Forecasting Flight Fares

A Step-by-Step guide to Build a Machine Learning Model to predict Flight Fares



**Article By:**

**Sneha Kurane**

07.07.2022

Data Trained Education

INTRODUCTION

In this Article, I will be guiding you to the step-by-step procedure in building a Machine Learning model in Python using popular machine learning libraries NumPy, Pandas & scikit-learn to predict Domestic Flight fares in India.

To help machines understand like humans do and to strengthen AI, machine Learning is required. One of the applications of machine Learning lies within the ‘Aviation industry’, to foresee the costs of flights. After the relaxation of norms since Pandemic, the Global Tourism Industry is now on track. The pandemic has changed a lot about travel, but one thing that hasn't changed is affordability. People like to visit places that are within their budget. Airfares have risen recently, impacting travelers' budgets.

### PROBLEM STATEMENT

On Thursday June 16th, ATF prices were increased by 16.3%. These factors like massive hike in ATF prices, together with plummeting rupee against dollar are set to raise the cost of airlines operations in turn increasing the air fares by up to 15%.

We have often heard travelers saying that flight ticket prices are so unpredictable. Although We cannot change the fact about rising fuel prices, there are some factors according to which we can predict the Prices of the flight and arrange the travel accordingly. In this Article, I have built the model predicting flight fares using the dataset of prices of flight tickets for various airlines between the months of March and June of 2019 and between various cities. Dataset can be downloaded from: <https://github.com/dsrscientist/Data-Science-ML-Capstone-Projects/blob/master/Flight_Ticket_Participant_Datasets-20190305T100527Z-001.zip>

There are 2 datasets in the link, Train and Test Dataset.

* Size of training set: 10683 records
* Size of test set: 2671 records

Using the Training set, I have built the model and predicted the Flight fares in the Test dataset. GitHub link for the Notebook in this blog is:

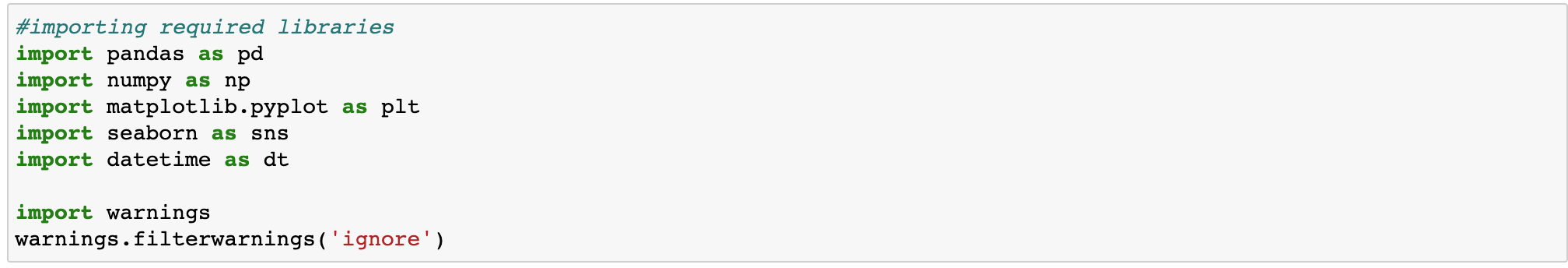
https://github.com/Razni-DS/Flight-Fare-Prediction/blob/main/Flight%20Fare%20Prediction\_%20Evaluation%20Project%207.ipynb

LIBRARIES IMPORTED

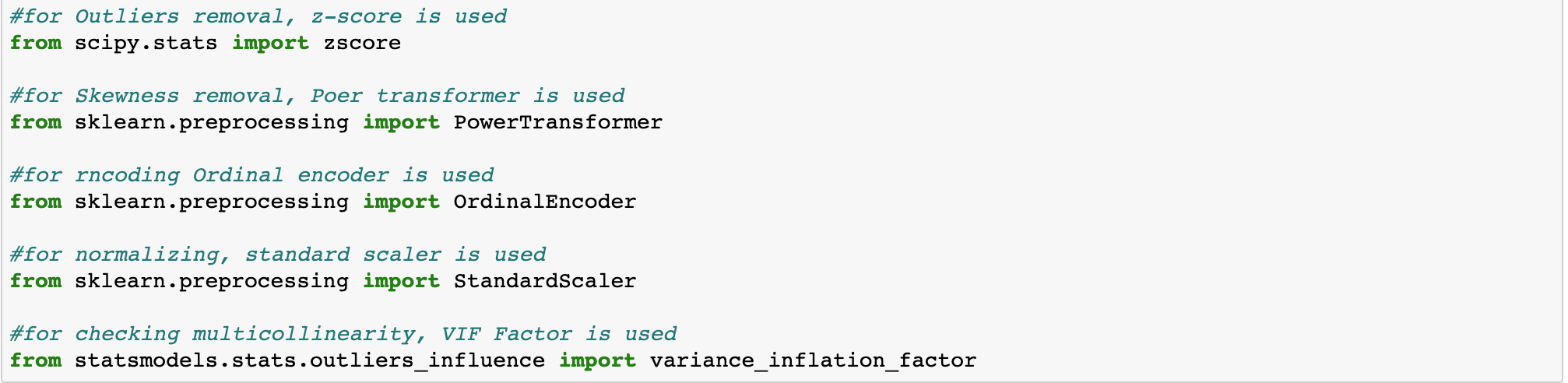
There are 3 sets of libraries used.

* Basic Libraries for Data Analysis and Visualization
* Libraries for Data Cleaning and Feature Engineering (Data preprocessing)
* Libraries for Building the ML Models

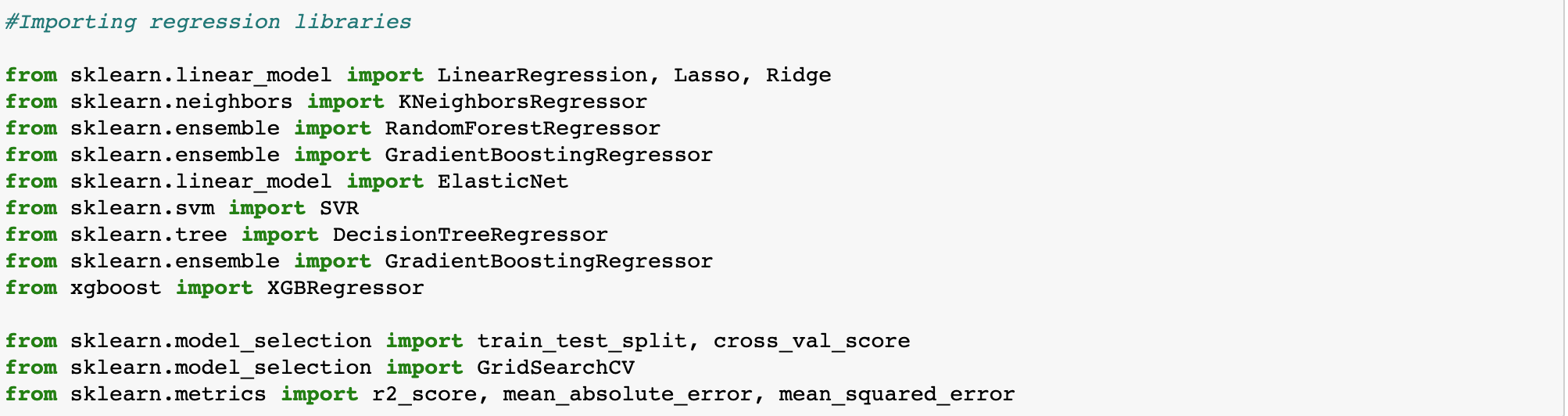
**Basic Libraries used are:**



**Libraries for Data Cleaning and Feature Engineering (Data preprocessing):**

****

**Libraries for Building the ML Models:**

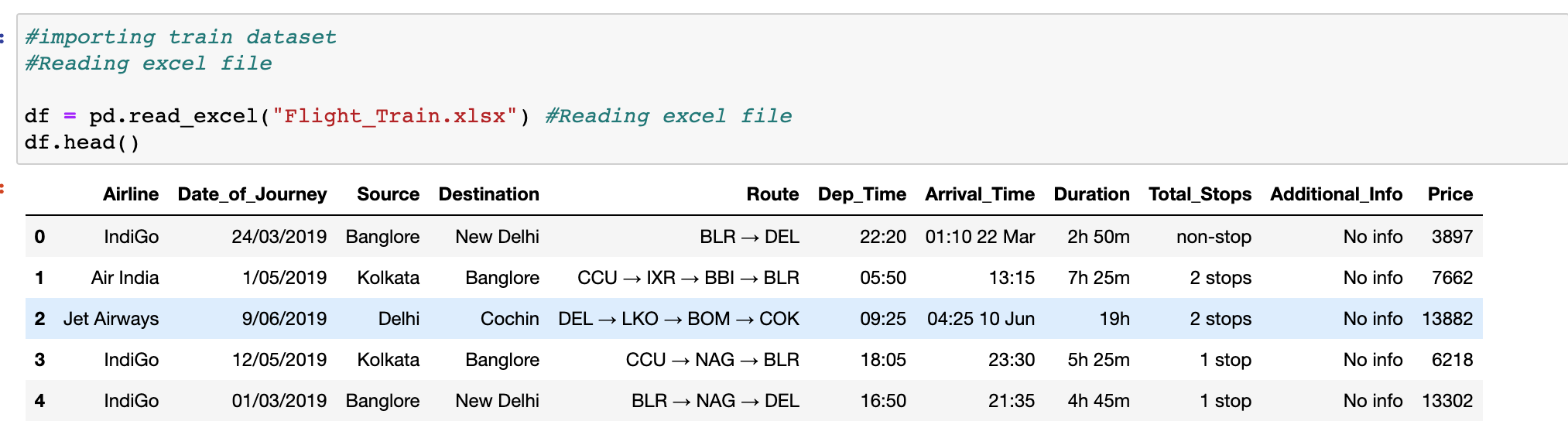
****

These are the libraries used in my Jupyter notebook for Prediction.

