**Flight Fare Prediction: Machine learning project**

In this article, I am going to walk you through how we can train a model that will help us to predict flight fare prices. We will practice the machine learning workflow.

**Problem Statement:**

Flight ticket prices can be something hard to guess, today we might see a price, check out the price of the same flight tomorrow, it will be a different story. We might have often heard travellers saying that flight ticket prices are so unpredictable. Here we will be provided with prices of flight tickets for various airlines between the months of March and June of 2019 and between various cities.

We can see dataset by going through the below URL-

<https://github.com/dsrscientist/Data-Science-ML-Capstone-Projects>

We have 2 datasets in our data: - Training and Test Dataset

The training set contains the features, along with the prices of the flights. It contains 10683 records, 10 input features and 1 output column — ‘Price’.

The test set contains 2671 records and 10 input features. The output ‘Price’ column needs to be predicted in this set. We will use Regression techniques here, since the predicted output will be a continuous value.

**Below are the Features dataset we have:**

**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

In this dataset, Price is our Target variable that we need to predict.

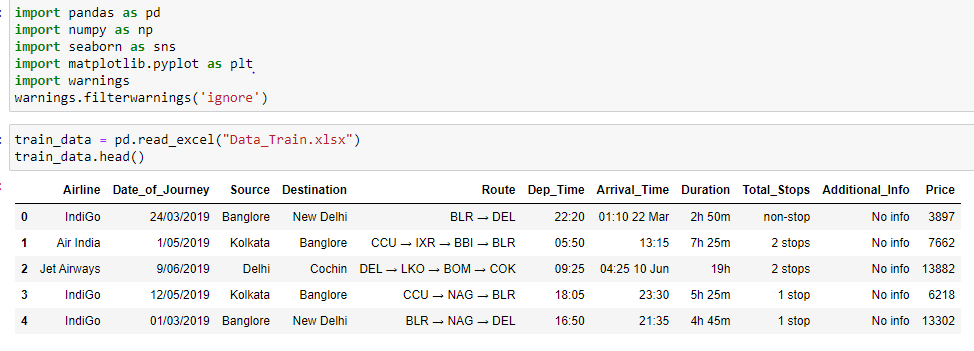
**Machine learning life cycle involves seven major steps, which are given below:**

1. Gathering Data.
2. Data preparation.
3. Data Wrangling.
4. Analyse Data.
5. Train the model.
6. Test the model.
7. Deployment.
8. **Gathering Data**

Now, we will start with the task of machine learning to predict Flight fare. I will start by importing all the necessary libraries that we need for this task and import the train dataset first.

1) Importing Required libraries

2) Importing the train dataset



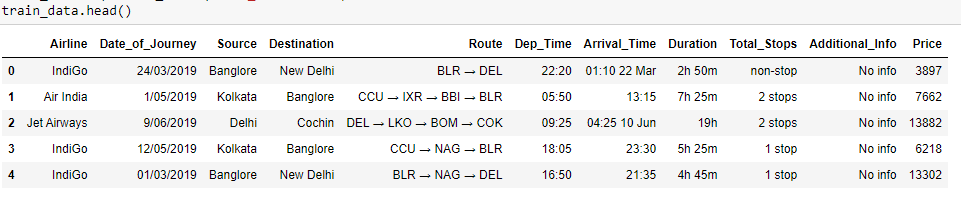
## **Data Preparation**

Data preparation is the process of cleaning and transforming raw data prior to processing and analysis. It is an important step prior to processing and often involves reformatting data, making corrections to data and the combining of data sets to enrich data.

## [**Exploratory Data Analysis**](https://medium.com/p/93da3958eb95/edit#e6a5)

EDA is the process of investigating the dataset to discover patterns, and anomalies (outliers), and form hypotheses based on our understanding of the dataset. EDA involves generating summary statistics for numerical data in the dataset and creating various graphical representations to understand the data better.

## The first thing that we can do when tackling a data science problem is getting an understanding of the dataset that we are working with. First we will see Key observations and trends in the data were noted down. For this we can use **train\_data.head(), train\_data.shape, train\_data.dtypes, train\_data.info(), train\_data.columns, train\_data.describe().**



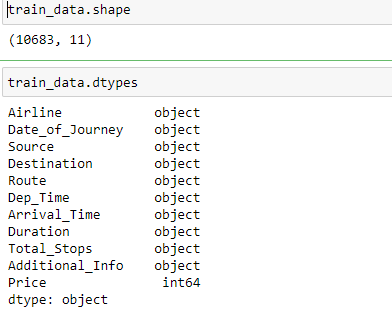
By observing the Train dataset, we can see that:

The **Route** column contains a list of cities which we will need to separate, since we would have multiple combinations in our dataset.

