**Flight Price Prediction**

****

Travelling is one of the most entertaining things that everybody wants to avoid city crowds.

At least in the metropolitan cities some of us might have thought we could go to the places through air to avoid the traffic.

Going to another island with a unique nature brings a new perspective about new things.

But the price sometimes seem to be too high that we prefer to stay home..haha..

How cool would it be if we can plan our journey get the most affordable price from using machine learning?

We would have heard people saying that "Seriously..!!', the price was way low yesterday, how come its this high now?

Let us get some help from machine learning to see the trend of price so that we can at least have some idea on when to book a ticket going forward.

Here we will be taking prices of flight tickets for various airlines between the months of March and June of 2019 and between various cities, let us see how the price of the flights varies based on the attributes that we have.

Let us have a check on the data that we have.

Size of training set: 10683 records

Size of test set: 2671 records

Nice, we have a test data as well to check if our model works, so guys, beware, you have to make all the pre processing steps in the test data as well.

Let us look at the features that 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.

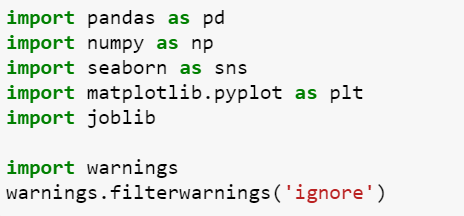
Total\_Stops: Total stops between the source and destination.

Additional\_Info: Additional information about the flight

Price: The price of the ticket

Let us get started !!..

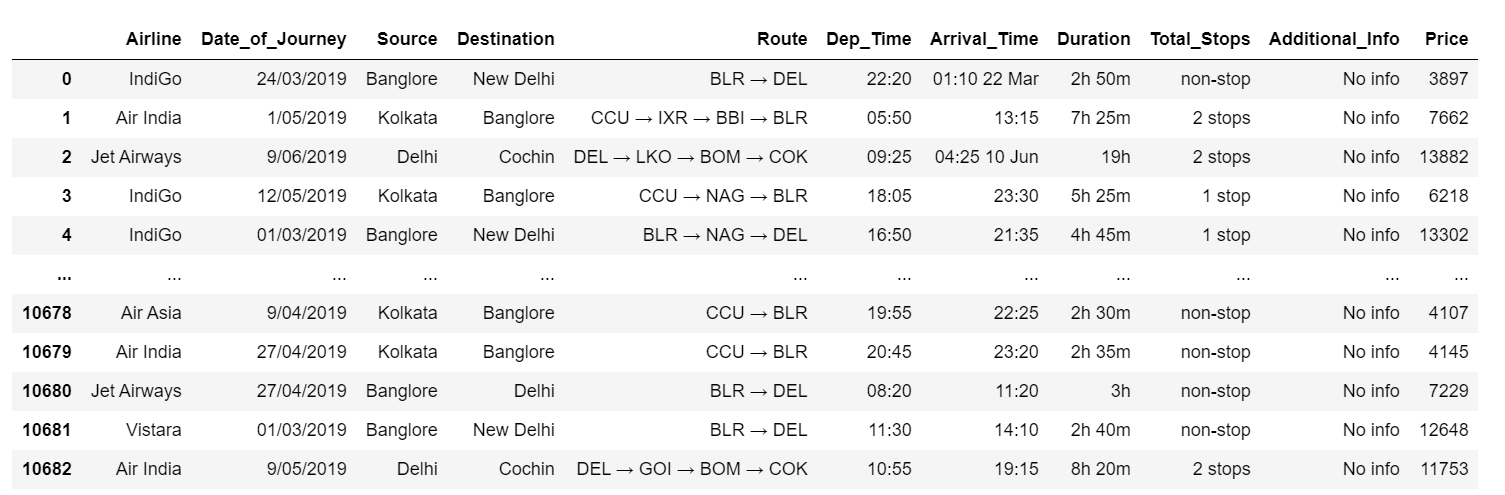
Get the data loaded..



I have imported the files whichever is necessary for now and moving on we will load more as and when needed.

We should thanks the pandas for making it easier to fetch the data using a single line that is pd.read\_csv/excel.

Here is our data:



There are two ways to approach the data here

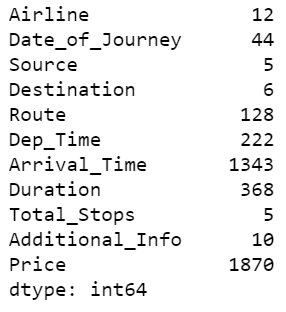
1. We can work on the train data and then preprocess the test data.

2. Parallelly work on test and train.

I'll go with the 2nd way as that is more comfortable for me.

We have taken the data as df for train and dft for test.