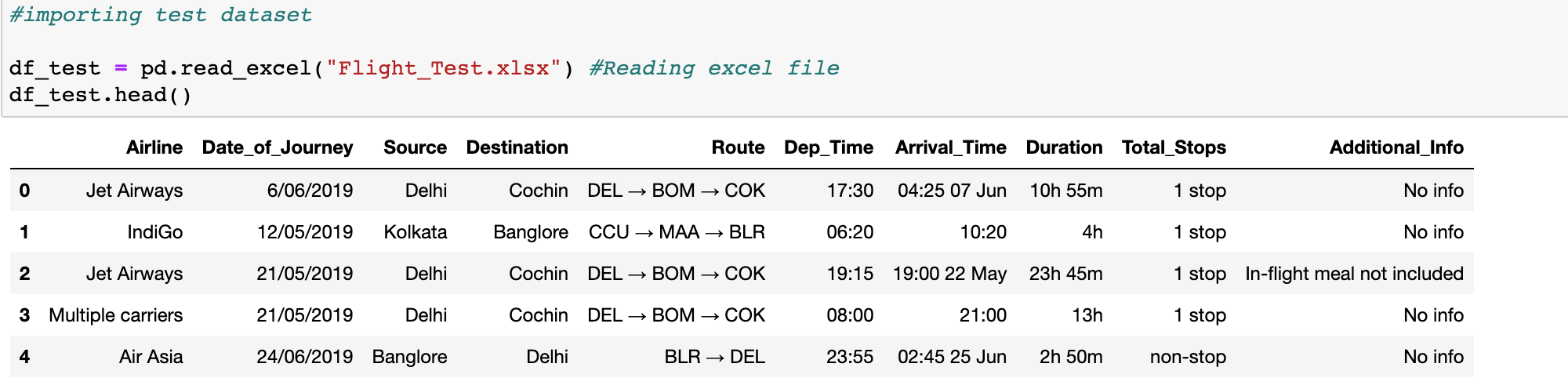
DATA ANALYSIS

* After loading the train and test datasets, I checked the first 5 elements in both sets.

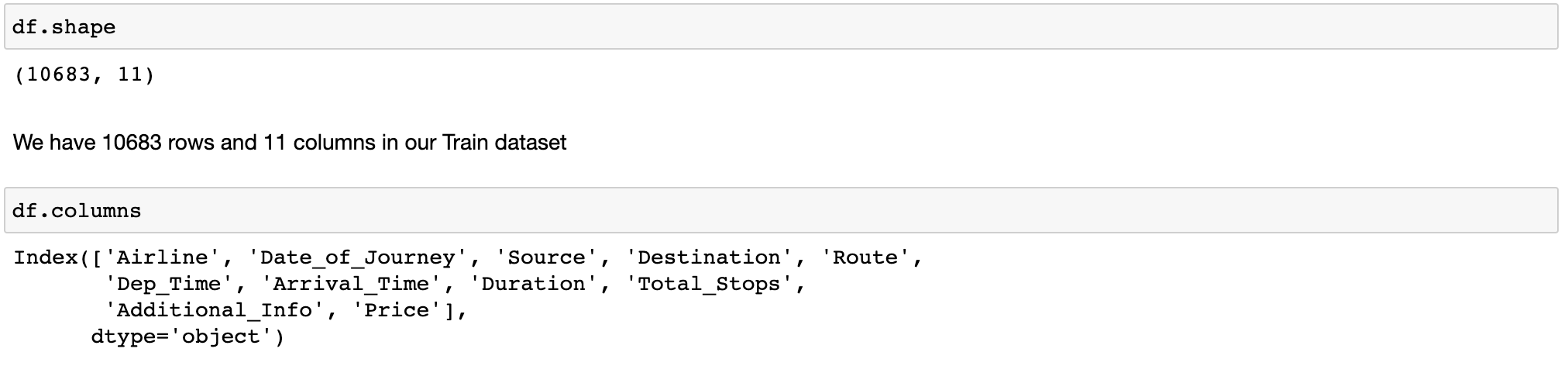
Train Dataset:



Test Dataset:



\* The next step is to check the shape and Variable names in the dataset.





We have 10,683 rows and 11 columns including our Target “Price” for the train dataset and 2,671 rows and 10 columns for the Test dataset. Target is not available for the Test dataset which we will be predicting at the end.

# 

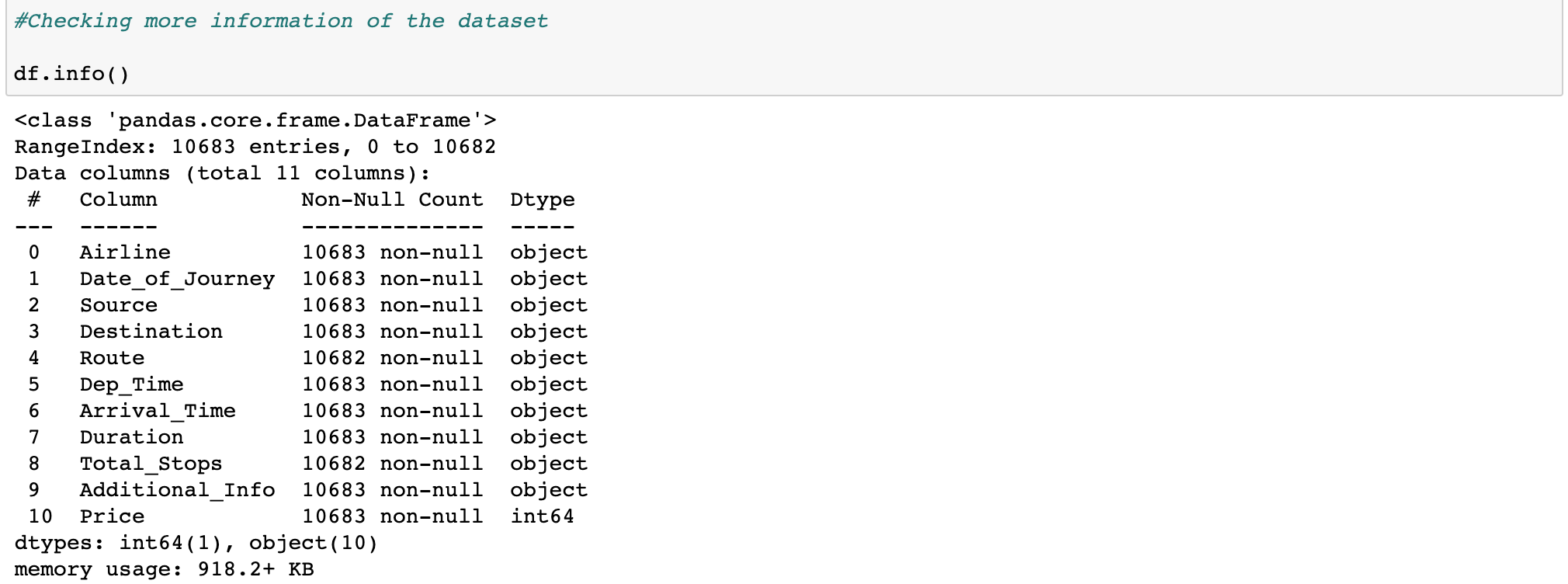
# **VARIABLE DESCRIPTION**

The 11 columns are:

1. **Airline**: The name of the airline. Indigo, Jet Airways, Air India, and many more.
2. **Date\_of\_Journey**: The date of the journey
3. **Source**: The source from which the service begins.
4. **Destination**: The destination where the service ends.
5. **Route**: The route taken by the flight to reach the destination.
6. **Dep\_Time**: The time when the journey starts from the source.
7. **Arrival\_Time**: Time of arrival at the destination.
8. **Duration**: Total duration of the flight.
9. **Total\_Stops**: Total stops between the source and destination.
10. **Additional\_Info**: Additional information about the flight like food, kind of food, and other amenities.
11. **Price**: The price of the ticket including all the expenses.

