Flight Fare Predictor

A

Project Report

Submitted for the partial fulfilment

of B.Tech. Degree

in

Information Technology

by

Abhishek Jaiswal (1805213002)

Neeraj Sharma (1805213031)

Niharika Chaudhary (1705213026)

Under the supervision of

Dr. Manik Chandra

Dr. Aditi Sharma

Department of Computer Science and Engineering

Institute of Engineering and Technology

Dr. A.P.J. Abdul Kalam Technical University, Lucknow, Uttar Pradesh.

June 2022

Contents

Declaration

We hereby declare that this submission is our own work and that, to the best of our belief and knowledge, it contains no material previously published or written by another person or material which to a substantial error has been accepted for the award of any degree or diploma of university or other institute of higher learning, except where the acknowledgement has been made in the text. The project has not been submitted by us at any other institute for requirement of any other degree.

Submitted by: - Date:

(1) Name: Abhishek Jaiswal

Roll No.: 1805213002

Branch: IT

Signature:

(2) Name: Neeraj Sharma

Roll No.: 1805213031

Branch: IT

Signature:

(3) Name: Niharika Chaudhary

Roll No: 1705213026

Branch: IT

Signature:

Certificate

This is to certify that the project report entitled “Flight Fare Predictor” presented by Abhishek, Neeraj Sharma and Niharika Chaudhary in the partial fulfillment for the award of Bachelor of Technology in Information Technology, is a record of work carried out by them under my sup--ervision and guidance at the Department of Computer Science and Engineering at Institute of Engineering and Technology, Lucknow.

It is also certified that this project has not been submitted at any other Institute for the award of any other degrees to the best of my knowledge.

(Dr Manik Chandra)
Department of Computer Science and Engineering
Institute of Engineering and Technology, Lucknow

(Dr Aditi Sharma)
Department of Computer Science and Engineering
Institute of Engineering and Technology, Lucknow

Acknowledgement

We would like to thank Dr. Manik Chandra and Dr. Aditi Sharma for their cooperation in completing our project on Flight Fare Predictor.

We would like to take this opportunity to express my gratitude to all of us - Abhishek Jaiswal, Neeraj Sharma and Niharika Chaudhary. We all together have completed this project with proper cooperation.

We would also like to thank my friends and family for their constant encouragement and support throughout the project.

Lastly, We like to thank all our supporters who have motivated us to fulfill their project before the timeline.

Abstract

Optimal timing for airline ticket buying from the consumer's perspective is challenging principally due to the fact that buyers have insufficient information for reasoning about future airline price fluctuations. In this project, we majorly focused to uncover underlying trends of flight fares in India using historical data,as well as to recommend the optimum time to buy a flight ticket.

We will analyze the flight fare prediction using Machine Learning dataset using essential exploratory data analysis methods then will make some predictions about the fare of the flight based on some features such as the type of airline, what is the arrival time, what is the departure time, what is the duration of the flight, source, destination and more.

Airline companies use complex algorithms to find flight prices given various conditions present at that particular time. To predict flight fares, these techniques take financial, marketing, and various type ofsocial factors into account.

Nowadays, the number of people using flights has raised significantly. Pricing change dynamically due to different conditions, making it difficult for airlines to maintain prices. As a result, we will attempt to solve this problem using machine learning. This can help airlines in determining what prices they can keep. Customers can also use it to predict future airline prices and plan their journey accordingly.

Remarkably, the trends of the prices are extremely dependent to the route, month of departure, day of departure, time of departure, if the day of departure is a holiday and airline provider. Most commercial routes (tier 1 to tier1 city like Mumbai-Delhi) are highly competitive routes, had a non-decreasing trend where prices increased as day to departure decreased, however other routes (tier 1 to tier 2 cities like Delhi - Guwahati) had a particular time frame where the prices are minimal. Furthermore, the data also revealed two basic categories of airline carriers operating in India - the economical group and the deluxe group, and in most cases, the affordable priced flight belonging to the economical group. The data also validated the fact that there are particular time periods of the day where the prices are expected to be their highest.

1. Introduction

Everyone knows that holidays always call for a much-needed vacation and planning the travel itinerary becomes a time-consuming task. The commercial aviation business has grown tremendously and has become a regulated marketplace as a result of the worldwide growth of the internet and E-commerce.

Hence, for Airline revenue management, different strategies like customer profiling, financial marketing, social factors are used for setting ticket fairs. When tickets are booked months in advance, airfares are often reasonable, but when tickets are booked in a hurry, they are often higher.

But, the number of days/hours until departure isn't the only factor which decides flight fare, there are numerous other factors as well. Customers find it quite difficult to obtain a perfect and lowest ticket deal due to the aviation industry's complex pricing methodology. Machine Learning and Deep Learning-based technologies and modals have been created to overcome this challenge, and substantial research is also happening. This study discusses a Machine Learning-based Flight Fare Prediction System that employs Random Forest Regression to predict airline ticket pricing. Various features influences prices are also studied along with the system's experimental analysis. Section II included a literature review that looked at technical papers as well as some current models and systems. Differences in the features considered are also mapped down, In Section III, the proposed system is described in detail along with the workflow and its features. In Section IV, the model' s implementation is explained. In Section V, the results as well as various comparisons between findings are reported.

In Section VI, conclusions are provided, as well as prospective advancements for further research.

2. Literature Review

A flight price prediction system has been created [1]. The paper begins with some broad information regarding machine learning, after which the authors further proceed to the methodology. The methodology consists of four-phase process that influences flight prices, collection based on data from Greek Aegean Airlines, selection and evaluation of an accurate ML Regression model.

Key to its success is the integrity of the system where consumers accurately represent the segments for which prices have been differentially determined.[2]

Today, airlines price tickets "as much as the customer and market will bear," according to consultant and former airline planning executive. Airlines also profile their customers to help them adjust prices.[3]

This often means placing passengers into one of two groups: leisure or business. And the way each group is priced is very different. Most studies on airfare price prediction have focused on either the national level or a specific market. Research at the market segment level, however, is still very limited. We define the term market segment as the market/airport pair between the flight origin and the destination.[4]

The airline dataset included the following eight characteristics: departure and arrival times, free luggage, days before departure, number of intermediate stops, holiday, time of day, and any day of the week. The authors performed prediction using eight state-of-art regression Machine Learning models that including, MLP, GRNN, ELM, Random Forest Regression Tree, Bagging Tree, Regression Tree, Regression SVM, Bagging Regression Tree, and Linear Regression. Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.[5]

All in [7] present a review of deep learning and social media data-based Airline ticket price prediction model. The authors introduce the current airline ticket pricing situation with the factors that affect ticket prices.

A Random Forest operates by constructing several decision trees during training time and outputting the mean of the classes as the prediction of all the trees. A Random Forest Regression model is powerful and accurate. It usually performs great on many problems, including features with non-linear relationships.[8]

With an accuracy of 87.42 percent, the Bagging Regression Tree model outperforms other models. All states the problem of market segment level airline price prediction and propose a novel application based on Machine learning. Random Forest Model is used for development since it outperforms other models such as LR SVM and Neural Network in terms of data performance.

