

FLIGHT PRICE PREDICTION

Submitted by
Steffin Varghese

Batch - 26

In partial fulfilment of Data Science – Internship

At



Flip Robo Technologies

AI and Software Development company

Flip Robo Technologies | Indiranagar, Bengaluru - 560 038, Karnataka, India

Month of Submission

July 2022

Acknowledgement

I am highly indebted to FlipRobo Technologies for giving me this opportunity to work on a project and for the guidance and persistent supervision, as well as for providing necessary project information and assistance in completing the project.

I would want to convey my gratitude to the members of FlipRobo Technologies for their kind encouragement and support in completing this project.

I would like to extend my heartfelt gratitude and appreciation to SME, Ms Khushboo Garg for spending such close attention to me and assisting me during the project's completion, as well as towards others who have volunteered to assist me with their skills.

I acknowledge my gratitude towards the authors of papers: "Flight Price Prediction for Users by Machine Learning Techniques", "Aircraft Ticket Price prediction using Machine Learning" and "Flight Fare Prediction System" for the insights I could attain through the extensive research which enhanced my knowledge in development of the project.

CONTENTS

INTRODUCTION

- Business Problem Framing
- Conceptual Background of the Domain Problem
- Review of Literature
- Motivation for the Problem Undertaken

Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem
- Data Sources and their formats
- Data Pre-processing
- Data Inputs- Logic- Output Relationships
- Assumptions related to the problem under consideration
- Hardware and Software Requirements and Tools Used

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)
- Testing of Identified Approaches (Algorithms) and evaluation of selected models
- Key Metrics for success in solving problem under consideration
- Visualizations
- Interpretation of the Results

CONCLUSION

- Key Findings and Conclusions of the Study
- Learning Outcomes of the Study in respect of Data Science
- Limitations of this work and Scope for Future Work

INTRODUCTION

Business Problem Framing

Anyone who has booked a flight ticket knows how unexpectedly the prices vary. The cheapest available ticket on a given flight gets more and less expensive over time. This usually happens as an attempt to maximize revenue based on –

1. Time of purchase patterns (making sure last-minute purchases are expensive).
2. Keeping the flight as full as they want it (raising prices on a flight which is filling up in order to reduce sales and hold back inventory for those expensive last-minute expensive purchases).

As the flight ticket price is fluctuating, we have to work on a project where you collect data of flight fares with other features and work to make a model to predict fares of flights.

Conceptual Background of the Domain Problem

Air travel has evolved into a popular, necessary, and quick mode of transportation thanks to the ever-increasing interconnectedness of air routes around the globe. For airlines, predicting prices is a crucial yet difficult issue because prices are always fluctuating and are known to depend on a wide range of factors.

Given that more and more individuals are choosing to travel faster, flying has become an essential aspect of modern life. The cost of airline tickets fluctuates depending on a number of variables, including the scheduling, destination, and duration of the flight. Various times, including holidays or the festive season. Therefore, many people will undoubtedly benefit from saving money and time by having a basic understanding of the flight costs before to planning the trip.

With extensive research in the domain, it has been found that an estimation of flight costs at a given time can be obtained within seconds utilising machine learning and artificial intelligence approaches.

Review of Literature

Flight Price Prediction for Users by Machine Learning Techniques (2021),

Pavithra Maria K, Anitha K L

People that take flights frequently will be more knowledgeable about the greatest deals and the best times to purchase tickets. Many airline firms adjust their fares in accordance with the seasons or time periods for commercial reasons. When more people travel, the cost will go up. Data for the route is gathered using features like Duration, Source, Destination, Arrival, and Departure to estimate the highest airline fares. In this study, authors have employed machine learning methods and regression methodologies to predict the price where the cost of an airline ticket varies over time. Features were extracted from a chosen dataset. They have created decision tree and random forest algorithms for flight price prediction for users. For the statistical analysis, this study as also conducted correlation tests and ANOVA test.

Aircraft Ticket Price prediction using Machine Learning (2022), Janhvi Mukane , Siddharth Pawar , Siddhi Pawar , Gaurav Muley

A comprehensive investigation was conducted for this research using dataset collected from Kaggle, and the Random Forest Machine Learning model was applied. The study was able to identify the factors that have the most effects on airline ticket pricing using visualisation. The results of experimental study show that the Random Forest Regression model has good accuracy.

As a long-term or future goal, the objective is to improve model accuracy and feature selection. In order to obtain more accurate airfares, authors also intend to expand the study by working with larger datasets and doing additional experiments on it. This will enable users to get an idea of the cost of their upcoming flights and enable them to negotiate the best price.

Flight Fare Prediction System (2021), Vinod Kimbhaune, Harshil Donga, Asutosh Trivedi, Sonam Mahajan and Viraj Mahajan

The idea of this research is to create an application that uses machine learning to forecast flight costs for various flights. The user will receive the expected values, and using these as a guide, they may determine how to purchase their tickets. A proper implementation of this project can help the travellers by giving them information on the patterns that flight costs follow and projected value of the price, which they can use to determine whether to book a ticket now or later, a properly executed version of this project can enable unskilled people to save money.

Motivation for the Problem Undertaken

To increase their profitability, airline companies currently attempt to influence the cost of airline tickets. Many people take flights frequently, so they are aware of the optimum times to get affordable tickets. However, a lot of people who are inexperienced in this and who is not a regular traveller are finding up falling into the companies' discount traps and wind up spending more money than they should have. By providing clients with the knowledge to purchase tickets at the appropriate moment, the proposed system can help them save millions of rupees.

Anyone who has booked a flight ticket knows how unexpectedly the prices vary. The cheapest available ticket on a given flight gets more and less expensive over time. This usually happens as an attempt to maximize revenue based on –

1. Time of purchase patterns (making sure last-minute purchases are expensive)
2. Keeping the flight as full as they want it (raising prices on a flight which is filling up in order to reduce sales and hold back inventory for those expensive last-minute expensive purchases)

Prediction-based services are currently employed in a wide range of industries. Examples include stock price forecasting tools used by stockbrokers and services like Zestimate, which provides an estimated property price worth. As a reason, the aviation sector needs a service like this one that can assist clients in making reservations.

With the help of data science, machine learning and Artificial Intelligence, we can build a powerful model that can predict the price of flight fare using the various factors impacting the ticket price such as airline, source, destination, time of flight, duration, class, time of booking etc. Since the flight price is always changing due to various factors, a machine learning model which has an option to recalibration would be a best way as it can save time and cost for all the travellers.

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

The study is divided as three phases: -

1. Data Collection
2. Data Analysis
3. Model Building

We have to collect the data which are required for making the analysis and building a predictive model.

With this dataset, we have to draw inferences using analysis of variables and characteristics of the dataset such as: -

Do airfares change frequently? Do they move in small increments or in large jumps? Do they tend to go up or down over time? What is the best time to buy so that the consumer can save the most by taking the least risk? Does price increase as we get near to departure date? Is Indigo cheaper than Jet Airways? Are morning flights expensive?

After collecting the data, you need to build a machine learning model. Before model building do all data pre-processing steps. Try different models with different hyper parameters and select the best model.

Data Sources and their formats

We had to scrape minimum 1500 records of flight journeys including the ticket price from websites. There was no limit to the number of columns. Generally, these columns include, are airline name, date of journey, source, destination, route, departure time, arrival time, duration, total stops, and the target variable price.

We have scraped flight ticket records from the website www.ixigo.com. The data was collected for the popular cities: -

New Delhi Mumbai Chennai Bangalore Kochi

We have collected 8700 records for flight journeys including the following information: -

Airline Name	Date of Journey	Source	Destination	Duration
Departure Time	Arrival Time	Total Stops	Ticket Price	

The data was collected for the month of August 2022 and the data collected was on 20 July 2022.