Let us check the uniqueness in the data using df.nunique()



Here is what we got, we have 12 Airlines which we will look into.

5 Sources, 6 Destinations, 5 is the maximum stops, and 10 varieties in Additional information.

Here are able to see few attributes on which we will have to dig deep to get sorted like Date of journey, Dep time, Arrival time etc., we will take them one by one.

Make sure that we check if there is any Na values in the data, luckily for me there are none.

And when we describe the price, we are also able to see that the maximum price that we have got is 79512 and the minimal price is 1759.

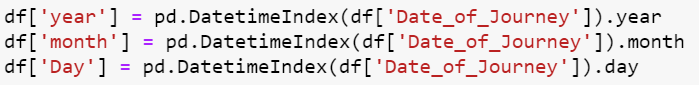
Wow, now that is a huge difference considering we are only taking the flights in India.

Let us work on creating more columns with the details that we have to clean the data.

**Feature Engineering:**

Let us start with the Date of journey.

we can use the function in pandas to get the data segregated.



This way we can get the date of journey segregated to Year month and date.

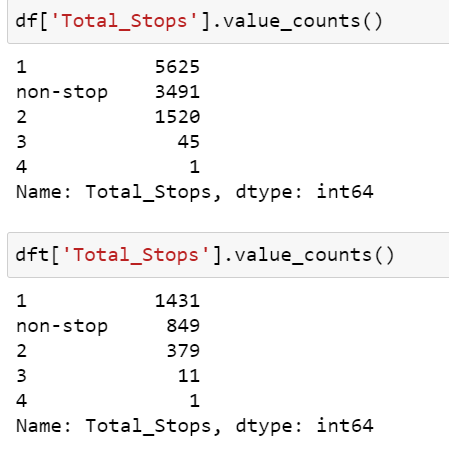
I will also do the same for test data as well here.

As we have got the date segregated, we can remove the column date of journey from test and train.

Total stops:



This will remove the string data and we get just the number of stops here.

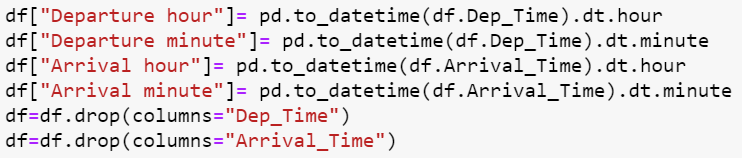


As you see, there is a column, nonstop , we can substitute that with 0.

Some of you might be thinking that the order is not right, no worries, we will change it later.

Departure and Arrival time:

Using to date\_time, we can get the date and time separated from the column.



We will get the details segregated here and dropping as well.

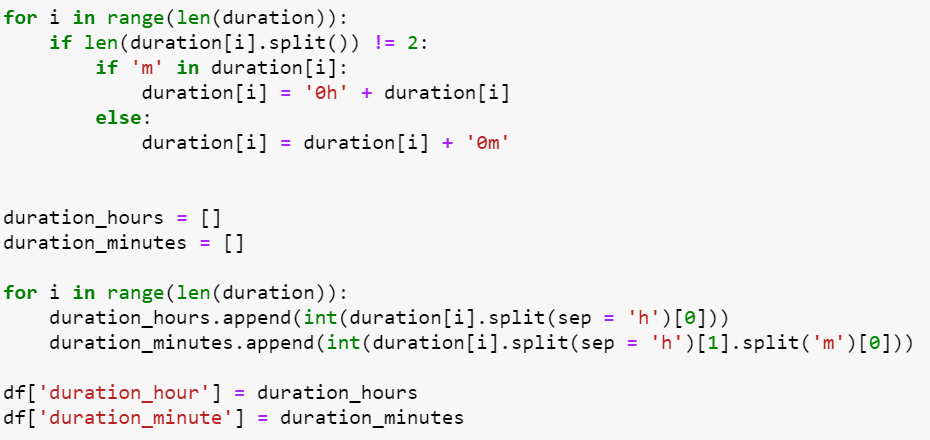
Make sure that you do the same for test data as well.

And as I had mentioned above, the details that we are taking is for 2019 data so we will also remove the Year column from the data as it will not help in any way for prediction.

Duration:

Duration is a part in data which needed to be dealt carefully to separate the hour and minute.

Here we will create a separate list for duration and then use a for loop to get the length and split the data to hours and minutes. then create a new list of duration hours and minutes and then use a for loop again to get the hours and minutes to separate columns in dataframe.



This way he has got the data separated to hours and minute.

I have done the same for test data as well so that going further, it should not create much problem.

Now we can remove the duration column from the data.

And also, I had seen that we have number of stops available in the data, so taking that into consideration, we can remove the route from the main data.

