**Flight Price Prediction**

**Problem Definition**

Air transport, being one of the safest and most reliable modes of transportation might come as costly for the major share of population. The fare for the airlines are very dynamic and could always depend on multiple factors. We can make use of the scope of data science and machine learning to build up a reliable model to analyze and predict the prices for each flights.

We have availed the help of a data set with prices of flight tickets for various airlines between the months of March and June of 2019 and between various cities. Here, we will be going through each feature to find the interdependence and correlation which would be later transformed into building a machine learning model that could predict the flight prices of a test data.

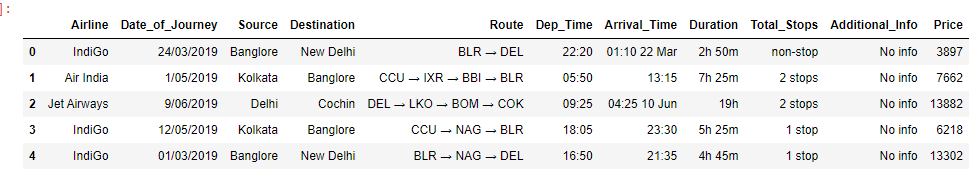
**Data Analysis**

We are importing the data set using the pandas library to a data frame in order to conduct the data analysis. As an initial observation, The data set is found to be having,

* 10683 rows and 11 columns

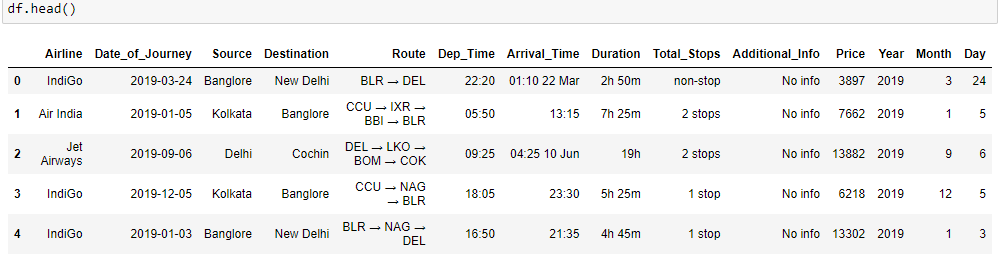
As we check the initial 5 rows of the data set. We have the following features,

1. Airline: The name of the airline.
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
11. Price: The price of the ticket



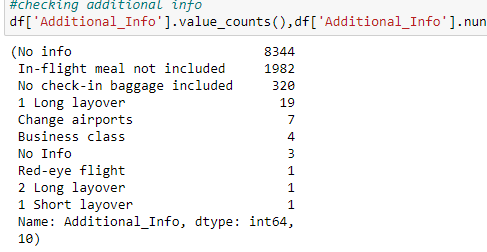
The data types of the given data has been checked and the price column obviously showed as an integer variable while the res of the columns were in ‘object’ datatype.

As the data has been shown, The Date of journey column, even though in object format has to be in ‘Date’ format in order for us to extract meaningful data, so as an initial step we are converting the same using the Datetime feature and also separating the Date, Month and year to different columns. After the transformation, the dataset looks like this with the newly added columns.



Now, the Date of Journey column is obsolete and we are dropping the same from the data set.

Looking further into the additional\_info column,



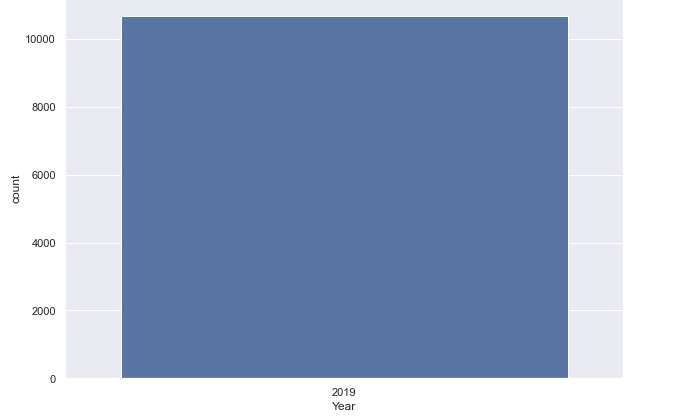
flight tickets for various airlines between the months of March and June of 2019 and between various cities.

