

# *A Presentation on Flight Projects price predictions...*

**Submitted To:** Khusboo Garg  
(SME of Internship Batch no-28)

**Submitted By:** Abhishek Behera  
Internship Batch no-28



# INTRODUCTION

Business Requirement

# PROBLEM STATEMENT

The airline industry is considered as one of the most sophisticated industry in using complex pricing strategies. Nowadays, ticket prices can vary dynamically and significantly for the same flight, even for nearby seats. 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)



# PHASES OF THE PROJECT

This project is done in three parts:

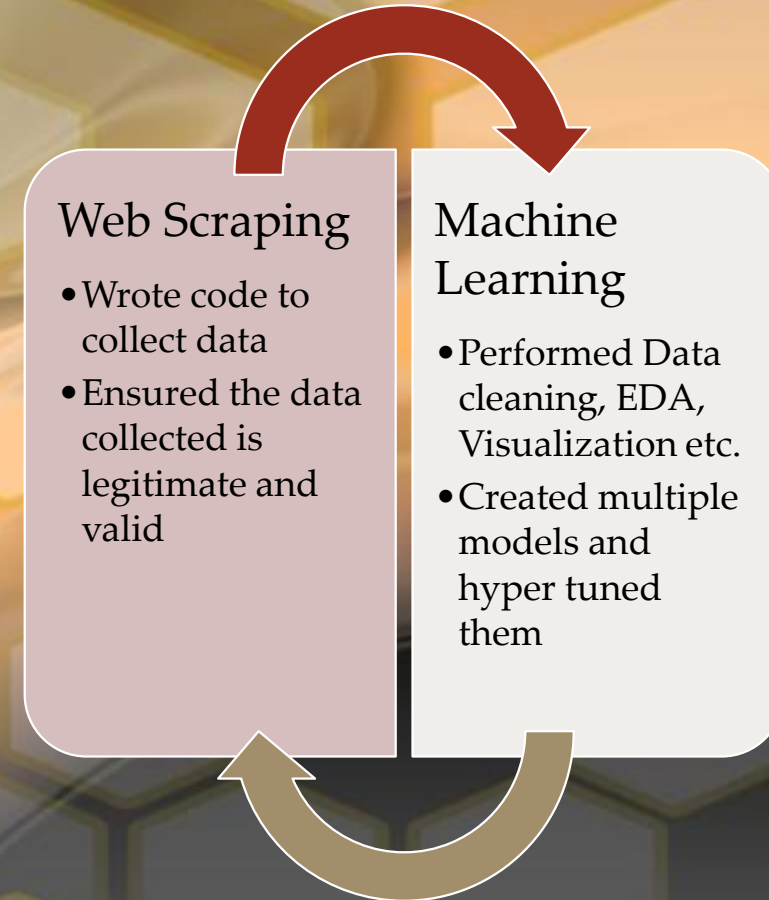
- Data Collection
- Data Analysis
- Model Building

I created two different Jupyter Notebook files to performed the required actions.

As per the requirement of client, I have scrapped the data from online sites and based on that data I have performed analysis like based on which feature of my data does flight prices change and checked the relationship of flight prices with all the other features to get a gist on what flight a passenger should choose.

# JUPYTER NOTEBOOK USAGE

- Used the Python programming in Jupyter Notebook for 2 separate files
- In the first notebook I wrote down the code to extract data for Flight prices and details from various web pages and stored them in a comma separated value file
- Then the second notebook was created to make a Flight Price Prediction project and analyze various ways to get better predicted results



