**Flight Price Prediction**

**Using**

**Machine learning**

So after completing few months of learning Data Science with Machine Learning its finally time for getting into some project works, which improve with experience and use of data….

“What does Machine Learning Do?”

*Machine learning algorithms build a model based on sample data (training data), and make predictions or decisions using the model without being programmed to do so.*

I have basically been assigned with few projects among which I will be explaining about my “Flight Price Prediction Project”.

**Introduction:**

Here i we will discussing about the whole end to end project and its utility towards the future. There are various factors/features which impact the prices of flights — distance, flight time, number of stops etc. These factors help create a pattern to decide the price of a flight, and the machine learning models get trained on this pattern to make the predictions in future, automating the process and making the process quicker.

**Problem Definition:**

Flight ticket prices can be something hard to guess, today we might see a price, check out the price of the same flight tomorrow, and it will be a different story.

To solve this problem, we have been provided with prices of flight tickets for various airlines between the months of March and June of 2019 and between various cities, using which we aim to build a model which predicts the prices of the flights using various input features.

Content of the Blog:

1. Load the Dataset
2. Summarize the data set:
3. Data Inspection
4. Data Cleaning and Handling
5. Handling Categorical variables
6. Exploratory Data Analysis
7. LabelEncoding
8. Outlier Detection
9. Model Building
10. HyperTuning
11. Conclusion
12. **Load the Dataset:**

* The dataset (in csv format)needs to be imported . But before we import the dataset we need to import the required and essential libraries such as
* Pandas
* Numpy
* Matplotlib
* Seaborn

The dataset is available on the below link:

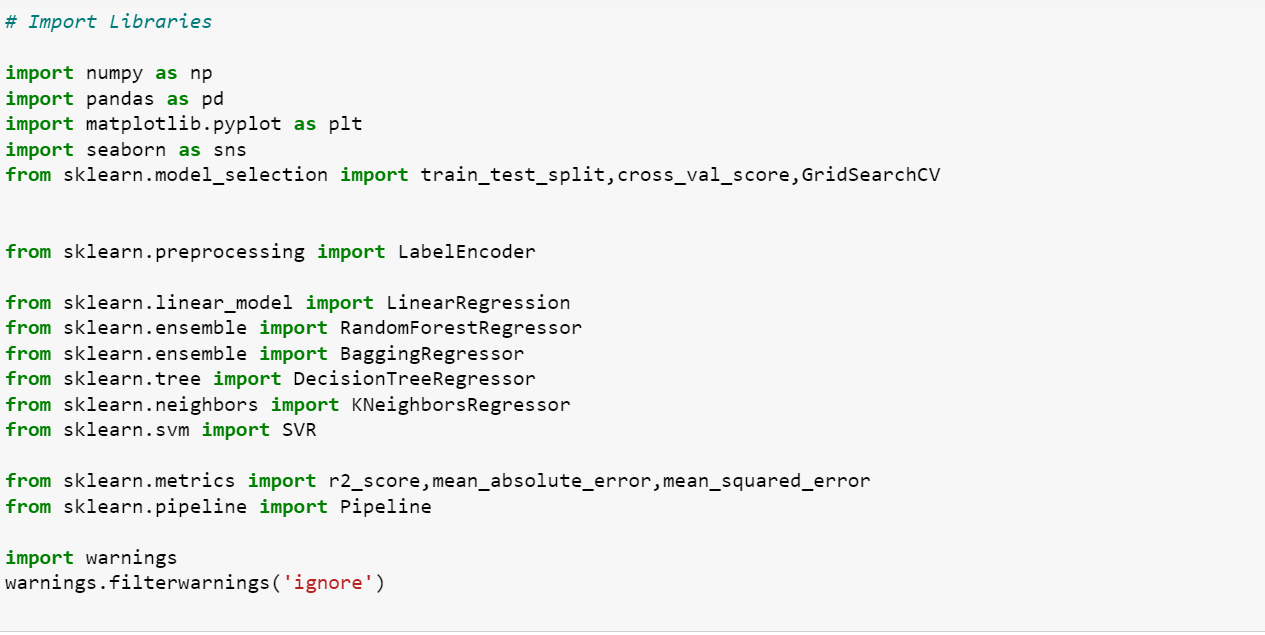
<https://github.com/dsrscientist/Data-Science-ML-CapstoneProjects/blob/master/Flight_Ticket_Participant_Datasets-20190305T100527Z-001.zip>

**The variables of the dataset are as following:**

* Airline: The name of the airline.
* Date\_of\_Journey: The date of the journey
* Source: The source from which the service begins.
* Destination: The destination where the service ends.
* Route: The route taken by the flight to reach the destination.
* Dep\_Time: The time when the journey starts from the source.
* Arrival\_Time: Time of arrival at the destination.
* Duration: Total duration of the flight.
* Total\_Stops: Total stops between the source and destination.
* Additional\_Info: Additional information about the flight
* Price: The price of the ticket
* 10 Independent variables with Price as target variables

I have two datasets Train and Test, so I need to do all the pre-processing on both the datasets. The difference between both the datasets is that train dataset has dependent variable(y) and test dataset does not have (y).

* 1. Import important libraries
  2. Import Regressor Libraries

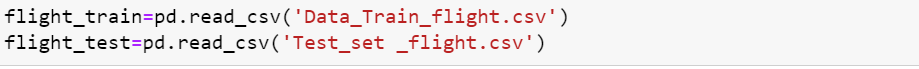


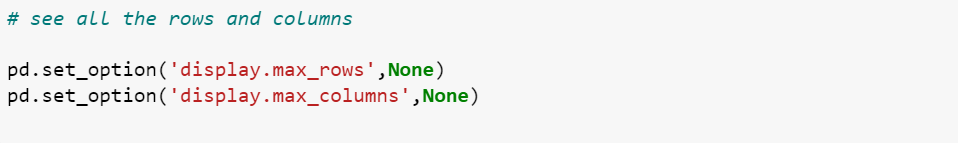
1. **Summarize the data set:**

Now it is time to take a look at the data.

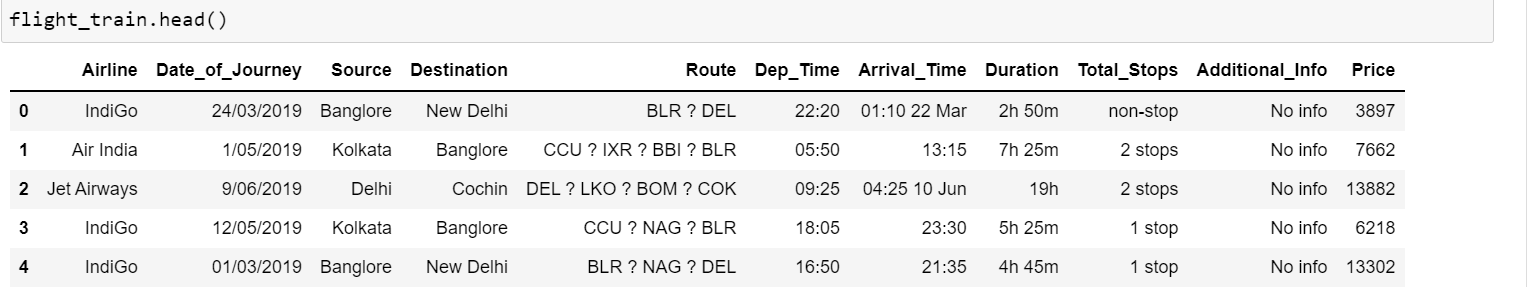
In this step we are going to take a look at the data a few different ways:

1. Dimensions of the dataset.
2. Peek at the data itself.
3. Statistical summary of all attributes.
   1. Load the Dataset and view all rows and columns

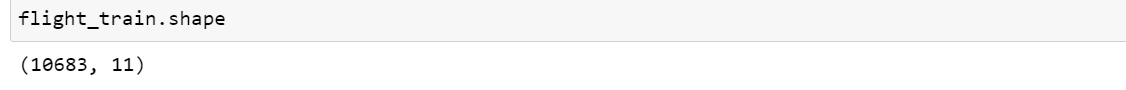




Its always good to look up at the data ,so used head function:



* 1. **Dimension of my Dataset**



After running the above code you will get a report as shown in the above figure. This report contains various sections or tabs.  ‘Overview’ section of this report provides us with all the basic information of the data we are using. For the current data we are using we got the following information:

Number of variables = 11  
 Number of rows = 10683  
 Number of categorical type of feature = 10  
 Number of numerical type of feature = 1