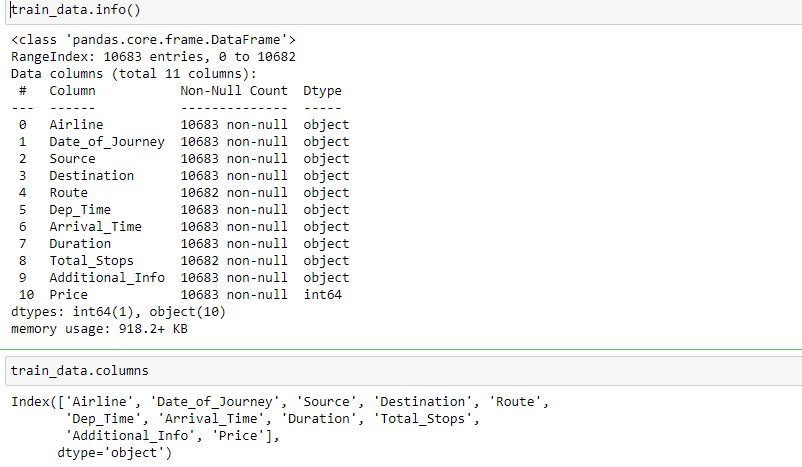
The**Arrival\_time**column has dates attached along with, which we will need to separate.

The **Duration, Date\_of\_Journey** column are in a string format, which need to convert to integer type.

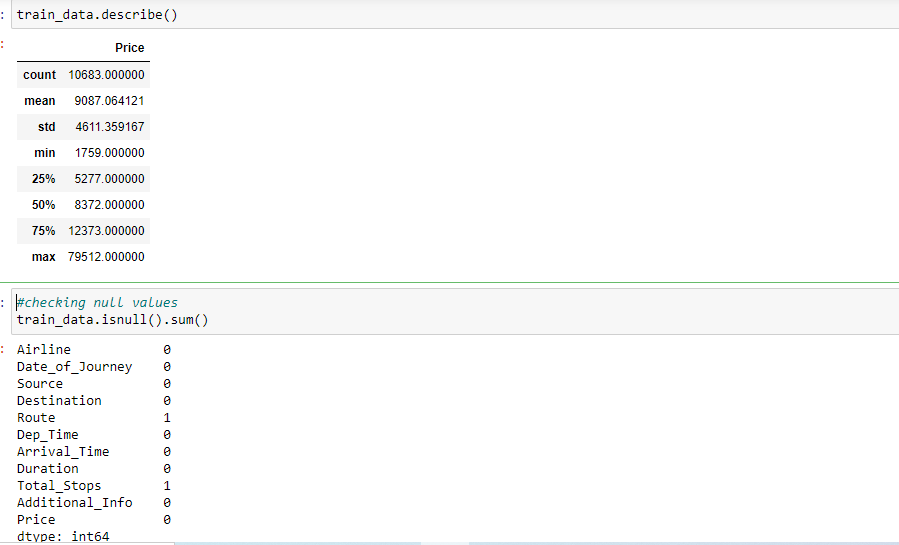
The**Total\_stops** column has word ‘stops’ added along with the number of stops, and **Dep\_Time, Duration** columns are also not in an appropriate form which we need to convert into integer.



By using shape() method, we can see the numbers of rows and columns of our dataset. And dtypes() method will show the data types of every column. we can see that in train\_data.info() all the dependent columns are “object” data type so as to use these columns properly for model we have to convert these data type into appropriate form which we’ll see in feature engineering.

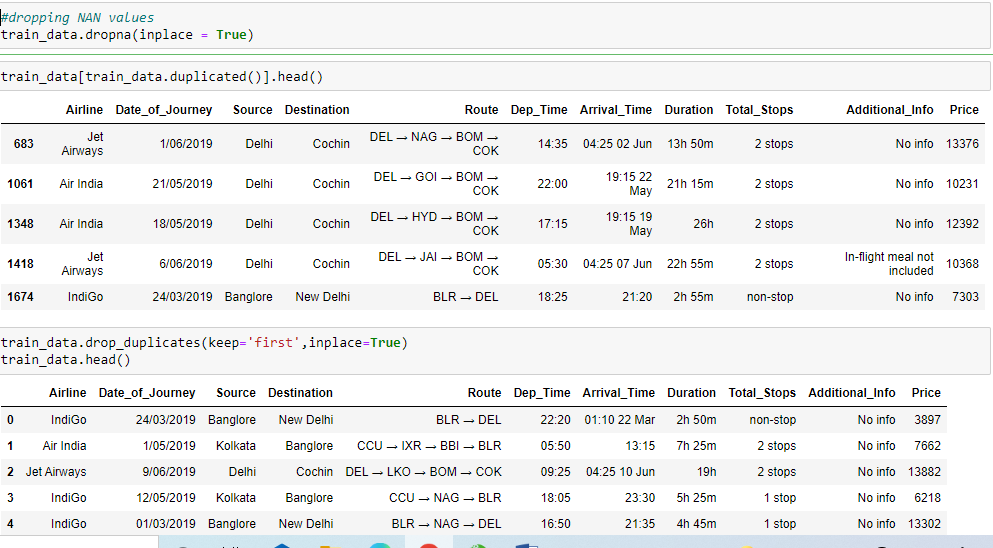


There is no missing value in our train dataset because all columns has 10683 entries. We can see all the columns name by using columns method.