Same for test!!

Let us have a check on how the data looks now...



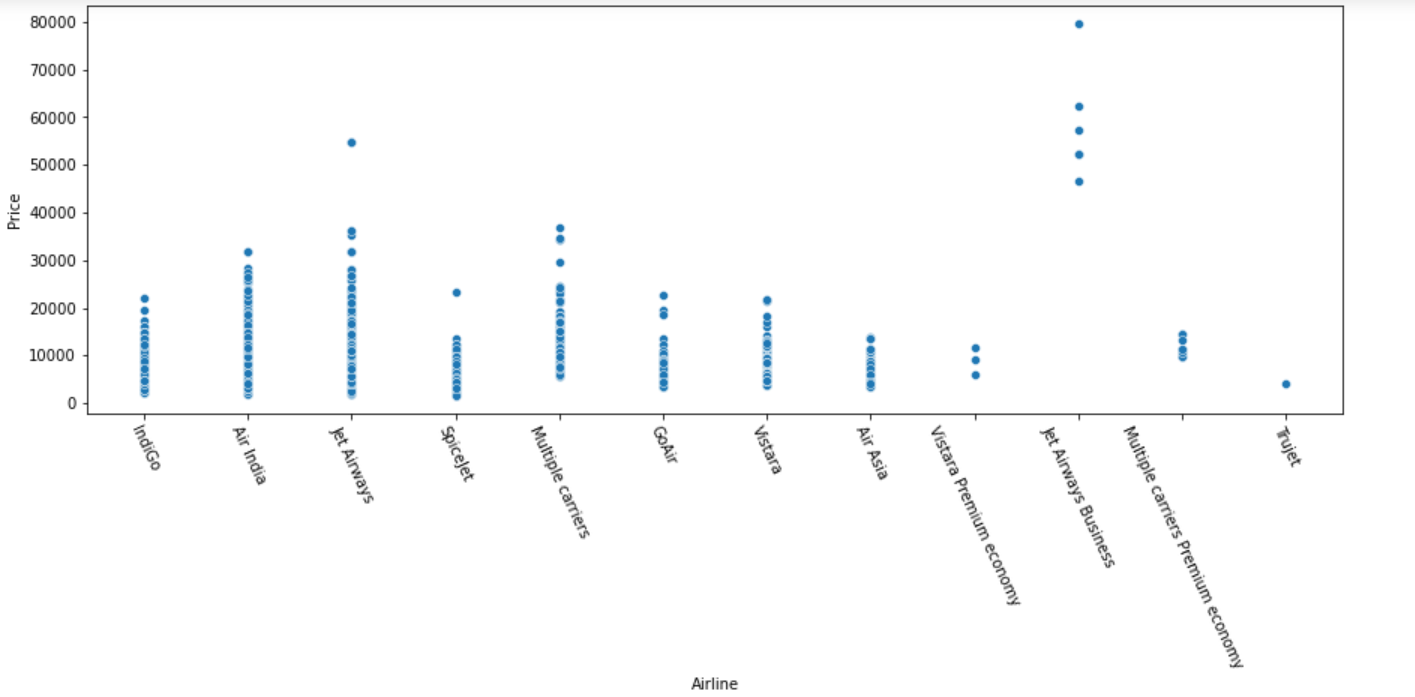
Looks better I guess...what do you guys think?

Oh, yes we have an column in it --> Additional information

If we see the value counts of this column, we will be able to see that more than 80% of the data are the one with no info, so there is no relevance for that data as well and hence we will remove the same.

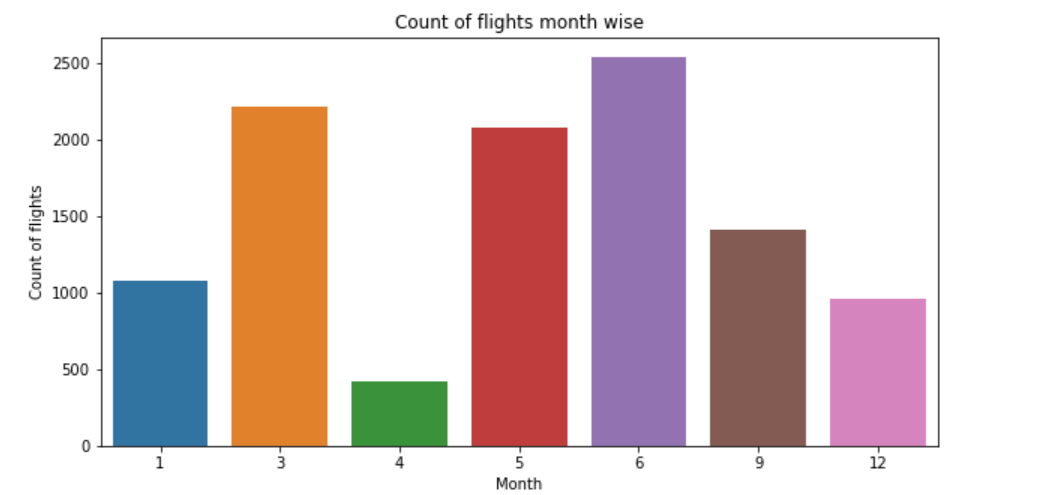
As we have a clean data now, let us work with visualization now.