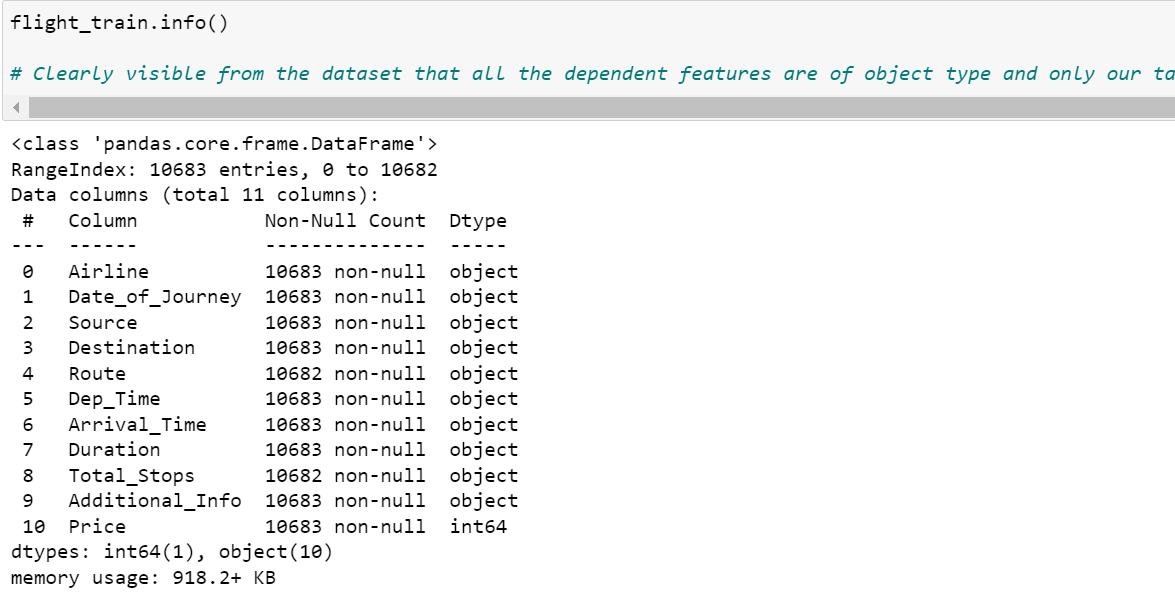
Few observations about some of the variables:

1. ‘**Price**‘ will be our dependent variable and all remaining variables can be used as independent variables.

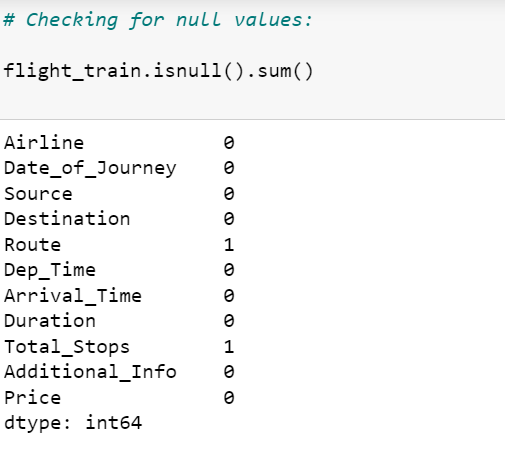
2. ‘**Total\_Stops**‘ can be used to determine if the flight was direct or connecting.

3. **Data Inspection**

* 1. Getting information regarding datatypes and non-null values



* 1. **Checking for null values**

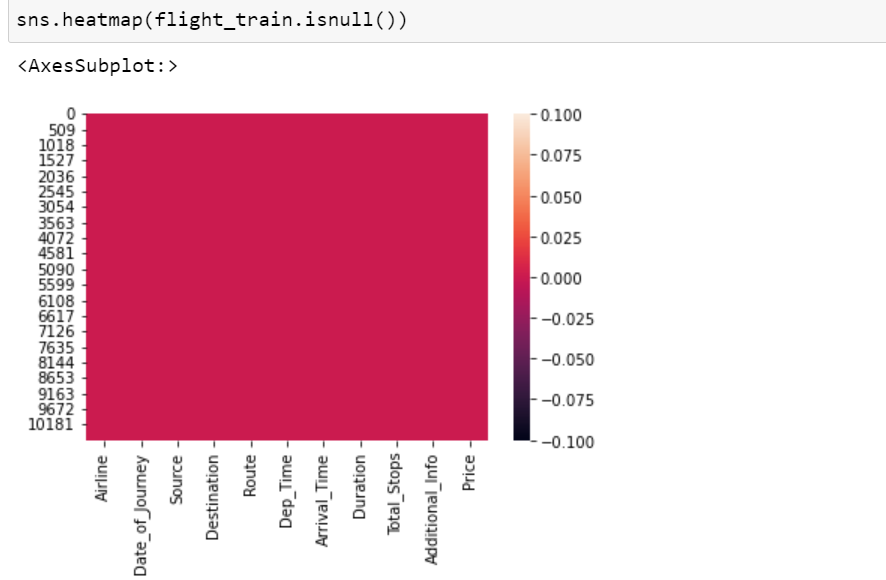


Very less i.e 1 in route column and 1 in total stop we have missing values which will be replaced or can be dropped even as the data is negligible.

* 1. **Handling Missing Values**

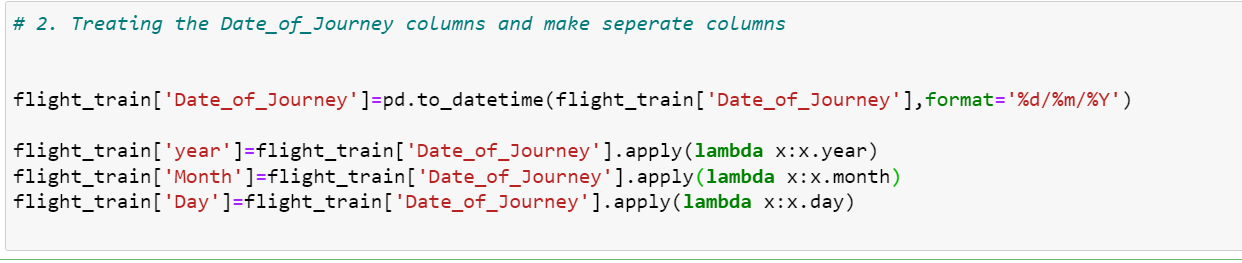


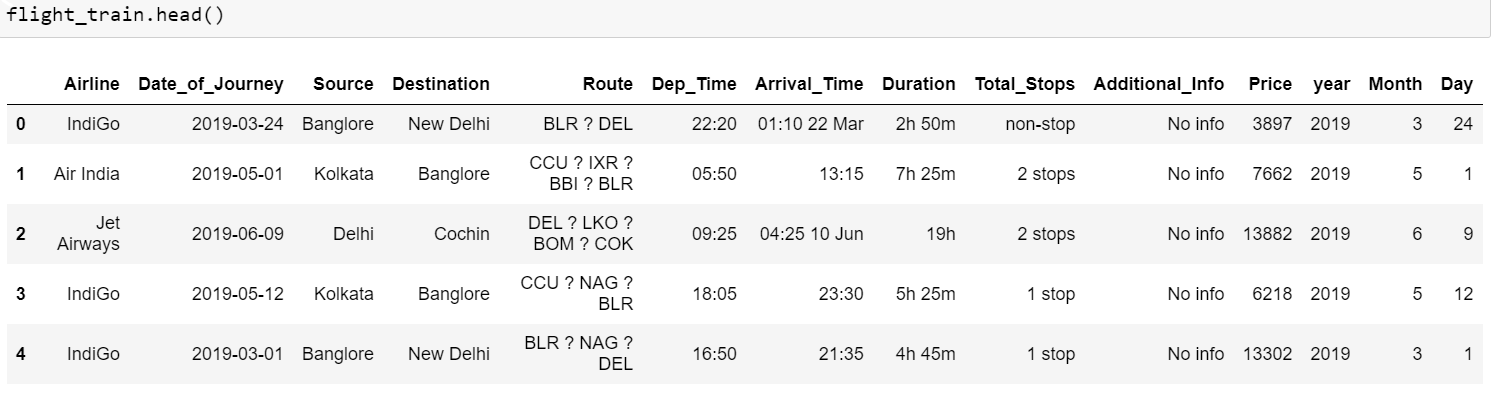
Using the above steps, we were successfully able to treat all the missing values from our data.