Airline_name	Date_of_journey	Source	Destination	Dep_time	Arr_time	Duration	Total_stops	Ticket_price
AIR INDIA	Mon, 01 Aug	New Delhi	Mumbai	10:00	12:35	2hr 35min	non-stop	5955
AIR INDIA	Mon, 01 Aug	New Delhi	Mumbai	21:15	23:35	2hr 20min	non-stop	5955
AIR INDIA	Mon, 01 Aug	New Delhi	Mumbai	14:00	16:15	2hr 15min	non-stop	5955
GO FIRST	Mon, 01 Aug	New Delhi	Mumbai	05:20	15:10	9hr 50min	1 stop	7108
GO FIRST	Mon, 01 Aug	New Delhi	Mumbai	17:05	23:55	6hr 50min	1 stop	7108
...
AIRASIA INDIA	Mon, 29 Aug	Cochin	Bengaluru	22:55	00:05	1hr 10min	non-stop	3494
ALLIANCE AIR	Mon, 29 Aug	Cochin	Bengaluru	15:30	17:55	2hr 25min	1 stop	3494
INDIGO	Mon, 29 Aug	Cochin	Bengaluru	05:45	06:45	1hr	non-stop	4123
AIR INDIA	Mon, 29 Aug	Cochin	Bengaluru	20:10	07:50	11hr 40min	1 stop	9059
AIR INDIA	Mon, 29 Aug	Cochin	Bengaluru	20:10	18:20	22hr 10min	1 stop	9059

Dataset

- We have 8700 rows and 9 columns in the dataset.
- We have string and integer type of data in the dataset.
- We have 8700 non null values in all the columns of the dataset.

Description of the dataset

Features in Dataset (Independent Variable)

Airline_name = Name of the airline

Date_of_journey = Date of the journey

Source = Source place of the journey

Destination = Destination place of the journey

Dep_time = Departure time

Arr_time = Arrival Time

Duration = Total travel duration

Total_stops = Total stops

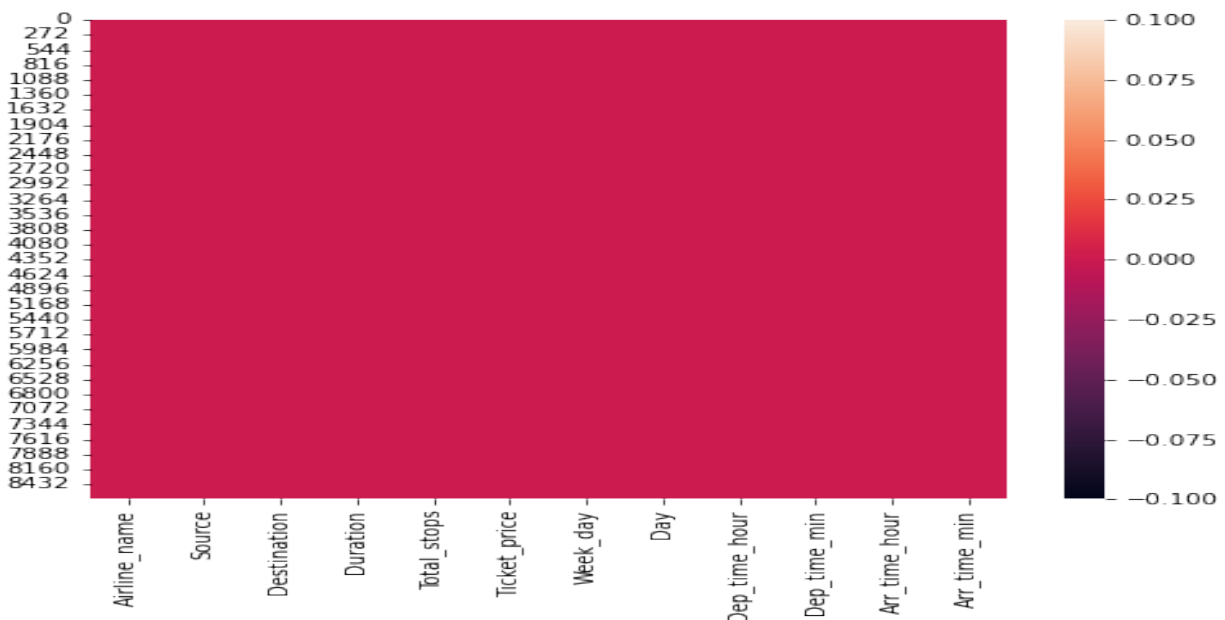
Target in dataset (Dependent Variable)

Ticket_price = Price of Flight Ticket

Data Pre-processing

- We have extracted the day of month and weekday from the column 'Date_of_journey' and added them as new column.
- We have extracted the hour and minute from the columns 'Dep_time' and 'Arr_time' and added them to dataset as new columns.
- We also converted the values in the column 'Duration' to minutes.

Checking for Missing Values



Heatmap for checking missing values in the dataset.

We don't have any missing values in the dataset.

Encoded the Categorical Columns

Since our data is having categorical variables, we have to encode them. Because machine will be able to understand numerical values for making better predictions.

```
from sklearn.preprocessing import OrdinalEncoder
onc = OrdinalEncoder()
for i in data.columns:
    if data[i].dtypes == 'object':
        data[i] = onc.fit_transform(data[i].values.reshape(-1,1)).astype('int64')    #Ordinal Encoding for the features.
```

Encoding the Categorical Variables

Data Cleansing

Removing the skewness with power transform

Only the column 'Duration' had skewness which was beyond the standard limit.

```
: #We can set the skewness standard limit as +/-0.5.  
x.drop(categorical_columns,axis =1).skew().sort_values(ascending=False)[np.abs(x.skew())>0.5]  
: Duration      4.6685
```

Columns having skewness which was beyond standard limit

After reducing the skewness using power transform, our data was like: -

```
Duration      0.097467  
Arr_time_min  0.052444  
Dep_time_hour 0.026863  
Day           0.000000  
Dep_time_min -0.014145  
Arr_time_hour -0.338474  
...
```

Skewness after power transform

Removed Outliers in the datasets

After reducing the skewness, we are not having any outlier data in the dataset.

Checked and Removed Multicollinearity from the Datasets.

	Column Name	VIF Factor
0	Duration	2.667756
9	Total_stops	2.635722
7	Source	1.228518
8	Destination	1.174500
6	Airline_name	1.149210
4	Arr_time_hour	1.138275
2	Dep_time_hour	1.110947
5	Arr_time_min	1.030024
3	Dep_time_min	1.025358
1	Day	1.018179
10	Week_day	1.012418

Variance of Inflation of Variables

We can see that the variance of inflation is least for the columns. That means there is not much multicollinearity present between the variables in the dataset.