**Exploratory Data Analysis:**

****

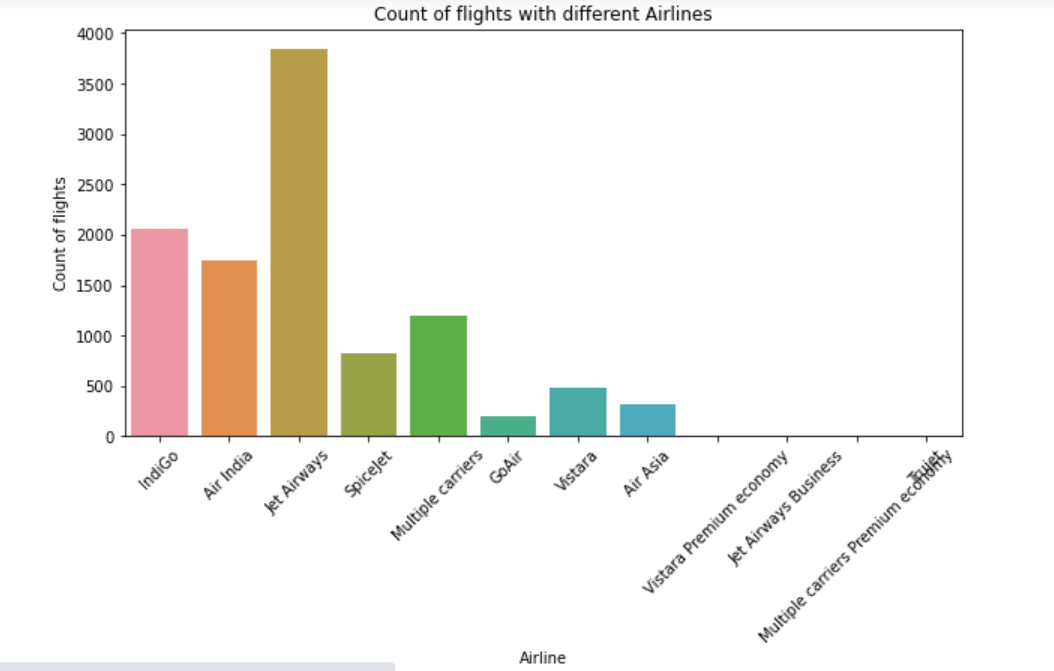
Now this was quite expected, the Jet Airways business is having the most price compared to other counterparts.

Let us check the number of flights per month.

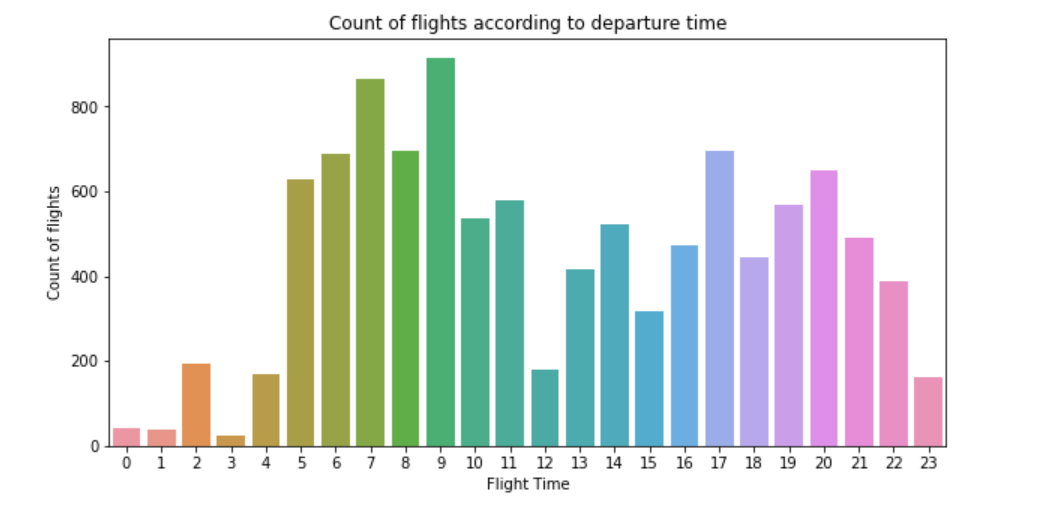


We see here that the greatest number of flights have been taken in 3rd 5th and 6th month.