4. **Data Cleaning Handling**

4.1 **Handling Date and Time Variables**





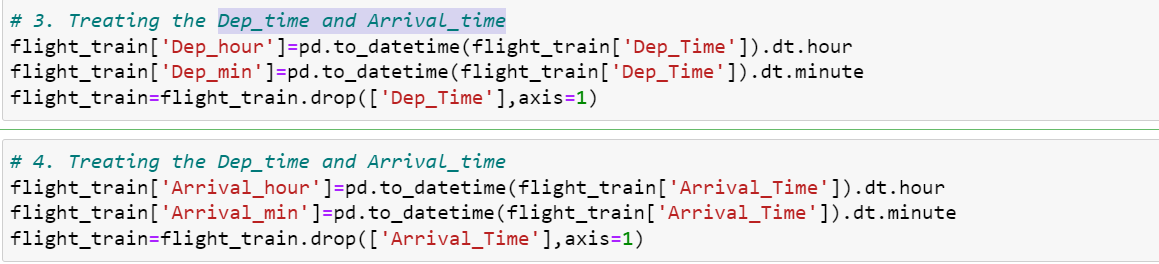
I have “Date\_of\_Journey” a date type variable and ‘Dep\_Time’, ‘Arrival\_Time’ that captures the time information.

We can extract ‘Day’ and ‘Month’ from the ‘Date\_of\_Journey’ variable. ‘Journey day’ shows the day of the month on which the journey was started.

Then drop the original column “Date\_of\_Journey”.

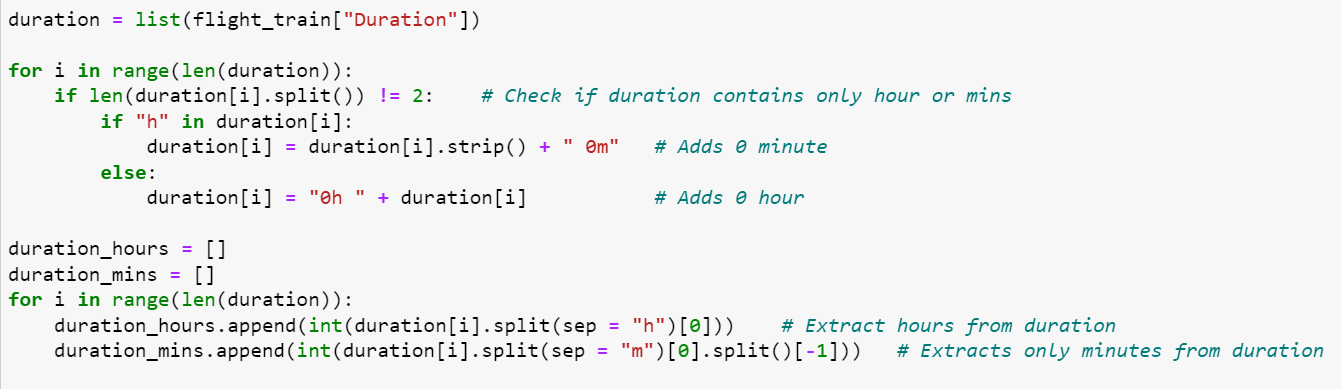
4.2 **Handling Dep\_time and Arrival\_time**

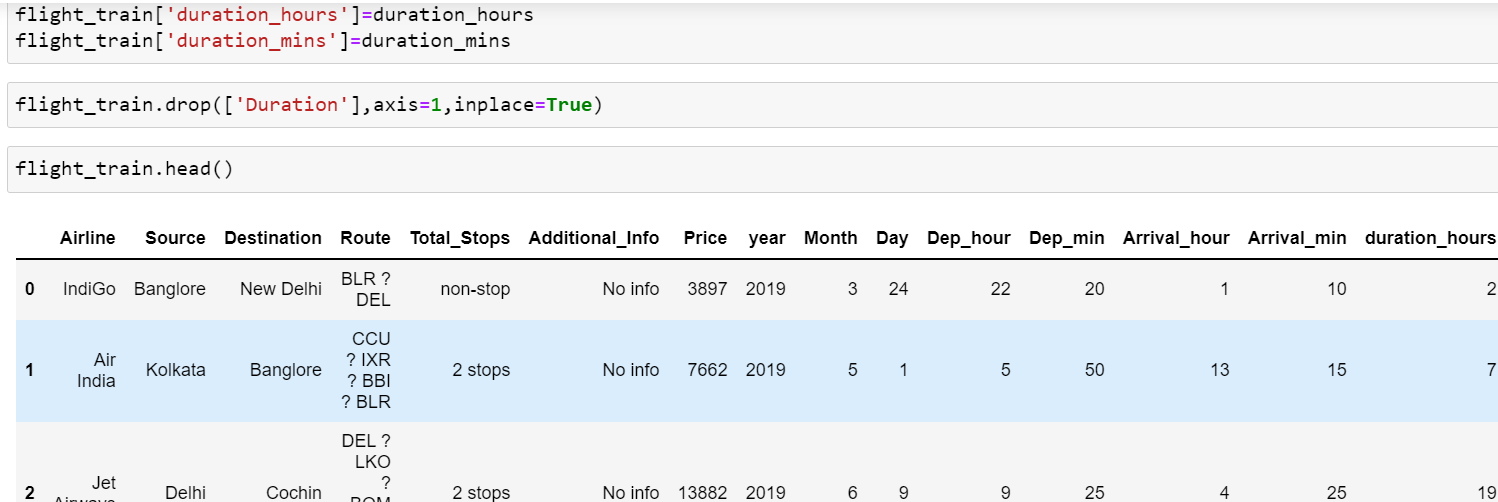
Similarly, we can extract ‘Dep\_hour’ and ‘Dep\_min’ as well as ‘Arrival\_hour and ‘Arrival\_min’ from ‘Dep\_Time’ and ‘Arrival\_Time’ variables respectively.



4.3 **Handling Duration**

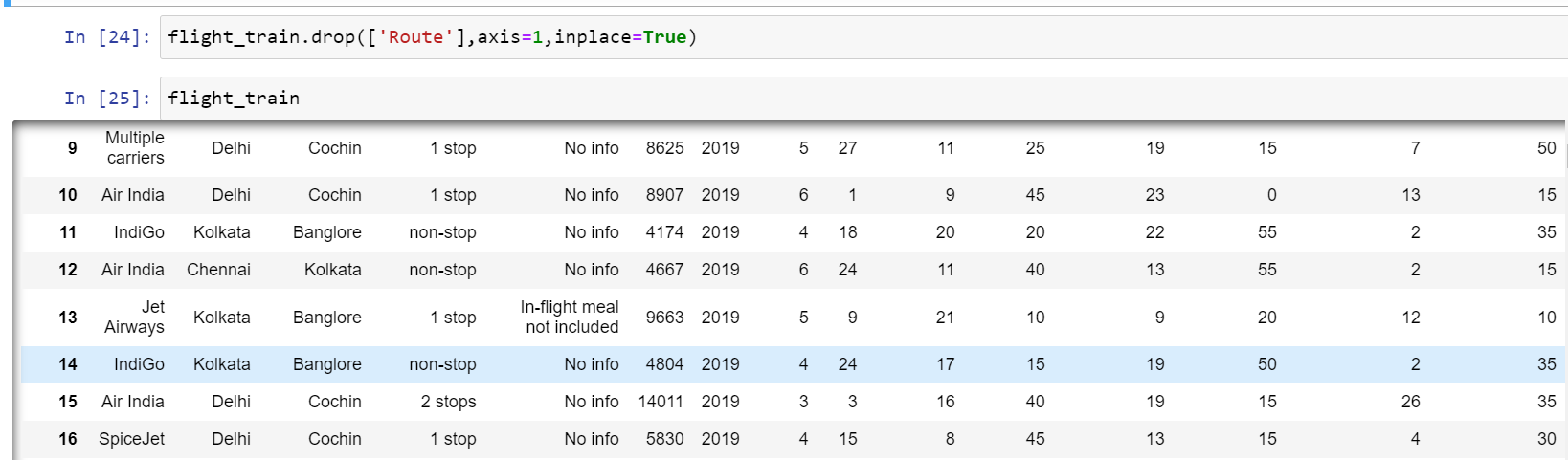
We can extract ‘Duration\_hours’ and ‘Duration\_minutes’ separately from the ‘Duration’ variable.



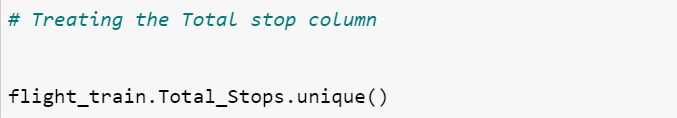


4.4 **Handling Route variable**

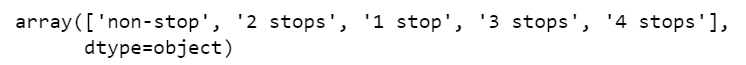
Route variable represents the path of the journey. Since the ‘Total\_Stops’ variable also gives the information if the flight is direct or connected so I have decided to drop this variable instead of treating it.