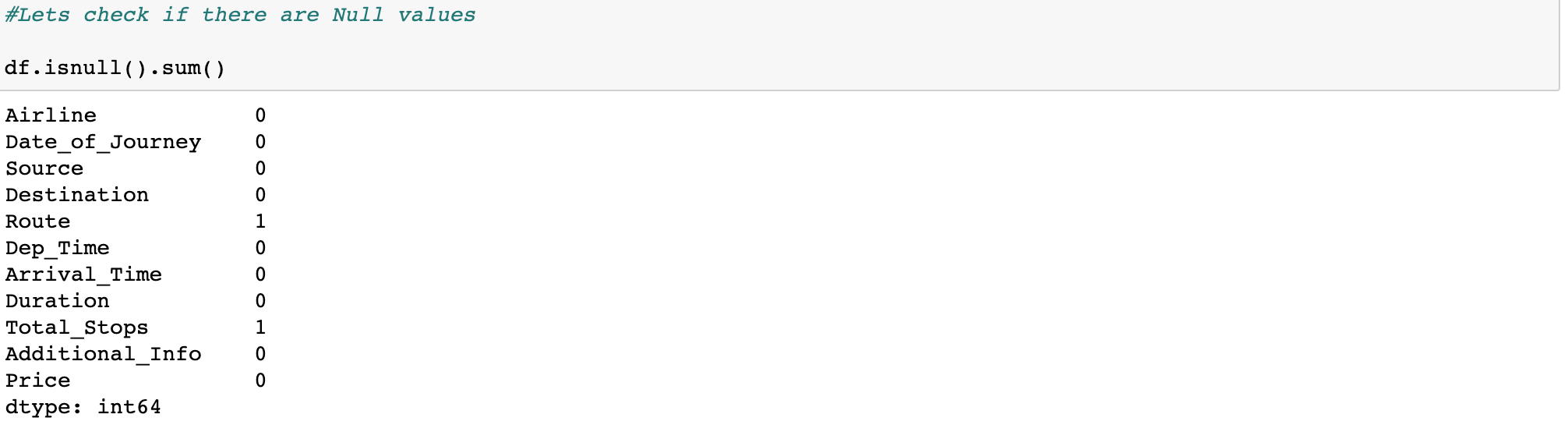
# **DATA EXPLORATION**

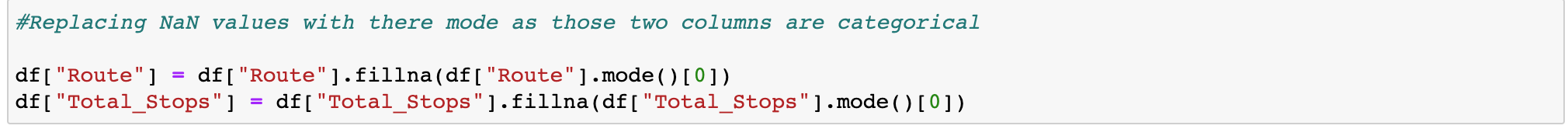
* I have checked the information of the dataset regarding null values and datatypes.



I have 1 INT datatype (Target- ‘Price’) and 10 Object data types. I can see that Date\_of\_journey, Dep\_time and Arrival\_Time is not in datetime datatype. I will change it to the format my Machine can understand.

* I then checked the missing data.



There is only **1 missing data,** and the variables are categorical. So, I have used the Imputation technique and used the mode of the column to replace the missing value.

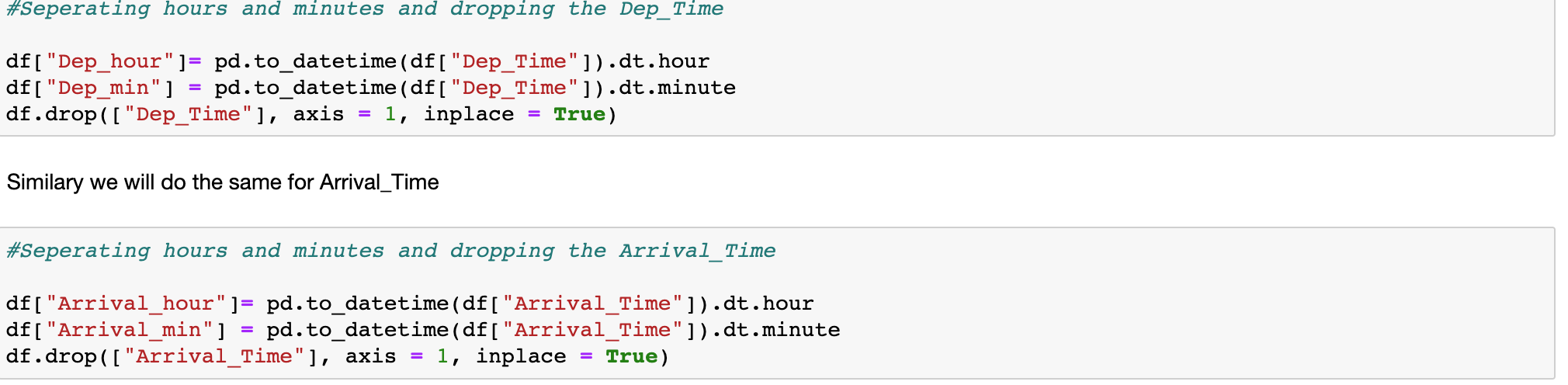
Then I checked the Unique number of values in my dataset:

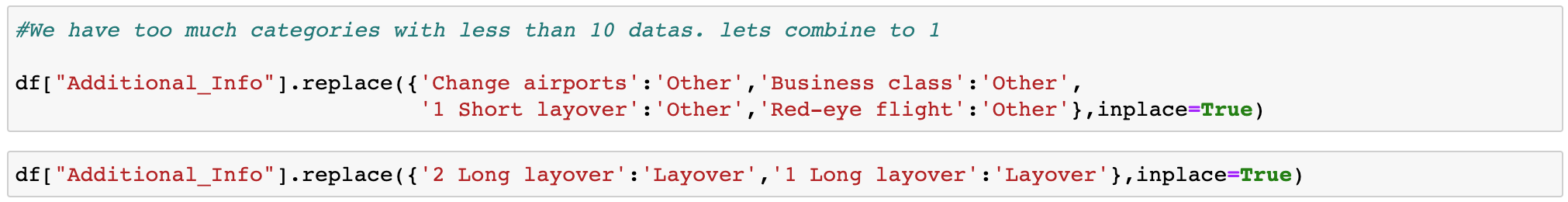
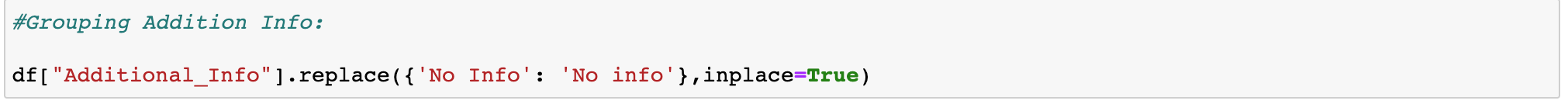
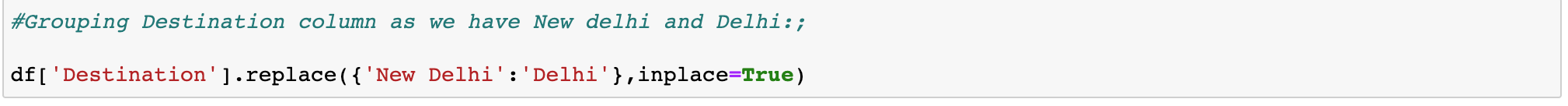
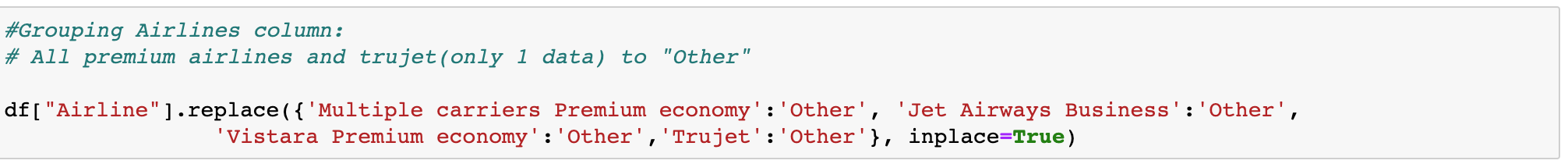
* 12 Airlines mentioned in the dataset.
* Date of Journey of 44 different dates.
* 5 Sources and 6 Destinations.
* 128 Routes of travel.
* 222 departure times mentioned, and 1,343 Arrival time Mentioned in the dataset.
* 368 Durations mentioned in the dataset.
* 5 different categories of Total\_stops.
* 10 different categories of Additional Info.
* 1,870 different prices in the dataset of 10,683 rows.