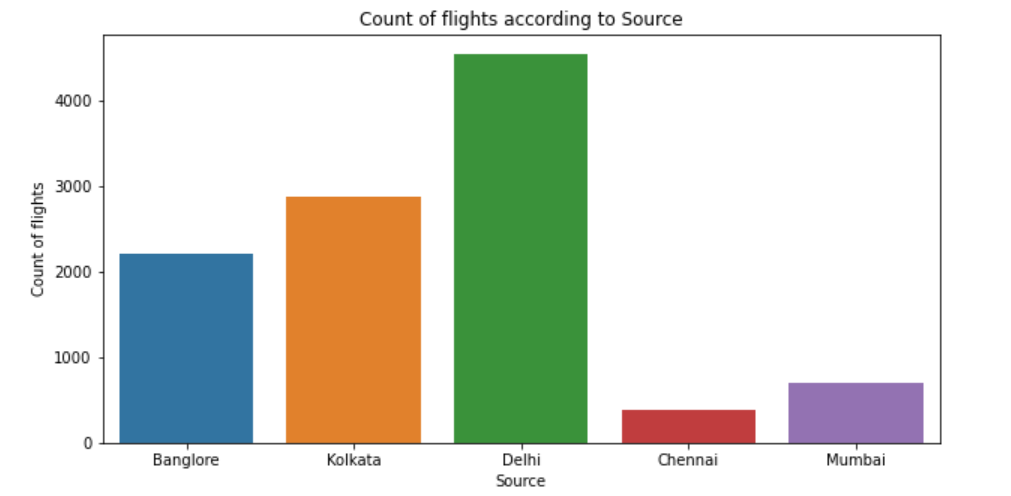
Looking at the number of flights taken per airline.



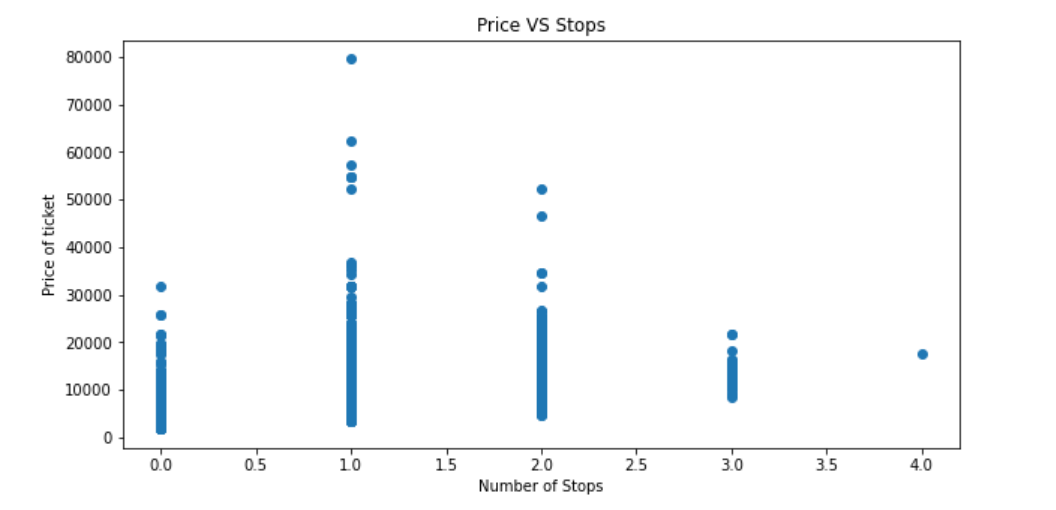
Jet Airways is having the greatest number of flights, it could possibly be because of the low price, taking that intuition, JetAirways Business is having the least.



We see that the greatest number of flights are in the morning time, there are few in evening as well but number of flights in morning at different times are more compared to evening.



