***Flight Price Predictio***

* Engineered features from the Departure Time, Date of Journey, to quantify the data and make it more understandable.
* Optimized multiple Regression models using GridsearchCV to reach the best model.

**Codes and Resources Used:**

* Python Version: 3.7.6
* Packages: pandas, numpy, sklearn, matplotlib, seaborn, flask, pickle.
* Dataset: F:\Datasets\Flight data\Data\_Train.xlsx

**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 you will be provided with prices of flight tickets for various airlines between the months of March and June of 2019 and between various cities.

#### 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

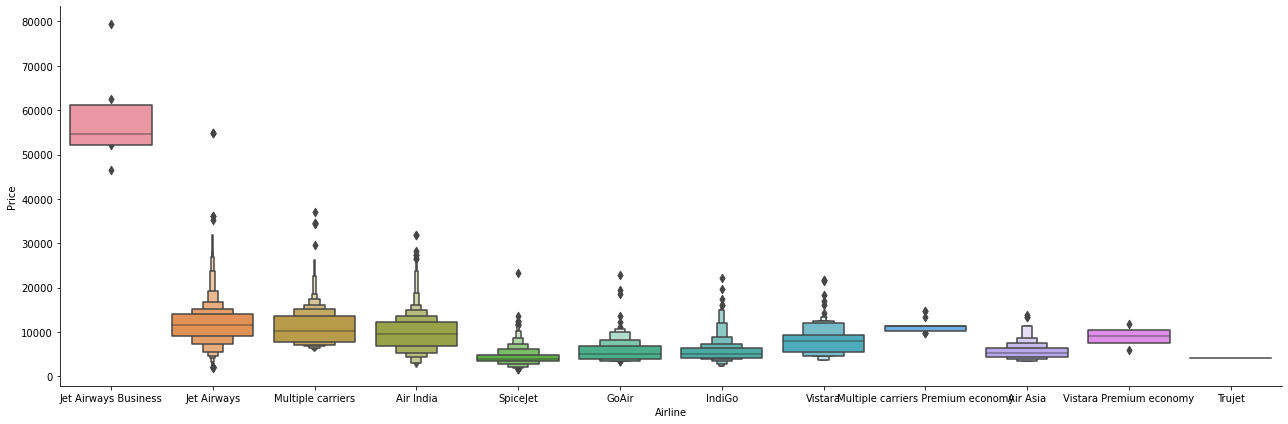
# Cleaning the Data:

I needed to clean it up so that it was usable for our model. I made the following changes and created the following variables:

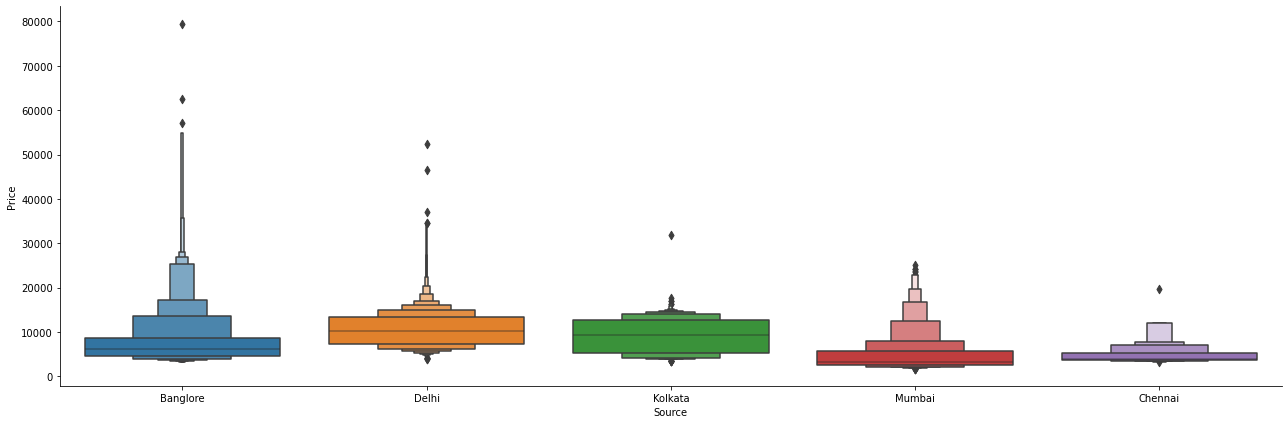
* Made Columns for Day and Month out of Date of Journey
* Made columns for dep hrs and dep min out of dep\_time
* Made columns for arrival hrs and arrival min out of Arrival\_Time
* Made columns duration hrs and duration min out of Duration
* Remove the Additional\_Info due to large amount of no\_info
* Removed the null values
* Removed the outliers