The column seems to be having information regarding the in flight perks and a large amount of rows are left as No info. Even though the additional perks could be a factor in predicting the price, the non-uniform nature of the column could affect the model. Thus, We are currently removing the same from the data set.

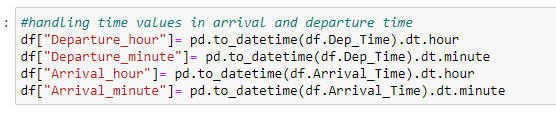
Since we have extracted the year data from the date column, We have to check the distribution of the data in the same.

All the data in the data set was collected in the same year. As this could not benefit in analyzing year wise disparity in the flight prices, we are removing the Year data from further analysis.

The graphical representation of the data distribution in the ‘Year’ Column is given below,



There are other date and time columns in the data set, first we will be converting the arrival and departure time to date time format and then we will be extracting data by minutes and hours.

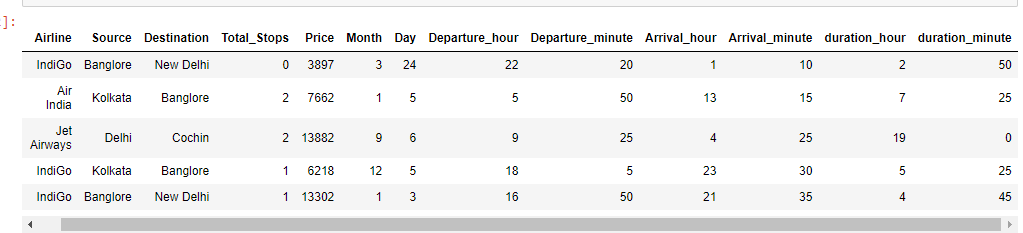
After which we can drop the actual columns, departure and arrival time from the data set.

The Total stops column will have to be cleaned as many of the values are followed by a suffix ‘stops’ which wont be necessary going forth, so we will be removing the suffix from the column.

We are removing the ‘Route’ column as I am assuming that the total stops column gives us almost the same information.

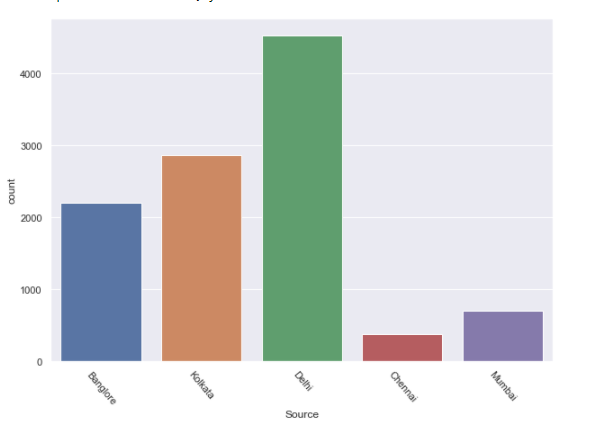
The Duration of the flights would be another important feature, the column has to be separated to hours and minutes, we will be using ‘h’ and ‘m’ suffixes to split the data into two new columns.

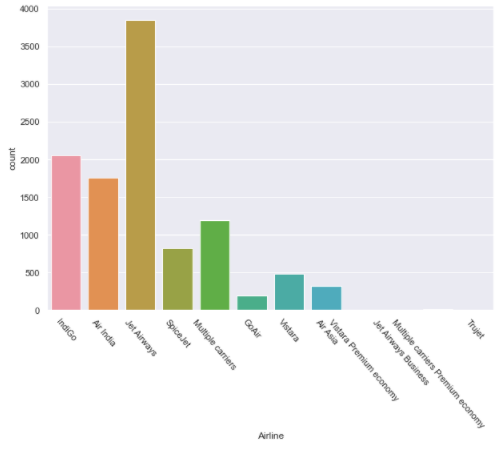
After the transformations the first 5 rows of he data set will be,