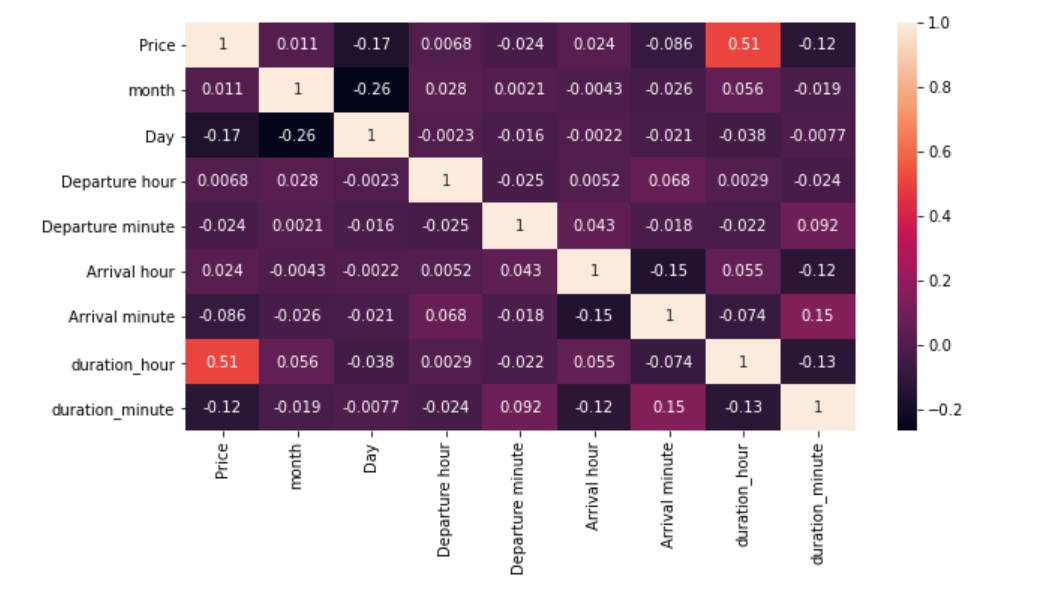
We also see here that the greatest number of flights are taken from Delhi whereas the least from Chennai.



As we see from the data, the flights having at least one stops will have more price than others.

Let us check the correlation now.

Correlation coefficients quantify the association between variables or features of a dataset. The correlation coefficient is determined by dividing the covariance by the product of the two variables' standard deviations.



Here are the observations:

1. Duration hours is highly correlated to Price and total stops.

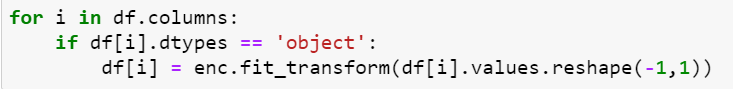
2. Price and total stops are positively correlated.

3. There is a negative correlation between Destination and source which is quite obvious lol!

As we have got some very needed insights here, we can work to encode the data. Our first step is to get our data out of the object data type land and into a numeric type, since nearly all operations we’d like to apply to our data are going to rely on numeric types.

We have a very useful encoder here which is ordinal encoder which treats every column individually and encodes it to avoid machine to consider every encoded value to be the same.

If we use label encoder, it will not consider same values in different column as separate, it will consider that to be the same which can yield a poor result afterwards.



I have created an instance for OrdinalEncoder under the name of enc and then use the for loop to change the object variables to encode the data.

Oh, forgot to tell you, while doing these steps I notes that the total number of stops were mix of Object and int hence I had used astype to change the whole column to int format.

Now let us split the data to X and Y.

And now let is straight forward move ahead to modelling section as we only have categorical sections in the x data, we do not have to check for outliers as that will be a waste of time and energy and will not contribute in any way towards the learning.

**Modelling:**

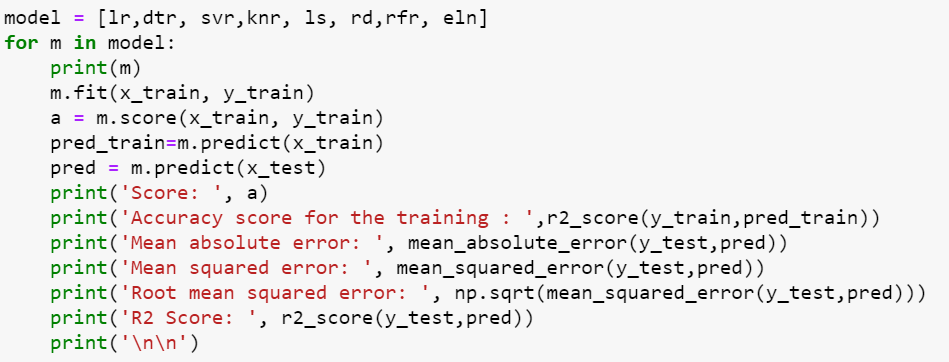


Not using for loop for test and random state as we usually work on getting the best random state and use that for other models as well, however there is a possibility the random value change with the model we use to get the best random state or test size hence not using that and assuming mostly used 30 7- split for test and a random value for random state as I presume that it might not bring a big difference in the data.