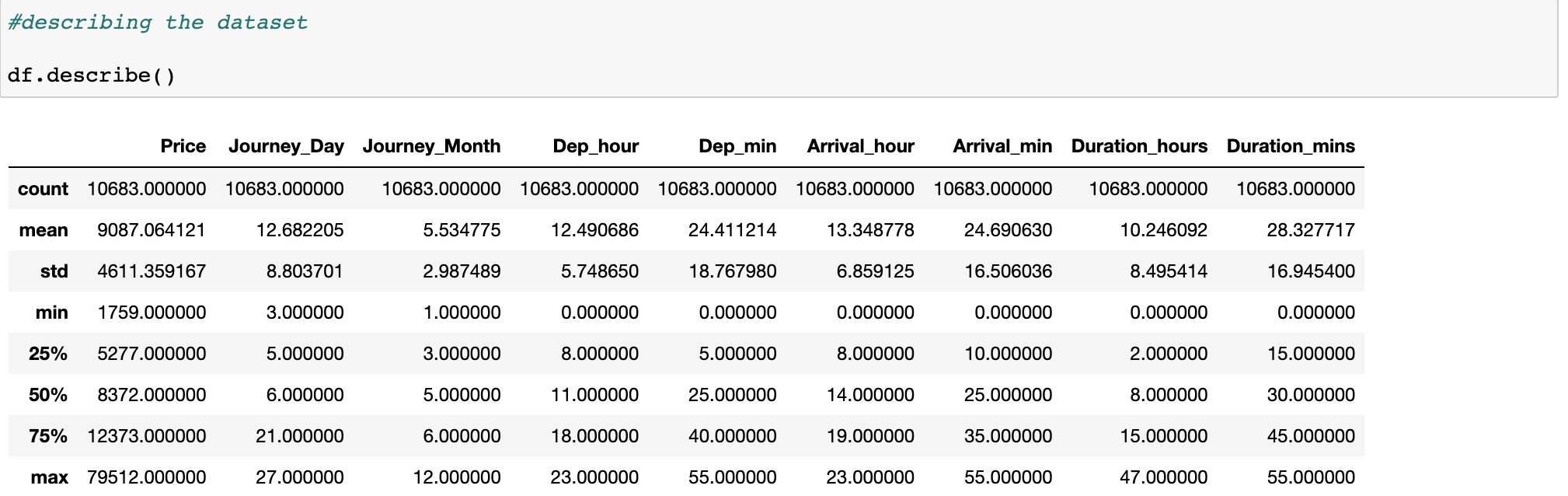
**FEATURE EXTRACTION**

The dataset contains some features to be changed for our Machine to understand well. Date, Duration and Dep and arrival Time format are not recognized by our Machine. We can convert it into meaningful data. After changing it to **datetime type**, I then extracted Day and Month from Date of journey. Since the year is only single data-2019, I have omitted it.



\* In the same way, I have extracted Departure and Arrival time into hours and minutesI have dropped Date\_of\_journey, Dep\_time and Arrival\_time from the dataset after extracting necessary data. I then extracted duration\_hours and minutes from the column Duration and dropped the Duration column as well. 

\* After checking the value\_counts of the column I have found that there are some **duplicate values** or similar values in columns Airline, Destination and Additional\_info. 

Our dataset is now comprehensible and easy to visualize the columns. I then described the dataset: 

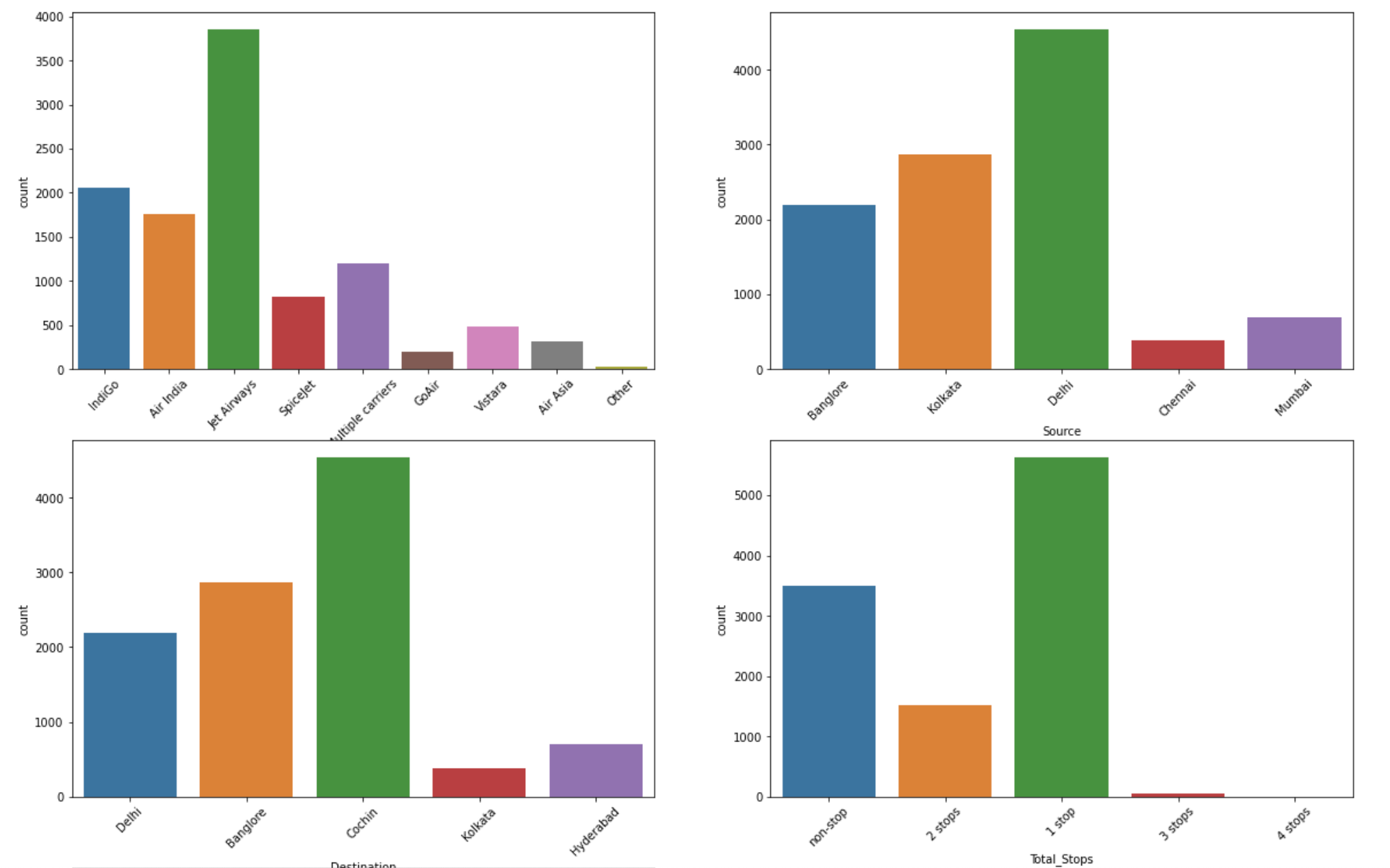
These are the numerical columns description. We can see some slight skewness in the dataset as mean and median are different. No major outliers present except for Price.

DATA VISUALIZATION

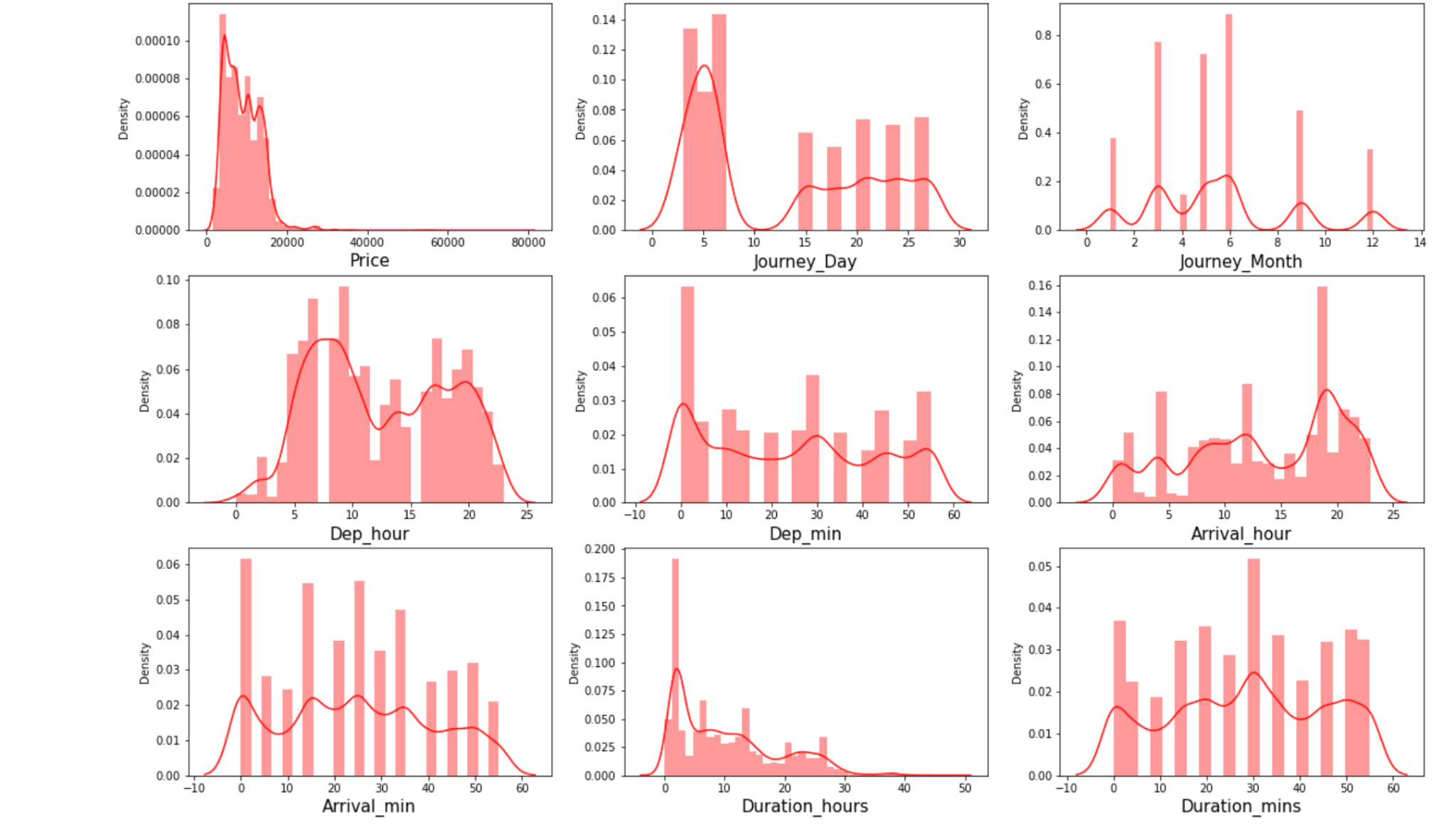
I have used Univariate, Bivariate and Multivariate Analysis in my dataset.

