

|  |
| --- |
| FLIGHT PRICE PREDICTION |
| This article is generally on ‘Prediction on flight price’. 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 travelers saying that flight ticket prices are so unpredictable.  Let’s takes you through each and every step-in detail how to predict the flight price and helps to understand the whole ML model building process. So, let’s get started.  Here some data will be provided with prices of flight tickets for various airlines between the month of march and June of 2019 in between various cities. |
| April 18  *COMPANY NAME*: Data Trained Education  *Authored by*: Deepesh Singh |



# *FEATURES:*

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.

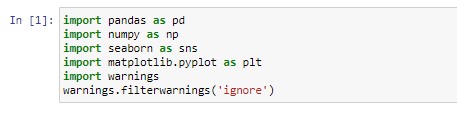
Duration: Total duration of the flight.

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

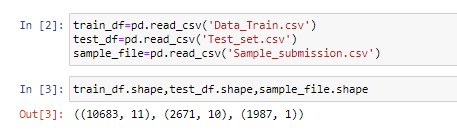
Additional\_Info: Additional information about the flight

Price: The price of the ticket

***Importing Necessary Libraries:***

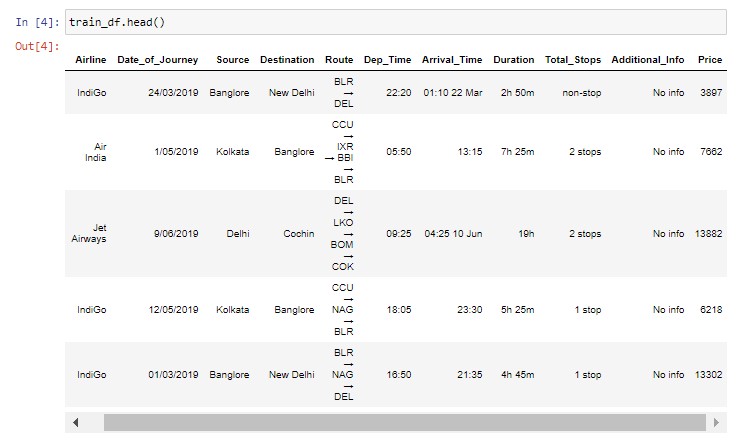
**

*Loading the Dataset:*

**

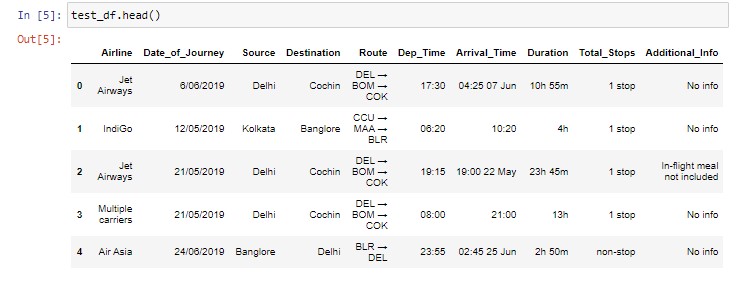
There are two types of datasets are provided – Train data and Test data

**Train data**:



Training data is combination of both categorical and numerical also we can see some special character also being used because of which we have to do data transformation on it before applying it to our model.

Test data:

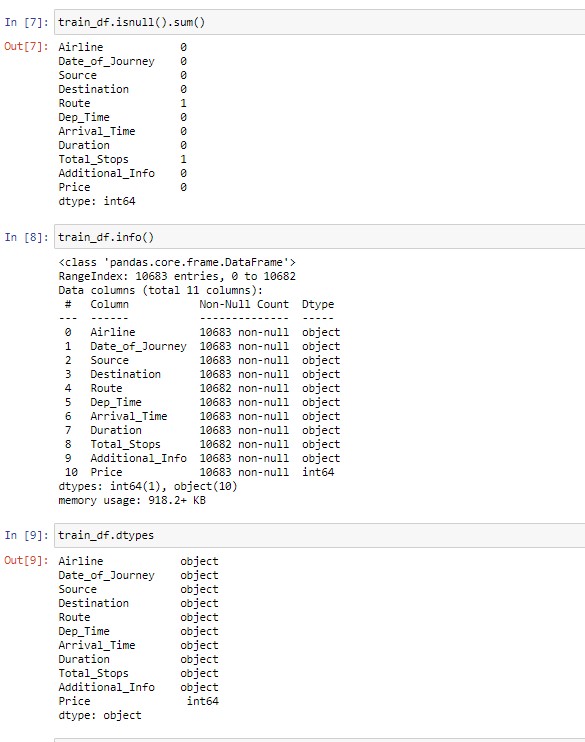


The test data is similar to the training dataset, only difference is minus of the ‘Price’

Column. That is to be predicted using the model.

Data Wrangling:

In this we find some information about the data set like null values, info (), data types, data columns, value\_counts of total\_stops, duration and etc. Screenshot of all the codes are as follows:



|  |
| --- |
| Also plot a countplot to see how many stops are.  After that now checking for a null values in both the dataset train and test remove the null values.  By dropping the null values, we remove the null values which are present in our dataset.  Loading the screen shot of the above process: |
| ***Feature Generation:***  In this step we mainly work on the data set and do some transformation like creating different bins of particular column’s, clean the messy data so that it can be used in our ML model. This step is very important because for high prediction score, we need to continuously make changes in it.  ***Date of journey:***  In the column ‘Date\_of\_Journey ‘, we can see that format is given as dd/mm/yyyy and as we can see that the datatype is given as object, so there is two ways to tackle this column, either we convert the column into the timestamp or separate the date and the month of the column. So here I am separating the columns. This process is done on the both of the dataset train and test. After that dropping the columns from both of the data.  Loading the screenshot of the whole process of the separating the columns as bellow:  Train data:    Test data:    ***Dep\_Time and Arrival\_Time:***  In Dep\_Time we see both combination time and hour, so we split it into dep\_hour and dep\_minute. And same with the Arrival\_time we only need hour and minute so we only split the hours and minute. And after converting them we drop those columns and we do this for both the Train and the Test data.      After doing that we check for the unique () in the Airline and do some **EDA** on the Airline like plotting the countplot of Airline, by plotting the countplot we see how many Airlines are present.  For showing how it is done I am putting a screenshot of the code:    After plotting that we plot a bar plot between Airline vs Price. And it shows the Price of the flight according to the Airline.    Above bar plot shows the flight price according to the Airline.  We can clearly see that **Jet Airways Business** has the highest flight price among all the Airlines. |
| Now converting all the categorical values of the the Airline into the Numeric value.  Replacing the airlines names with the integer value.  This is done with the both the dataset Train and Test.    And after Airline we do same for Total\_Stops, Source and Destination and perform some **EDA** on that. And here also this is done for the both Train and Test data. I will show all this by inserting the screen shot of all the python coding.    Train Data:    Test Data:    Now plotting the count plot of the ‘Source’ and the ‘Destination’    By above countplots we can clearly see that which city is the highest operating Source and which city has the highest Destination. From first graph it is clearly seen that Delhi is the place where the highest number of flights are take off as a Source and Cochin is the place where the highest number of flights are landed as Destination.  Now replacing the Source and Destination with the integer value. How it get replace and what python coding is used for that ar as below:    Now all of the data is now in numeric form which should be needed in model building. Now lets describe() the train dataand see what output comes:    Here we can see the mean and standard deviation of the data of all the columns.  After all process is done, we plot heatmap of correlation matrix of the train data and see the correlaion of the train data.    After this we see the shape of the Train data and Test data and checking for any null values in the Test data.    At last we plot the pairplot of the train data.    In pairplot of the data we can see there is lot of plots are shown, in which scatter plot, density plot, skewness of the data and many other data is shown. This pairplot plot all the columns present in the dat and showing the relation between them.  After all this Feature Engineering is done, we remove the null values of the data and we convert all the categorical values to the numerical values, and remove that columns which is not needed.  Now our data is ready for model building but before model building we have to split our data into X and Y. for splitting the data we asign the x and y variables.    Now after splitting the data into x and y variable, now we go for Train Test Split.  Here we assign how much of the our data go for training and testing, for doing so we have to import first train\_ test\_split and some other codes also which we needed later after train\_test\_split.    So above is the python coding for the train\_test\_split. Here we doing the training of the 70% data and testing will be done on the 30% data with random\_state of 42 i.e. taken randomly. And we can see the shape of the x\_train and y\_train of the data and also the shape of the x\_test and y\_test of the data.  Now our splitting and the training and testing of the data part is done, now our data is fully ready for the model building process. So lets build the model and predict the price of the flight.  ***Model Building:***  The goal in this step is to develop a benchmark model that serves us a baseline, upon which we will measure the performance of a better and more tunned algorithim. We are using different regressin technique and comparing them to see which algorithim is giving better performance then other and we will also go for a cross validation check for seeing that the model is not in overfitting/underfitting .  I am using Linear Regression from linear model, SVR from svm, Decidion Tree Regressor from the tree and also using some ensemble techinques like Random Forest Regressor, Gradient Boosting Regressor and Ada Boost Regressor.  Because our model is regression type model so we use r2\_score, mean\_squared\_error, and mean\_absolute\_error.    Above we can see that we are imported the models and define all the models.  Now we do the python coding for predicting the r2\_score and mean\_squared\_error, and mean\_absolute\_error. In this code we call all the algorithim at one time by using for loop, we also check cross\_val\_score in this coding itself. So lets put the screenshot of python coding for better understanding and see what happens and which model is giving us the best pridiction score.    We can see Random Forest Regressor is given best r2\_score.  Now we predict the x\_test for the selected model, and predict the total pridected price of the x\_test.    We can see the Total Predicted Prices for the x\_test by the selected model.    From above we can see the Actual Price and the Predicted price and what was the error in the Actual Price and the Predicted Price.  Lets see the Train score and the Test score of the selected Model.    We can se the test score of the Random Forest Regressor model, for better test score we can apply **Hyper Parameter Tuning** to the test score.    We can see our Test score is now increased by .02 % and removed error coming in the **Total Predicted Price.**  Now dropping the unusual columns from the Test data and checking the columns of the Test data and Predict the Final Price for the Test data.    We can see above the final prediction for the test data.  And loading the final prediction for the test data to the DataFrame and seeing the Price columns for the Test data which is predicted Price for the flight.  Now saving the model by importing joblib using .pkl method.    Here we can see that our model is saved.  ***END NOTES:***  In this type of problem **Feature Engineering** is the most crucial think. We can see how we have handled the categorical and numerical data and we also do EDA on the Train Data. After that we remove the unneccesary columns and clean the data, remove the null values from the data and prepare the data for the model building. After that we build the model by using different **ML Model** on the same dataset. We also check r2\_score, MSE, MAE for each model, so that we can understand how it should perform in our test dataset. After that we perform **Hyper Parameter Tuning** for the the selected model which has the highest r2\_score and after doing that we get better score, which help in removing the error in the Predicted Price also.  And at last we Predict the **Final Price** for the **test data.** And save the model**.** |

# 