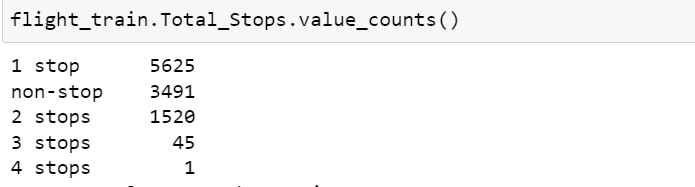


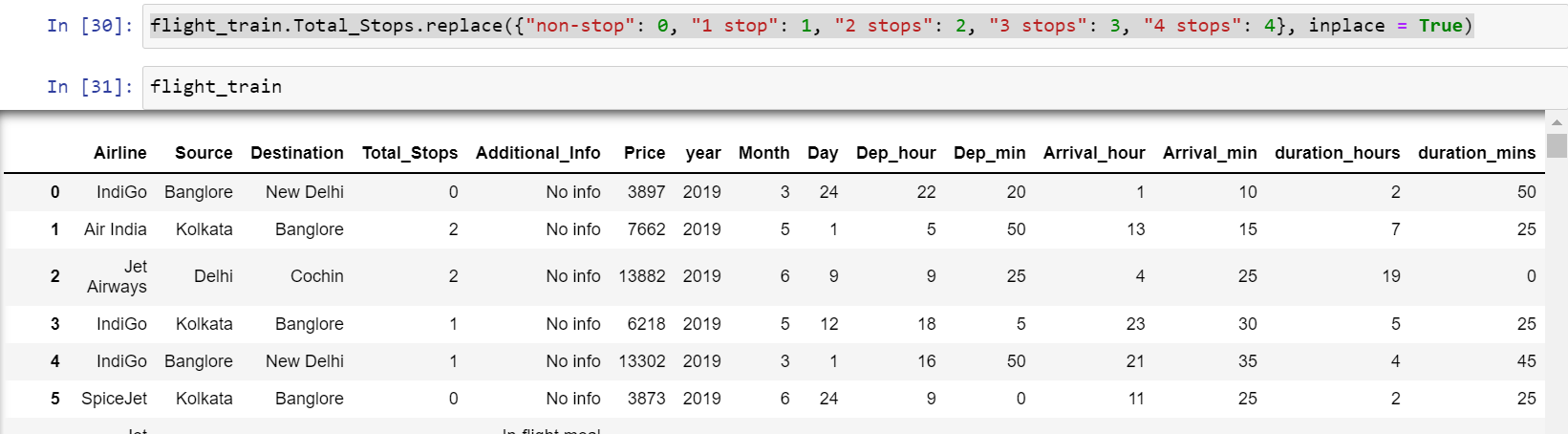
4.5 **Handling Total\_stops variable**



Output:



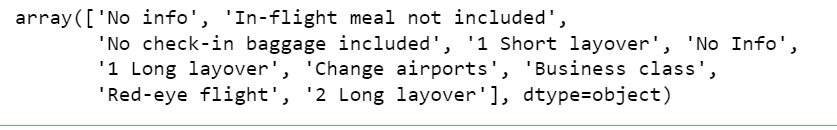




4.6 Handling Additional\_Info column



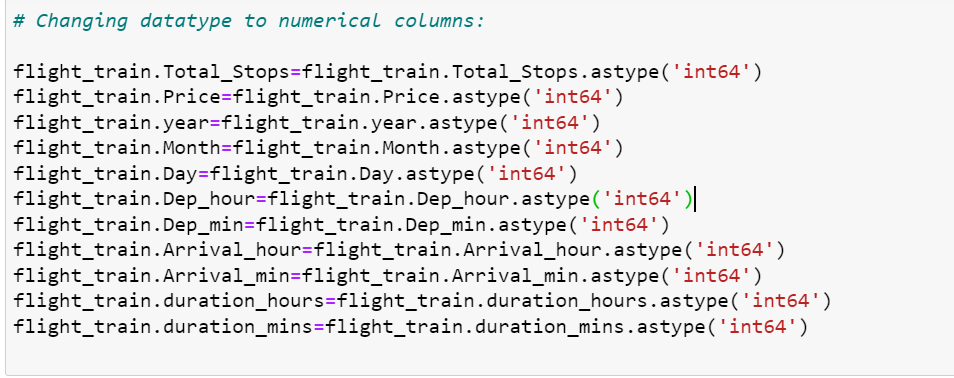
Output:





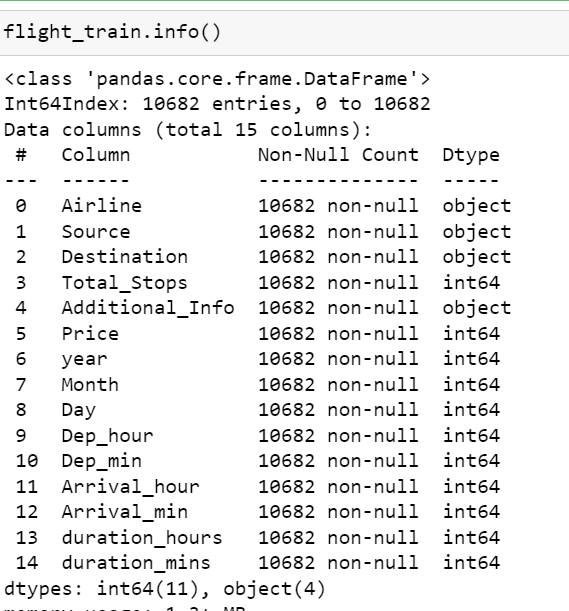
* 1. Handling Categorical variables

Now all the numerical variables are changed into int type



We again check the info in our data and find out that the dataset still has data types for multiple columns as ‘object’, where it should be int.

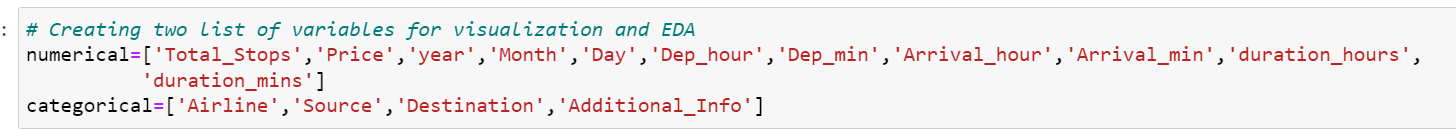
After we change the datatypes the data information looks like this:



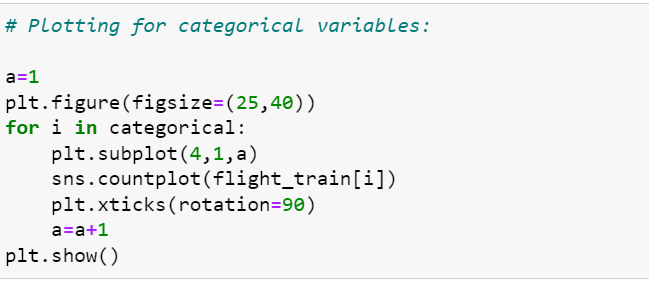
* 1. **Exploratory Data Analysis:**

For starting with Data Analysis the whole dataset we are dividing into two parts:

**Categorical and Numerical**

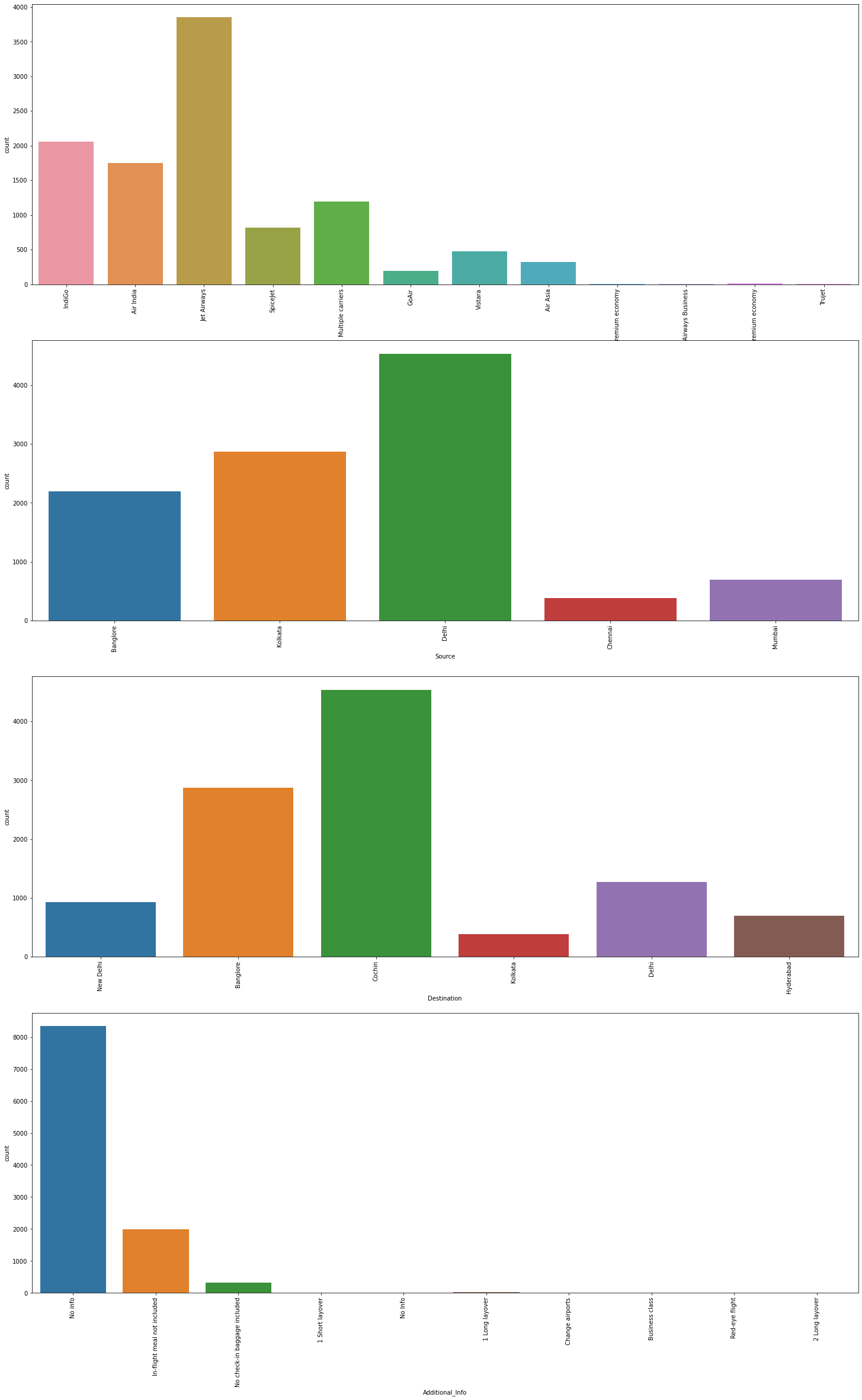


6.1 **I have plotted for categorical data(Univariate Analysis )**

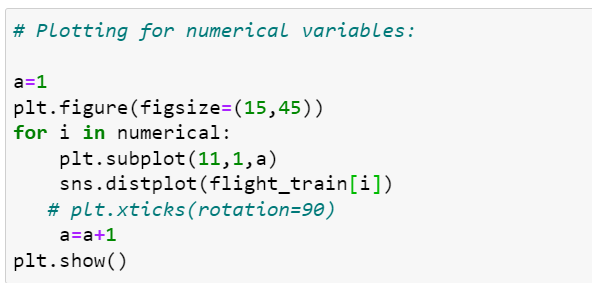


**Observation of the below plotting:**

* Jet Airways is having highest frequency of flights followed by Indigo , AirIndia and so on.
* Most of the flights take of from Delhi and least no from Chennai.
* Maximum no of flights land in Chochin whereas minimum no at kolkata.
* maximum no i.e 80 % of row with no info.

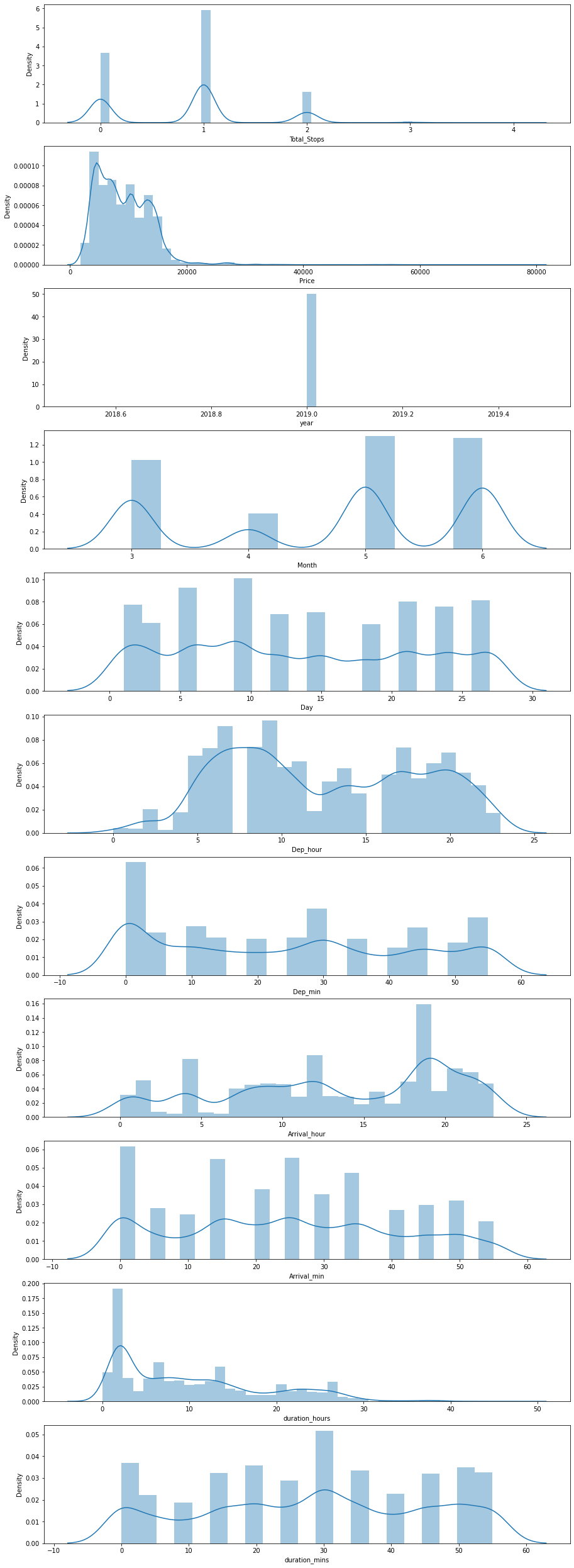


6.2 **I have plotted for numerical data(Univariate Analysis)**



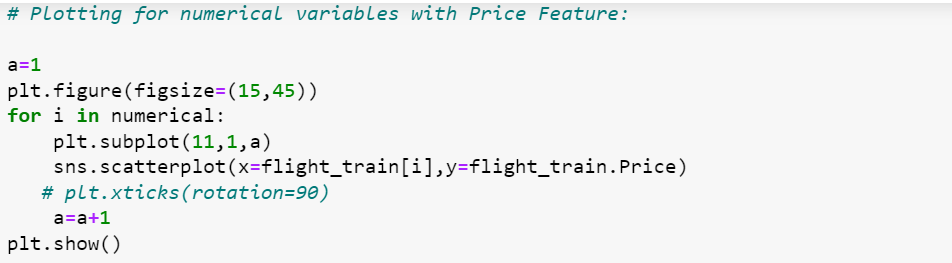
# **Observation from the below plotting:**

* One stop flights are more in number followed by no stop and 2 stop.
* Most of the flight have fare between 1500 to around 20000.
* Year column is having only one value that can be ignored nad dropped in further prediction.
* Mostly people travel in May and June whereas minimum in April.
* Mostly people are traveling between 2,3 day to 26,27 of the month.
* Early morning flights are more in number almost till 11am. Then afternoon flights are less in number.
* Most of the flights take off at 00 mins.
* Most of the flights arrive between 18:00 10 19:00 and its directly related to the take-off.
* Most of the flight arrive at 00 mins and between 00 to 50 min max.
* Most of the flights are taking 2 to 3 hrs followed by 5 to 6 hrs.
* Maximum minute duration is 30 mins.



