameters	Source data	Form should not be submit	PASS
Test 2	On Screen instructions	submiting form without any date of departure	no data	msg : * please fill out this field	PASS
		submiting form without any date of arrival	no data	msg : * please fill out this field	PASS

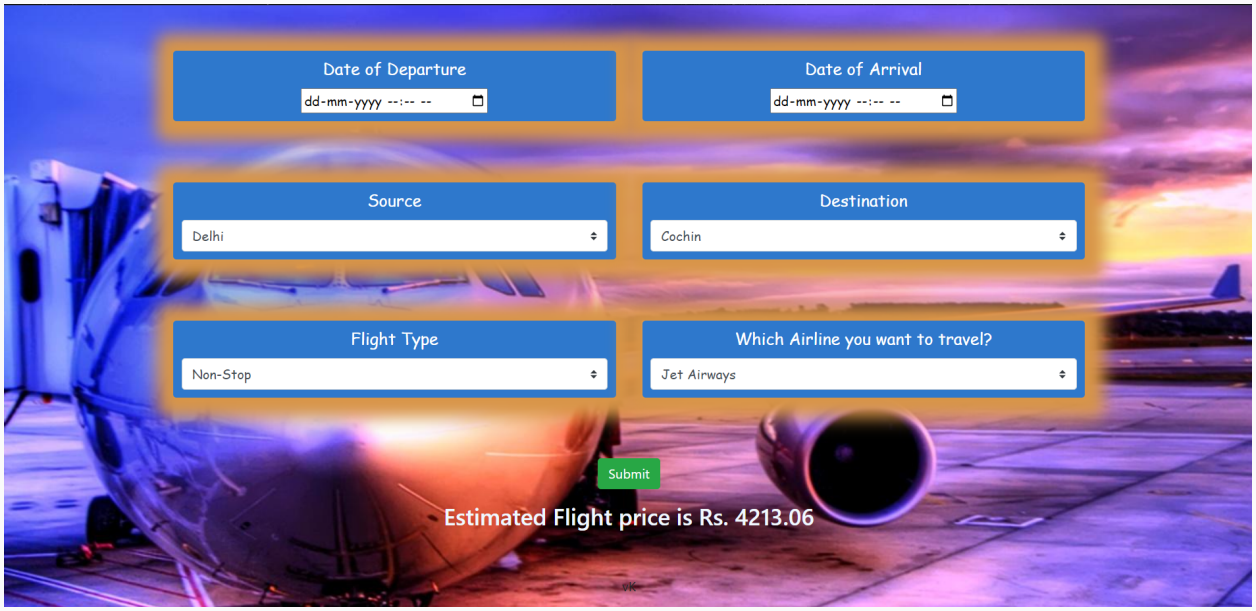
Testing

Before Hyperparameter tuning R2 value of test dataset was 79% and after apply hyperparameter tuning R2 value is increased by 81%

```
In [60]: metrics.r2_score(y_test, y_pred)
Out[60]: 0.7971708186549799
```

```
In [73]: y_prediction = forest.predict(X_test)
In [74]: metrics.r2_score(y_test, y_prediction)
Out[74]: 0.8120711829697359
```





As we can see above , on selecting everything correctly , our model is predicting the estimated price of flight ticket.

5. Conclusion & Future Enhancements

This report is “Flight Fare prediction”. We have collected data from Kaggle and show that it is possible to predict flight prices based on historical fare data. Experimental results show that ML models are a good tool to predict airfares. Other important factors in predicting airfares are data collection and selection of features from which we have drawn useful conclusions. From our tests, we concluded which feature has the most influence on flight fare prediction.

In addition to selected features, there are other features that can improve forecast accuracy, such as we can get weekend fares and night flight fares are often low. In the future, this work could be extended to predicting airfares for entire airline flight maps. Additional tests on larger airline ticket datasets are urgently needed, but this first pilot study highlights the potential of machine learning models to guide consumers in purchasing airline tickets during the period. best market. We can also use recent datasets or can take International dataset and implement multiple algorithms for more promising results.

5. References

1. Vaibhav Kumar,Flight Price ML,https://github.com/vkshirley/flightprice-ML
2. NIKHIL MITTAL,Flight Fare Prediction MH,Flight Fare Prediction Dataset by MachineHack,https://www.kaggle.com/datasets/nikhilmittal/flight-fare-prediction-mh
3. Subhajit Saha, Flask – (Creating first simple application)https://www.geeksforgeeks.org/flask-creating-first-simple-application/
4. Tutorialspoint, Flask application https://www.tutorialspoint.com/flask/flask_overview.htm
5. Python Basics : Flask Tutorial Hello World https://pythonbasics.org/flask-tutorial-hello-world/
6. Matt Clarke ,How to save and load machine learning models using Pickle,https://practicaldatascience.co.uk/machine-learning/how-to-save-and-load-machine-learning-models-using-pickle#:~:text=Pickle%20is%20a%20useful%20Python,ML%20model%20for%20future%20use.