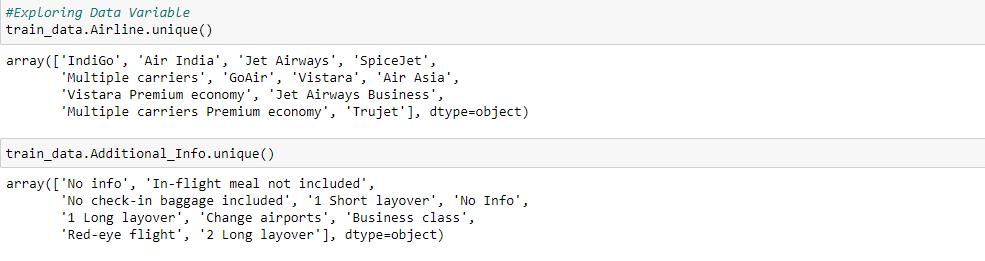
By using describe() method we can see the descriptive summary of our dataset i.e mean, min, max, percentiles, standard deviation etc.

As we can see isnull() method we can check if we have null values in our dataset or not. We have noticed that we have 1-1 null value in Route and Total\_Stops columns. We will drop these Null values.



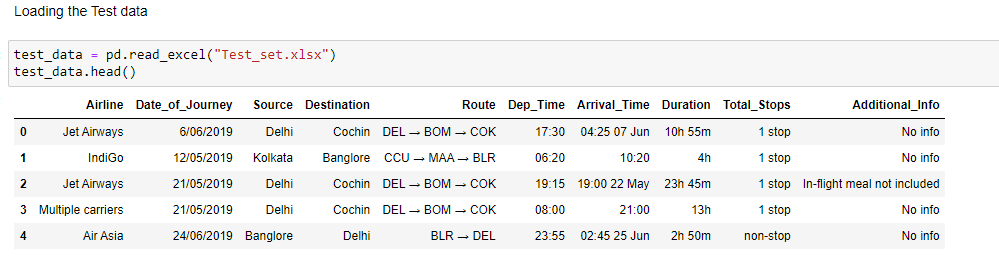
We have some duplicate entries also in our train dataset and we will keep the first values and remove further duplicate entries by using drop function.

By using unique() method, we can see all the unique (different-different) values in all the columns of our dataset as shown below.

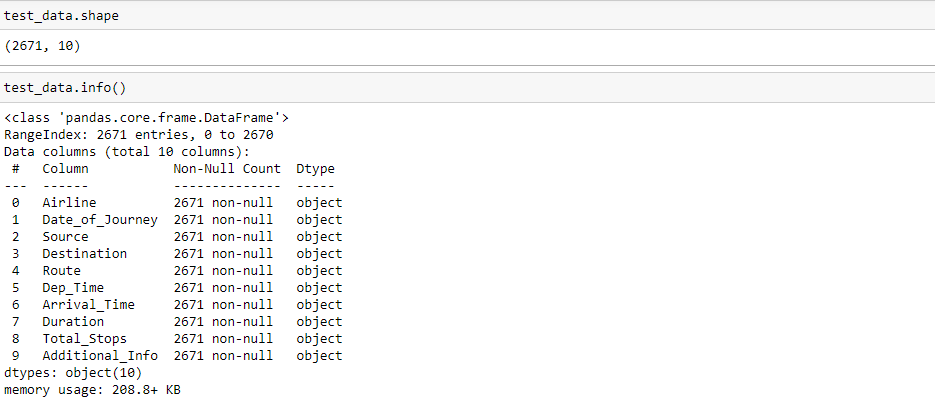


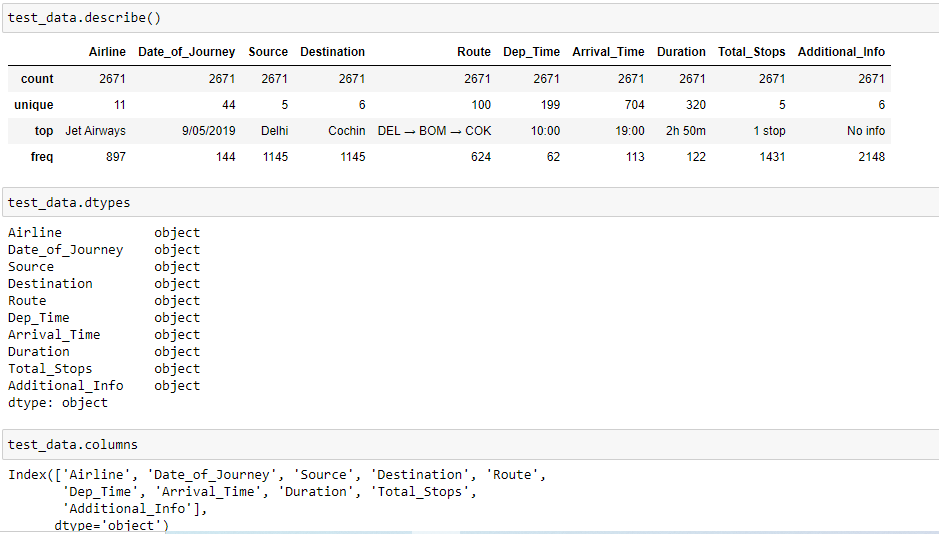
Airline column has 12 unique values — ‘IndiGo’ , ‘Air India’, ‘Jet Airways’ , ‘SpiceJet’ , ‘Multiple carriers’ , ‘GoAir’, ‘Vistara’, ‘Air Asia’, ‘Vistara Premium economy’ , ‘Jet Airways Business’, ‘Multiple carriers Premium economy’, ‘Trujet’.