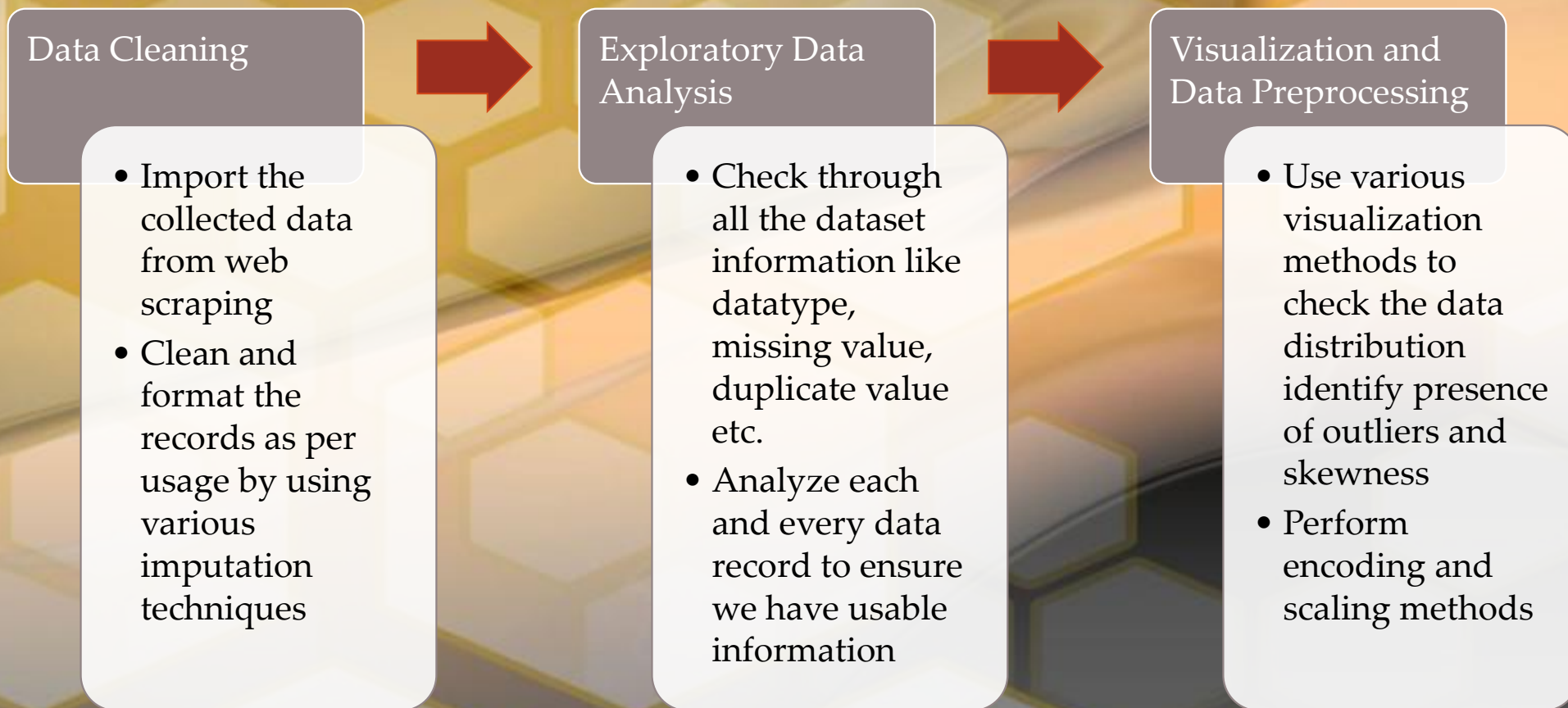
# MODEL BUILDING STEPS

1. Data Cleaning
2. Exploratory Data Analysis
3. Data Pre-processing
4. Model Building
5. Model Evaluation
6. Selecting the best model





# DATA SCIENCE LIFE CYCLE



# DATA SCIENCE LIFE CYCLE

## Model Building

- Create appropriate Regression Machine Learning model function
- Need to ensure that whenever the regression function is called it is able to process all the necessary parameters

## Model Evaluation



- Usage of evaluation metrics to check the accuracy of the models over trained and test data inputs
- Ensure the cross validation techniques helps in reducing over fitting and under fitting data

## Hyperparameter Tuning Best Model


- Choosing the appropriate Regression Machine Learning model to check various parameter permutation and combinations
- Using Grid Search CV to obtain the best parameters that can be plugged into the selected model



# WEB SCRAPING WEBPAGES FOR FLIGHTS

 | STAR ALLIANCE 

SEARCHFLIGHTPASSENGERSSEATSPAYMENT

 **SEARCH FLIGHTS**

English▼

☒ RETURN ☐ ONE WAY ☐ MULTI-CITY


FROM

Going From▼


TO

Going To▼

DEPART

Select Date

RETURN

Select Date


PASSENGERS

1 Passenger▼

ENTER PROMO CODE

Promotion Code

SEARCH FLIGHTS →

 **COVID ALERT**

For all passengers booking on this site,  
Please [Click Here](#) to go through undertaking  
requirements if not already done so prior to  
making your bookings as it is a mandatory  
requirement.

# DATA PREPROCESSING

- Importing the necessary dependencies and libraries.
- Reading the CSV file and converted into data frame.
- Checking the data dimensions for the original dataset.
- Looking for null values and accordingly renaming the values.
- Checking the summary of the dataset.
- Checking unique values.
- Checking all the categorical columns in the dataset.
- Ensuring that the values are good to use and discarding junk data.

# DATA PREPROCESSING

- Visualizing with the use of pandas profiling feature.
- Visualizing each features using matplotlib and seaborn.
- Performing encoding using the ordinal encoder on categorical features.
- Checking for co-relation/multi-collinearity in a heatmap.
- Checking for Outliers/Skewness using boxen plot and distribution plot.
- Checking for the final dimension of dataset to confirm the input details.
- Creating train test split and the best random state found in the range 1-1000.
- Taking a look at the importance of feature details to analyse further.



# TECHNOLOGY USED

- Hardware technology being used.

RAM : 8 GB

CPU : AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx 2.10 GHz

GPU : AMD Radeon™ Vega 8 Graphics and NVIDIA GeForce GTX 1650 Ti

- Software technology being used.

Programming language : Python

Distribution : Anaconda Navigator

Browser based language shell : Jupyter Notebook

- Libraries/Packages specifically being used.

Pandas, NumPy, matplotlib, seaborn, scikit-learn, pandas-profiling, missingno

# EXPLORATORY DATA ANALYSIS (EDA) AND VISUALIZATION

## 01. Univariate Analysis

**Univariate analysis** is the simplest form of analyzing data. “Uni” means “one”, so in other words your data has only one variable.

## 02. Multivariate Analysis

**Multivariate analysis** is a set of statistical techniques used for **analysis** of data that contain more than one variable.