With a R squared score of 0.868, this prediction framework has a good level of accuracy.

2. Methodology

The following steps were performed while building the system.

A. Data Collection

The important part of the project is data collection. Training and testing datasets, both have been extracted from website Kaggle.com. They include both categorical and nominal data for Indian Airlines from the year 2019. The dataset contains vital information on various factors that influence the price of a flight, such as the location of departures and arrivals, the time of departure and arrivals, the flight path, the number of passengers on board, and the number of halts along the way depending on those factors. There are 10683 rows and 11 columns (each representing one attribute) in this enormous dataset.

B. Data Pre-processing

While pre-processing the data, we converted the date of journey, departure time and the arrival time from string datatype to date-time object and then extracted the numeric values from the m, the month-date numeric value from the date of journey attribute and hour-minute numeric value from the departure time and arrival time attributes accordingly. After that, we used the 'One hot encoding' method for nominal categorical data and the label encoding method for ordinal categorical data in both the training and testing datasets. It is a process of converting the categorical data variables into numerical values, making them acceptable for machine learning algorithms. One hot encoding approach was applied to nominal categorical data attributes such as the source, the destination and the airline company chose n by the user. The nominal categorical data elements such as the 'total number of halts' were encoded using the label encoding approach. The columns were re-arranged in the final step.

C. Data Cleaning

In the training dataset, the null values present were removed. There were a few columns which were completely useless for the feature selection process, and were deleted from the dataset. After the new columns with the numerical values extracted from the dataset, were stored for the prediction and the columns of attributes with categorical data were eliminated from the dataset. As a result, the training dataset suitable for use was obtained with the following attribute columns.

Table I

Description of the Attributes

Data Attribute	Description
Total Stops	The number of halts in the journey
Journey Day	The numerical value of 'day' selected from the calendar
Journey Month	The numerical value of 'month' selected from calendar
Dep_hour	The numerical value of 'hour' in departure time
Dep_Min	The numerical value of 'minutes' in departure time
Arrival_hour	The numerical value of 'hour' in arrival time
Arrival_Min	The numerical value of 'minutes' in arrival time
Duration_Hour	The numerical value of 'hours' in duration time
Duration_Min	The numerical value of the minutes in duration time
Airline Company (One hot encoding applied)	Display '1' for the chosen Airline company and display 'O' for the rest
Source (One hot encoding applied)	Display '1' for the chosen Source and display 'O' for the rest
Destination (one hot encoding applied)	Display '1' for the chosen Destination and display 'O' for the rest

D. Presenting the final prediction

The user input fields will be provided on a webpage developed using the flask framework. HTML5 was used to create the webpage body, while CSS3 was used to style it. After the user fill out all the required input fields and submits the form, the data will be sent to the generated random forest regression model and the predicted valueof the ticket price will be displayed.

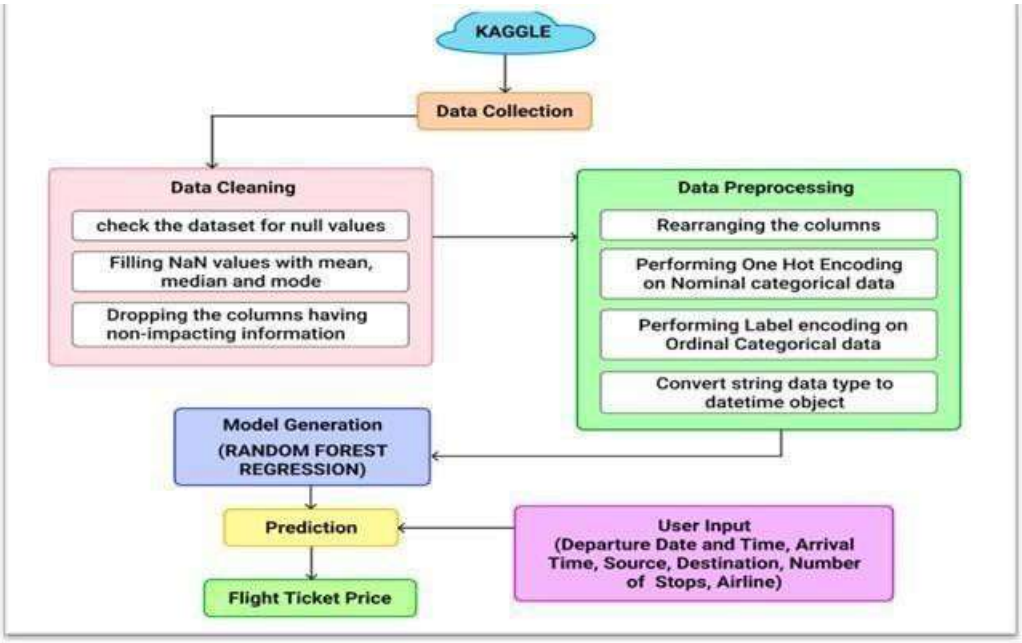


Fig1. Methodology block diagram

Implememtation

3.1.1 Importing Libraries

```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error as mse
from sklearn.metrics import r2_score
from math import sqrt
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import KFold
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import RandomizedSearchCV

from prettytable import PrettyTable
```

3.1.2 Reading Training Dataset

```
train_df = pd.read_excel("Data_Train.xlsx")
train_df.head(10)
```

Output:

	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
5	SpiceJet	24/06/2019	Kolkata	Banglore	CCU → BLR	09:00	11:25	2h 25m	non-stop	No info	3873
6	Jet Airways	12/03/2019	Banglore	New Delhi	BLR → BOM → DEL	18:55	10:25 13 Mar	15h 30m	1 stop	In-flight meal not included	11087
7	Jet Airways	01/03/2019	Banglore	New Delhi	BLR → BOM → DEL	08:00	05:05 02 Mar	21h 5m	1 stop	No info	22270
8	Jet Airways	12/03/2019	Banglore	New Delhi	BLR → BOM → DEL	08:55	10:25 13 Mar	25h 30m	1 stop	In-flight meal not included	11087
9	Multiple carriers	27/05/2019	Delhi	Cochin	DEL → BOM → COK	11:25	19:15	7h 50m	1 stop	No info	8625

3.2 Exploratory Data Analysis (EDA)

Now here we will be looking at the kind of columns our dataset has.

```
train_df.columns
```

Output:

```
Index(['Airline', 'Date_of_Journey', 'Source', 'Destination', 'Route',
       'Dep_Time', 'Arrival_Time', 'Duration', 'Total_Stops',
       'Additional_Info', 'Price'],
      dtype='object')
```

Here we can get more information about our dataset

```
train_df.info()
```

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   Airline          10683 non-null  object
 1   Date_of_Journey  10683 non-null  object
 2   Source           10683 non-null  object
 3   Destination      10683 non-null  object
 4   Route            10682 non-null  object
 5   Dep_Time         10683 non-null  object
 6   Arrival_Time     10683 non-null  object
 7   Duration         10683 non-null  object
 8   Total_Stops      10682 non-null  object
 9   Additional_Info  10683 non-null  object
 10  Price            10683 non-null  int64 
dtypes: int64(1), object(10)
memory usage: 918.2+ KB
```

To know more about the dataset