Now, We will load the Test Dataset -



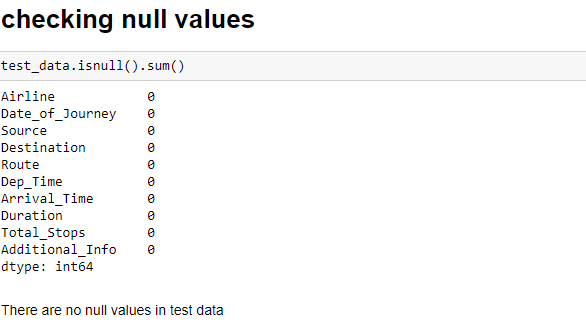
We will do the same data analysis with test dataset too by using **test\_data.head(), test\_data.shape, test\_data.dtypes, test\_data.info(), test\_data.columns, test\_data.describe().**





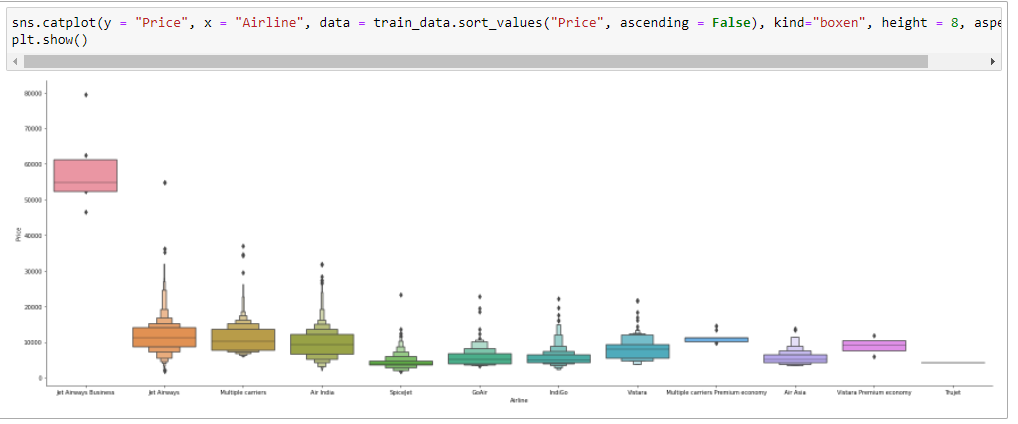
**Checking Null Values-**

By using isnull() method we can check whether we have any null values in our test dataset or not as shown below -



We can see there are no null values our test dataset.

We have created Catplot to show the frequencies of all the data variables for all categorical values. Catplot show the relationship between a numerical and one or more categorical variables using one of several visual representations.



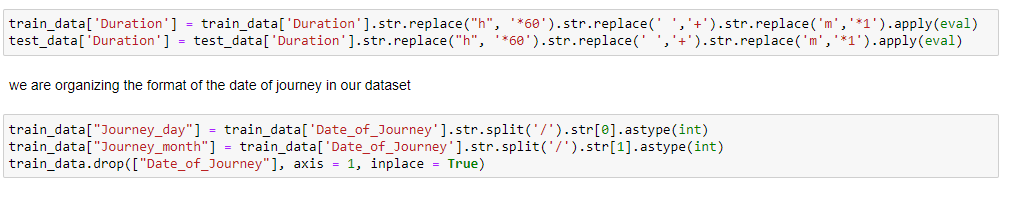
1. **Data Wrangling**

Data wrangling is the process of cleaning and unifying messy and complex data sets for easy access and analysis. With the amount of data and data sources rapidly growing and expanding, it is getting increasingly essential for large amounts of available data to be organized for analysis.

In our dataset, Duration column is not in an appropriate form to help predict machine learning model. We have to bring it in a same format. In our dataset some flight duration could be just “30m” so we will write it as “0h 30m” . So we extract two new cloumns “Duration\_hours” and “Duration\_mins” from “Duration”.

We will split our Date\_of\_Journey to date and month by using date() method.

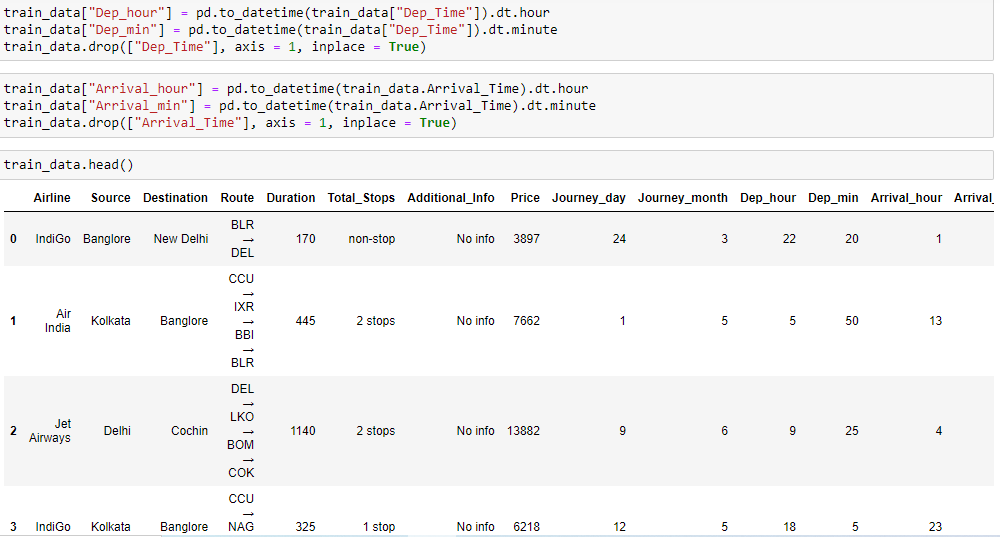
From .info() we know that Date\_of\_Journey is a object data type. To deal with this categorical data we use feature extraction method where we derived new features using **Date\_of\_Journey** .For this we require pandas **to\_datetime** to convert object data type to datetime dtype. so we get two new feature that is.