## 03. Correlation of Dataset

**Correlation** is used to test relationships between quantitative variables or categorical variables.

## 04. Correlation with Target variable

**Correlation** with the target variable to know how the data is related.

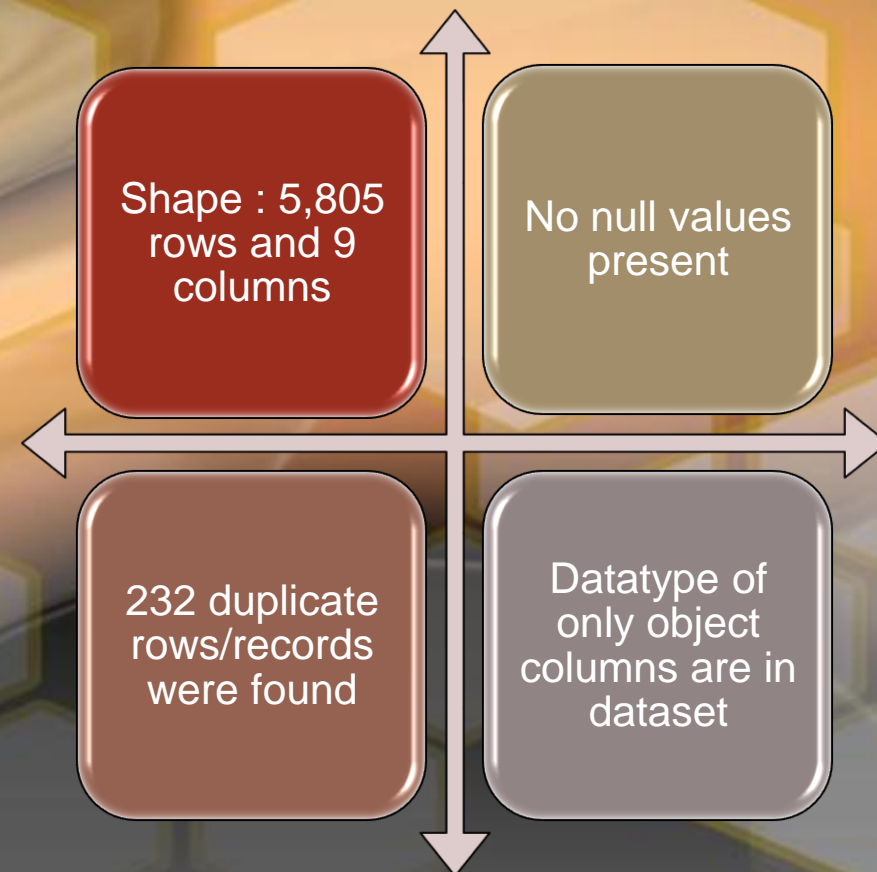
## 05. Conclusion

**Summary** with the conclusion of all the analysis



# EXPLORATORY DATA ANALYSIS (EDA)

- First I have imported the necessary libraries and loaded the entire dataset in our Jupyter Notebook and renamed the project file from untitled.
- Then I checked the shape of our dataset and found that we have a total of 5,805 rows and 9 different columns.
- We don't have any null values or missing values present in our dataset from the web scraping.
- There were 232 duplicate rows/records in our dataset but I decided to retain them instead of deleting it.
- By checking the data types I came to know that our data set consists of columns having only object datatype even those there were numeric information present.





Summarize dataset: 100%  28/28 [00:25<00:00, 1.38s/it, Completed]

Generate report structure: 100%  1/1 [00:06<00:00, 6.65s/it]

Render HTML: 100%  1/1 [00:04<00:00, 4.28s/it]

Pandas Profiling Report

Overview

Variables

Interactions

Correlations

Missing values

Sample

Duplicate rows

## Overview

Overview

Warnings **38**

Reproduction

### Dataset statistics

Number of variables	15
Number of observations	5805
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	232
Duplicate rows (%)	4.0%
Total size in memory	680.4 KiB
Average record size in memory	120.0 B

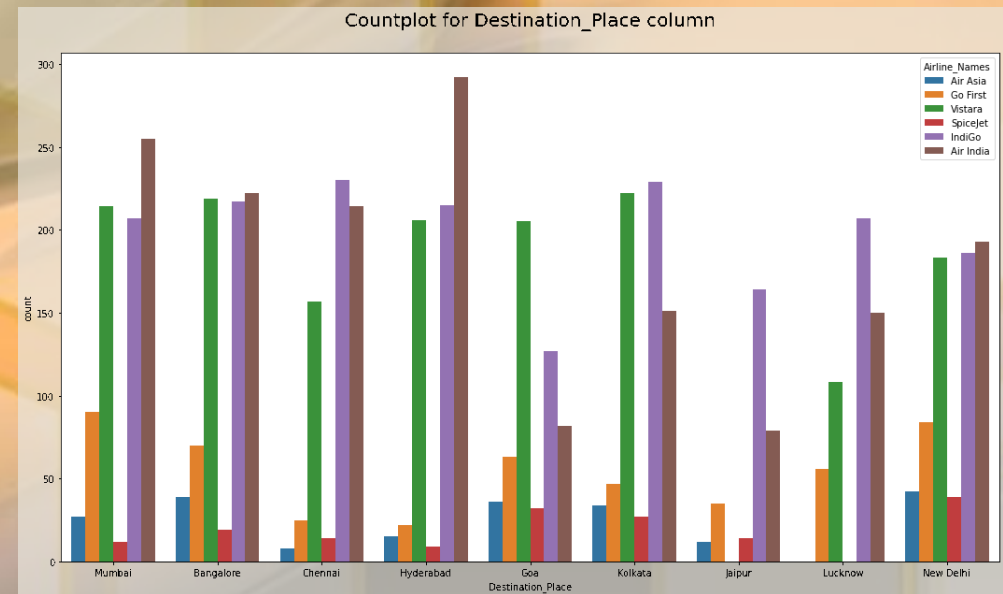
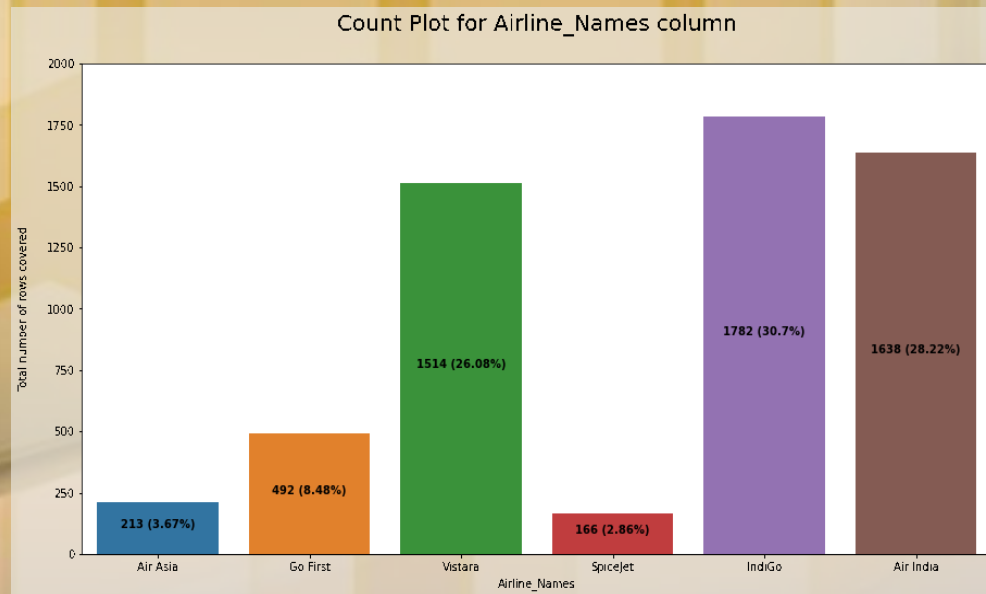
### Variable types