\* Univariate Analysis:

Countplot of Categorical columns:

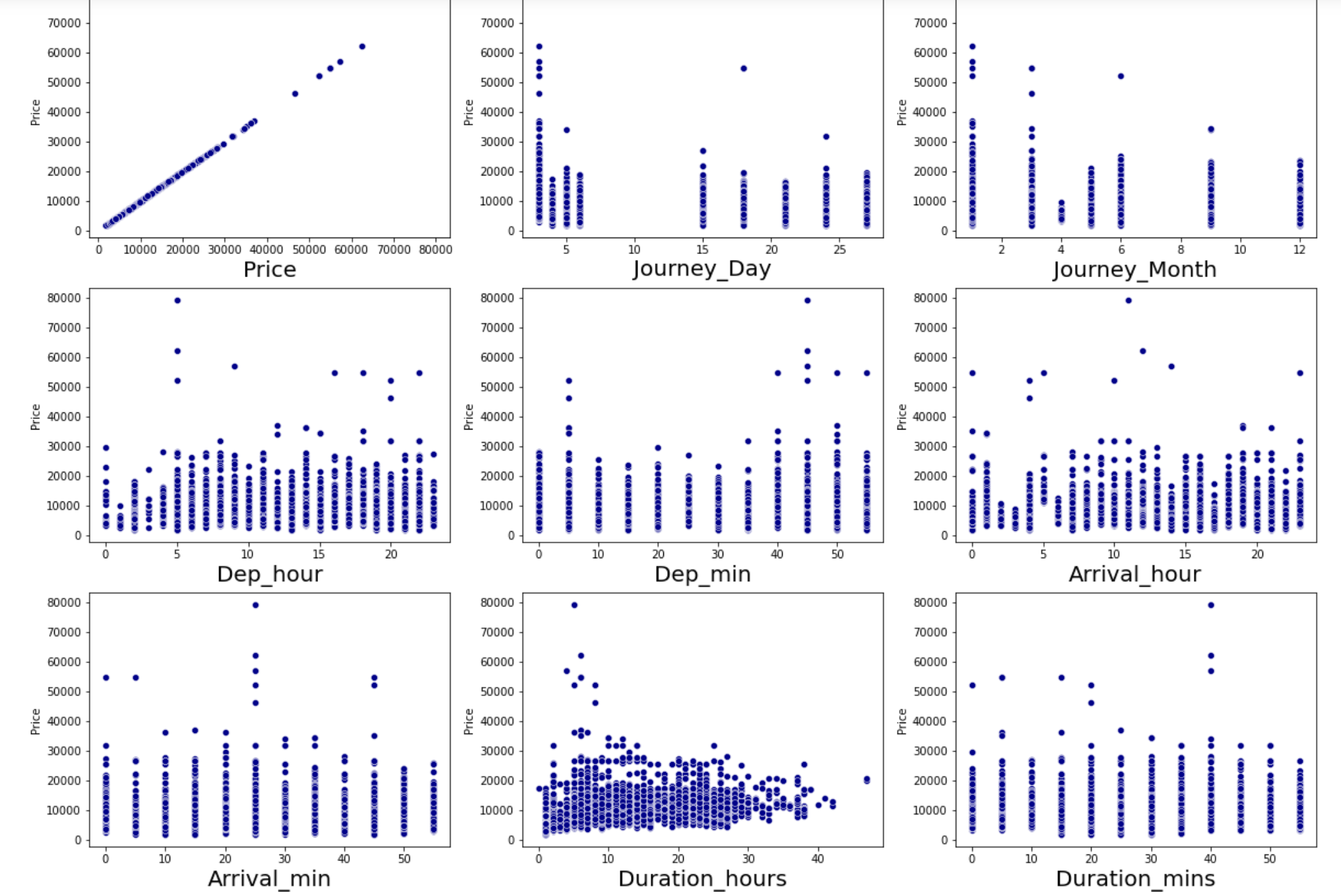


Distplot of numerical columns:

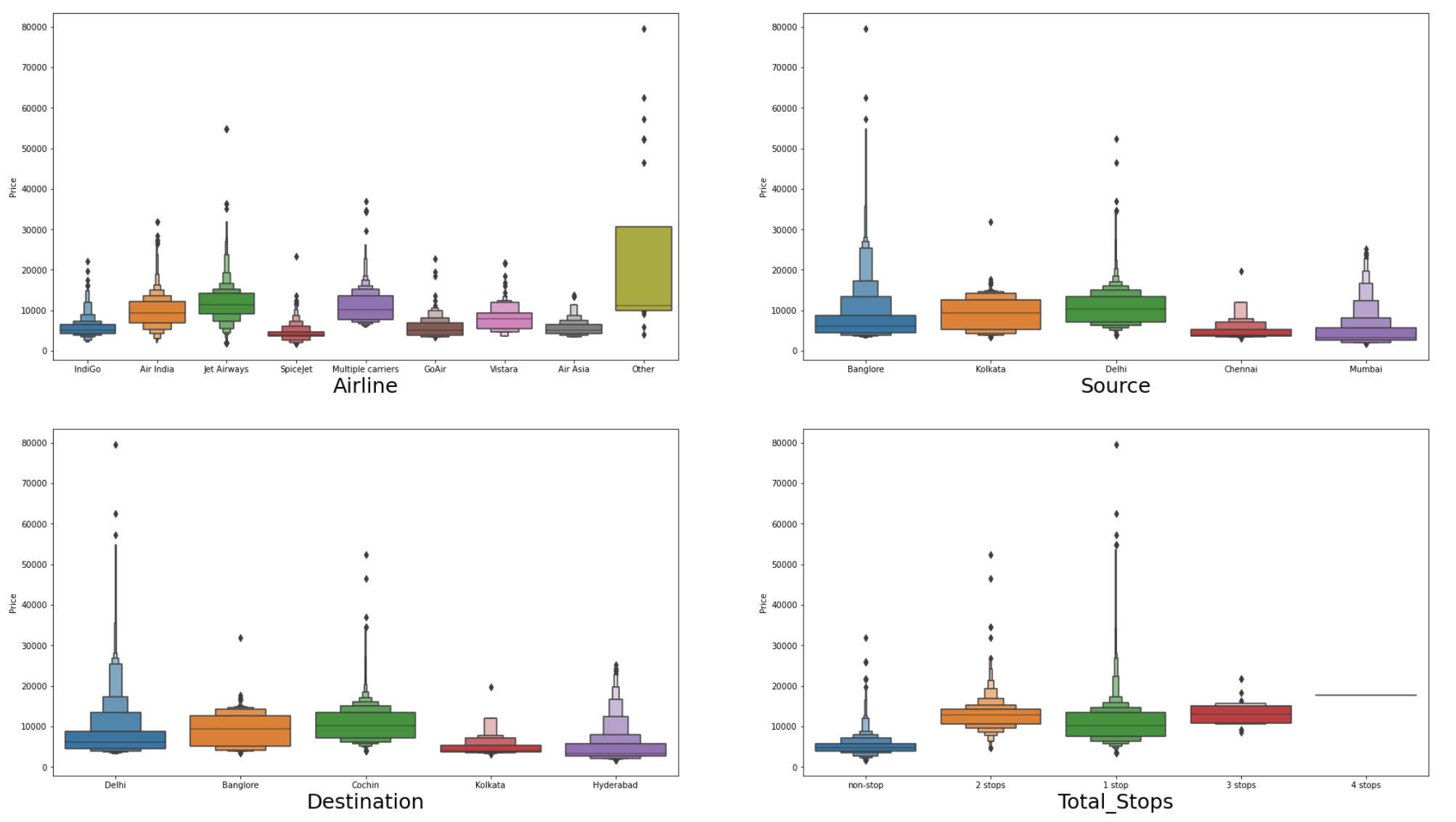
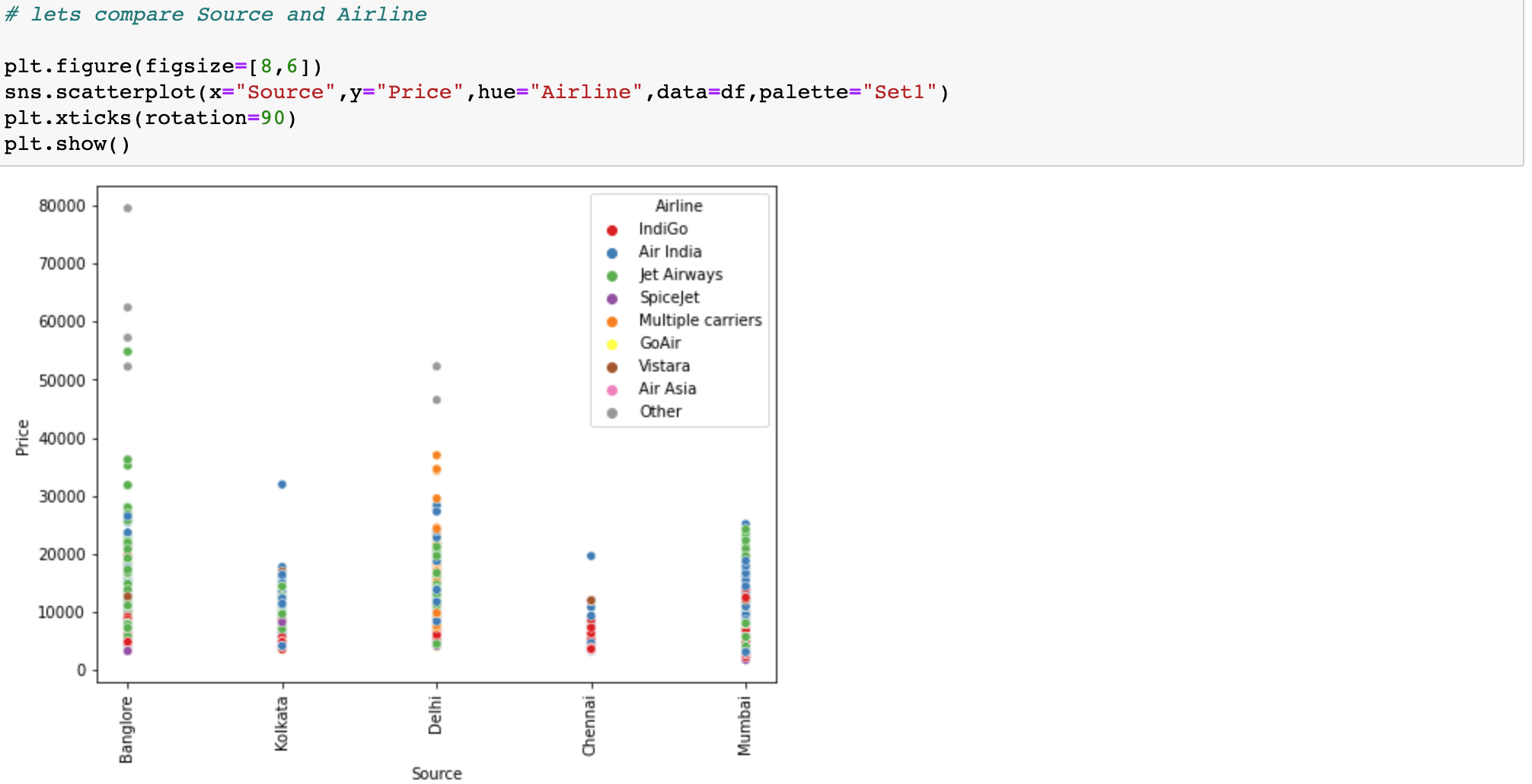