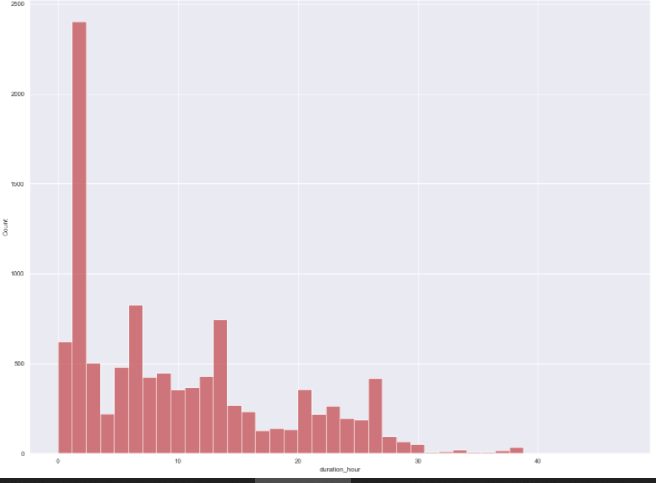
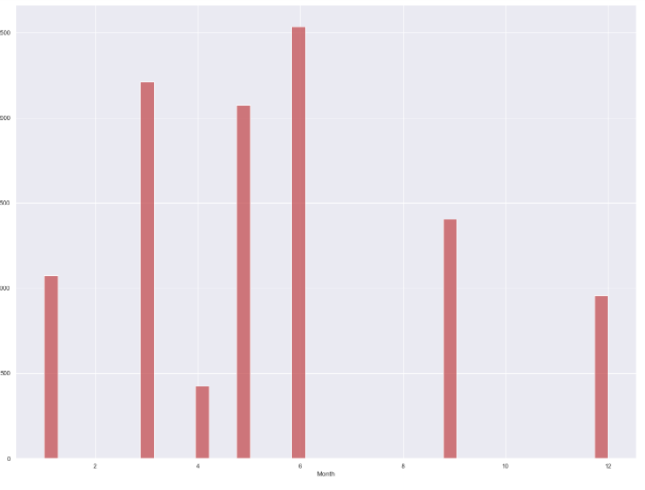
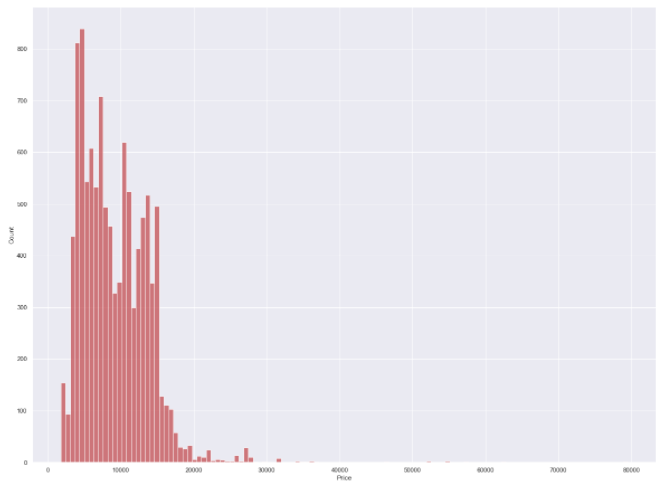
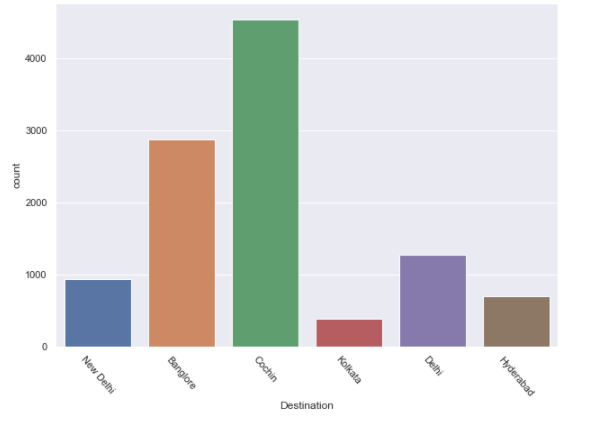


The data set is checked for null values and are removed as the total data loss is minimal and is less likely to affect the model building.