```
train_df.describe()
```

Output:

	Price
count	10683.000000
mean	9087.064121
std	4611.359167
min	1759.000000
25%	5277.000000
50%	8372.000000
75%	12373.000000
max	79512.000000

Now while using the IsNull function we will gonna see the number of null values in our dataset

```
train_df.isnull().head()
```

Output:

Airline	Source	Destination	Route	Duration	Total_Stops	Additional_Info	Price	Journey_day	Journey_month	Dep_hour	Dep_min	Arrival_hour	Arrival_min
False	False	False	False	False	False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False	False	False	False	False	False	False

Now while using the IsNull function and sum function we will gonna see the number of null values in our dataset

```
train_df.isnull().sum()
```

Output:

```
Airline            0
Date_of_Journey    0
Source             0
Destination        0
Route              1
Dep_Time           0
Arrival_Time       0
Duration           0
Total_Stops        1
Additional_Info    0
Price              0
dtype: int64
```

Dropping NAN values

```
train_df.dropna(inplace = True)
```

Duplicate values

```
train_df[train_df.duplicated()].head()
```

Output:

Airline	Source	Destination	Route	Duration	Total_Stops	Additional_Info	Price	Journey_day	Journey_month	Dep_hour	Dep_min	Arrival_hour	Arrival_min
Vistara	Banglore	New Delhi	BLR → DEL	175	non-stop	No info	7608	3	3	21	10	0	5
Air Asia	Banglore	New Delhi	BLR → DEL	165	non-stop	No info	4482	24	3	23	25	2	10

Here we will be removing those repeated values from the dataset and keeping the in-place attribute to be true so that there will be no changes.

```
train_df.drop_duplicates(keep='first',inplace=True)
train_df.head()
```

Output:

	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

```
train_df.shape
```

Output:

```
(10462, 11)
```

Checking the Additional_info column and having the count of unique types of values.

```
train_df["Additional_Info"].value_counts()
```

Output:

```
No info                         8182
In-flight meal not included     1926
No check-in baggage included     318
1 Long layover                    19
Change airports                    7
Business class                     4
No Info                            3
1 Short layover                    1
2 Long layover                     1
Red-eye flight                     1
Name: Additional_Info, dtype: int64
```

Checking the different Airlines

```
train_df["Airline"].unique()
```

Output:

```
array(['IndiGo', 'Air India', 'Jet Airways', 'SpiceJet',
       'Multiple carriers', 'GoAir', 'Vistara', 'Air Asia',
       'Vistara Premium economy', 'Jet Airways Business',
       'Multiple carriers Premium economy', 'Trujet'], dtype=object)
```

Now let's look at our testing dataset

```
test_df = pd.read_excel("Test_set.xlsx")
test_df.head(10)
```

Output:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info
0	Jet Airways	6/06/2019	Delhi	Cochin	DEL → BOM → COK	17:30	04:25 07 Jun	10h 55m	1 stop	No info
1	IndiGo	12/05/2019	Kolkata	Banglore	CCU → MAA → BLR	06:20	10:20	4h	1 stop	No info
2	Jet Airways	21/05/2019	Delhi	Cochin	DEL → BOM → COK	19:15	19:00 22 May	23h 45m	1 stop	In-flight meal not included
3	Multiple carriers	21/05/2019	Delhi	Cochin	DEL → BOM → COK	08:00	21:00	13h	1 stop	No info
4	Air Asia	24/06/2019	Banglore	Delhi	BLR → DEL	23:55	02:45 25 Jun	2h 50m	non-stop	No info
5	Jet Airways	12/06/2019	Delhi	Cochin	DEL → BOM → COK	18:15	12:35 13 Jun	18h 20m	1 stop	In-flight meal not included
6	Air India	12/03/2019	Banglore	New Delhi	BLR → TRV → DEL	07:30	22:35	15h 5m	1 stop	No info
7	IndiGo	1/05/2019	Kolkata	Banglore	CCU → HYD → BLR	15:15	20:30	5h 15m	1 stop	No info
8	IndiGo	15/03/2019	Kolkata	Banglore	CCU → BLR	10:10	12:55	2h 45m	non-stop	No info
9	Jet Airways	18/05/2019	Kolkata	Banglore	CCU → BOM → BLR	16:30	22:35	6h 5m	1 stop	No info

Now here we will be looking at the kind of columns our testing data has.

```
test_df.columns
```

Output:

```
Index(['Airline', 'Date_of_Journey', 'Source', 'Destination', 'Route',
       'Dep_Time', 'Arrival_Time', 'Duration', 'Total_Stops',
       'Additional_Info'],
      dtype='object')
```

Information about the dataset

```
test_df.info()
```

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2671 entries, 0 to 2670
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   Airline          2671 non-null   object
 1   Date_of_Journey  2671 non-null   object
 2   Source           2671 non-null   object
 3   Destination      2671 non-null   object
 4   Route            2671 non-null   object
 5   Dep_Time         2671 non-null   object
 6   Arrival_Time     2671 non-null   object
 7   Duration         2671 non-null   object
 8   Total_Stops      2671 non-null   object
 9   Additional_Info  2671 non-null   object
dtypes: object(10)
memory usage: 208.8+ KB
```

To know more about the testing dataset

```
test_df.describe()
```

Output:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info
count	2671	2671	2671	2671	2671	2671	2671	2671	2671	2671
unique	11	44	5	6	100	199	704	320	5	6
top	Jet Airways	9/05/2019	Delhi	Cochin	DEL → BOM → COK	10:00	19:00	2h 50m	1 stop	No info
freq	897	144	1145	1145	624	62	113	122	1431	2148

Now while using the IsNull function and sum function we will gonna see the number of null values in our testing data

```
test_df.isnull().sum()
```

Output:

```
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
dtype: int64
```

3.3 Data Visualization

Plotting Price vs Airline plot

```
sns.catplot(y = "Price", x = "Airline", data = train_df.sort_values("Price", ascending =
plt.show()
```

Output:

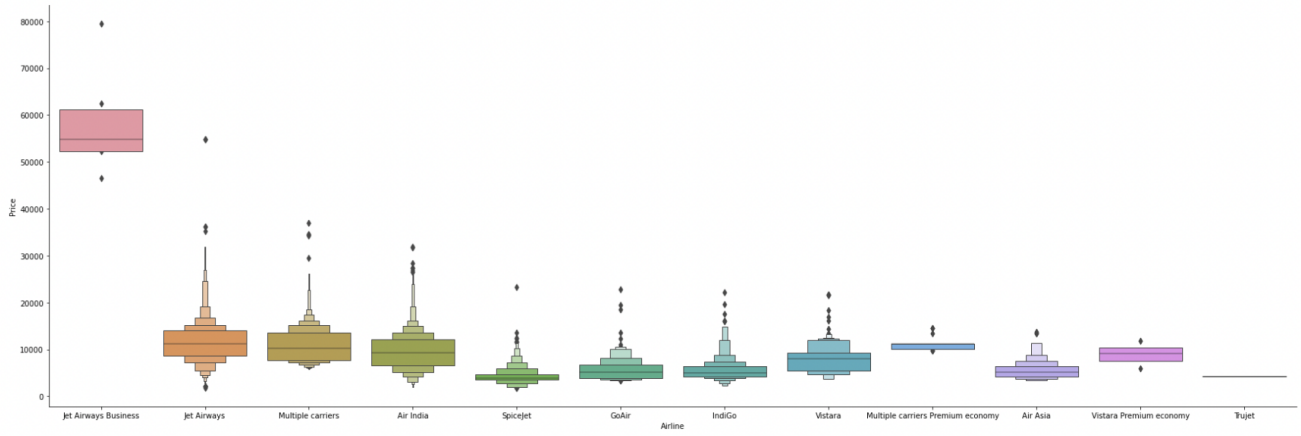


Fig. 3.3.1

3.4 Feature Engineering

Let's see our processed data first

```
train_df.head()
```

Output:

	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

Here first we are dividing the features and labels and then converting the hours in minutes.

```
train_df['Duration'] = train_df['Duration'].str.replace("h", '*60').str.replace(' ','+')
test_df['Duration'] = test_df['Duration'].str.replace("h", '*60').str.replace(' ','+').s
```

Date_of_Journey: Here we are organizing the format of the date of journey in our dataset for better preprocessing in the model stage.

```
train_df["Journey_day"] = train_df['Date_of_Journey'].str.split('/').str[0].astype(int)
train_df["Journey_month"] = train_df['Date_of_Journey'].str.split('/').str[1].astype(int
train_df.drop(["Date_of_Journey"], axis = 1, inplace = True)
```

Dep_Time: Here we are converting departure time into hours and minutes

```
train_df["Dep_hour"] = pd.to_datetime(train_df["Dep_Time"]).dt.hour
train_df["Dep_min"] = pd.to_datetime(train_df["Dep_Time"]).dt.minute
train_df.drop(["Dep_Time"], axis = 1, inplace = True)
```

```
train_df["Arrival_hour"] = pd.to_datetime(train_df.Arrival_Time).dt.hour
train_df["Arrival_min"] = pd.to_datetime(train_df.Arrival_Time).dt.minute
train_df.drop(["Arrival_Time"], axis = 1, inplace = True)
```

Now after final preprocessing let's see our dataset

```
train_df.head()
```

Output:

Airline	Source	Destination	Route	Duration	Total_Stops	Additional_Info	Price	Journey_day	Journey_month	Dep_hour	Dep_min	Arrival_hour	Arrival_min
IndiGo	Banglore	New Delhi	BLR → DEL	170	non-stop	No info	3897	24	3	22	20	1	10
Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	445	2 stops	No info	7662	1	5	5	50	13	15
Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	1140	2 stops	No info	13882	9	6	9	25	4	25
IndiGo	Kolkata	Banglore	CCU → NAG → BLR	325	1 stop	No info	6218	12	5	18	5	23	30
IndiGo	Banglore	New Delhi	BLR → NAG → DEL	285	1 stop	No info	13302	1	3	16	50	21	35

Dealing with Categorical Data and Numerical Data

```
train_categorical_data = data.select_dtypes(exclude=['int64', 'float','int32'])
train_numerical_data = data.select_dtypes(include=['int64', 'float','int32'])

test_categorical_data = test_df.select_dtypes(exclude=['int64', 'float','int32','int32']
test_numerical_data  = test_df.select_dtypes(include=['int64', 'float','int32'])
train_categorical_data.head()
```

Output:

	Airline	Source	Destination	Route	Total_Stops	Additional_Info
0	IndiGo	Banglore	New Delhi	BLR → DEL	non-stop	No info
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	2 stops	No info
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	2 stops	No info
3	IndiGo	Kolkata	Banglore	CCU → NAG → BLR	1 stop	No info
4	IndiGo	Banglore	New Delhi	BLR → NAG → DEL	1 stop	No info

Label Encode and Hot Encode for Categorical Columns

```
le = LabelEncoder()
train_categorical_data = train_categorical_data.apply(LabelEncoder().fit_transform)
test_categorical_data = test_categorical_data.apply(LabelEncoder().fit_transform)
train_categorical_data.head()
```

Output:

	Airline	Source	Destination	Route	Total_Stops	Additional_Info
0	3	0	5	18	4	8
1	1	3	0	84	1	8
2	4	2	1	118	1	8
3	3	3	0	91	0	8
4	3	0	5	29	0	8

Concatenating both Categorical Data and Numerical Data

```
X = pd.concat([train_categorical_data, train_numerical_data], axis=1)
y = train_df['Price']
test_set = pd.concat([test_categorical_data, test_numerical_data], axis=1)
X.head()
```

Output:

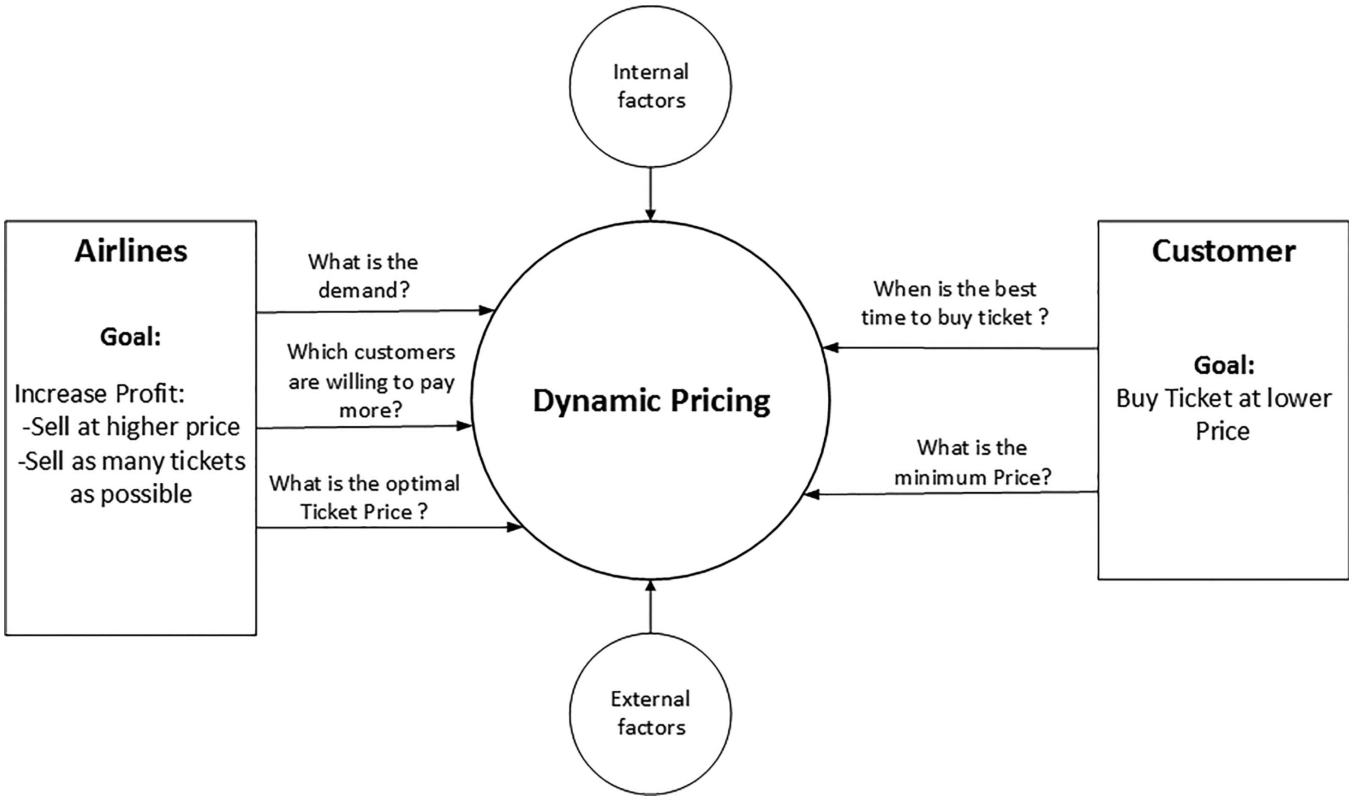
	Airline	Source	Destination	Route	Total_Stops	Additional_Info	Duration	Journey_day	Journey_month	Dep_hour	Dep_min	Arrival_hour	Arrival_min
0	3	0	5	18	4	8	170	24	3	22	20	1	10
1	1	3	0	84	1	8	445	1	5	5	50	13	15
2	4	2	1	118	1	8	1140	9	6	9	25	4	25
3	3	3	0	91	0	8	325	12	5	18	5	23	30
4	3	0	5	29	0	8	285	1	3	16	50	21	35

```
y.head()
```

Output:

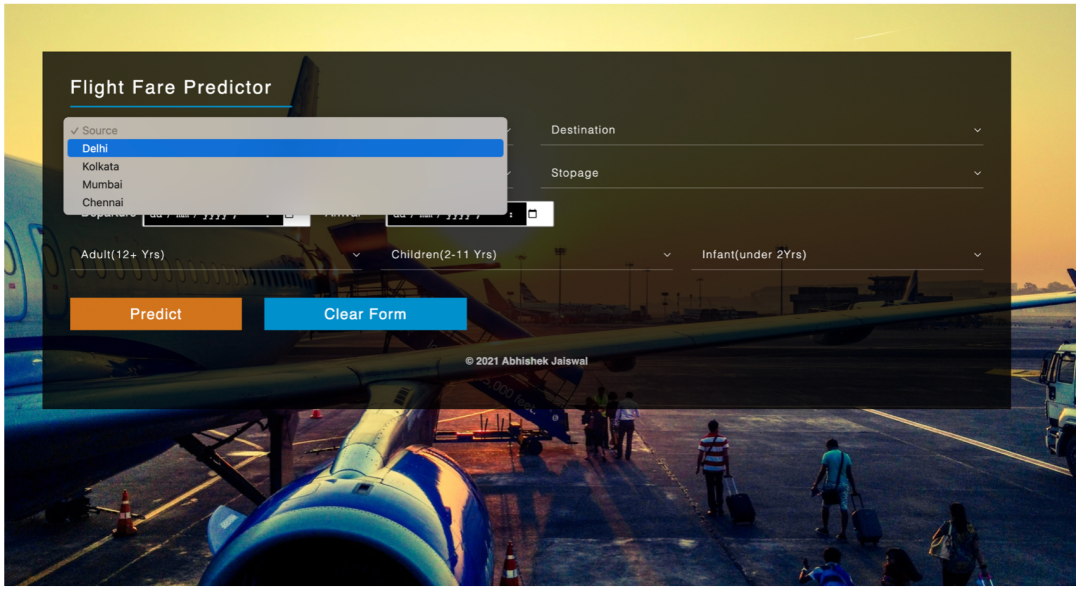
```
0     3897
1     7662
2    13882
3     6218
4    13302
Name: Price, dtype: int64
```

4. DATA FLOW DIAGRAM



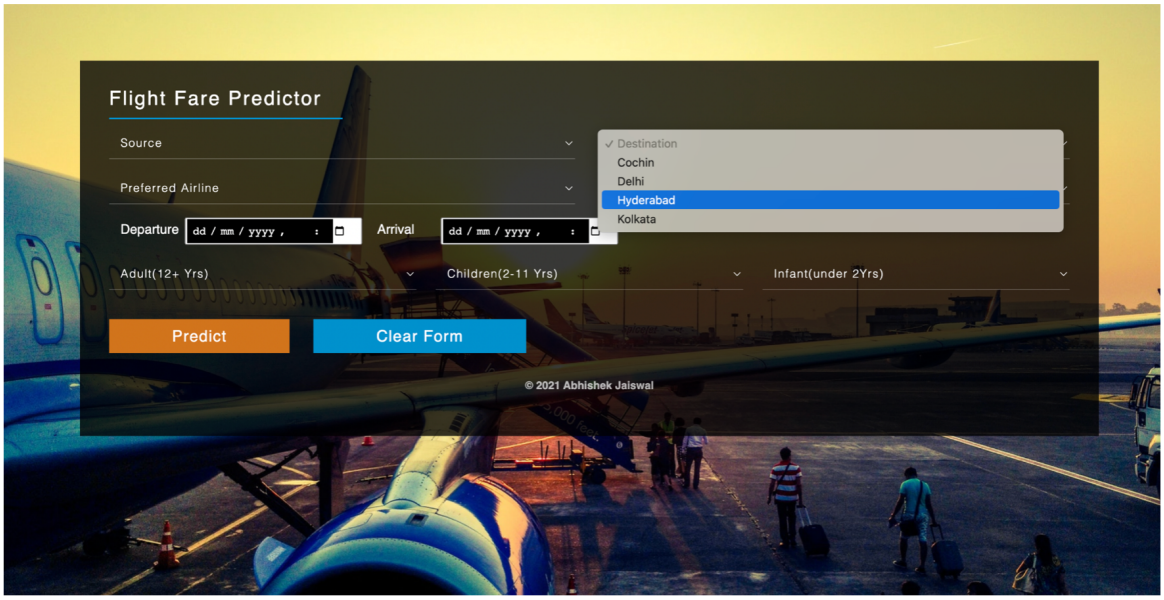
Customer's communication flow with our web app-