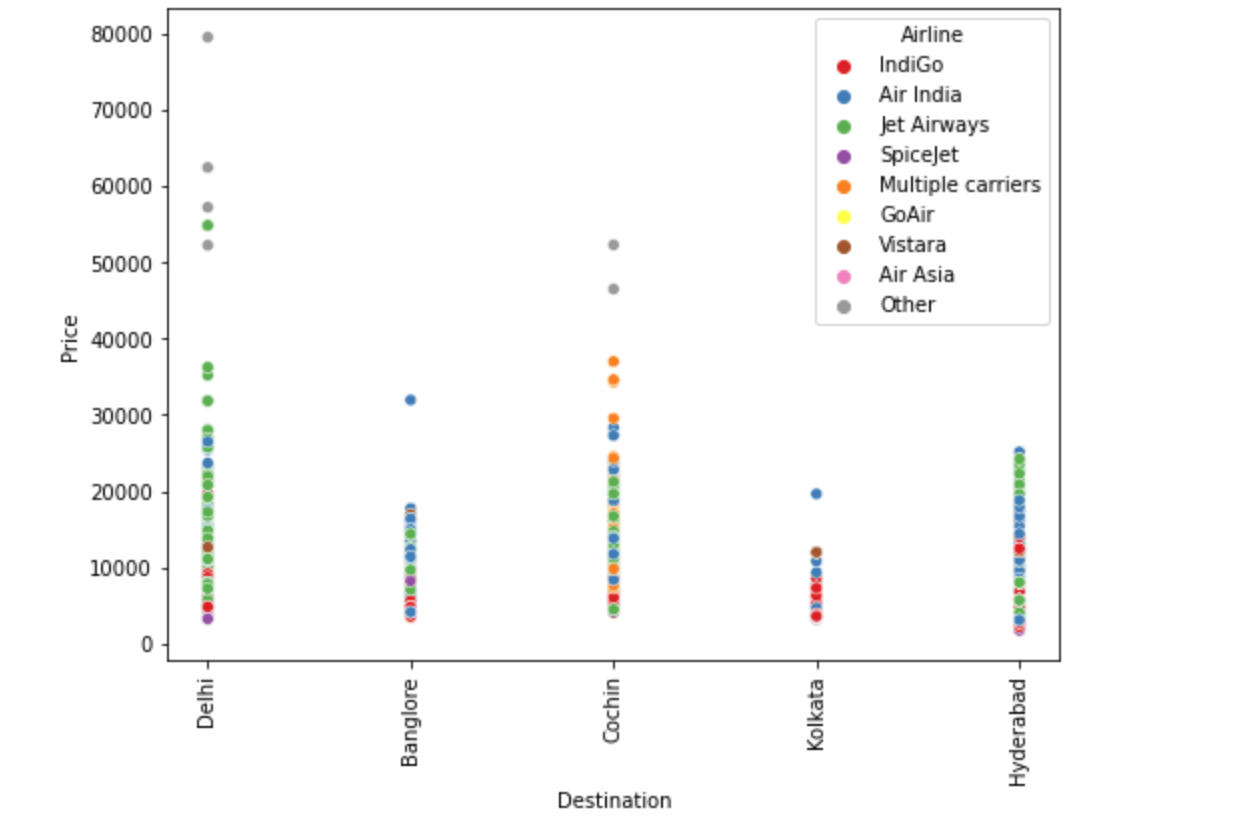
\* Bivariate Analysis:

Scatterplot of numerical columns with Target “Price”



Boxenplot of Categorical columns with Price:

\* While comparing Source and Price with hue as Airline we can see that Jet Airways is the most used flight and Flight from Bangalore in other category flights which is premium, or Business class is the most expensive. 

\* While comparing Destination and Price with hue as Airline we can see that Highest price is for destination as Delhi and cheapest for Destination Hyderabad.

EDA Concluding Remark

From Univariate analysis, Major findings are:

1. Most of the data have Source as Delhi and Destination as Cochin.
2. The top Airline is Jet Airways followed by Indigo.
3. We can see skewness present in numerical columns.

Checking the categorical columns with Target ”Price”, major findings are:

* The cheapest flight is SpiceJet, and the top fare is between 2-5,000.
* Source Mumbai and Destination Hyderabad are cheaper.
* Destination as Delhi has the Highest price of 80,000.
* “Other” Flights like Premium and Business Class have the highest fare.
* Flights with No Check-in baggage have the least Fare range.

Checking the numerical columns with the Target, major findings are:

* Price is higher during the start of a month. Also, prices are higher in January.
* Prices are the highest at 1 to 10 hours duration of flight.

### *We can assume that SpiceJet from Mumbai to Hyderabad with No-check in baggage and flying in the midnight on the 4th month is the cheapest flight.*

### *Also, Flights from Bangalore to Delhi in the Early morning in January in a Business class with 1 stop have the highest fare.*

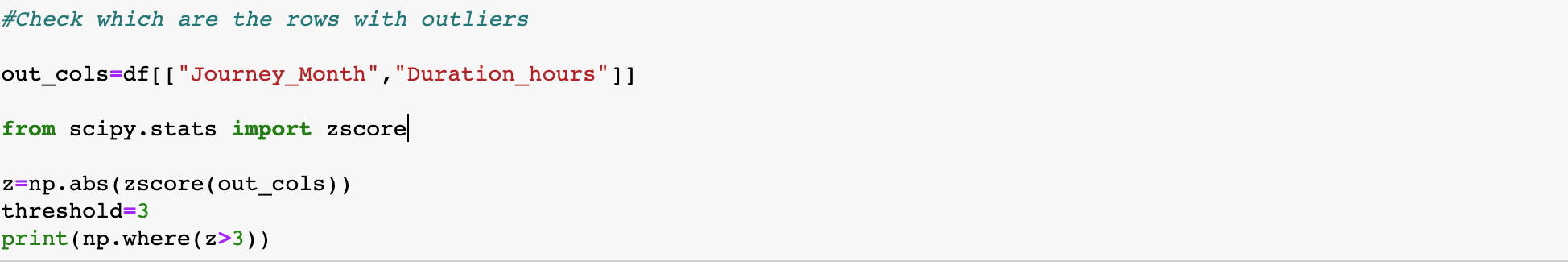
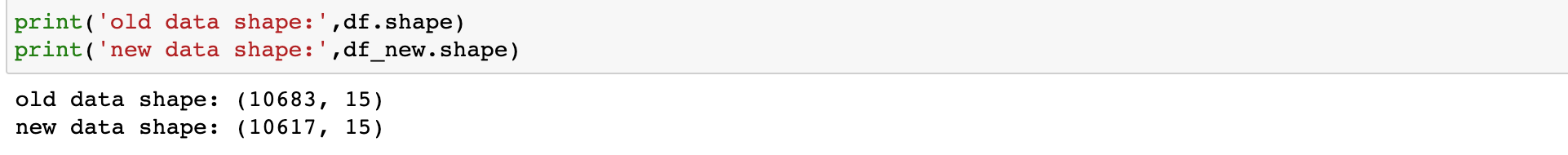
DATA PREPROCESSING