Final Dataset

	Airline_name	Source	Destination	Duration	Total_stops	Ticket_price	Week_day	Day	Dep_time_hour	Dep_time_min	Arr_time_hour	Arr_time_min
0	0	4	3	155	2	5955	1	1	10	0	12	35
1	0	4	3	140	2	5955	1	1	21	15	23	35
2	0	4	3	135	2	5955	1	1	14	0	16	15
3	3	4	3	590	0	7108	1	1	5	20	15	10
4	3	4	3	410	0	7108	1	1	17	5	23	55
...
8695	1	2	0	70	2	3494	1	29	22	55	0	5
8696	2	2	0	145	0	3494	1	29	15	30	17	55
8697	4	2	0	60	2	4123	1	29	5	45	6	45
8698	0	2	0	700	0	9059	1	29	20	10	7	50
8699	0	2	0	1330	0	9059	1	29	20	10	18	20

8700 rows × 12 columns

Final Dataset after data cleansing and pre-processing

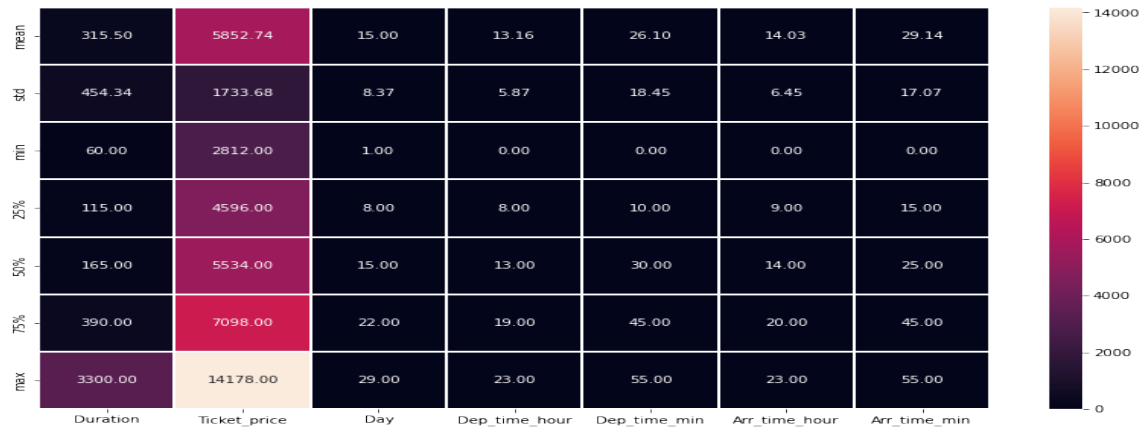
Data Inputs- Logic- Output Relationships

Statistical Summary

Describe of the data

	Duration	Ticket_price	Day	Dep_time_hour	Dep_time_min	Arr_time_hour	Arr_time_min
count	8700.000000	8700.000000	8700.000000	8700.000000	8700.000000	8700.000000	8700.000000
mean	315.497126	5852.737356	15.000000	13.157356	26.104023	14.034713	29.141954
std	454.339428	1733.682920	8.367081	5.866758	18.452366	6.453344	17.067180
min	60.000000	2812.000000	1.000000	0.000000	0.000000	0.000000	0.000000
25%	115.000000	4596.000000	8.000000	8.000000	10.000000	9.000000	15.000000
50%	165.000000	5534.000000	15.000000	13.000000	30.000000	14.000000	25.000000
75%	390.000000	7098.000000	22.000000	19.000000	45.000000	20.000000	45.000000
max	3300.000000	14178.000000	29.000000	23.000000	55.000000	23.000000	55.000000

Statistical Summary

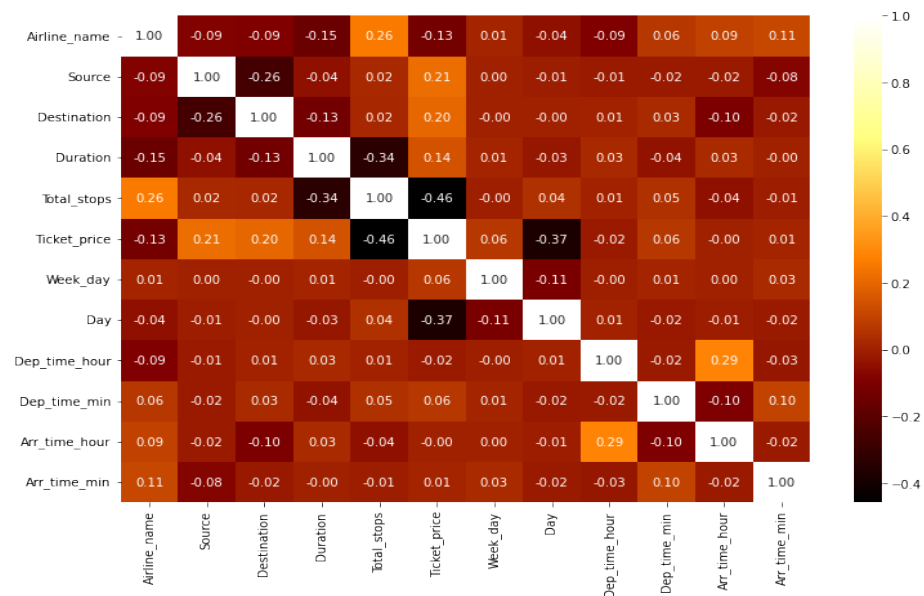


Heatmap of Describe of Data

Observations:

- All the columns except the columns ['Day', 'Dep_time_min'] are having higher mean value than the median. That means the distribution of values in these columns are not normal and skewness is present in the data distribution.
- The max value of the columns ['Duration', 'Ticket_Price'] are having huge difference between the 75%. Possible outliers are present in the data of these columns.

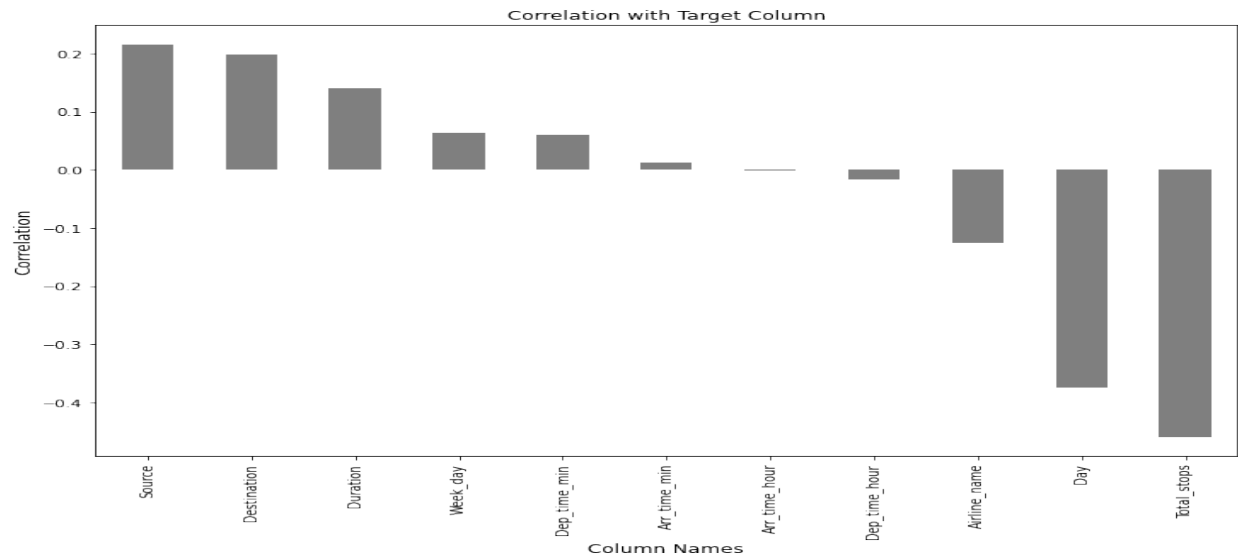
Correlation



Heatmap of Correlation between variables

Most of the columns are having moderate to least correlation to each other.