Step-1



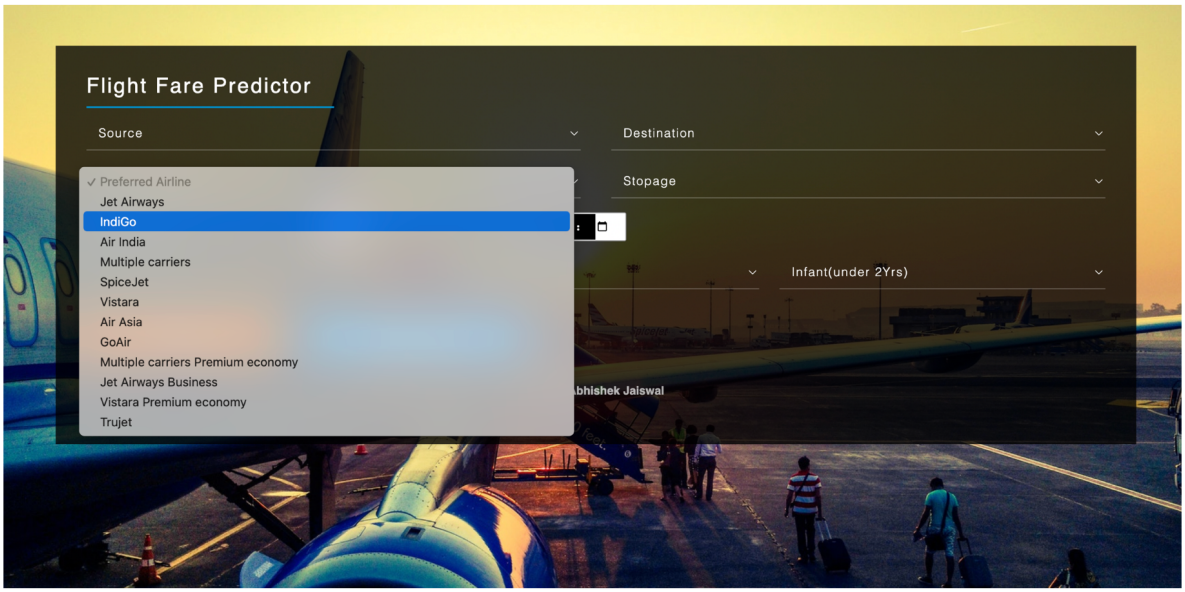
Choosing the Source

Step-2



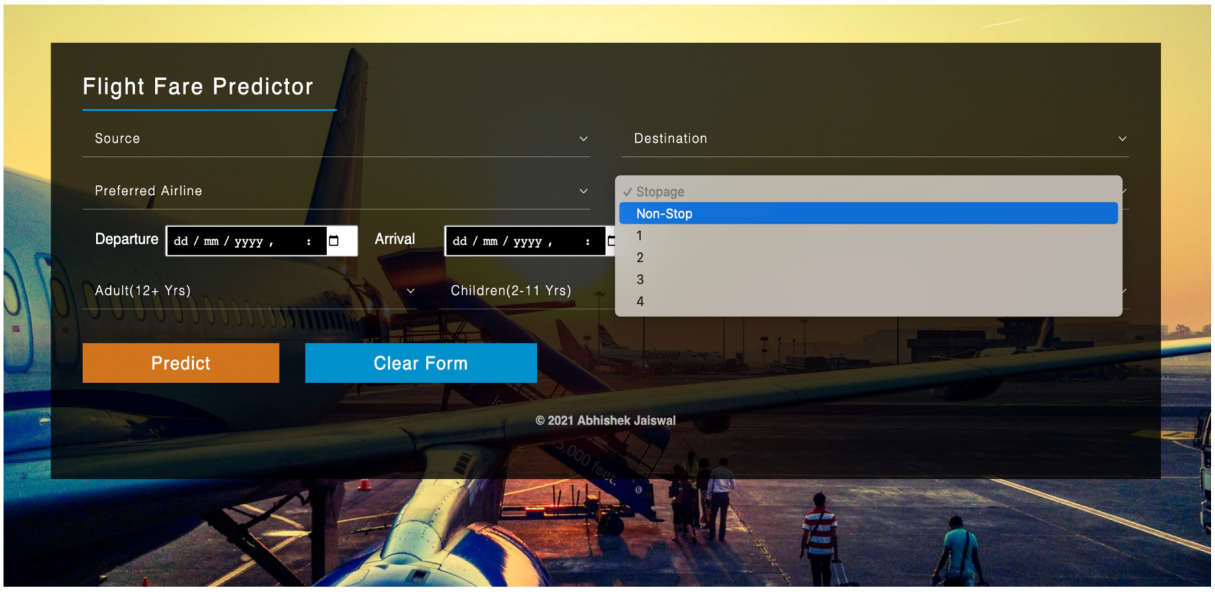
Choosing the Destination

Step-3



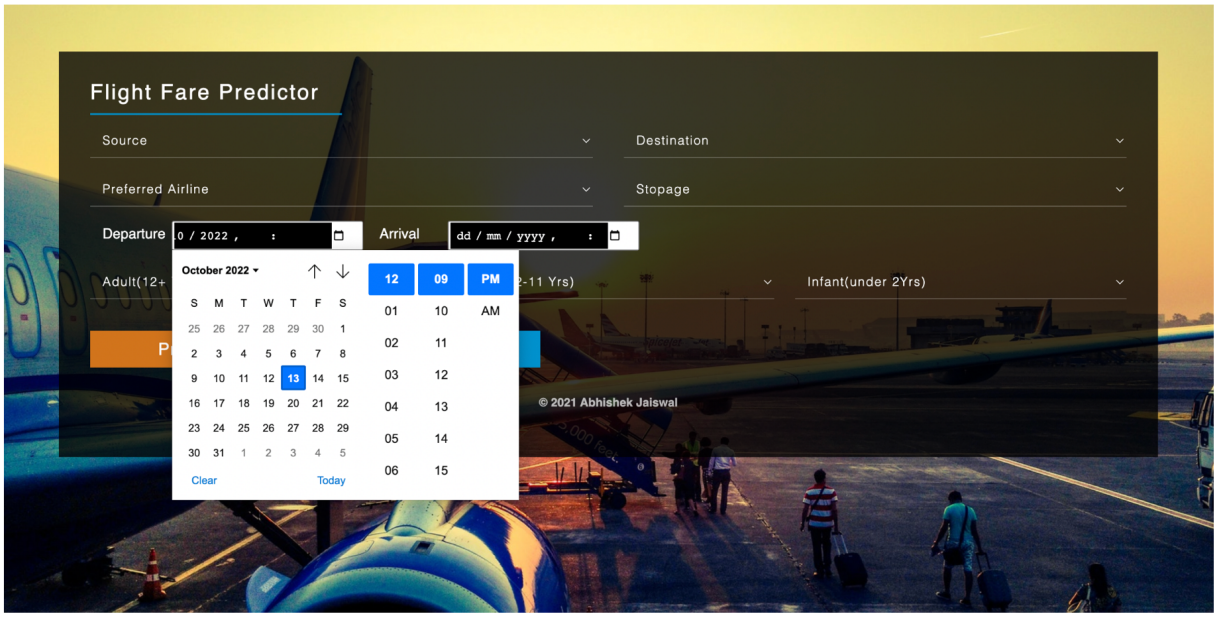