**Uni-variate Analysis**

****

****



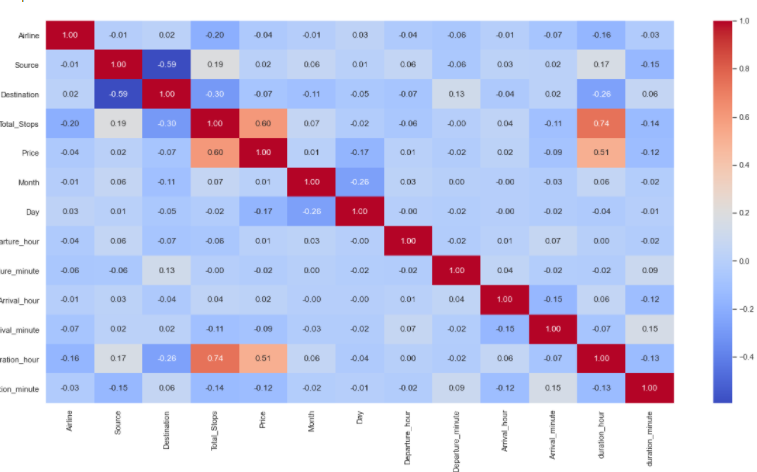
* As the data suggests majority of the airlines belongs to the jet airways, followed by indigo
* The source cities from where the journey starts with Delhi topping the list followed by Kolkata
* The destination cities, Cochin Tops the list
* The price range is around 5 to 20k, the plot also denotes the possibility of outliers
* The Months plot is showing that the most travels is around the month June, followed by march
* 2 hour flights are the most common, followed by 7 hour flights, the highest duration is 40 hour flight

**Bi-Variate Analysis**

|  |  |
| --- | --- |
|  | Jet airways business has the highest ticket price and spice jet and truejet the lowest |
|  | * Steady increase in the Price as the duration of the flights increases |
|  | * Some of the early morning flights costs more |
|  | * We can see that the price increasing as the number of stops goes up |

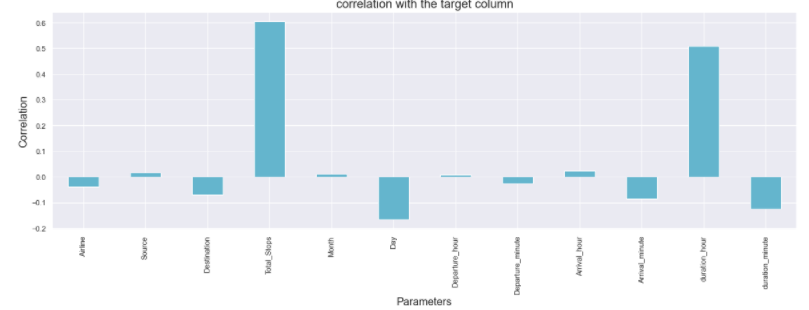
Checking the correlation between the columns.

* Total\_Stops,duration\_hour seems to be having a positive correlation to the price variable
* Airline seems to be having a negative correlation



The target variable Prices will be plotted in accordance with its correlation to other major features,

As we have observed earlier the number of stops in each flights and the total duration of the flights seems to be having the highest possitive correlation to the target variable.

**EDA Concluding Remarks**

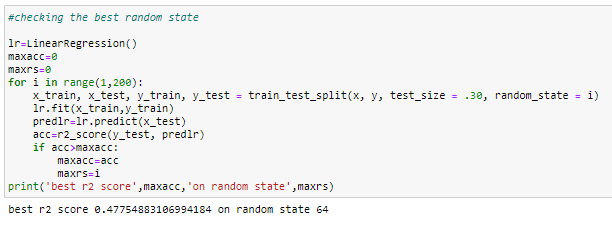
* The target column is the prices and we have identified the relationship with the target and other features
* The total number of stops and the total duration of the flights seems to be having the highest positive correlation to the target column.
* The Airline company seems to be having the highest negative correlation to the target column.
* Business class flights are the costliest.
* The early morning flights are the most pricey.

## **Pre-processing Pipelines**

* We have extracted the date, month and year from the Date column and have created new columns for the same.
* The time columns like arrival time, departure time and the total duration are cleaned and hours and minutes are extracted from each column and added to new columns.
* The null values are removed.
* Skew-ness has been checked and decided not to treat as the impact is minimal.
* Standard scaler was used in scaling the data

**Building Machine Learning Models**

* We are splitting the features as x and the target variable as y as the initial step.
* Since the target column has continues values, we are importing all the regression algorithms.
* We are implementing a small function to identify the best random state while splitting the data.