Correlation with Target Variable



Correlation with Target Variable

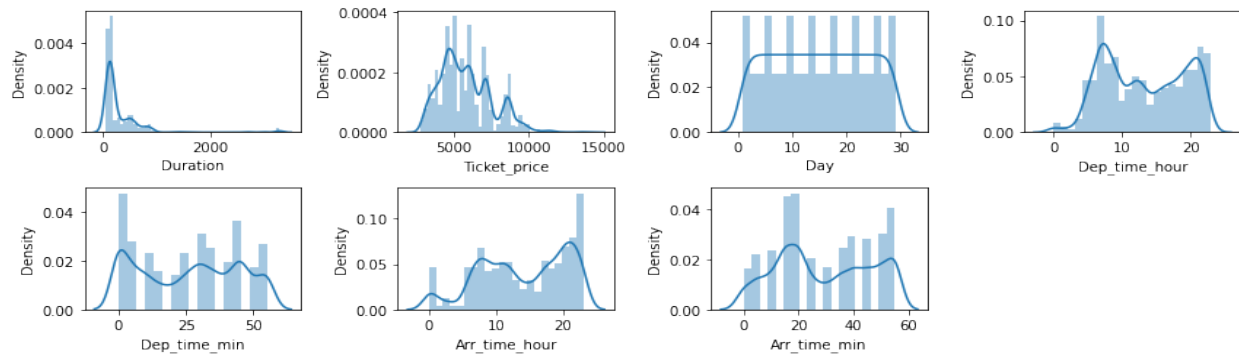
Source	0.214695
Destination	0.197541
Duration	0.139878
Week_day	0.063736
Dep_time_min	0.060005
Arr_time_min	0.012615
Arr_time_hour	-0.001085
Dep_time_hour	-0.017314
Airline_name	-0.125710
Day	-0.374815
Total_stops	-0.458598

Observations:

- The columns ['Source', 'Destination', 'Duration', 'Week_day', 'Dep_time_min', 'Arr_time_min'] are positively correlated to the target variable 'Ticket_price', while the rest of the columns in the dataset are having negative correlation to the target variable 'Ticket_price'.
- The column 'Source' is having highest positive correlation to the target variable whereas the column 'Total_stops' is having highest negative correlation to the target variable 'Ticket_price'.
- The column 'Arr_time_min' is having the least positive correlation to the target variable 'Ticket_price', while the column 'Arr_time_hour' is having the least negative correlation to the target variable 'Ticket_price'.

Assumptions Related to the Problem Under Consideration

Data Distribution (Skewness)

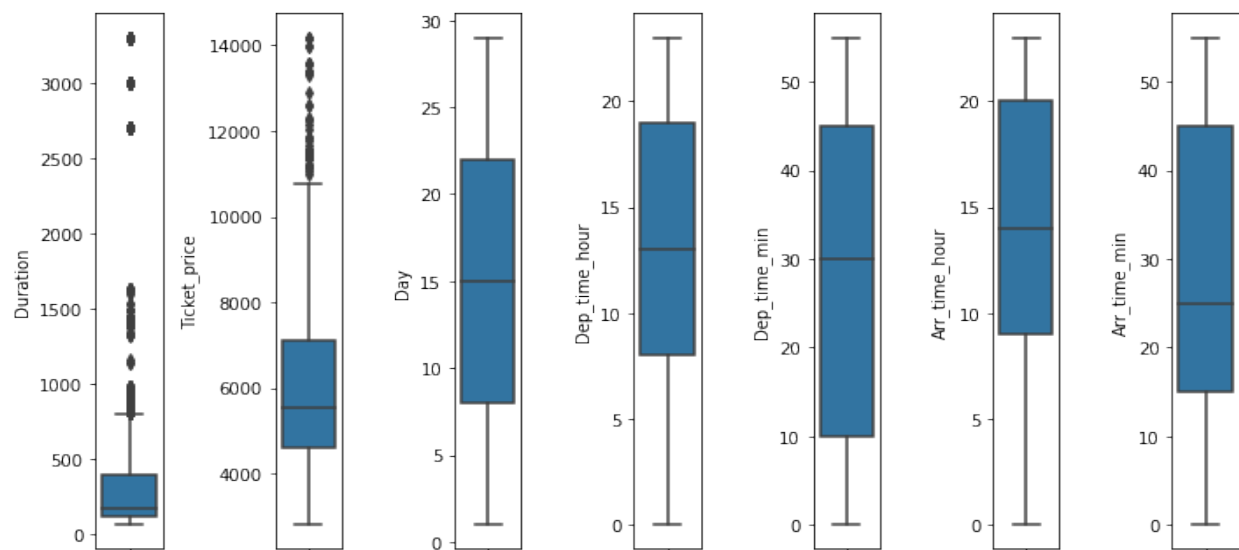


Data Distribution

Observations:

- The columns except the column 'Day' are not having normal distributions.
- Skewness is present in the data of all numerical columns except the column 'Day'.

Outliers



Presence of outliers in the dataset using boxplot

- The columns ['Duration', 'Ticket_price'] are having outliers present in the data.
- The column 'Ticket_price' is our target variable, so we will not consider the outliers in this column as it is required for building our model.

Hardware and Software Requirements and Tools Used

Hardware Requirement:

```
System Manufacturer: Dell Inc.  
System Model: Inspiron 5520  
BIOS: A17  
Processor: Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz (4 CPUs), ~2.5GHz  
Memory: 8192MB RAM  
Page file: 10586MB used, 2993MB available  
DirectX Version: DirectX 12
```

Hardware Configuration

Software Requirements:

- Windows Version: Windows 10 Pro
- Anaconda Navigator: 2.0.3
Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine.
- Jupyter Notebook: 6.3.0
Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine.
- Python3: Python 3.9.9
Python3 is used as the base environment for performing the machine learning and data analysis.

Python Libraries Used:

- Pandas: Data manipulation and analysis
- NumPy: Adding support for large, multi-dimensional arrays and matrices, along with an enormous collection of high-level mathematical functions to operate on these arrays.
- Matplotlib, Seaborn: For visualization of variable relations and data distribution, and analysis.
- Sklearn: Simple and efficient tools for predictive data analysis.
- SciPy: SciPy provides algorithms for optimization, integration, interpolation, eigenvalue problems, algebraic equations, differential equations, statistics, and many other classes of problems.
- Statsmodels: Provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration.
- Xgboost, catboost, lightgbm: Gradient boosting framework that uses tree-based learning algorithms.
- Pickle: Implements binary protocols for serializing and de-serializing

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods)

```
from sklearn.linear_model import LinearRegression
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import AdaBoostRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.linear_model import SGDRegressor
from sklearn.ensemble import ExtraTreesRegressor
from xgboost import XGBRegressor
from catboost import CatBoostRegressor
from lightgbm import LGBMRegressor

from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

from sklearn.linear_model import Lasso, Ridge, ElasticNet #Regularization technique

from sklearn.model_selection import train_test_split
```

Importing required libraries for building the model

```
lr = LinearRegression()
knn = KNeighborsRegressor()
dtr = DecisionTreeRegressor()
rfr = RandomForestRegressor(n_estimators=100)
abr = AdaBoostRegressor()
gbr = GradientBoostingRegressor()
sgd = SGDRegressor()
etr = ExtraTreesRegressor()
xgb = XGBRegressor()
lgbmr = LGBMRegressor()
cbr = CatBoostRegressor(verbose=0, n_estimators=100)
```

Creating the instances for the algorithms

We created three functions for testing the model and for cross validations:

- best_ran: Finding the best random state for the selected model
- mod_test: Training the model with the train data using the best random state.
- cross_val: Finding the best cross validation mean score for each model.