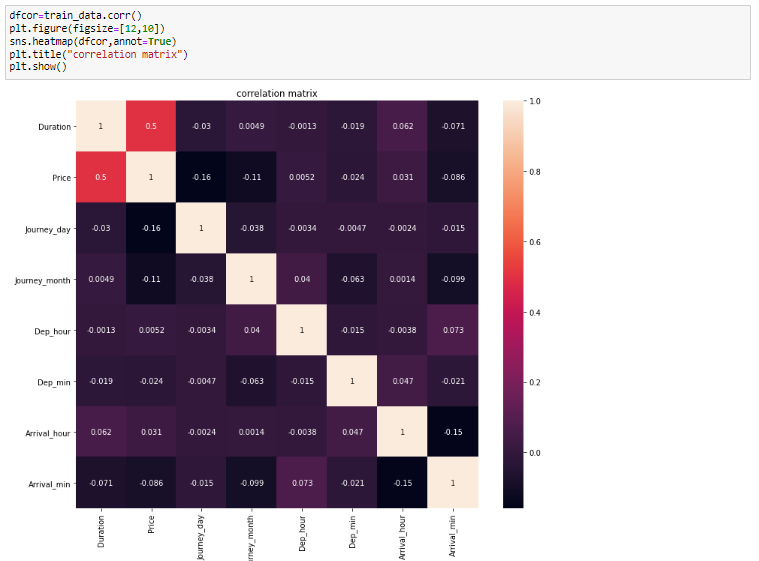
Similary, we extract values from Dep\_Time and Arrival\_Time and create separate columns for departure hours [”Dep\_hour”] and minutes [“Dep\_min”], Arrival\_hour, Arrival\_min.

“Duration” column and “Arrival\_hour” are not in an appropriate form to help predict machine learning model. We have to bring them in a same format. In our dataset some flight duration could be just “30m” so we will write it as “0h 30m” . So we extract two new cloumns “Duration\_hours” and “Duration\_mins” from “Duration”,

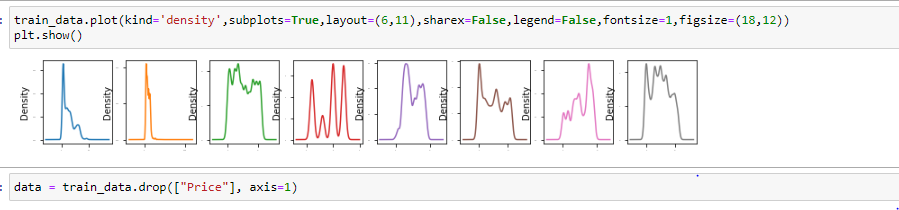
And we will extract two new columns “Arrival\_hour” and “Arrival\_min” from “Arrival\_Time”.



By using sns.heatmap() we will see the relation between our datasets. All the datasets are related to each other. We can see Arrival and Departure Time depends on the Duration.

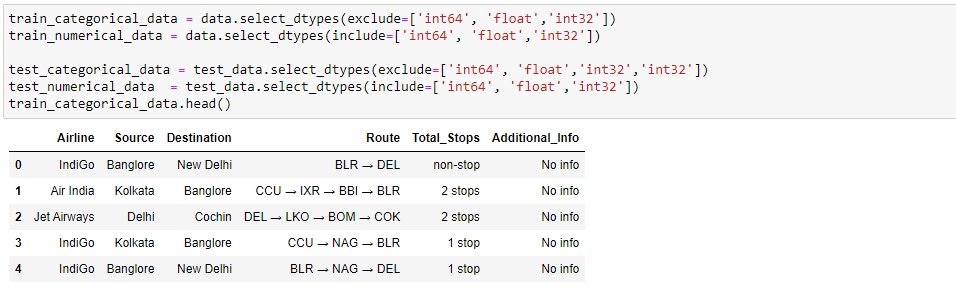


Now we will see the data distribution for our train dataset. By this we can see the overall pattern of the data (shape, center, spread), and any deviations from the pattern (outliers). We can see below that our data is showing deviation.