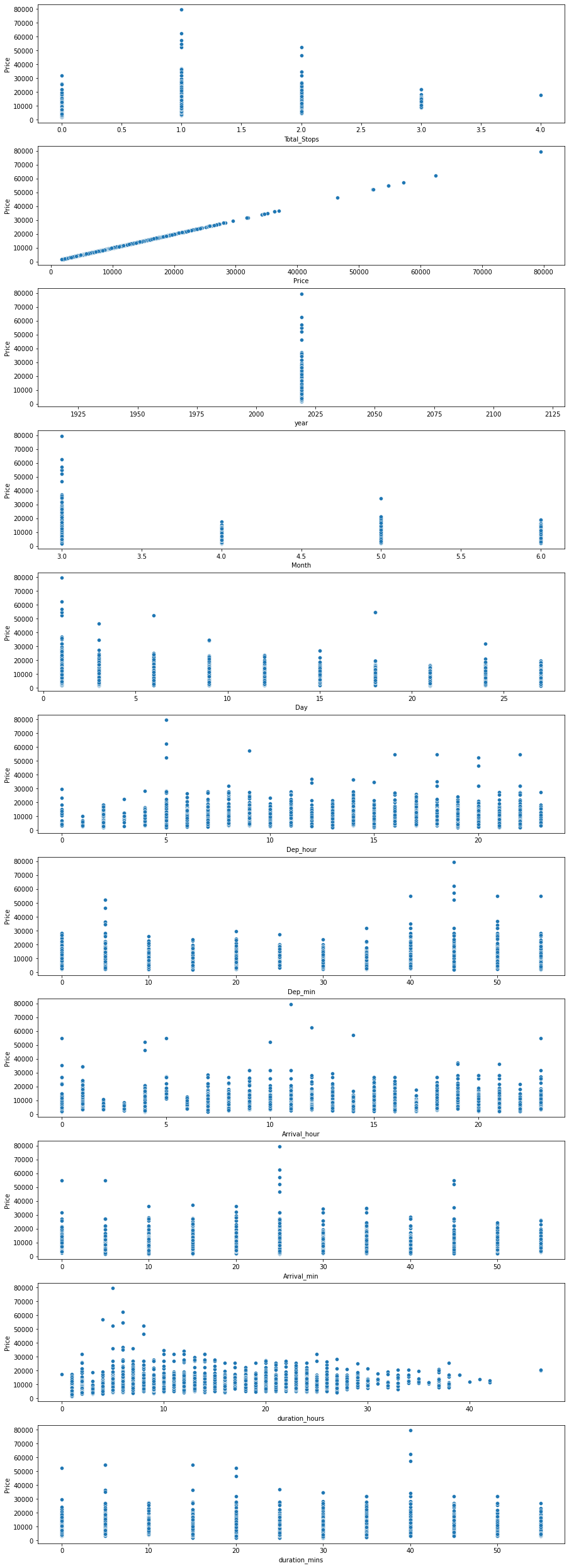
6.3 **Now I have plotted for Numerical variables with comparison with Price feature**

**(Bi-Variate Analysis)**



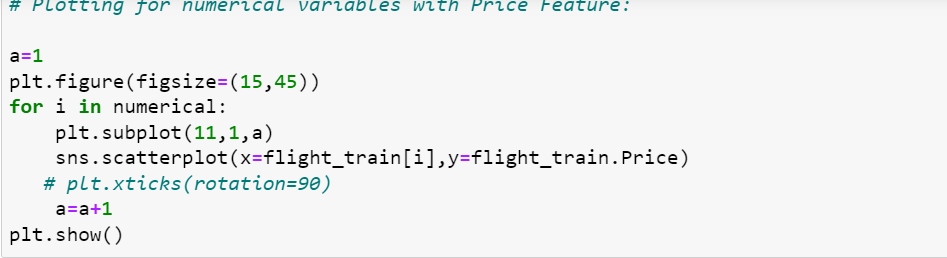
**Observation for the below plots:**

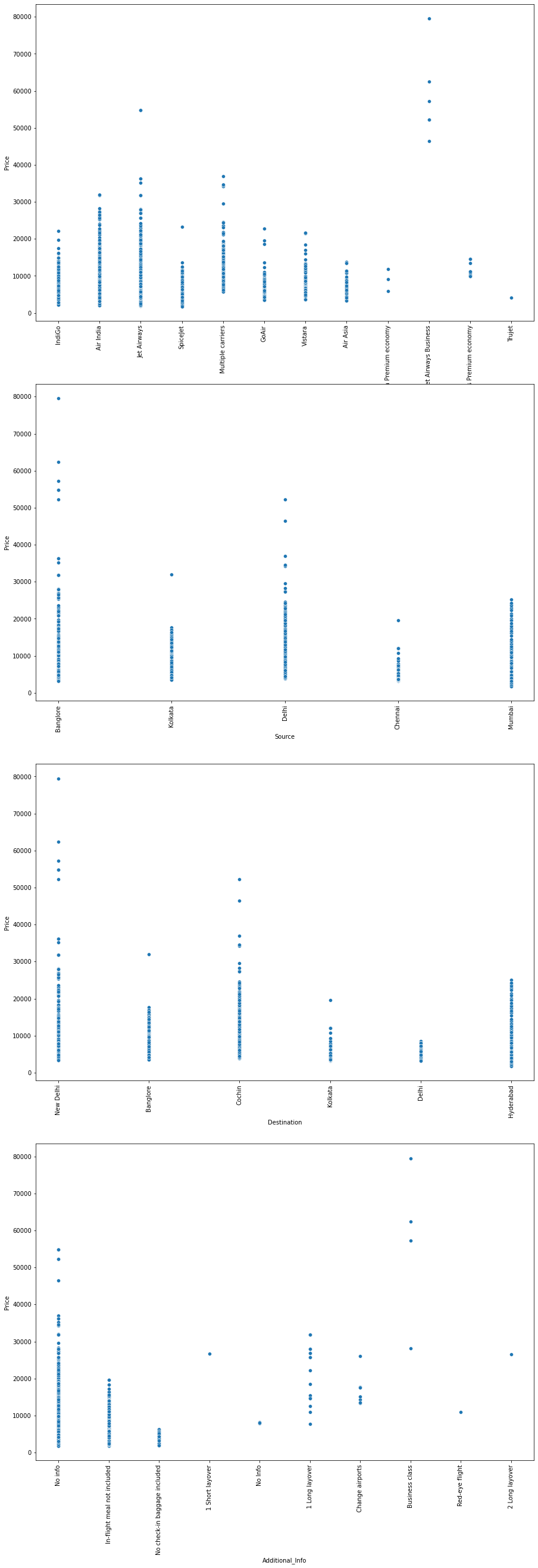
* Total stop is inversely proportional to the no. of stop.
* Price is highest in the month of march followed by May and April.
* Early morning flights are higher in price.
* Flights taking less duration are slightly higher in price.



6.4 **Now I have plotted for Categorical variables with comparison with Price feature**

**(Bi-Variate Analysis)**





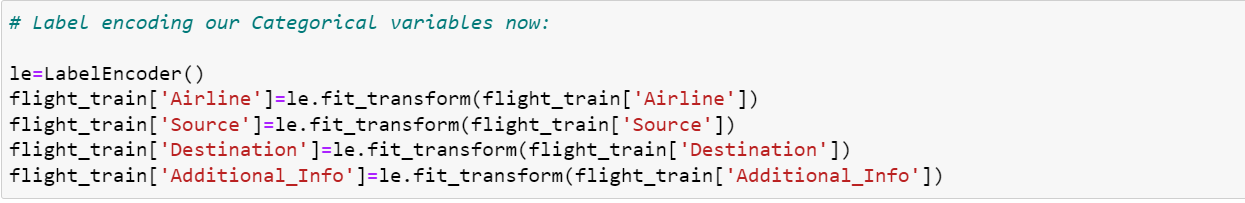
**Observation for the above plots:**

* JetAirways having maximum price followed by AirIndia..
* Flights taking off from Banglore are having high price followed by Delhi and Mumbai.
* Flights with destination as New Delhi has highest price followed by Chochin ,Hyderabad.
* Additional Info is not helping us much to extract information as it shows price highest with no info.

7. **LabelEncoding**

The categorical data are LabelEncoded to change it into numerical data.