Categorical	5
Numeric	10

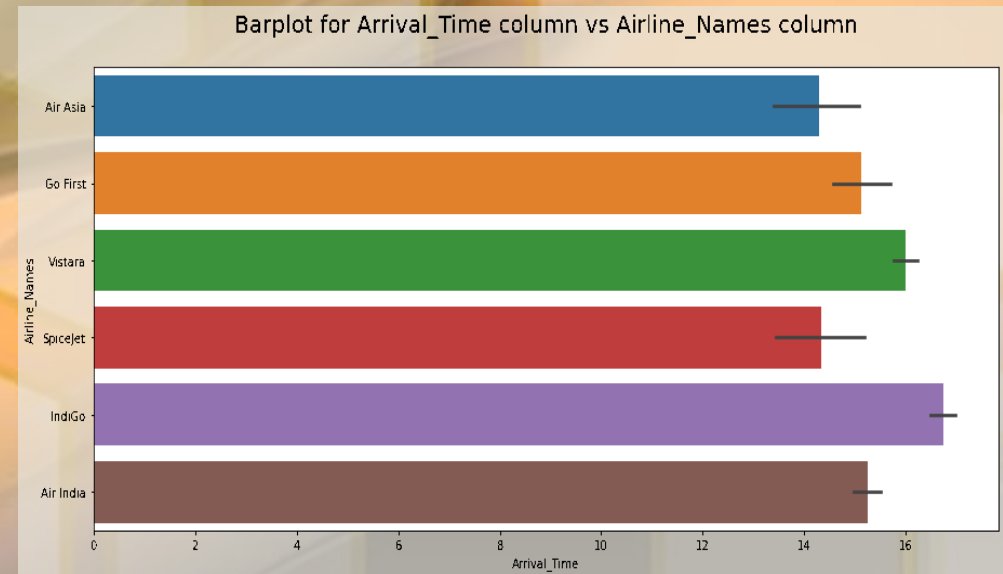
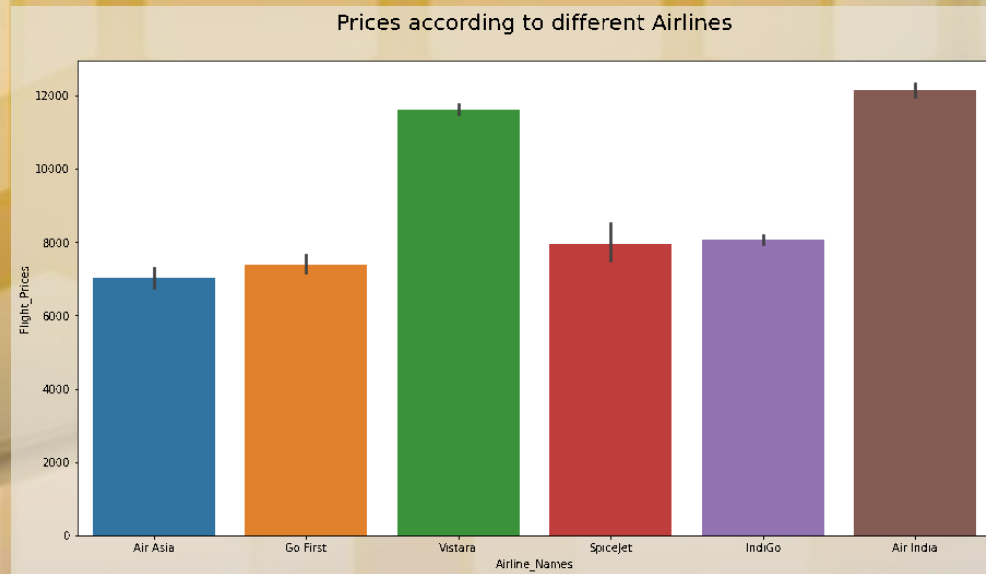
# VISUALIZATION USING PANDAS PROFILING REPORT

Here I have made use of pandas profiling to get a gist of my pre processed data and get a insight on the basic overview of my dataset values.

# COUNT PLOTS

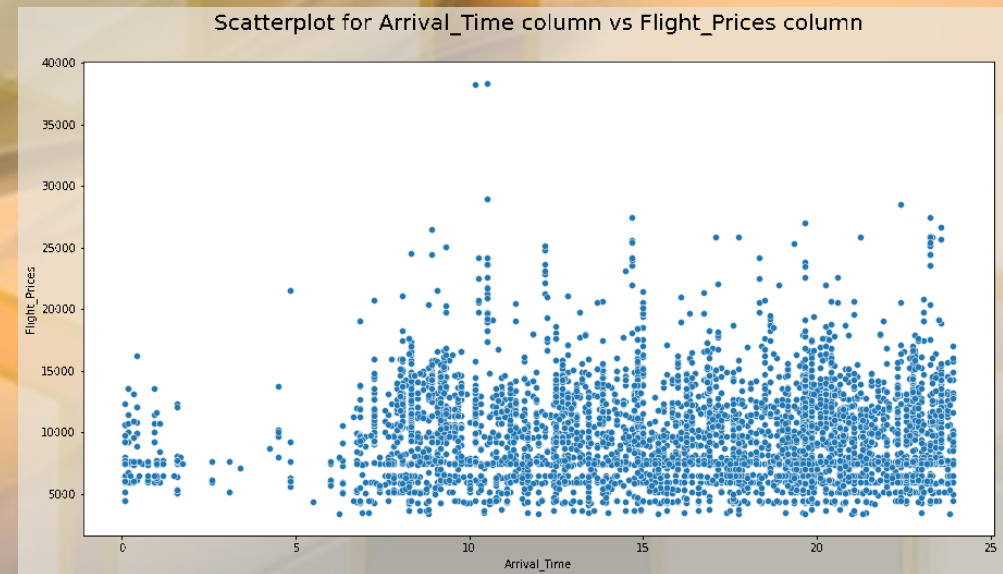
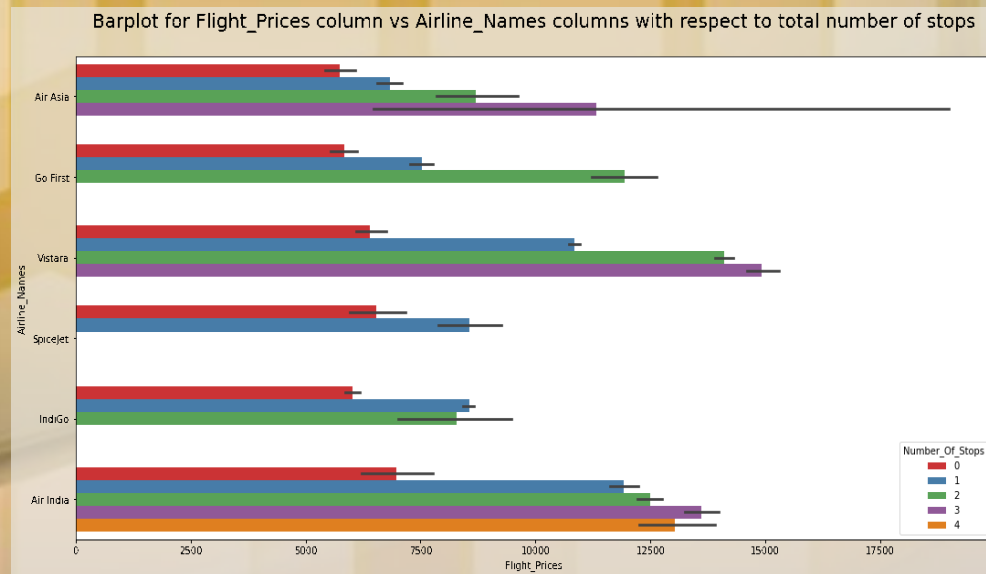