* Splitting the data for training and testing with the previosly identified random state
* Defining the functions for model building and cross validation.
* Calling different models and checking the model r2 score and cross validation score

**Selecting the best model:**

After Using **Linear regression** algorithm we were able to achieve the following results

r2score: 0.48813597334339986

Mean squared error 9501582.752697084

Root Mean squared error 3082.4637471829387

mean absolute error: 2304.7200652152655

mean\_cross\_validation: 0.4066160171894794

The R2 score seems vay lesser however, the difference between the actual score and the cross validation score is minimal

Using **Lasso**

r2score: 0.4881359732776561

Mean squared error 9501582.753917467

Root Mean squared error 3082.4637473808943

mean absolute error: 2304.7183862688717

mean\_cross\_validation: 0.40661601998535896

We are getting a similar results as Linear regression using Lasso

The **Ridge** Model also produced a similar result of,

r2score: 0.4881359733370888

Mean squared error 9501582.752814235

Root Mean squared error 3082.4637472019417

mean absolute error: 2304.7195911349622

mean\_cross\_validation: 0.4066160222544295

We are getting the best r2 score till now using the **Random forest regressor,**

r2score: 0.9609056908960714

Mean squared error 725696.2665198211

Root Mean squared error 851.8780819576361

mean absolute error: 527.2356824332021

mean\_cross\_validation: 0.7739437830225496

However, we could still look for a better model as the above has slightly larger cross validation score difference

The **decision Tree Regressor** gave as the bet R2 score of them all,

r2score: 0.9839264119874443

Mean squared error 298369.32990119606

Root Mean squared error 546.2319378260448

mean absolute error: 146.3185647425897

mean\_cross\_validation: 0.5931890948806563

As the scores suggest the stability of this model is questionable.

After running almost 6 different regression algorithms.

The Kneighbors regressor gives the best stable model based on the lowest variance with the cross validation score, even though algorithms like random forest regressor gave as the better r2 scores.

The scores are,

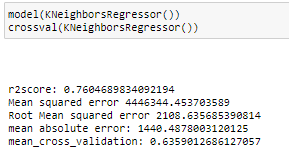
r2score: 0.7604689834092194

Mean squared error 4446344.453703589

Root Mean squared error 2108.635685390814

mean absolute error: 1440.4878003120125

mean\_cross\_validation: 0.6359012686127057

****

We are selecting the Kneighbors regressor for hyper parameter tuning.

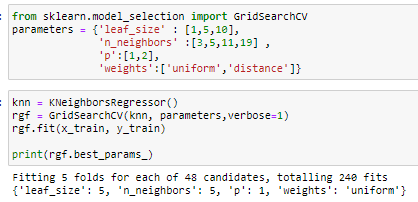
Hyper parameter Tuning using **GridsearchCv:**

GridSearchCV tries all the combinations of the values passed in the dictionary (which I’ve set manually) and evaluates the model for each combination using the Cross-Validation method. Hence after using this function, we get accuracy/loss for every combination of hyperparameters and we can choose the one with the best performance.

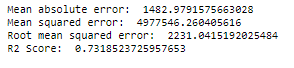
It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, you can select the best parameters from the listed hyperparameters

We are running gridsearchcv for Kneighbors regressor. The parameters are declared as a dictionary and checking the best values for 4 different features. After running the same the best values for the parameters we passed to the algorithm are,

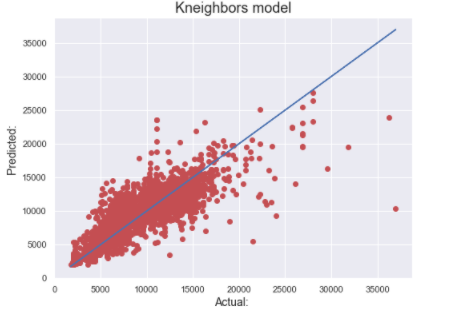
'leaf\_size': 5, 'n\_neighbors': 5, 'p': 1, 'weights': 'uniform'

****

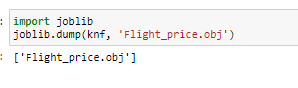
We are running the model agin using the best parametes and getting an R2 score of 0.7318523725957653



Plotting the actual values against the predicted values, we are getting the below given plot.



**Saving the model**: I’ve used the Joblib module to build the ML model. It provides utilities for saving and loading Python objects that make use of NumPy data structures, efficiently.



## **Concluding Remarks**

## The pricing of flight tickets depends upon various factors. However, two of the major factors are the flight duration and the total number of stops in each flights.

## The features like additional info which gives us more information regarding the perks in each flight can also be a valid feature if we can get our hands on much more balanced sample

## We were able to build a model that has an R2 score of 0.7318523725957653

## The final model was saved using the help of Job-lib library.