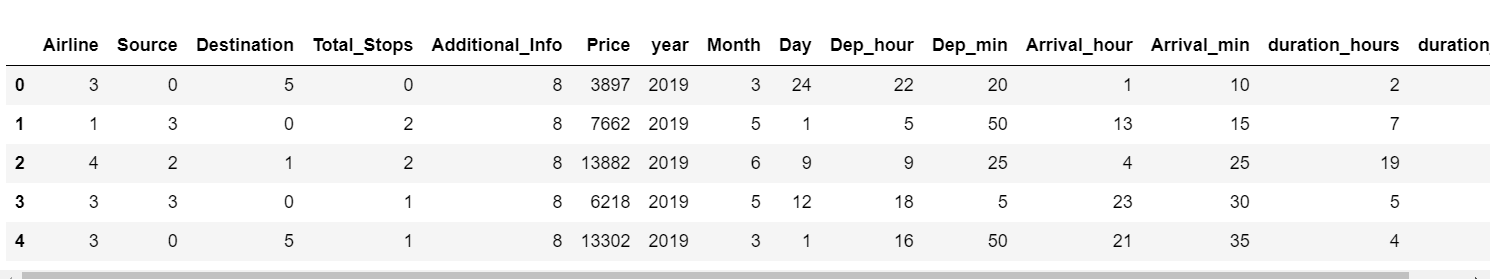
Its always important to change the categorical data as because we cannot fit categorical data for training the our model.



Lets peep on the dataset once now:

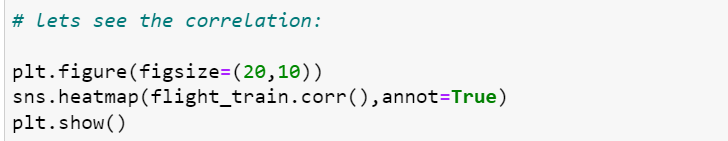


**Output:**

\

Now our seems to be absolutely clean and numerical ,as well as ready for model building

7.1. See the correlation once before starting the Model building:



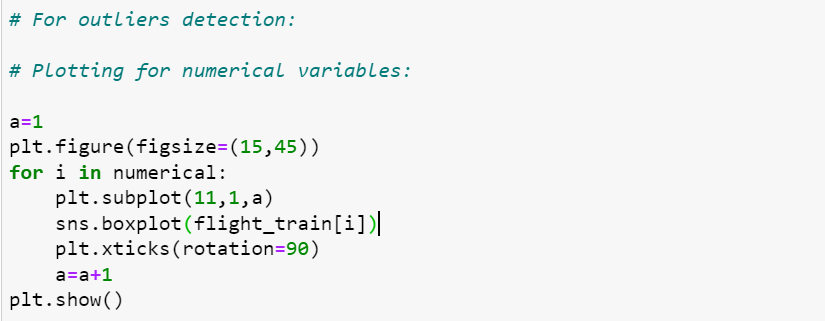
Output

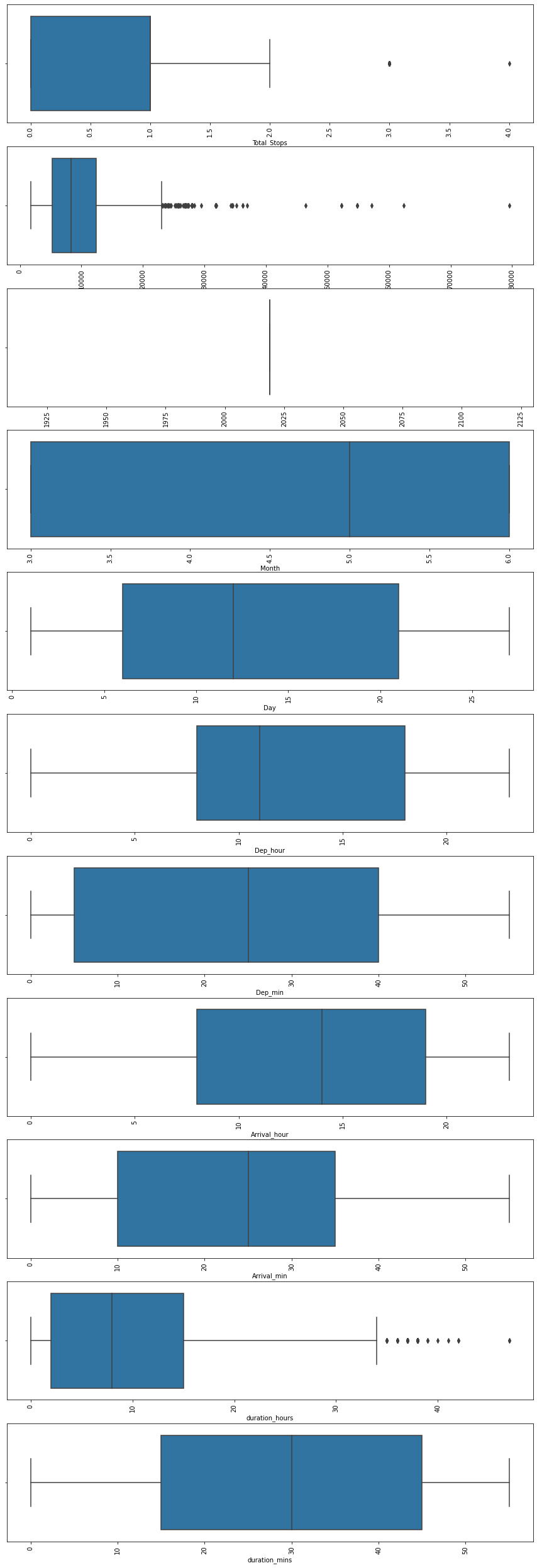


**Observation:**

From the above correlation map we can make out that Total\_stops and Duration are highly related.

1. **Outlier Detection:**



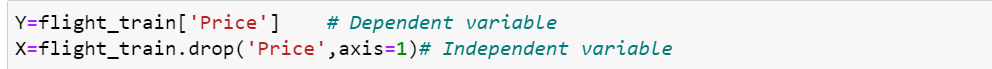


**Observation:**

* We make the below conclusions –
* Outliers are present in Total hours, Total stops and price
* We will not remove outliers from total stops since price is impacted by number of stops
* We will not remove the data with high number of hours, increase in number of hours shows a price pattern in the above graphs plotted for EDA.

1. **Model Building**

**9.1 Splitting the Train Data and Test Data**

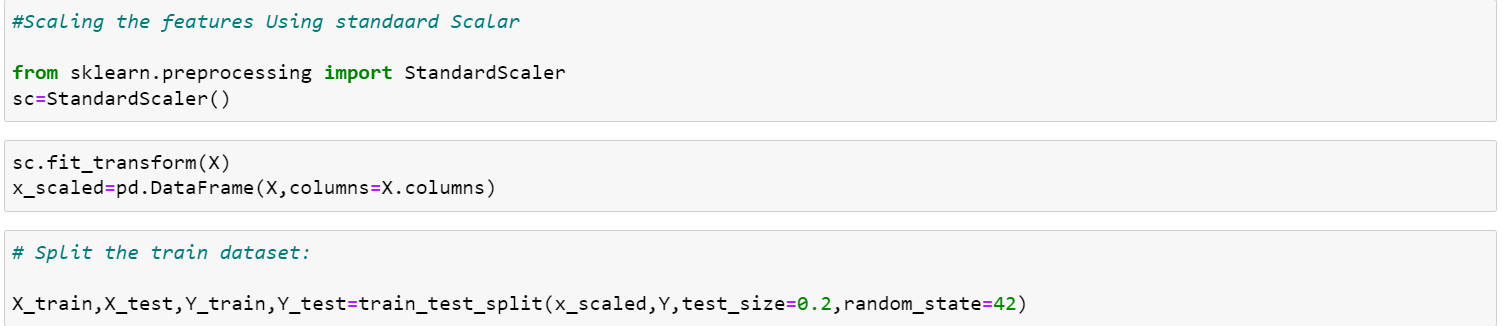
****

**9.2 Scaling the data**

The next step is to bring the data to a common scale, since there are certain columns with very small values and some columns with high values. This process is important as values on a similar scale allow the model to learn better.