Choosing the Airline

Step-4



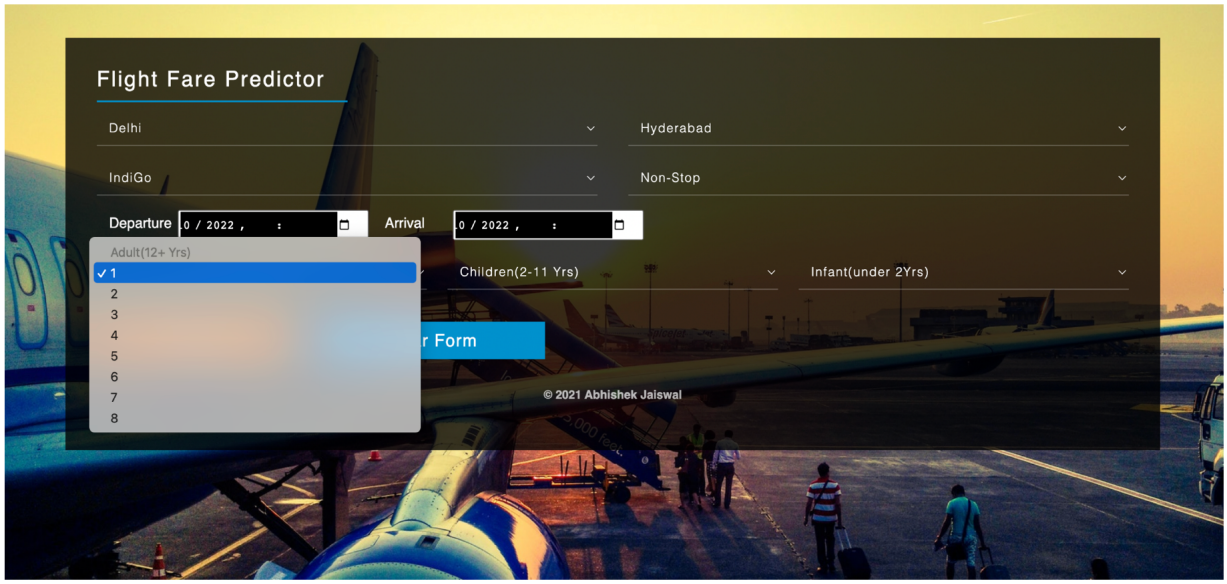
Choosing the Number of Stopages

Step-5



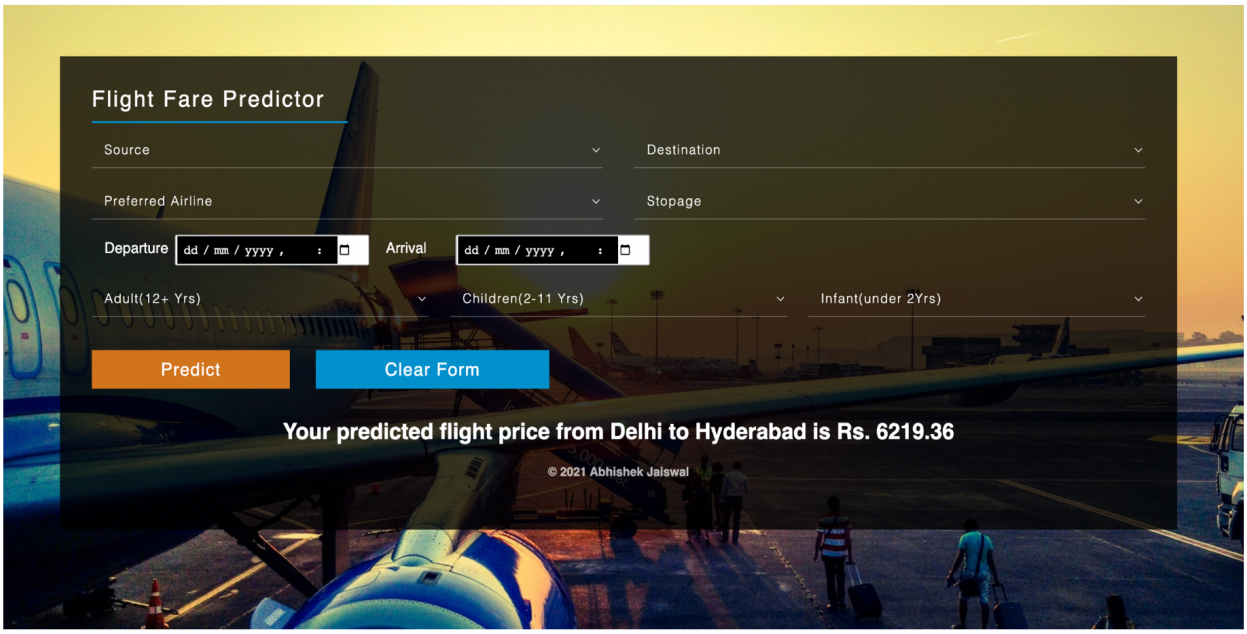
Choosing the Arrival and Departure Date and Time