Testing of Identified Approaches (Algorithms) and evaluation of selected models

```
#User defined function for finding the best random state
def best_ran(model):
    maxacc = 0
    maxrs = 0
    print(model)
    for i in range(1,100):
        features_train, features_test, target_train, target_test = train_test_split(features, target, test_size = 0.20, random_state = i)
        model.fit(features_train, target_train)
        pred_test = model.predict(features_test)
        acc = r2_score(target_test, pred_test)
        if acc > maxacc:
            maxacc = acc
            maxrs = i
    print("At random state ", maxrs, 'the model is having r2 score of ', maxacc)
```

Code Snippet for function to find best random state

```
#User defined Function for training and testing the model with best random state
def mod_test(model, ran):
    model
    print(model)
    features_train, features_test, target_train, target_test = train_test_split(features, target, test_size = 0.20, random_state = ran)
    model.fit(features_train, target_train)
    pred_test = model.predict(features_test)
    acc = r2_score(target_test, pred_test)
    mse = mean_squared_error(target_test, pred_test)
    mae = mean_absolute_error(target_test, pred_test)
    print("R2 score is ", acc)
    print("_"*50)
    print("Mean Squared Error is ", mse)
    print("_"*50)
    print("Mean Absolute Error is ", mae)
    print("_"*50)
```

Code Snippet for function to test the model

➤ Linear Regression

LinearRegression()

R2 score is 0.527009516175881

Mean Squared Error is 1487823.3860520374

Mean Absolute Error is 878.2206749544725

Coefficient is [523.10653695 -590.72066241 -8.84896411 128.45766926 19.18291727
25.93514153 36.96868397 407.14767623 413.96665875 -395.61220262
29.44912273]

Intercept is 5843.46834689682

Testing the model performance

➤ KNeighbors Regressor

```
KNeighborsRegressor()  
R2 score is  0.8476736785594783  
  
Mean Squared Error is  441730.54232183914  
  
Mean Absolute Error is  421.3079310344827
```

Testing the model performance

➤ DecisionTree Regressor

```
DecisionTreeRegressor()  
R2 score is  0.9560811792433538  
  
Mean Squared Error is  126264.02586206897  
  
Mean Absolute Error is  108.40632183908046
```

Testing the model performance

➤ RandomForest Regressor

```
RandomForestRegressor()  
R2 score is  0.9697383109141254  
  
Mean Squared Error is  87000.57577913284  
  
Mean Absolute Error is  125.43368122605362
```

Testing the model performance

➤ AdaBoost Regressor

```
AdaBoostRegressor()  
R2 score is  0.6332858519329794  
  
Mean Squared Error is  1103702.4968724176  
  
Mean Absolute Error is  890.0885705475964
```

Testing the model performance

➤ GradientBoosting Regressor

```
GradientBoostingRegressor()  
R2 score is  0.8976894085335864  
  
Mean Squared Error is  312587.6736244026  
  
Mean Absolute Error is  403.7299161452732
```

Testing the model performance

➤ SGD Regressor

```
SGDRegressor()  
R2 score is  0.5253409909956601  
  
Mean Squared Error is  1493071.8442520385  
  
Mean Absolute Error is  879.1301802163991
```

Testing the model performance

➤ ExtraTrees Regressor

```
ExtraTreesRegressor()  
R2 score is  0.974804219841895  
  
Mean Squared Error is  72436.38564718391  
  
Mean Absolute Error is  108.23430459770115
```

Testing the model performance

➤ XGB Regressor

```
R2 score is  0.9692296414576231  
  
Mean Squared Error is  95324.12949473114  
  
Mean Absolute Error is  181.1027625774515
```

Testing the model performance

➤ LGBM Regressor

```
LGBMRegressor()  
R2 score is  0.9583050790946543  
  
Mean Squared Error is  127389.72711377613  
  
Mean Absolute Error is  225.41025266676056
```

Testing the model performance

➤ CatBoost Regressor

```
R2 score is  0.9636998893569437  
  
Mean Squared Error is  104360.68250619466  
  
Mean Absolute Error is  211.60298710390592
```

Testing the model performance

After testing the data with the regression algorithms, most of the models are performing well and providing the best R2 Score.

Cross Validation

```
#User defined function for checking cross validation for each model
from sklearn.model_selection import cross_val_score





def cross_val(model,ran):    #ran = random_state
    cv_mean = 0
    cv_fold = 0
    features_train, features_test, target_train, target_test = train_test_split(features,target,test_size = 0.20, random_state =
    model.fit(features_train,target_train)
    pred_test = model.predict(features_test)
    for j in range(2,10):
        cv_score = cross_val_score(model,features, target, cv = j)
        a =cv_score.mean()
        if a>cv_mean:
            cv_mean = a
            cv_fold = j
    print(model)
    print("At cv fold",cv_fold," the cv score is ", cv_mean, "and the R2 score is ",r2_score(target_test,pred_test))
```

Code Snippet for function to find the cross validation mean score

- Linear Regression
At cv fold 5 the cv score is 0.07036419325235899 and the R2 score is 0.527009516175881
- KNeighbors Regressor
At cv fold 6 the cv score is 0.3116648860583661 and the R2 score is 0.8476736785594783
- DecisionTree Regressor
At cv fold 5 the cv score is 0.3441344925813712 and the R2 score is 0.9594829955818878
- RandomForest Regressor
At cv fold 5 the cv score is 0.29547815277943074 and the R2 score is 0.9697482782184201
- AdaBoost Regressor
At cv fold 2 the cv score is 0.22613569274209 and the R2 score is 0.6190540964323343
- GradientBoosting Regressor
At cv fold 7 the cv score is 0.4492890940990274 and the R2 score is 0.8976894085335864
- SGD Regressor
At cv fold 5 the cv score is 0.06829978503296896 and the R2 score is 0.5275310053010887
- ExtraTrees Regressor
At cv fold 9 the cv score is 0.49297123686094024 and the R2 score is 0.9747227889478558
- XGB Regressor
At cv fold 7 the cv score is 0.4802321705474825 and the R2 score is 0.9692296414576231

- LGBM Regressor
At cv fold 5 the cv score is 0.3826905302044449 and the R2 score is 0.9583050790946543
- CatBoost Regressor
At cv fold 9 the cv score is 0.46628797931931476 and the R2 score is 0.9636998893569437

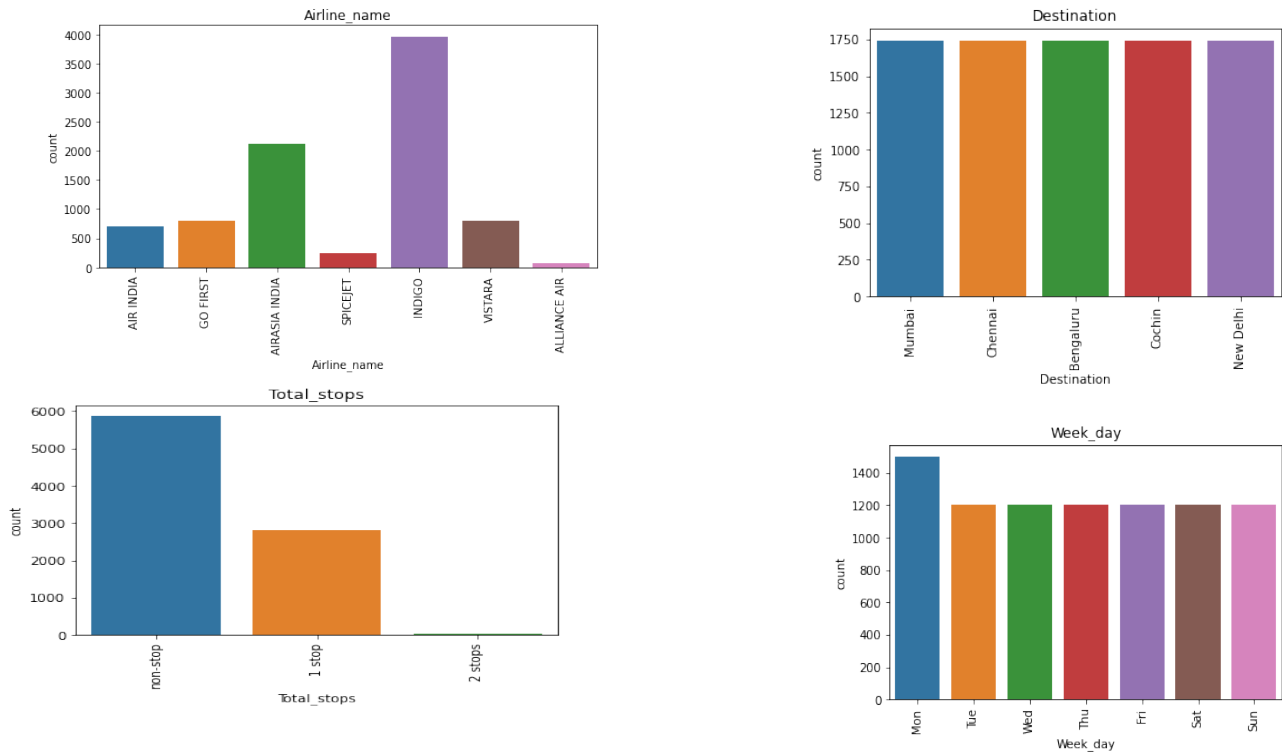
Key Metrics for success in solving problem under consideration

-  R2 Score
The proportion of the variance in the dependent variable that is predictable from the independent variable(s). It is the one of the key metrics for analysing the performance of regression models.
-  Mean Squared Error
It is the average of the square of the errors. The larger the number the larger the error. **Error** means the difference between the observed value and predicted values.
-  Mean Absolute Error
It is the average of difference between the measured value and “true” value.
-  Regularization
Regularization is a technique used to reduce the errors by fitting the function appropriately on the given training set and avoid overfitting.