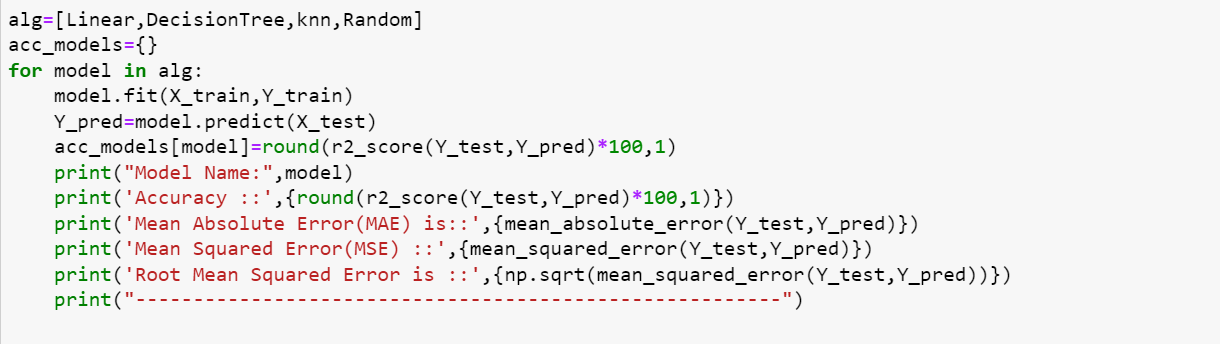
We use standard scaler for this process –

**‘**StandardScaler follows Standard Normal Distribution (SND). Therefore, it makes mean = 0 and scales the data to unit variance’

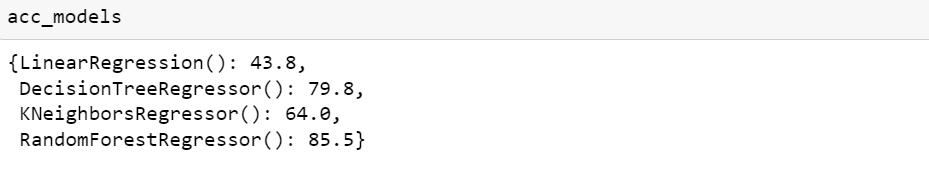
****

**9.3 Fitting the Regressor Models**

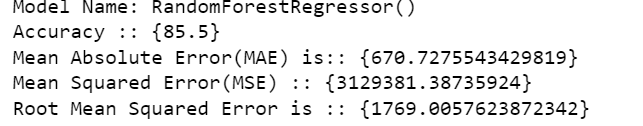
****

****

**The above code I am using to fit all my regressor algorithms.**

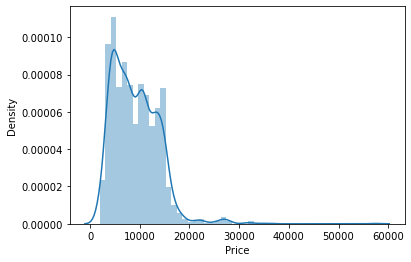
****

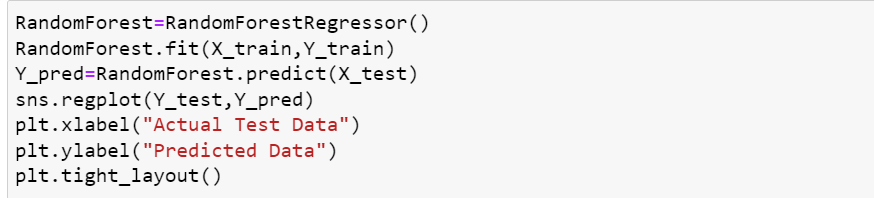
**We achieve the best score with Random Forest i.e 85.5%**

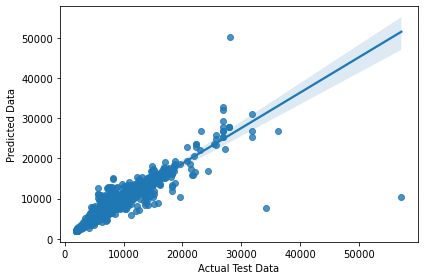
****

**Lets see the plot for test data**

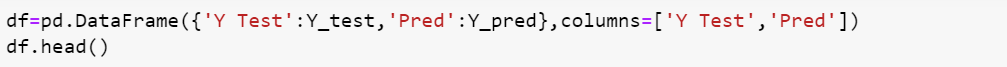
****

****

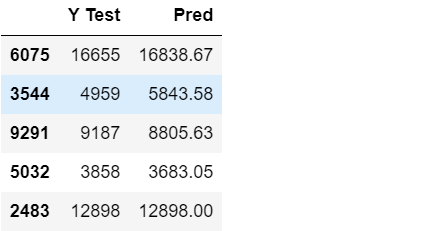
****

****

**9.4 Built a dataframe to compare the test data and predicted data**

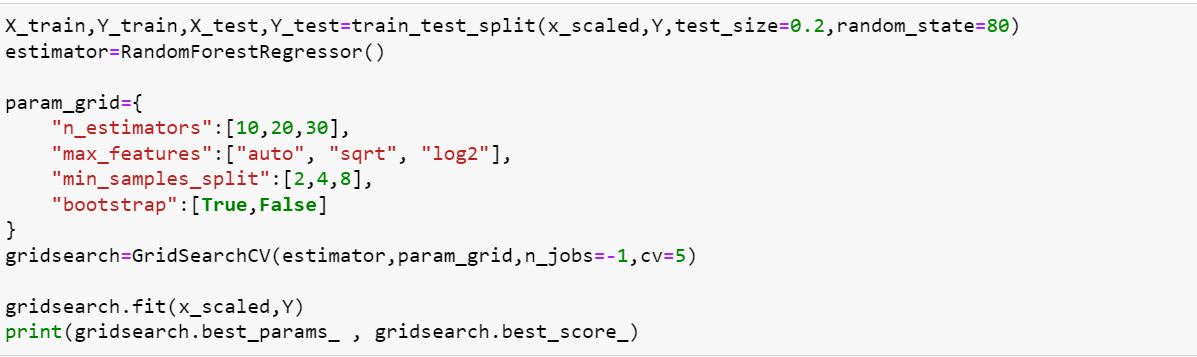
****

**Output:**

****

**10. HyperTuning the Random Forest Model**

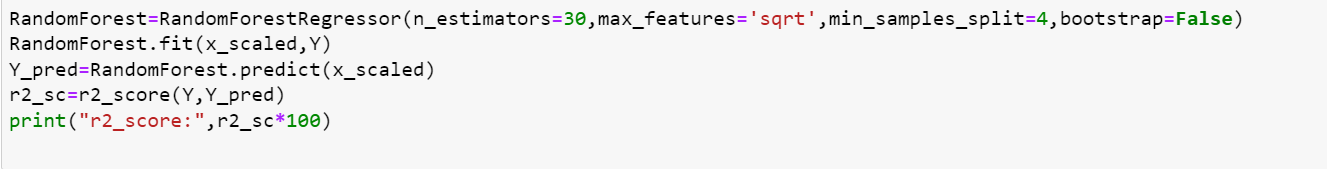
GridSearch CV is a technique used to validate the model with different parameter combinations, by creating a grid of parameters and trying all the combinations to compare which combination gave the best results. We apply grid search on our model.

****

**The above code give the best param as given below**

****

**Now fitting the model with our best param and see the accuracy**

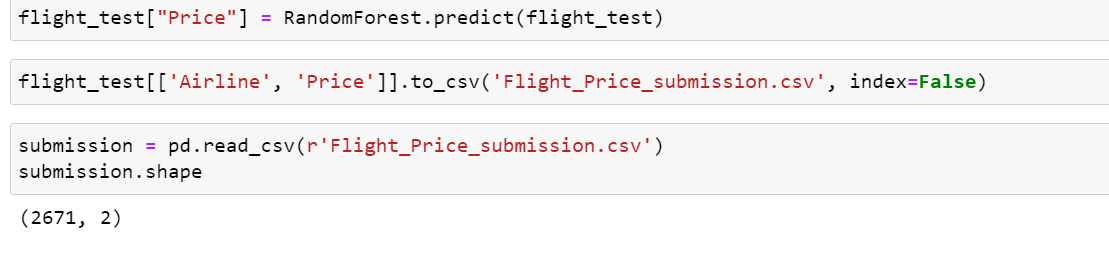
****

****

**The accuracy increases from 85% to 98%**

**Now I have to work on Test Dataset.**

**Same data cleaning part,labelEncoding should be followed for Test Data as well.**

****

**11.Conclusion**

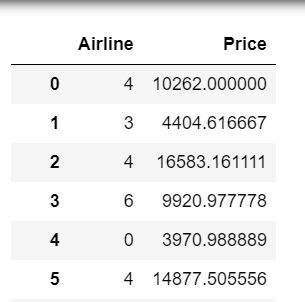
**After performing all data cleaning and label encoding and all the test data is given to the Random Forest Model for predicting the Price .**

**We load the test file, apply all the data modeling processes and operations on our test data similar to what we did with the train data, and then make the final prediction using the saved model object.**

**A new file name “submission” was made to store the Price along with the Airline name.**

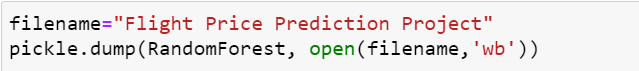
****

**Output:**

****

**10.1 Importing the pickle and saving the model**

****

****

**Hence, at the end, we were successfully able to train our regression model ‘Random Forest Regressor’ to predict the flights of prices with an r2\_score of 98%, and have achieved the required task successfully.**