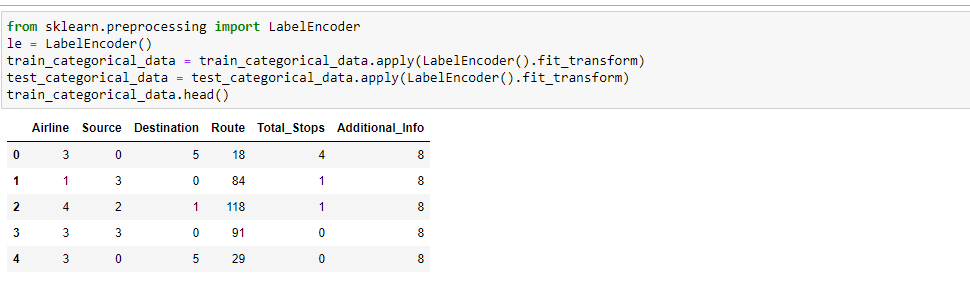


We drop the “price” column from train dataset and make (y) independent variable to find correlation between dependent and independent data later. After cleaning the data, we can visualize data and better understand the relationships between different variables. There are many more visualizations that you can do to learn more about your dataset, like scatterplots, histograms, boxplots, etc.

We have numeric and categorical data in our both datasets. We will exclude categorical values and include all the data which is numeric i.e. int, float values.



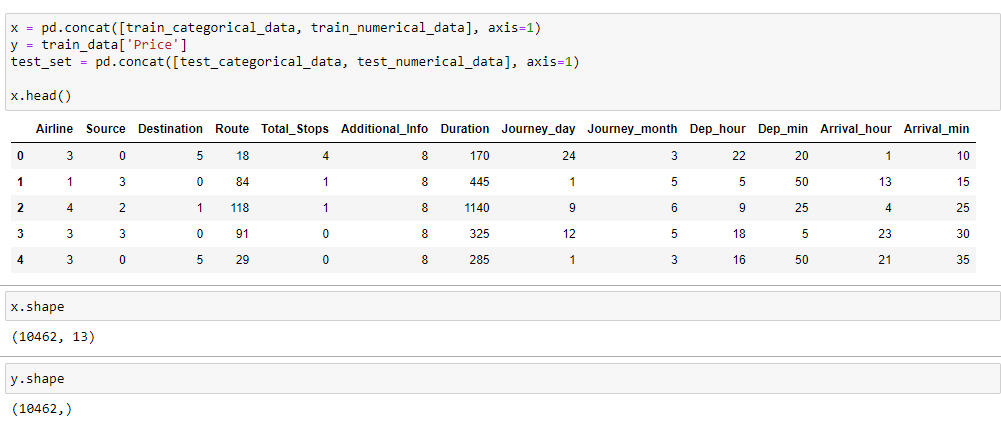
We will use the transform function and LabelEncoder library to convert all the categorical values into numeric data. We will convert all the categorical values into numeric values of our test and train both dataset values.



1. **Data Analysis**

Now we are Splitting Dataset into x and y to perform the model.

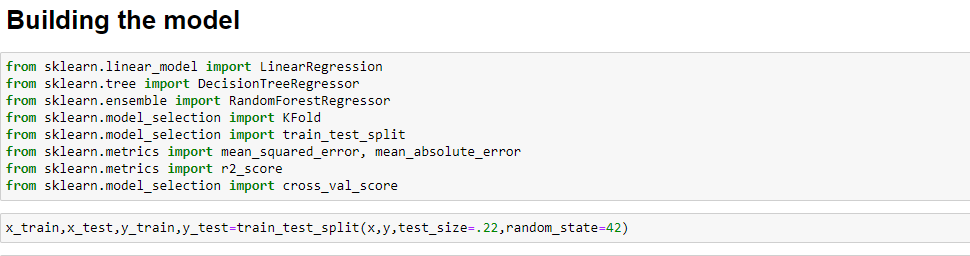
Now, we will concate both of our datasets together and make it x variable and price is our y variable.



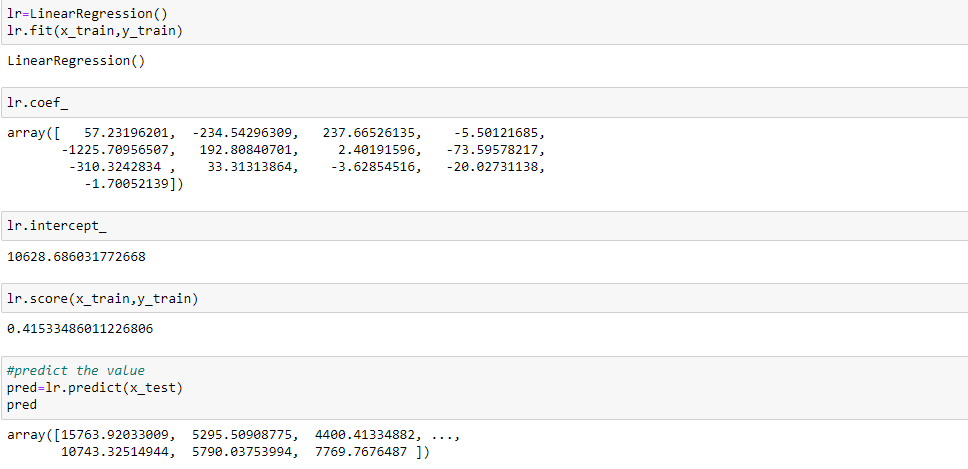
1. **Train the model**

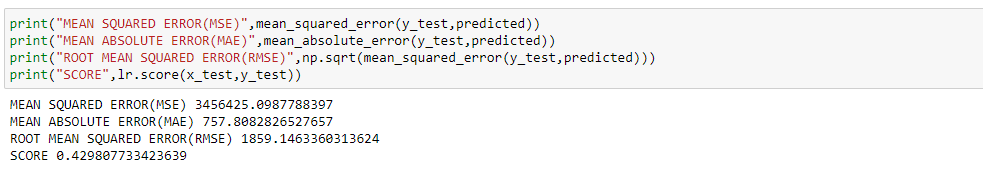
We will use regression to build the model. Now we will import all the required Libraries and functions.