Before building the Model, I have used certain techniques to transform the data so that it may be easily parsed by the machine. Steps involved are:

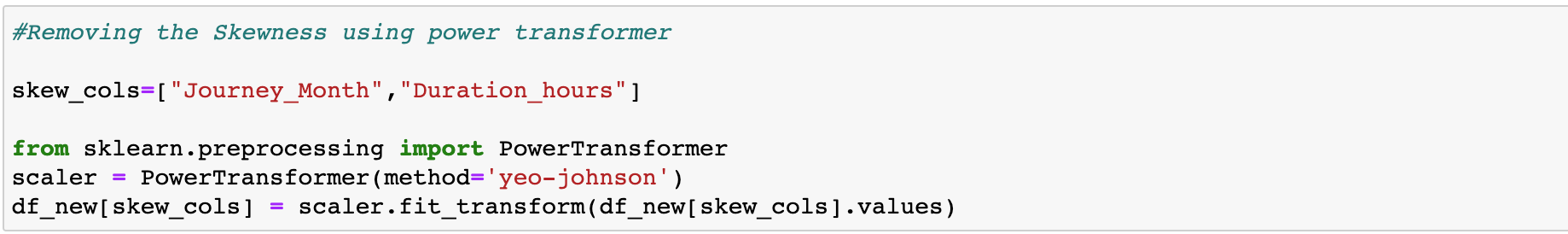
1. Outliers’ removal
2. Skewness removal
3. Encoding the Data
4. Checking the Correlation
5. Standardization
6. Checking the Multicollinearity using VIF Factor

**OUTLIERS**

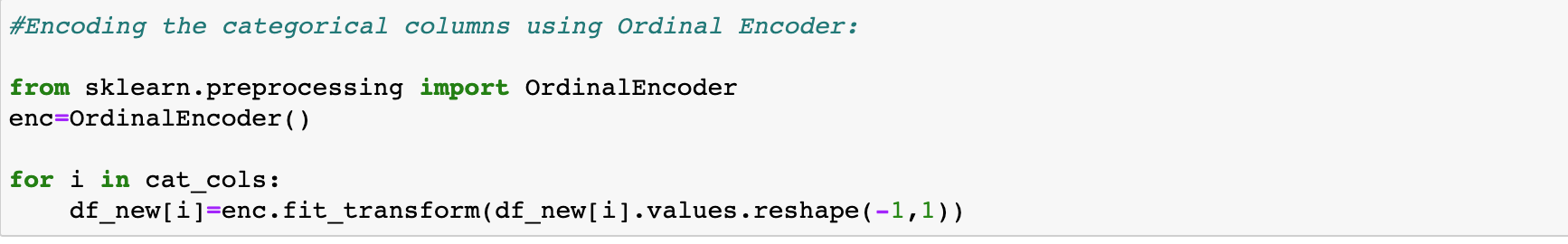
I checked Outliers using Boxplot and found that I have outliers in Journey\_Month and Duration\_hours. I used the Z-score method to remove Outliers.

After removing Outliers, my new dataset shape is:

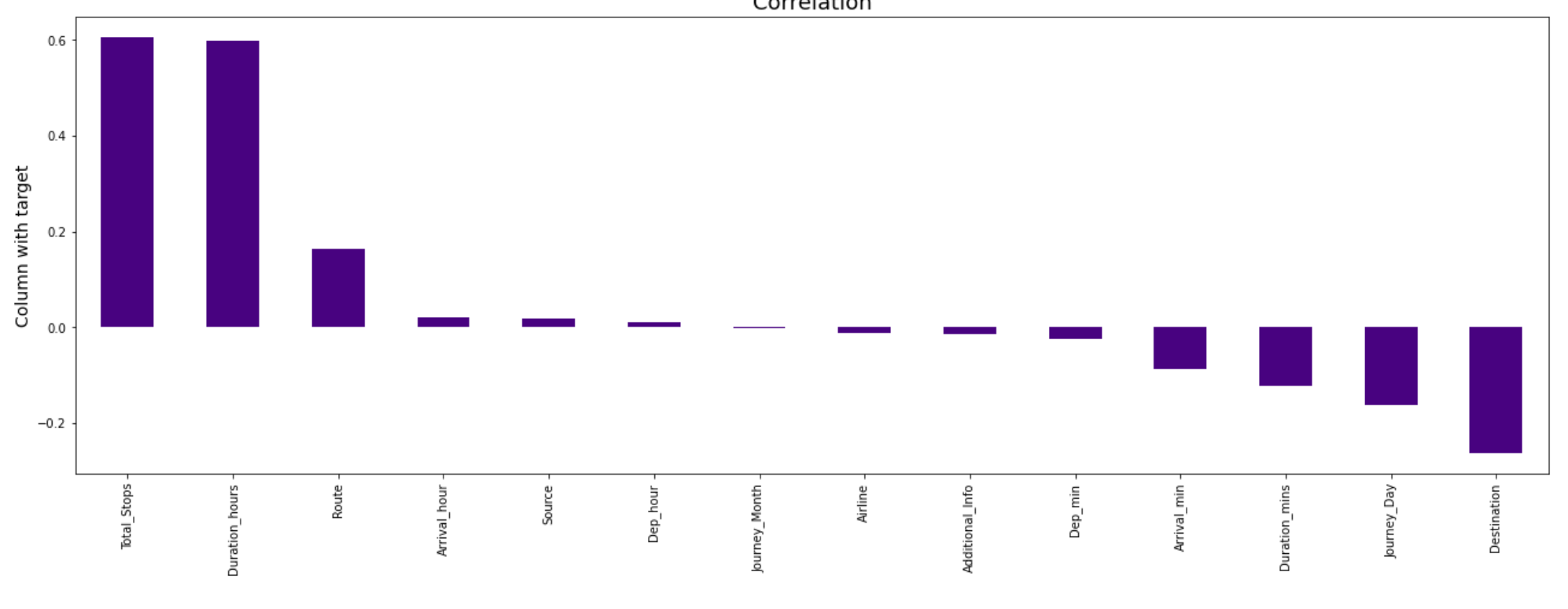
**SKEWNESS**

Using the skew() function, I checked skewness in my new dataset and found that I have skewness in Journey\_month and duration\_hours. I used “power transformer” to transform both columns alone.

**ENCODING**

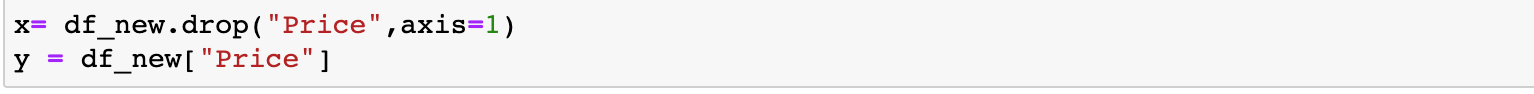
Using Ordinal Encoder, I have encoded the categorical columns. 

**CORRELATION**

Using the corr() function I checked the Correlation between Variables and Target. The visualization correlation coefficient shows that our target is highly correlated with Total\_stops and Duration\_hours and least correlated with Journey\_month.The correlation heatmap:There are some multicollinearities between variables like Total stops and Duration hours which is normal in this flight fare case.

**DATA SPLITTING**

Splitting the data to x and y



**STANDARDIZATION**

I used a standard scaler to normalize the data.

**VIF FACTOR**

I first defined a function and then called it to check Multicollinearity.Since all values are below 5, My dataset is free from multicollinearity, and I can proceed with Model building.

***All the above preprocessing steps except for data splitting has been performed in the Test dataset as well.***

BUILDING MACHINE LEARNING MODEL

Our Target “Price” is a Continuous variable, we have a **Regression problem**. I have imported all necessary regression libraries from sklearn.