Step-6



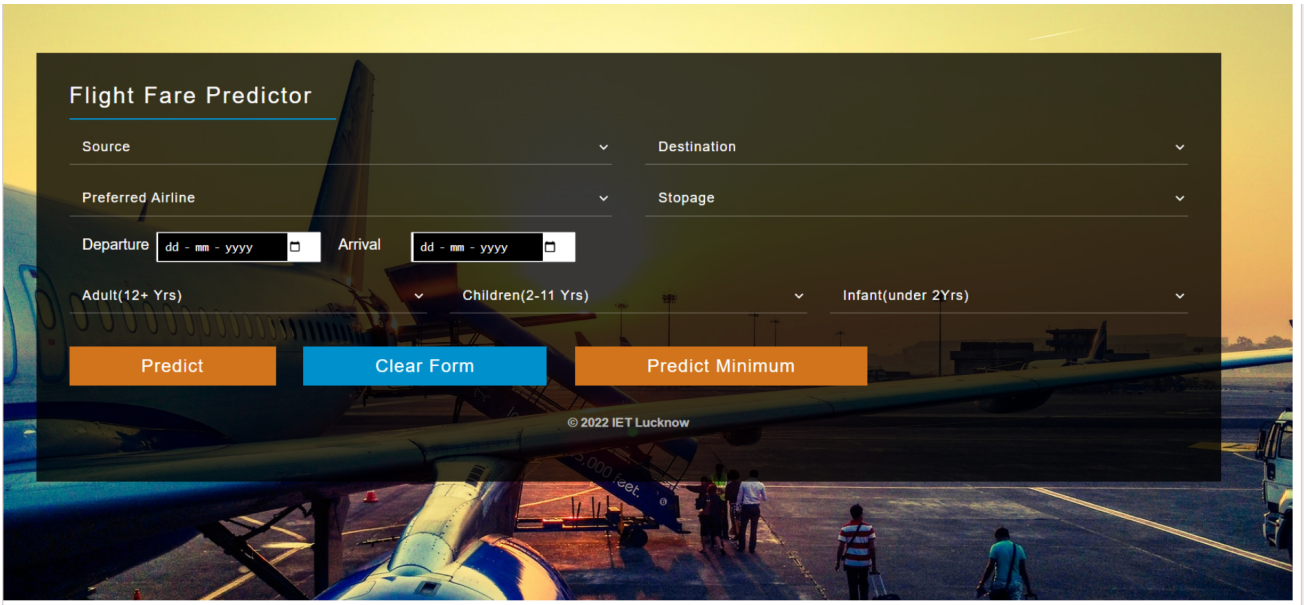
Choosing the Number of Passengers

Step-7



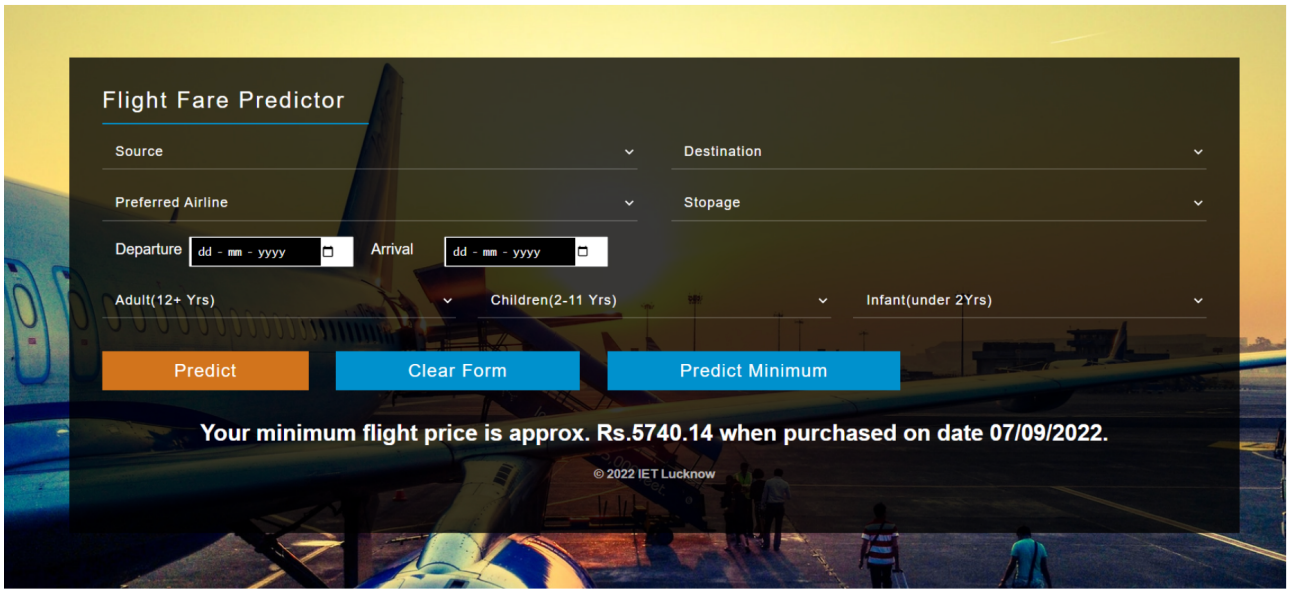
Prediction of Fare

Step-8



Adding Functionality of Predicting Minimum flight price

Step-9



Minimum Flight Prediction between Current Date to Departure Date

5. Conclusion

From our detailed analysis of each of the 18 routes, we can determine the following:

- Flight prices almost always remain constant or increase between the major cities.
- Tourist routes and routes that provide services to Tier-2 cities in the country have varying patterns in terms of pricing increases and decreases.
- The model in the worst case almost breaks even with the profits and losses, and most case saves an average of about Rs. 200 per transaction when predicting to wait.
- Routes with data collected over a longer period of time likely to result in more accurate predictions in the model and, as a result it leads to higher average savings. We were successfully able to analyse each route and categorized the entire project based in terms of the sector to which the route belonged, and classified them into three major subsections: Business Routes, Tourist Routes and Tier-2 Routes. We've also successfully busted some of the typical myths and misconceptions related to the airline industry and backed them up with data and analysis.

Finally, we have created a User Interface for the entire process of purchasing an airline ticket and given a proof of our predictions based on the previous trends with our prediction. Thus, leaving itas a battle between 'The risk appetite of the user' vs 'Our understanding of the airline industry'.

REFERENCES

1. K. Tziridis, Th. Kalampokas, G. A. Papakostas, *"Airfare Prices Prediction Using Machine Learning Techniques", 25th European Signal Processing Conference (EUSIPCO), IEEE, October 26, 2017.*

2. Moira McCormick, BlackCurce, *"Behind the Scenes of Airline Pricing Strategies", September 19, 2017.Available: https://blog.blackcurve.com/behind-the-scenes-of-airline-pricing-strategies.*

3. Tom Chitty, CMBC Business News, *"This is how airplanes price tickets", August 3, 2018. Available: https://www.cnbc.com/2018/08/03/how-do-airlines-price-seat-tickets.html/.*

4. Tianyi Wang, Samira Pouyanfar, Haiman Tian, Yudong Tao, Miguel Alonso Jr., Steven Luis and Shu-Ching Chen,*"A Framework for Airfare Price Prediction: A Machine Learning Approach", 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI), September 9, 2019.*

5. Supriya Rajankar, Neha Sakharkar, Omprakash Rajankar, *"Predicting the price of a flight ticket with the use of Machine Learning algorithms", international journal of scientific & technology research volume 8, December, 2019.*

6. Tao Liu, Jian Cao, Yudong Tan, Quanwu Xiao, *"ACER: An Adaptive Context- Aware Ensemble Regression Model for Airfare Price Prediction".*

7. Juhar Ahmed Abdella, Nazar Zaki and Khaled Shuaib, *"Automatic Detection of Airline Ticket Price and Demand: A Review", 13th International Conference on Innovations in Information technology (IIT), January 10, 2019.*

8. Chaya Bakshi, Medium, *"Random Forest Regression", June 9, 2020, Available: https://levelup.gitconnected.com/random forest-regression-209c0f354c84*.

9. Zach, Statology, "How to calculate mean Absolute Error in Python", January 8. 2021, *Available: https://www.statology.org/mean-absolute-error-python/*.

10. Wikipedia, *"Mean Squared error", Available: https://en.wikipedia.org/wiki/Mean_squared_error*.

11. Science Direct, *"Root-Mean Squared Error", Available: https://www.sciencedirect.com/topics/engineering/root-mean-squared-error/*.

12. NCL.AC.UK, "Coefficient of Determination, R-squared", Available: *https://www.ncl.ac.uk/webtemplate/ask-assets/external/maths-resources/statistics/regression-and-correlation /coefficient-of-determination r-squared.html/*.