We are going to use Linear Regression, Decision Tree Regressor and Random Forest Regressor. We will split our data in x\_train, y\_train, x\_test, y\_test and train\_test\_split. We will use general random state i.e. 42. Below, I have used three Regression method to build our prediction. We should always train our data in at least 3-4 Models.



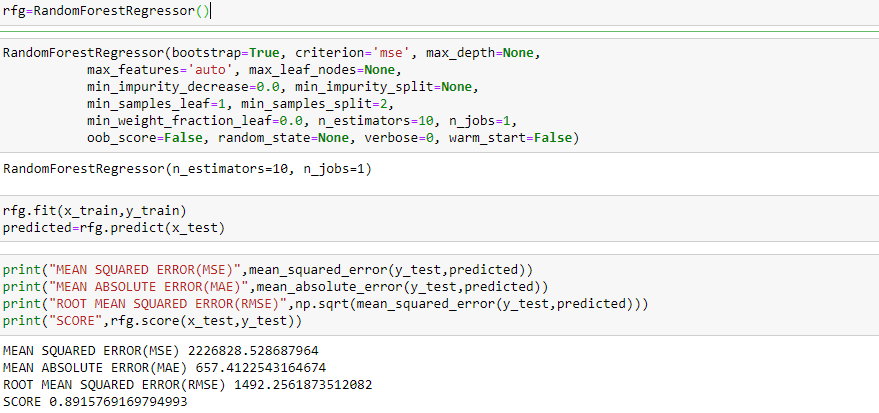
1. **Linear Regression Model**





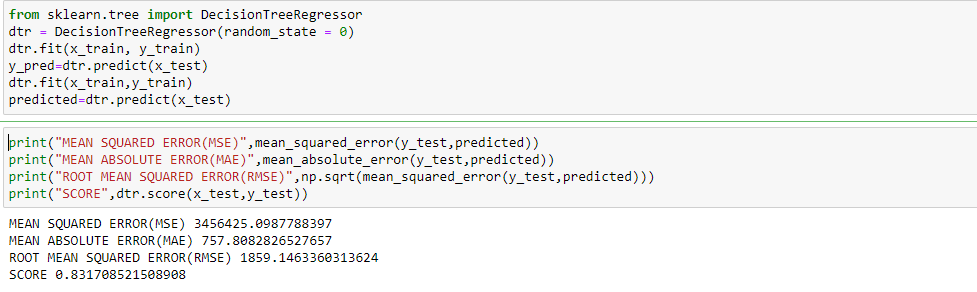
We can see we got Mean Squared error 3456425.09877, Mean Absolute Error 757.8082, Root Mean Squared Error 1859.146 and Linear Regression score is 0.42980.

1. **Random Forest Regressor Model**



We can see we got Mean Squared error 2226828.52868, Mean Absolute Error 657.4122, Root Mean Squared Error 1492.2561 and Random Forest Regression score is 0.8915.

1. **Decision Tree Regressor model**



We can see we got Mean Squared error 3456425.0987, Mean Absolute Error 757.8082, Root Mean Squared Error 1859.146 and Decision Tree Regression score is 0.8317.

1. **Test the Model**

We will test our dataset by using Hyperparameter Tuning and cross validation score.

**Hyperparameter Tuning**

We use the Randomized Search CV for the best hyper parameters. A random search of parameters, using 5 fold cross-validation search across 100 different combinations.

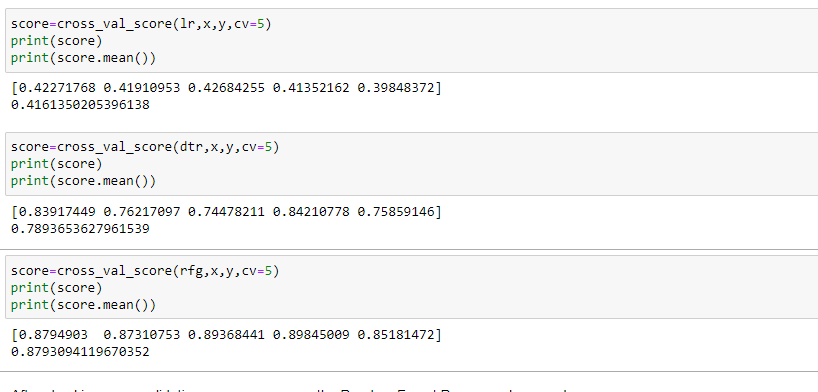
**Checking accuracy of the model**

Evaluating the model accuracy is an essential part of the process of creating machine learning models to describe how well the model is performing in its predictions. The MSE, MAE, and RMSE metrics are mainly used to evaluate the prediction error rates and model performance in regression analysis.