# BAR PLOTS





# BAR PLOTS AND SCATTER PLOTS

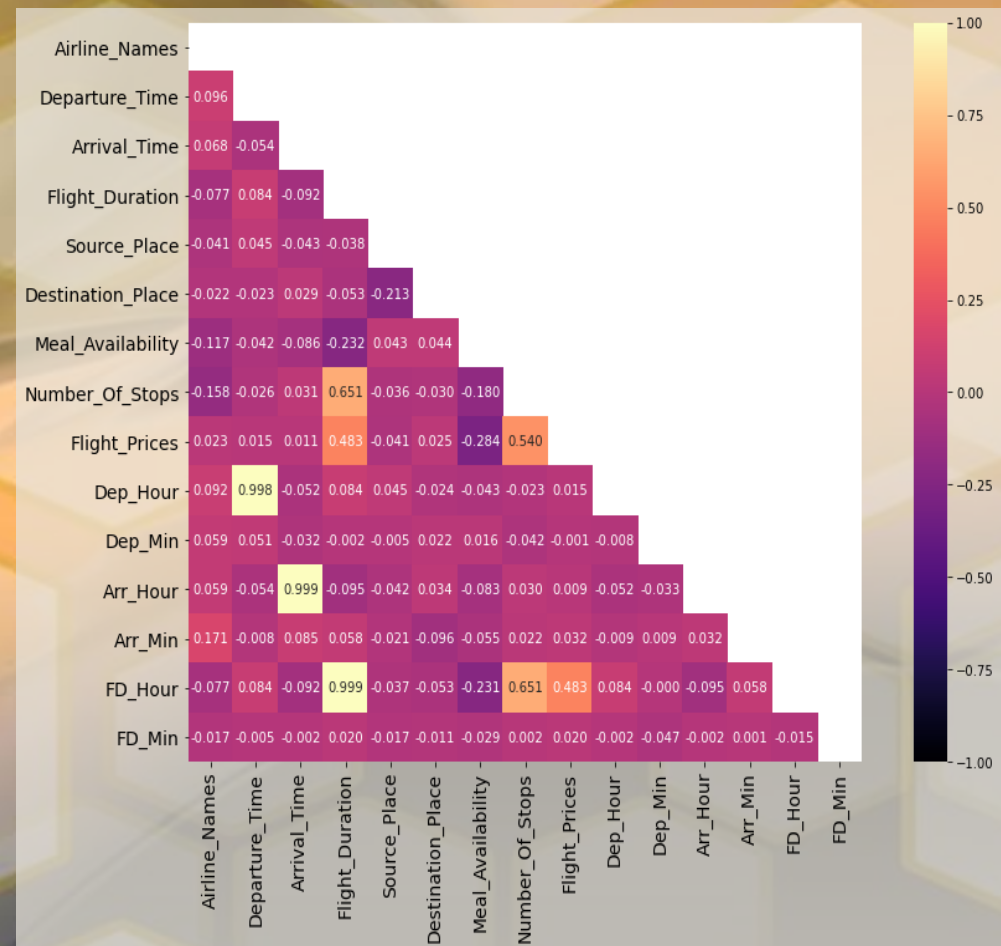
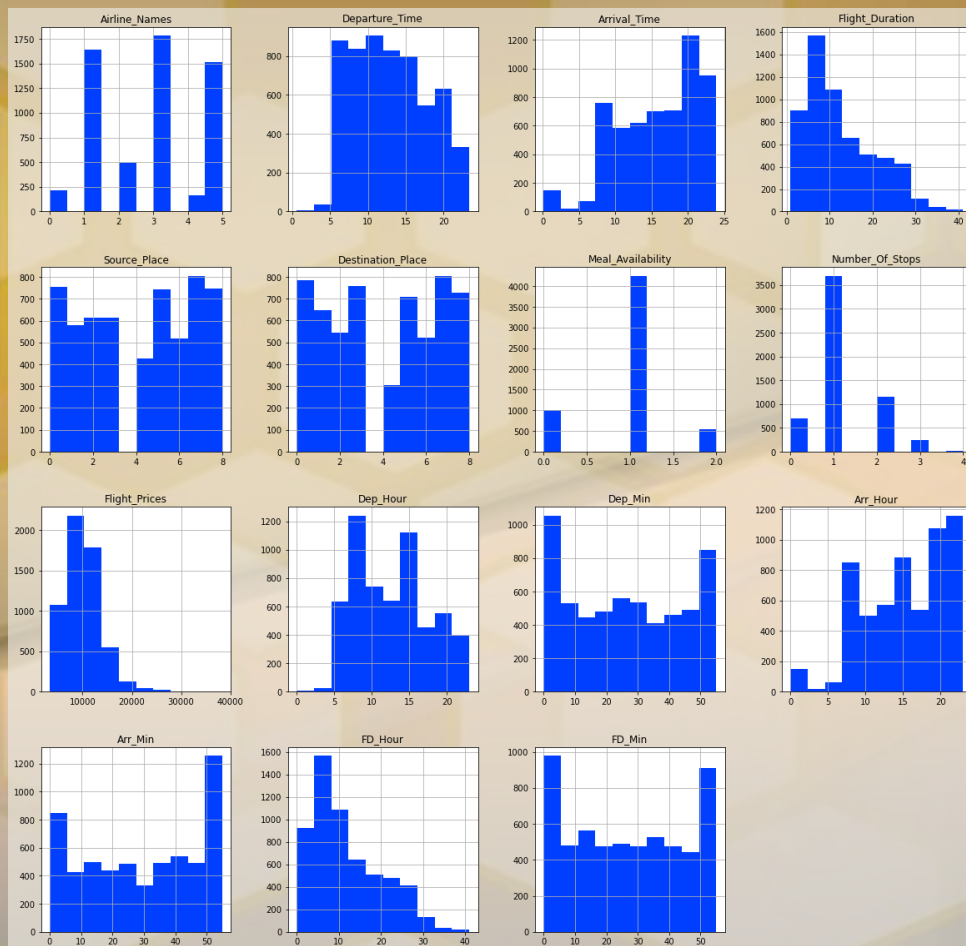