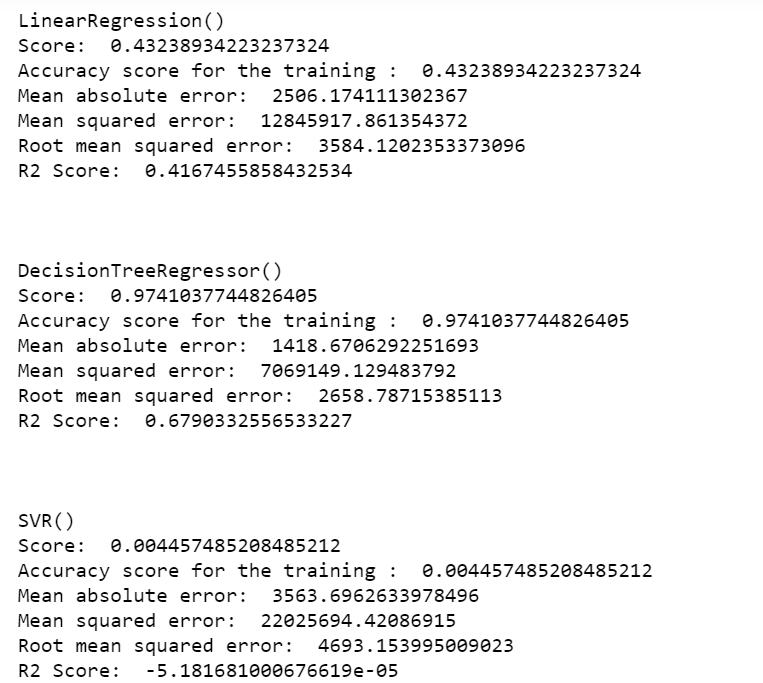
Importing necessary data.



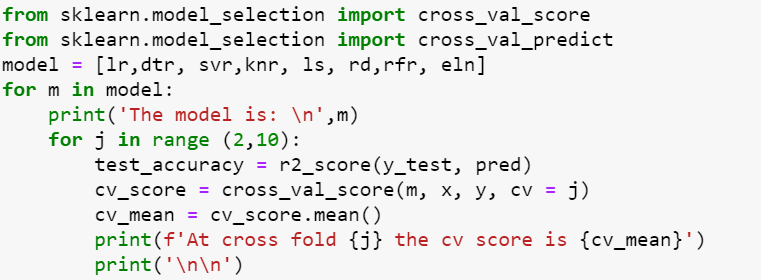
I have imported all the necessary models and created instances to run a for loop for all.



Like this and we will get a result somewhat on what is mentioned in the image below.

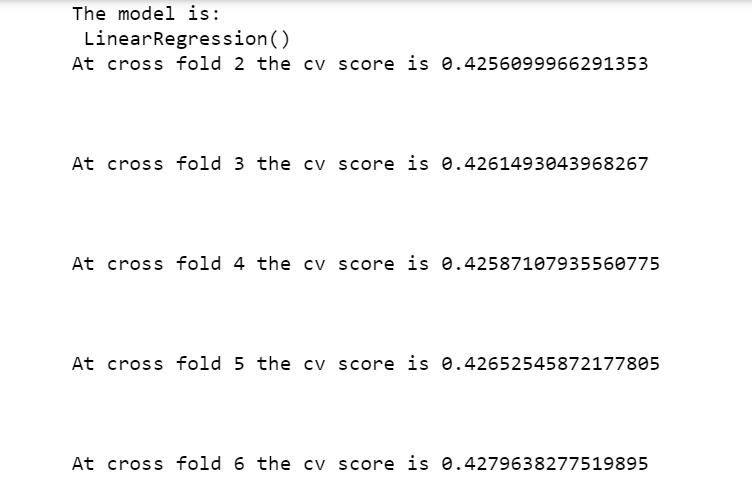


Now there is a possibility that the data will over fit or underfit, we can use the cross validation to reduce that. Let us work on that.



I have used a for loop to check all the possible cv's from 2 to 10.