* **MAE** (Mean absolute error) represents the difference between the original and predicted values extracted by averaged the absolute difference over the data set.
* **MSE** (Mean Squared Error) represents the difference between the original and predicted values extracted by squared the average difference over the data set.
* **RMSE** (Root Mean Squared Error) is the error rate by the square root of MSE.

**Cross-Validation Score**

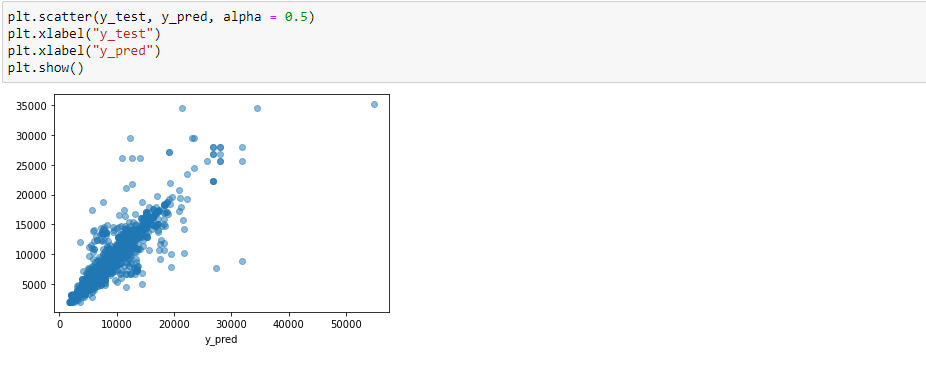
Now, we will check the cross-validation score for our model. Cross-validation is **a resampling procedure used to evaluate machine learning models on a limited data sample**. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation.



We will use cross-validation score for all 3 regression model we have built and will compare the best fit model for our prediction. As we can see we have higher cross-validation score for random forest Regressor. So we can predict that Random Forest Regressor is the best model to predict the flight price.

## **Plotting y\_test vs predictions.**

Scatter plots are used to observe relationships between variables and uses dots to represent the relationship between them. Here, we can see all the points are nearly aligned in a line.



**Saving the Model in Pickle format**

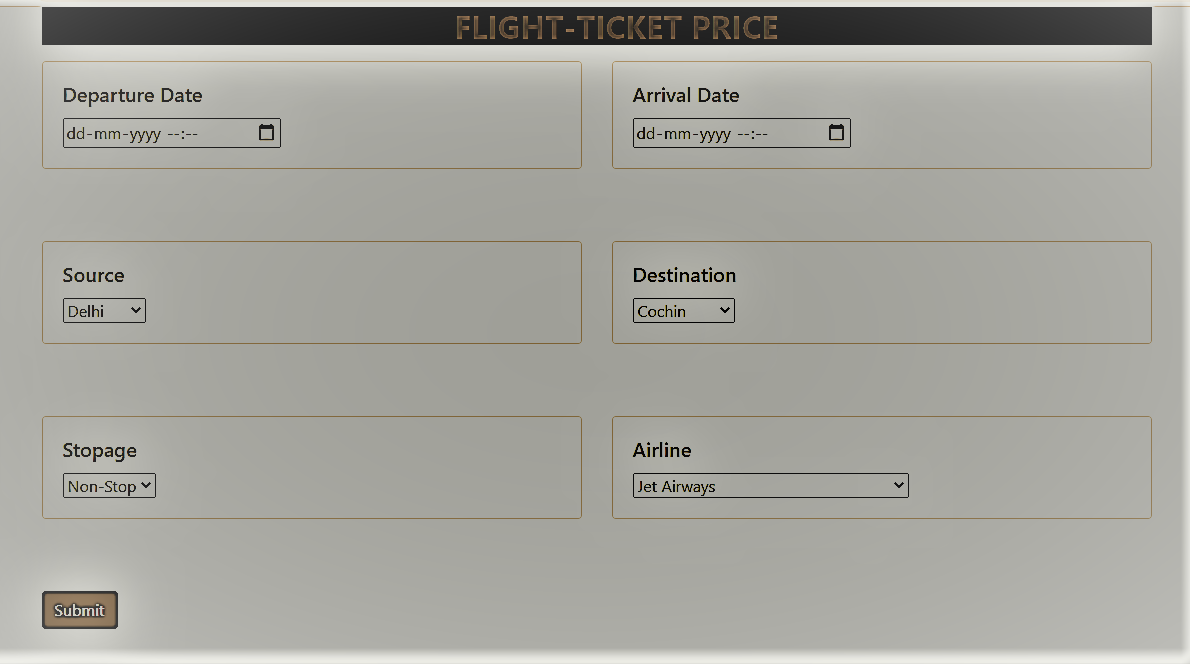
Now, We will save our model in pickle format by using dump method as follow-



## **Model Deployment**

Model Deployment is one of the last stages of any machine learning project. Here, we will design a user interface. We used a flask to make an HTML file for flight price prediction. This will take the input value for each feature and calculate the price for a flight as shown in the image below.

You can deploy your model on Heroku through the GitHub link.



I hope you liked this article on how to predict fare of flight with Machine Learning.