# MISSING VALUES AND DESCRIBE DATA



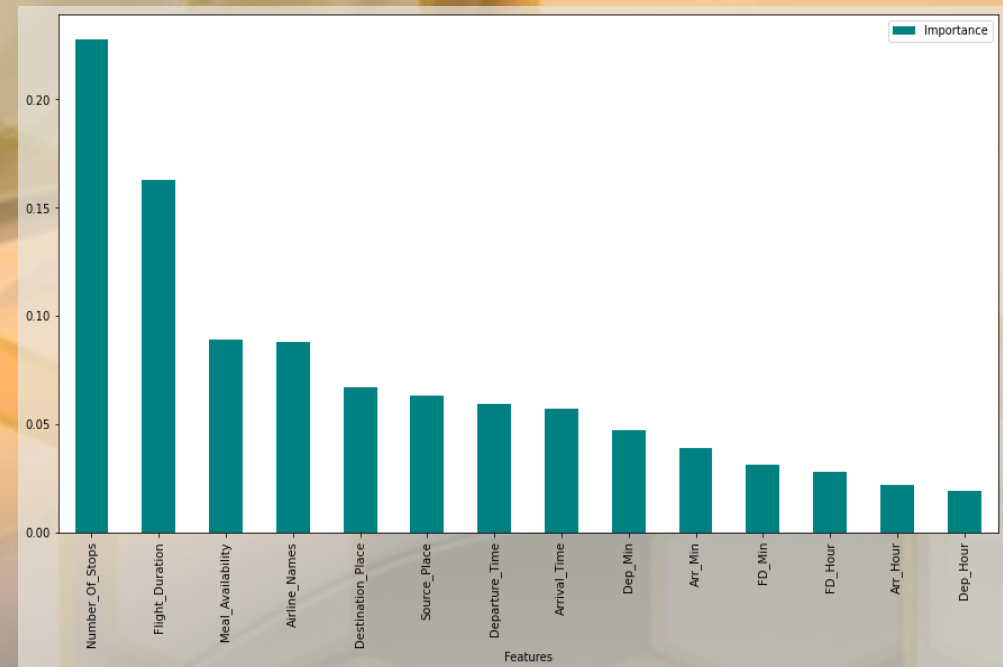
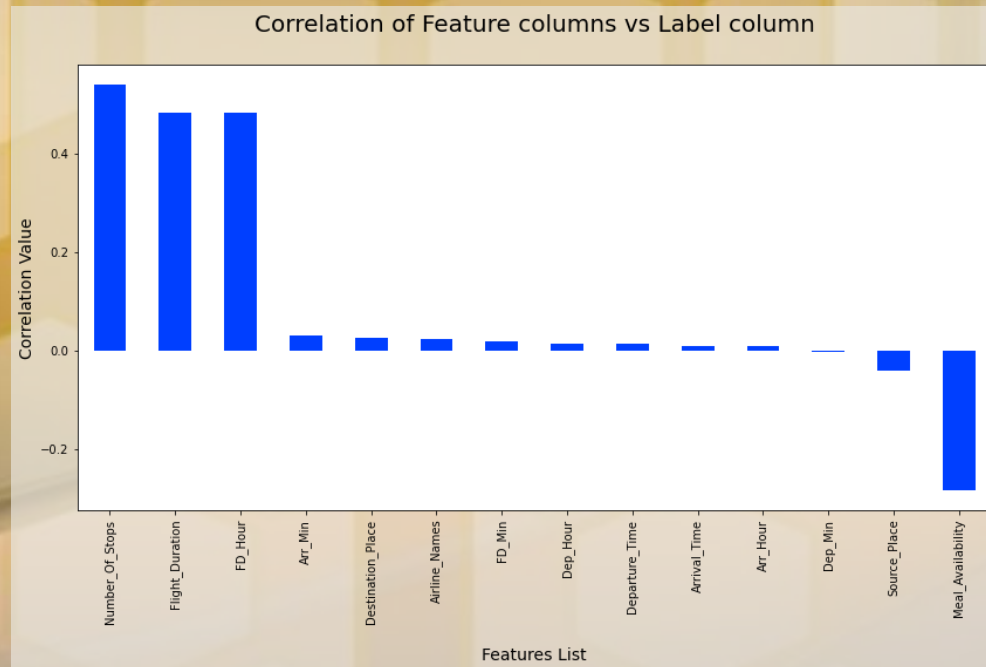
Statistical Report of Numerical Columns							
Departure_Time	13.02	4.89	0.58	8.92	12.92	16.83	23.42
Arrival_Time	15.82	5.54	0.08	11.50	16.67	20.42	23.92
Flight_Duration	12.45	8.14	0.83	6.33	10.25	17.83	41.08
Number_Of_Stops	1.17	0.70	0.00	1.00	1.00	1.00	4.00
Flight_Prices	10046.68	3667.75	3361.00	7425.00	9747.00	12249.00	38348.00
Dep_Hour	12.59	4.88	0.00	8.00	12.00	16.00	23.00
Dep_Min	26.36	17.29	0.00	10.00	25.00	40.00	55.00
Arr_Hour	15.33	5.53	0.00	11.00	16.00	20.00	23.00
Arr_Min	29.47	17.49	0.00	15.00	30.00	45.00	55.00
FD_Hour	12.00	8.14	0.00	6.00	10.00	17.00	41.00
FD_Min	26.94	17.05	0.00	10.00	25.00	40.00	55.00
	mean	std	min	25%	50%	75%	max