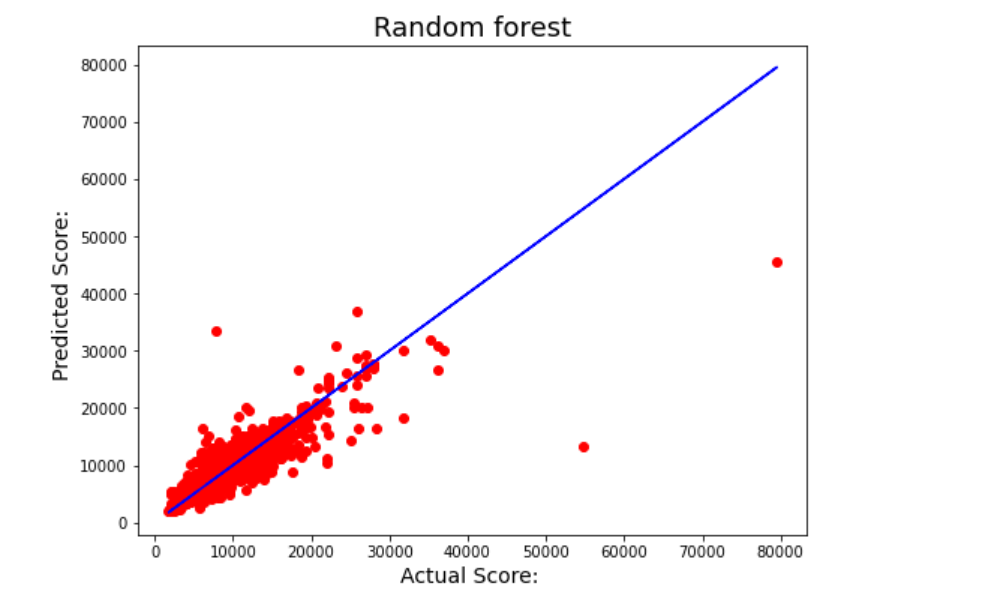
We will get a result like this:



Now we can compare the models and get the details.

In here the best R2 score that I got was for Random forest.

When we check the regression plot, we get a pretty good model, let me show you:

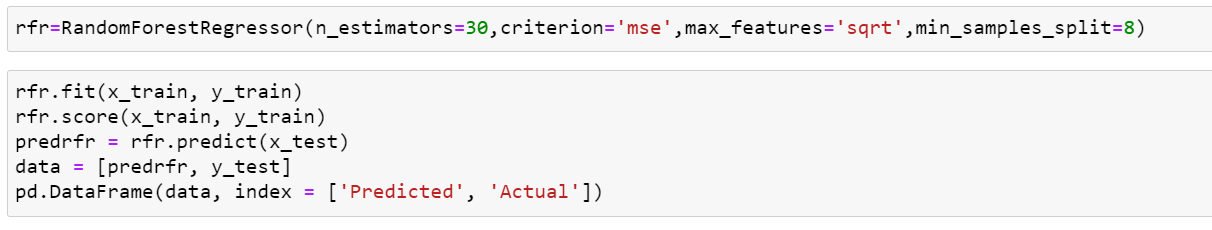


I know that you guys are able to see few data that are not close to the predicted score, its okay, we can manage that much.

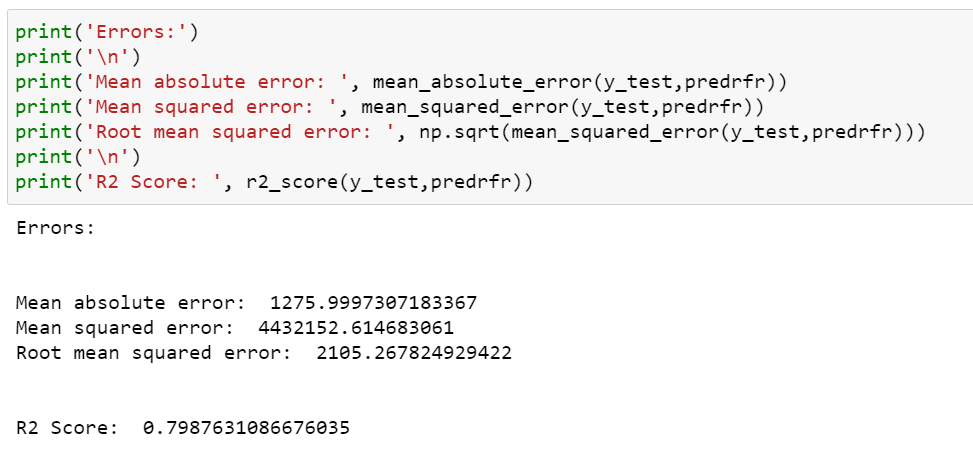
Let us try to hypertune the data using Gridsearch.



Sorry couldn’t accommodate more here, though you can get the details from scikit learn in google about all the parameters that you have in the Randomforest regressor.



After getting the best parameters, I have finalised the model.



I hope that it’s not a bad score, it’s quite good enough.

Saving the model:

Yay, that was fun to go with the whole data and get quite good accuracy at the end. Now we can save the model using Joblib and test that data on the pre-processed test data that we had.

**Conclusion:**

In this case study, a Machine Learning model is developed to predict the airlines fare. Here several features were mined from the dataset and combined together with the help of Machine Leaning, to do the flight price prediction.

You can see how we have handled the categorical and numerical data and also how we build different ML model on the same dataset.

Please let me know if you find anything that can be improved, I will be glad to look into that.

Thanks.