Visualization

Univariate Analysis

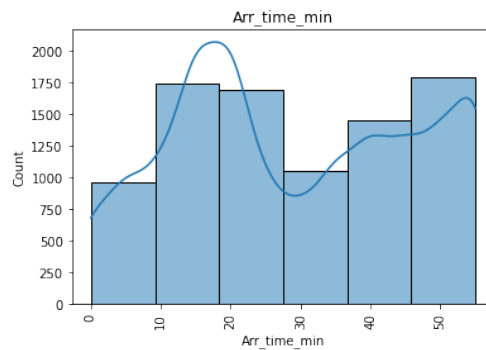
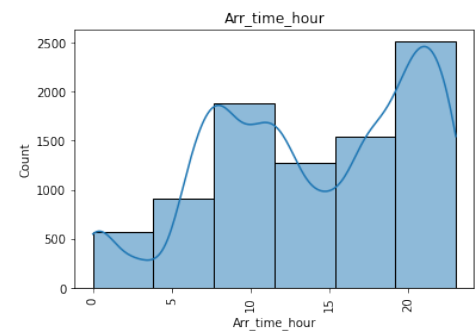
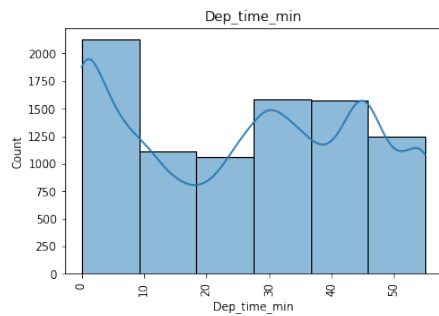
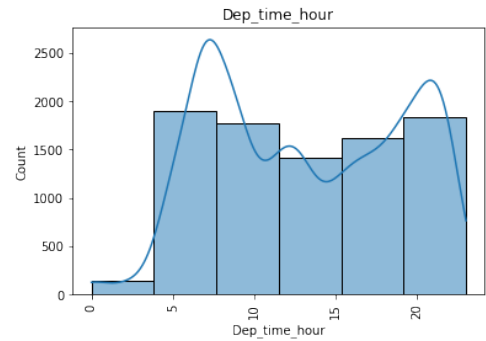
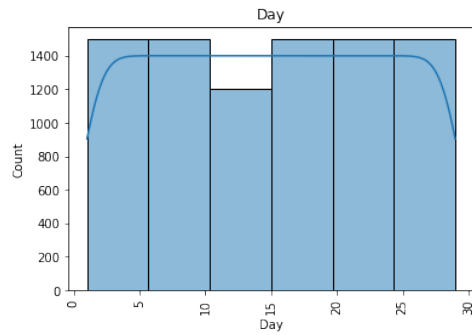
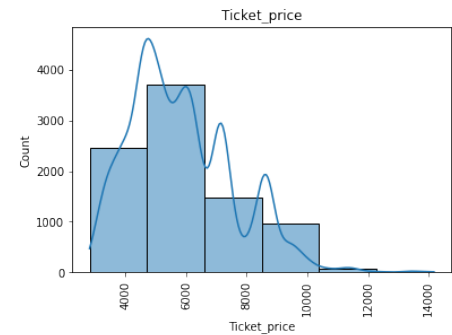
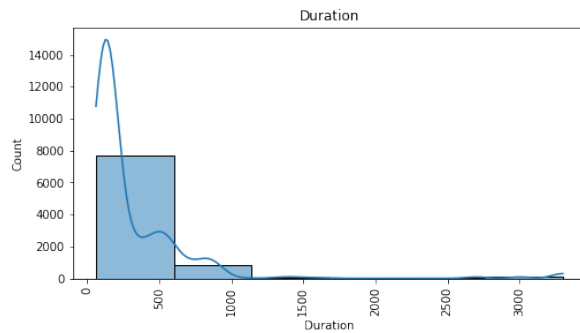
Categorical Variables – Using Countplot



Observations:

- Most of the flights in the dataset are of Indigo Airlines. Second highest flights in the dataset are of AirAsia India Airlines.
- Most of the flights are taking off from New Delhi airport while the least number of flights are taking off from the Mumbai Airport.
- We have 5 destination airports in the dataset. We don't have any difference in the number of data for the flights which are landing off in these cities.
- Most of the flights in the dataset are having no stop in between the journey.
- Most of the flights are having the journey on Monday compared to other weekdays.

Numerical Variables – Using histogramplot

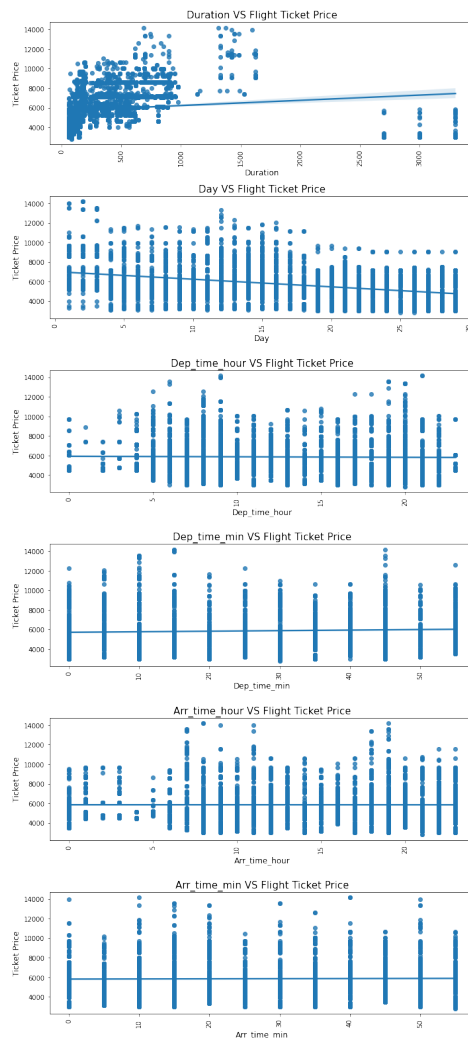


Observations:

- Most of the flights are having a duration time between 60 to 600 minutes.