# HISTOGRAM AND HEATMAP

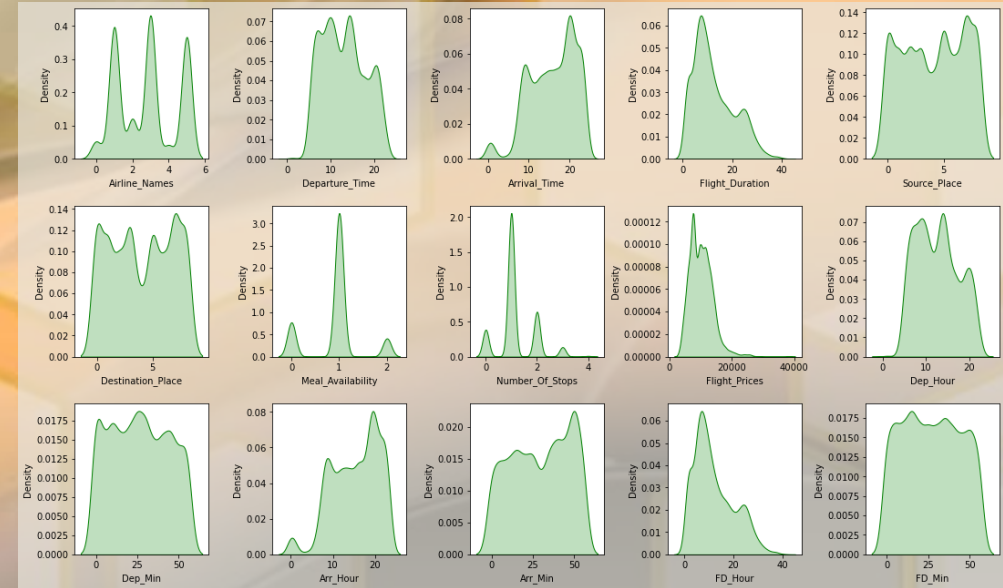
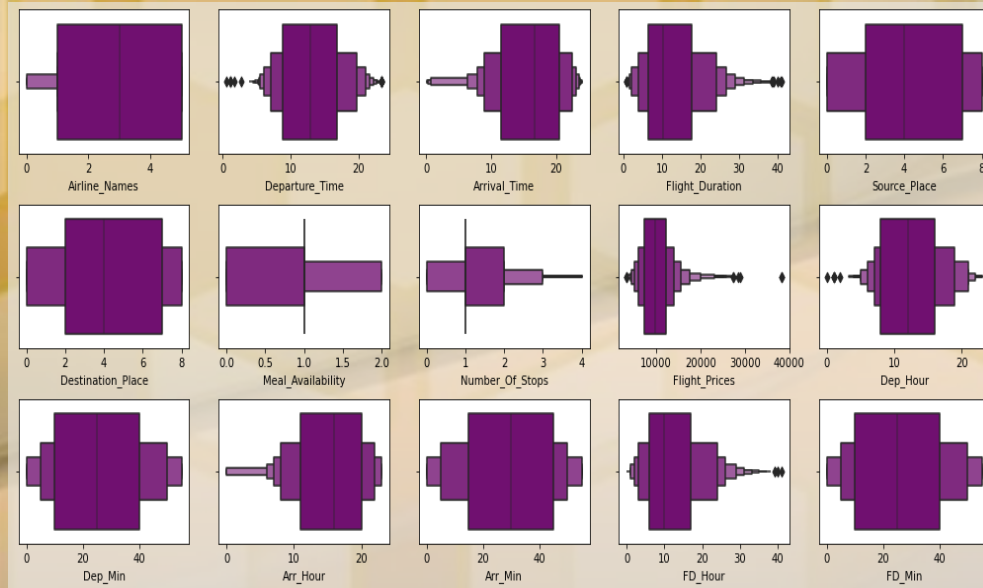