# EDA:

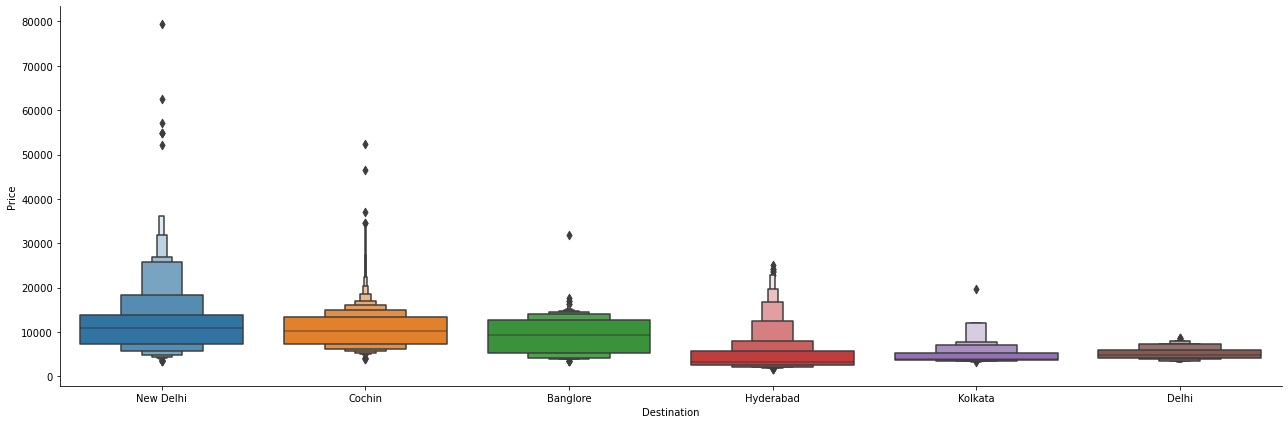
* Create catplot between price and airlie



* Create the dummies columns for Airline feature.
* Create catplot between price and source



* Create the dummies columns for Source feature.
* Create catplot between price and Destination.



* Create the dummies columns for Destination feature.
* Convert the total stop feature from object to int64.
* Create the new dataset by concatenating all the dummies data and old dataset. Then drop the Airline, Source and destination feature and create the final dataset with no object. There is only int data after do all these things.
* Load the test dataset and apply all same EDA which I applied on the train dataset.

# Model Building:

 I split the data into train and tests sets with a test size of 20%.

I tried four different models and evaluated them using Root Mean Squared Error. I chose RMSE because it is relatively easy to interpret and outliers aren’t particularly bad in for this type of model.

Different models I tried:

( Root Mean Squared Error before hyperperameter tunning)

LinearRegression : 2863.9705929717807

XGBRegressor : 1820.2734760131596

RandomForestRegressor : 2096.4787415802357

DecisionTreeRegressor : 2401.22784029413

(Mean Squared Error before hyperperameter tunning)

LinearRegression : 8202327.557407134

XGBRegressor : 3313395.5274770306

RandomForestRegressor : 4395223.113897848

DecisionTreeRegressor : 5765895.141003613

DecisionTreeRegressor, RandomForestRegressor and gave the lowest RMSE so I chose these model and performed hyper parameter tuning(RandomizedSearchCV).

Finaly RandomForestRegressor gave lowest RMSE.

# Model Accuracy:

# RandomForestRegressor:

**MAE: 1156.8417477850164**

**MSE: 4072272.0660317563**

**RMSE: 2017.98713227606**