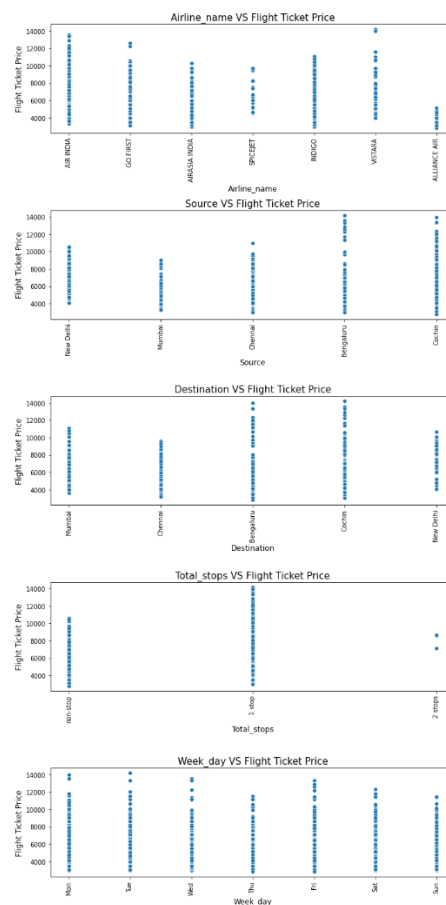
- Most of the flights tickets are having a flight fare between the range Rs 4700 to Rs. 6600.
- The number of flights which are having higher number of journeys are during the beginning and end of a month. The number of flights between the 10th and 15th day of month is comparatively less.
- The number of flights which are taking off between 12.00 AM to 4.00 AM is comparatively less compared to other hours during the day. Rest of the hours in a day are having similar number of flight departure time.
- Most of the flights are taking off in first 10 minutes of any hour for a day.
- Most number of flights are landing off between the 07.00 PM. to 11.59 PM for a day.
- As per the dataset, the number of flights which are landing off between 10 to 28 minutes and 45 to 59 minutes of an hour is more compared to other time intervals of any hour in a day.

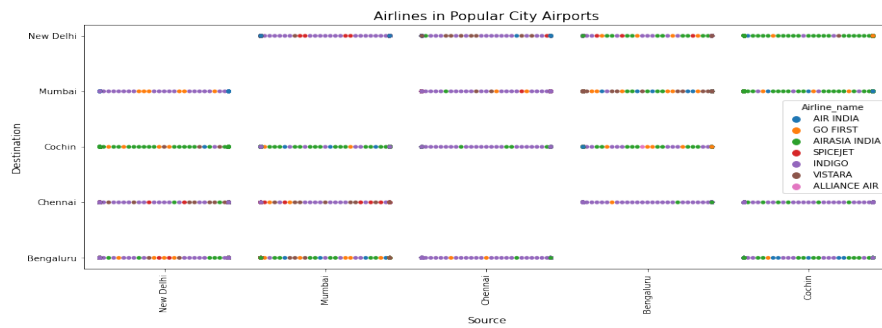
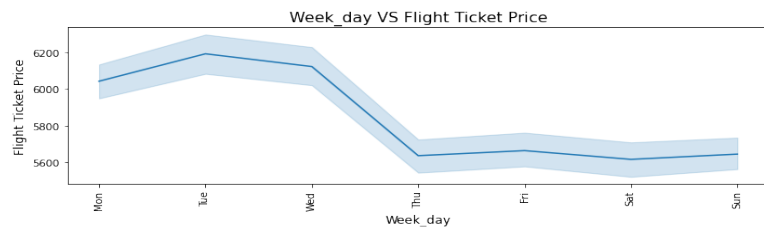
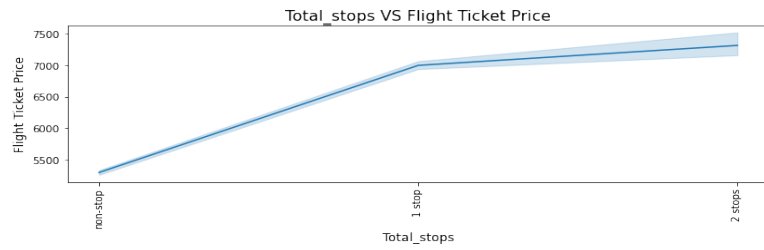
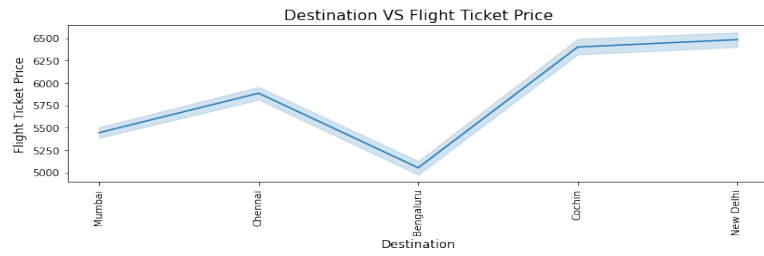
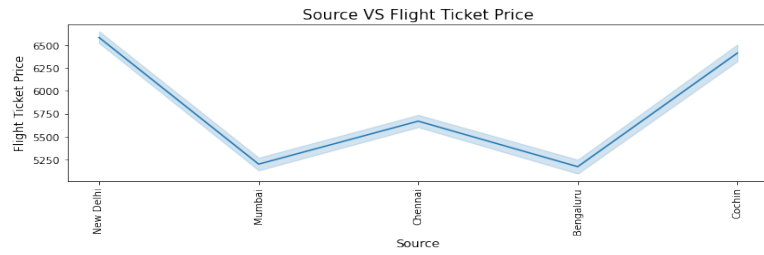
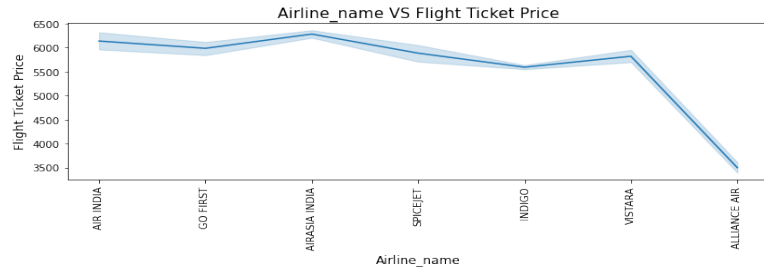
Bivariate and Multivariate Analysis



Observations:

- The flight journey duration is having a slight impact to the increase in flight ticket price. But the data is not uniform between each interval of duration.
- The day of flight journey in a month is having a slight impact to the decrease in flight ticket price. The flight ticket price is higher during the start of month compared to the flight ticket price end of month. So, it is better to travel in flights during the end of month than the beginning of the month.
- The flight ticket price is higher between 05.00 AM to 10.00 A.M and between 05.00 PM to 09.00 PM of the day. The flight price is low for the rest of the hours in a day. So, it is better to choose a flight which is not in the busy hours.
- The flight ticket price is higher for flights which are taking off at 10,15 and 45 minutes of an hour of any day
- The flight ticket price is higher if the flight is arriving at the destination between 06.00 AM to 11.00 AM and 06.00 PM to 07.00 PM of a day.
- The arriving time minute of a flight is not having much impact on the price of flight ticket as the price is almost uniform in different interval of minutes of hours in a day.





Observations:

- The flight ticket price is higher for the Vistara and Air India airlines while SpiceJet and Alliance Air flights are having comparatively less flight fare.
- Flights fare is higher for flight which are taking off and landing in Bengaluru and Cochin airports compared to other airports.
- The flight ticket price is higher for flights which are having one stop in between journey.
- The flight ticket price is high during the start of a week and decreasing to weekend.

Interpretation of the Results

- The columns ['Source', 'Destination', 'Duration', 'Week_day', 'Dep_time_min', 'Arr_time_min'] are positively correlated to the target variable 'Ticket_price', while the rest of the columns in the dataset are having negative correlation to the target variable 'Ticket_price'.
- The column 'Source' is having highest positive correlation to the target variable whereas the column 'Total_stops' is having highest negative correlation to the target variable 'Ticket_price'.
- The column 'Arr_time_min' is having the least positive correlation to the target variable 'Ticket_price', while the column 'Arr_time_hour' is having the least negative correlation to the target variable 'Ticket_price'.