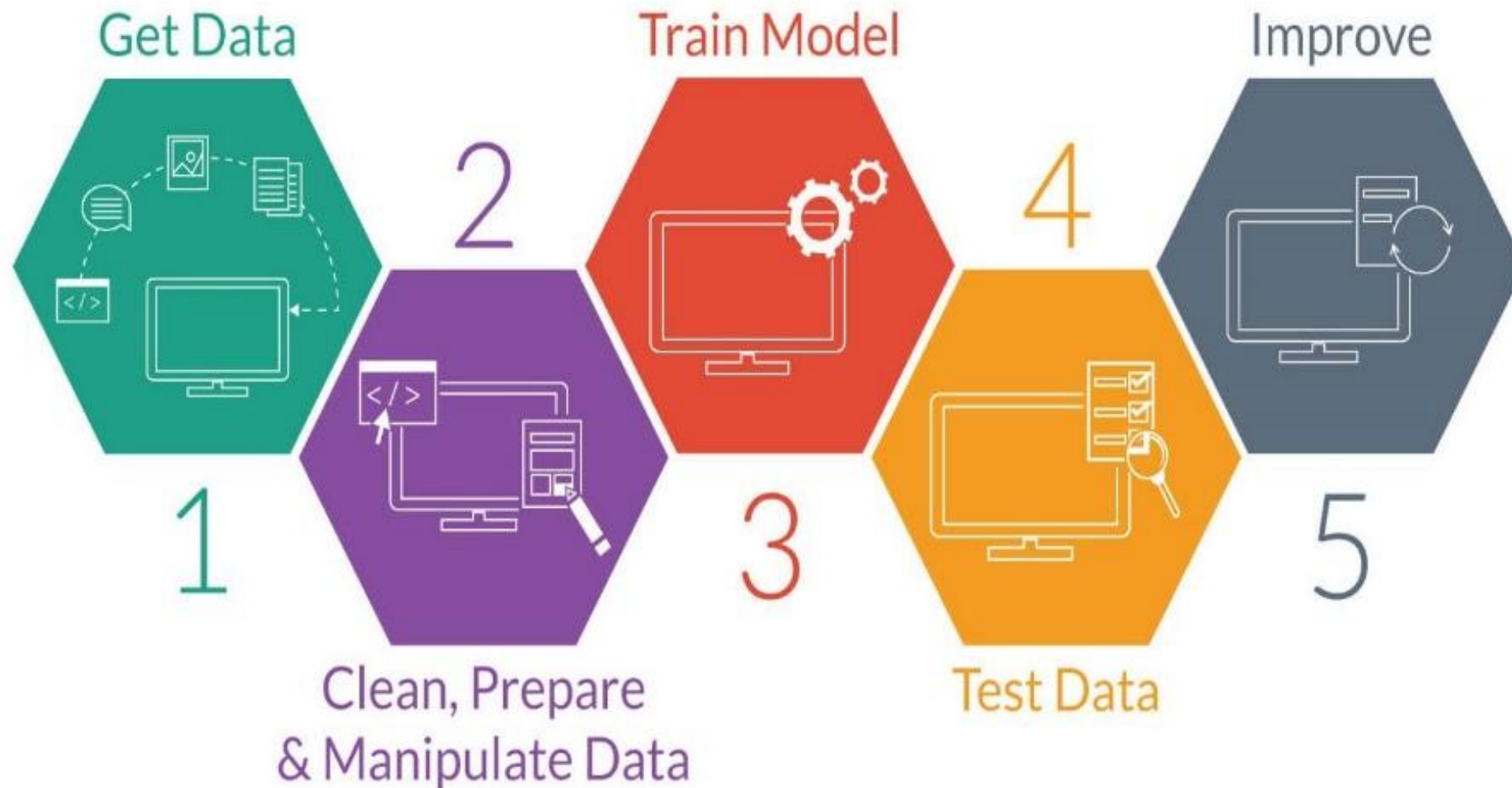
# CORRELATION AND IMPORTANCE BAR GRAPHS



# OUTLIERS AND SKEWNESS



# MODEL TRAINING PHASES





# REGRESSION MACHINE LEARNING MODEL/S USED

- Linear Regression Model
- Ridge Regularization Model
- Lasso Regularization Model
- Support Vector Regression Model
- Decision Tree Regression Model
- Random Forest Regression Model
- K Neighbours Regression Model
- Gradient Boosting Regression Model
- Ada Boost Regression Model
- Extra Trees Regression Model



# REGRESSION MODEL FUNCTION WITH EVALUATION METRICS

```
# Regression Model Function

def reg(model, X, Y):
    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25, random_state=638)

    # Training the model
    model.fit(X_train, Y_train)

    # Predicting Y_test
    pred = model.predict(X_test)

    # RMSE - a lower RMSE score is better than a higher one
    rmse = mean_squared_error(Y_test, pred, squared=False)
    print("RMSE Score is:", rmse)

    # R2 score
    r2 = r2_score(Y_test, pred, multioutput='variance_weighted')*100
    print("R2 Score is:", r2)

    # Cross Validation Score
    cv_score = (cross_val_score(model, X, Y, cv=5).mean())*100
    print("Cross Validation Score:", cv_score)

    # Result of r2 score minus cv score
    result = r2 - cv_score
    print("R2 Score - Cross Validation Score is", result)
```

# RESULT OF MULTIPLE REGRESSION MODELS

```
# Linear Regression Model
```

```
model=LinearRegression()  
reg(model, X, Y)
```

```
RMSE Score is: 2741.9142718035005
```

```
R2 Score is: 41.30962845018751
```

```
Cross Validation Score: 33.96122010005579
```

```
R2 Score - Cross Validation Score is 7.34840835013172
```



# EVALUATION AND HYPER PARAMETER TUNING

The key metrics used here were:

- ✓ R2 score
- ✓ Cross Validation Score
- ✓ MAE
- ✓ MSE
- ✓ RMSE

We tried to find out the best parameters list to increase our accuracy scores by using Hyperparameter Tuning. In order to achieve a higher score we used the Grid Search CV method with 5 folds.



# Inference

Concluding the project outcome

# KEY FINDINGS AND CONCLUSIONS OF THE STUDY

In this project we have scraped the flight data from airline webpages. Features like flight duration, number of stops during the journey and the availability of meals are playing major role in predicting the prices of the flights.

It could also help customers to predict future flight prices and plan the journey accordingly because it is difficult for airlines to maintain prices since it changes dynamically due to different conditions. Hence by using Machine Learning techniques we can solve this problem.

The above research will help our client to study the latest flight price market and with the help of the model built he can easily predict the price ranges of the flight, and also will helps him to understand Based on what factors the flight price is decided.



# LEARNING OUTCOMES OF THE STUDY IN RESPECT OF DATA SCIENCE

Visualization part helped me to understand the data as it provides graphical representation of huge data.

It assisted me to understand the feature importance, outliers or skewness detection and to compare the independent-dependent features.

Data cleaning is the most important part of model building and therefore before model building, I made sure the data is cleaned.

I have generated multiple regression machine learning models to get the best model wherein I found Extra Trees Regressor Model being the best based on the metrics I have used.

Ensured that I at least get a decent prediction confidence percentage.

# LIMITATIONS OF THIS WORK AND SCOPE FOR FUTURE WORK

Some algorithms are facing over-fitting problem which may be because of a smaller number of features in our dataset.

Limitation of the study is that in the volatile changing market we have taken the data, to be more precise we have taken the data at the time of pandemic and recent data, so when the pandemic ends the market correction might happen slowly.

Therefore based on that again the deciding factors of it may change and we have shortlisted and taken these data from the important cities across India.

If the customer is from the different country our model might fail to predict the accuracy prize of that flight.



**Source**

**Destination**

**Thank you**