**1. Linear Regression:**

Chart, scatter chart

Description automatically generatedUsing the “for” loop, I found the best Random state. The R2 score for Linear regression Model was 50.87% and Cross validation score of 46.30%. Even after Hyperparameter tuning, my score did not improve. Regplot of Linear Regression is:

**2. Ridge regularization:**

Using the “for” loop, I found the best Random state. The R2 score for Ridge Model was 51.06% and Cross validation score of 46.30%. Regplot of Ridge is:

Chart, scatter chart

Description automatically generated

**3. KNeighborsRegressor:**

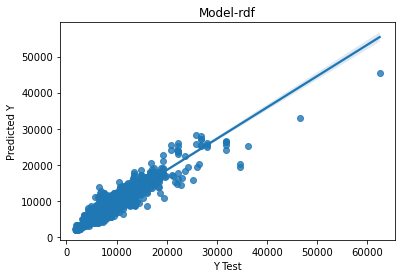
Using the “for” loop, I found the best Random state. The R2 score for KNN Regressor Model was 77.52%. After Hyperparameter tuning, my score increased to 82.59% and my cross-Validation score for the tuned model was 79.40%. Regplot os KNN Regressor is:

Chart, scatter chart

Description automatically generated

**4. Random Forest Regressor:**

Using the “for” loop, I found the best Random state. The R2 score for the Random Forest Model was 91.52%. After Hyperparameter tuning, my score increased slightly to 91.59% and my cross-Validation score for the tuned model was 88.64%. Regplot of random forest regressor is:



**5. XGB Regressor:**

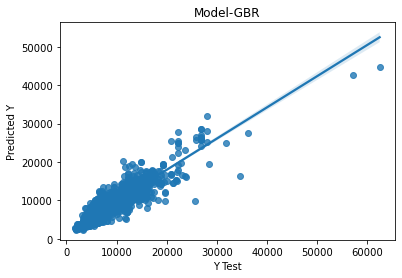
Using the “for” loop, I found the best Random state. The R2 score for the XGBRegressor Model was 92.56% and my cross-Validation score was 89.53%. After Hyperparameter tuning, my score did not increase. Regplot of XGB is:

Chart, scatter chart

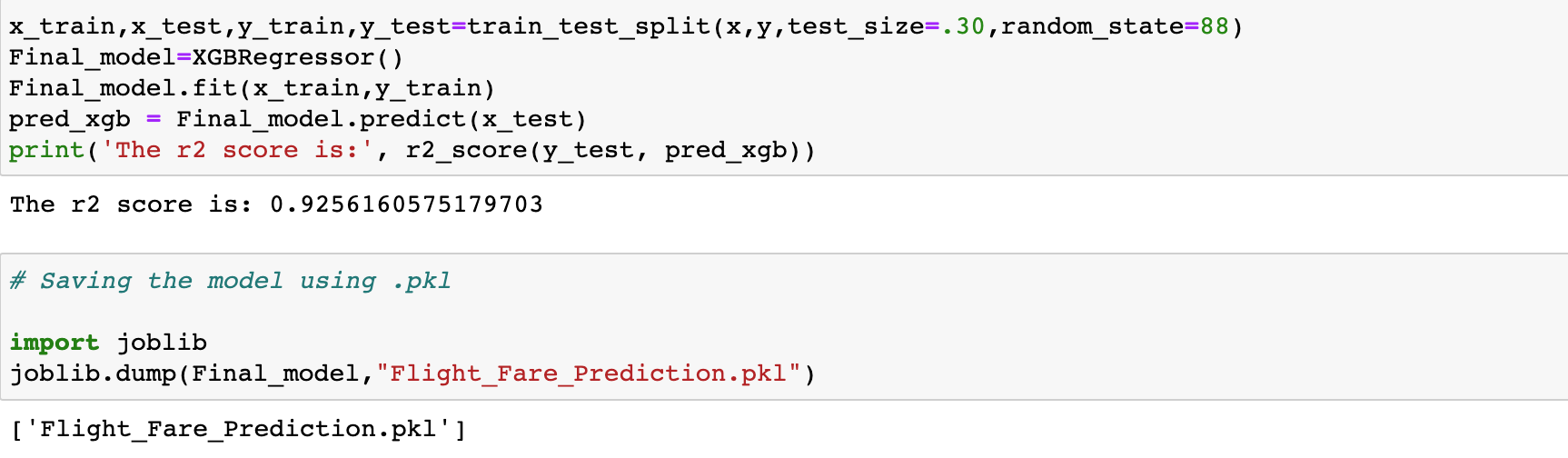
Description automatically generated

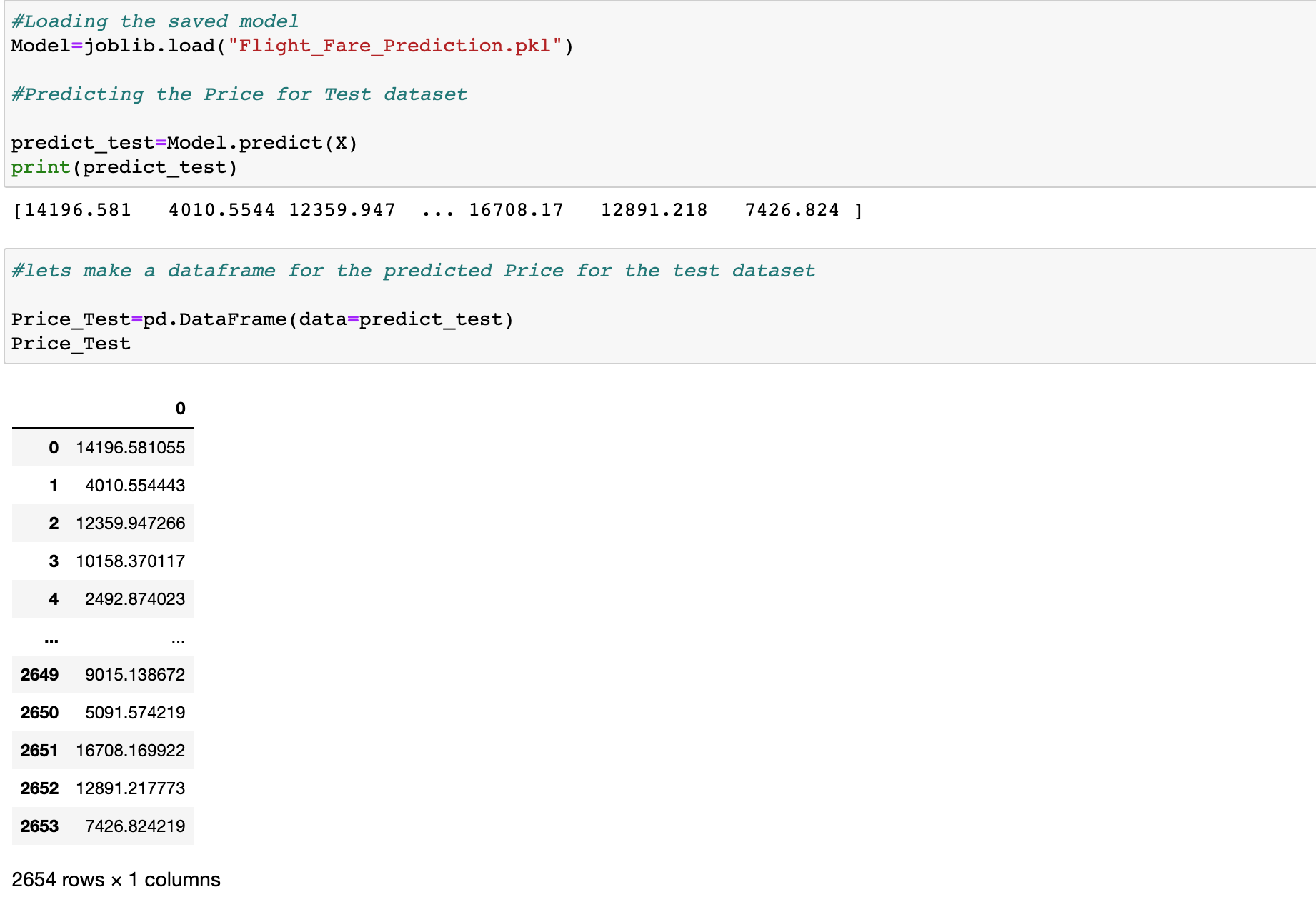
**6. Gradient Boosting Regressor:**

Using the “for” loop, I found the best Random state. The R2 score for the Gradient Boosting Regressor Model was 84.74% and my cross-Validation score was 81.40%. Regplot of Gradient Boosting is:



Saving the Best Model

I have concluded that the XGB Regressor model is my Final Model because the R2 score and Cross Validation score is the highest among all others.

Using joblib, Model is dumped in a pkl file and can be used to predict the flight fare in future. We have our test dataset flight fare to be predicted. After removing Outliers, our test dataset has 2654 rows. I have predicted Target “Price” for the Test dataset.

The predicted Price has been added to the test Dataset and saved as a csv file.

CONCLUSION

As we saw from above steps, I have cleaned and preprocessed my dataset well before building the Model. Using the Trained dataset, I have built the model and predicted Fares of the Test Dataset. All necessary steps have been explained in this article. My Final Model is XGBRegressor Model with R2 score of 92.56% and Cross Validation Score of 89.53%. I believe that I have successfully built the Model which anyone can predict the price of Domestic flights in India.

I hope you all liked the Article and please feel free to contact me through GitHub [Link](https://github.com/Razni-DS) or my LinkedIn [Profile](https://www.linkedin.com/in/razni-nazeem-2508b1a3/) if you have any Queries.

# **REFERENCES**

1. <https://www.kaggle.com/competitions/air-ticket-fare-prediction>
2. [https://www.techrepublic.com/article/machine-learning-the-smart-persons-guide](https://www.techrepublic.com/article/machine-learning-the-smart-persons-guide/)
3. <https://www.financialexpress.com/lifestyle/travel-tourism/travel-budget-set-to-take-hit-as-flying-becomes-expensive-heres-why/2564400/>
4. <https://navbharattimes-indiatimes-com.translate.goog/business/business-news/government-should-consider-raising-the-ceiling-on-the-maximum-fare-for-domestic-flights-indigo-ceo/articleshow/91933329.cms?_x_tr_sl=hi&_x_tr_tl=en&_x_tr_hl=en&_x_tr_pto=sc>

Text, letter

Description automatically generated