Regularization

Lasso(L1)

```
ls = Lasso(alpha = 0.0001, random_state = 0)
ls.fit(features_train,target_train)
pred_ls = ls.predict(features_test)

lss = r2_score(target_test,pred_ls)
lss
```

0.4903414584081053

```
cross_val(ls,0)
```

Lasso(alpha=0.0001, random_state=0)
At cv fold 5 the cv score is 0.07036458248460671 and the R2 score is 0.48989237620341886

Lasso Regularization Performance

Ridge(L2)

```
rd = Ridge(alpha = 1, random_state=0)
rd.fit(features_train, target_train)
pred_rd = rd.predict(features_test)

rds = r2_score(target_test, pred_rd)
rds
```

0.490345950577325

```
cross_val(rd, 0)
```

Ridge(alpha=1, random_state=0)

At cv fold 5 the cv score is 0.07084509752008901 and the R2 score is 0.4898934948997812

Ridge Regularization Performance

ElasticNet

```
en = ElasticNet(alpha = 0.01, random_state = 0)
en.fit(features_train, target_train)
pred_en = en.predict(features_test)

ens = r2_score(target_test, pred_en)
ens
```

0.4904886443286951

```
cross_val(en, 0)
```

ElasticNet(alpha=0.01, random_state=0)

At cv fold 5 the cv score is 0.08619348059523609 and the R2 score is 0.48992260913781294

ElasticNet Regularization Performance

The regularization techniques didn't provide any better results.

Hyperparameter Tuning

ExtraTree Regressor

```
grid.best_score_  
0.4934526430504786
```

```
grid.best_params_  
{'criterion': 'squared_error',  
 'max_depth': None,  
 'max_features': 'log2',  
 'n_estimators': 150}
```

Best Score and Parameters

XGBoost Regressor

```
: grid.best_score_  
: 0.451804472852324  
  
: grid.best_params_  
: {'base_score': 1,  
  'eval_metric': 'mae',  
  'objective': 'reg:squarederror',  
  'seed': 0,  
  'seed_per_iteration': False}
```

Best Score and Parameters

CatBoost Regressor

```
grid.best_score_  
0.4173356610286454  
  
grid.best_params_  
{'eval_metric': 'RMSE',  
 'n_estimators': 100,  
 'sampling_frequency': 'PerTreeLevel',  
 'sampling_unit': 'Object'}
```

Best Score and Parameters

After all the tests, cross validations, regularizations and hyperparameter tuning the XGBoost model is performing well. So, we can consider this model as the best performing model.

Finalized Model Performance with Tuned Parameters

```
xgb = XGBRegressor(base_score = 1,eval_metric = 'mae',objective = 'reg:squarederror',seed = 0,seed_per_iteration = False)
features_train, features_test,target_train,target_test= train_test_split(features,target,test_size = 0.20, random_state = 17)
xgb.fit(features_train, target_train)
pred_test_xgb = xgb.predict(features_test)

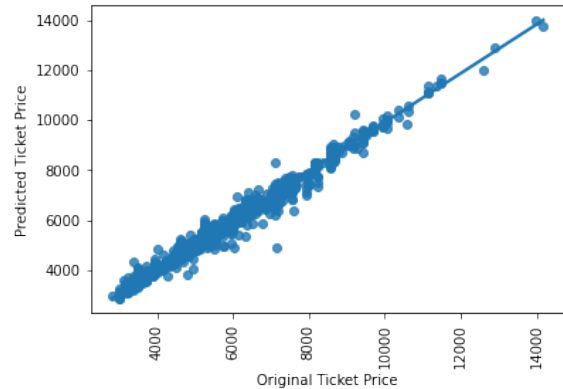
print('R2 Score',r2_score(target_test,pred_test_xgb))
print('Mean Squared Error',mean_squared_error(target_test,pred_test_xgb))
print('Mean Absolute Error',mean_absolute_error(target_test,pred_test_xgb))
```

R2 Score 0.9692296741063277
Mean Squared Error 95324.02835163505
Mean Absolute Error 181.10267642667924

Final Model Performance

Now we have trained our model and it is ready to test with the actual data to cross verify the performance.

	Original Ticket Price	Predicted Ticket Price
102	3208	3293.0
419	4018	3950.0
1424	5518	5480.0
319	3943	4427.0
1451	7116	7010.0
780	3975	4245.0
947	4505	4538.0
1588	3980	4201.0
403	9419	9386.0
1643	7022	6974.0
1399	6048	6068.0
805	6610	6616.0
687	7098	7062.0
1415	4469	4504.0
945	9418	9244.0
572	5689	5553.0
630	6043	6000.0
1661	5146	5138.0
24	7326	7009.0
1691	4998	5270.0



Model performance with predictions vs actual price

Model Predictions and Actual Price

Our model is performing well with predictions and provided almost accurate results. The XGBoost model(xgb) is providing a final R2 Score of 96.92%.

Saving the best model

```
import pickle

filename = 'flight_ticket_price_prediction_model.pkl'
pickle.dump(xgb, open(filename, 'wb'))
```

We have saved the machine learning model for future predictions. We have serialized and saved the binary file as “flight_ticket_price_prediction_model.pkl” using the pickle library.

CONCLUSION

Key Findings and Conclusions of the Study

With the help of data science and machine learning, we were able to create a machine learning model using XGBoost algorithm, which can predict the price of flight ticket price.

Now this model can be used to predict the price of flight ticket price in India with the following variable information about the flight journey and flight. (Important Variables)

Duration	Total_stops	Source	Destination	Airline_name
Arr_time_hour	Dep_time_hour	Arr_time_min		
Dep_time_min	Day	Week_day		

Impact of Variables on Target Variable (Correlation)

- The columns ['Source', 'Destination', 'Duration', 'Week_day', 'Dep_time_min', 'Arr_time_min'] are positively correlated to the target variable 'Ticket_price', while the rest of the columns in the dataset are having negative correlation to the target variable 'Ticket_price'.
- The column 'Source' is having highest positive correlation to the target variable whereas the column 'Total_stops' is having highest negative correlation to the target variable 'Ticket_price'.
- The column 'Arr_time_min' is having the least positive correlation to the target variable 'Ticket_price', while the column 'Arr_time_hour' is having the least negative correlation to the target variable 'Ticket_price'.

Learning Outcomes of the Study in respect of Data Science

According to a report, India's civil aviation industry is on a high-growth trajectory. India aims to become the largest aviation market by 2030. Cheap Air Tickets is the most often searched term in India, according to Google Trends. Additionally, as India's middle class becomes more accustomed to flying, the number of customers looking for bargains rises.

As the data science field is booming and the data is accumulating day by day, we can always rely on the data science, machine learning and artificial intelligence for making analysis and predictions on the flight ticket price which will help a large group of potential customers in saving their time and money. With the implementation of a powerful predictive model that can predict the flight fare, customers can plan their journeys by exploiting the maximum benefits from flight ticket booking.

Limitations of this work and Scope for Future Work

Limitations

- We tried to include as many features as possible which were available at the time of data collection. But still this is not enough to build a powerful model as the aviation market is always fluctuating and are affected by various factors.
- The data is missing other factors which can make an impact on the flight fare such as fuel price, class, weather, festival or seasons at source and destinations, time, and date of booking the flight etc.
- The source data was limited to one website and the data was only collected for one month. For building a powerful model, we have to include flight fare records for different timeframes and also different time which we are checking the flight fares.
- We had only included the major airports in India. There are other airports in India, and we also have to include international journeys which will increase the volume of the dataset.

Scope

- In future, to build a powerful predictive model for flight fare prediction, we can include a large dataset including much more important features that are making impact on the flight ticket price
- We should also consider international flight journey records and include more airlines from various sources and websites. Our model is scalable and can be improved by including more data and adding more features.
- As the data science is advancing, in future, we will be able to use much more powerful algorithms and approaches such as deep learning and artificial intelligence.

Thank You