# Project Report on-

# Flight Price Prediction

# By:

# Simran Kumari

**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. As data scientists, we are going prove that given the right data anything can be predicted. Here we were provided with prices of flight tickets for various airlines between the months of March and June of 2019 and between various cities.

Size of training set: 10683 records

Size of test set: 2671 records

**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

**Data Analysis**

We identify the below mentioned points in the first look –

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

2. The Arrival time column has dates attached along with, which we will need to separate. These are the cases when the flight takes off from the source on a date and reaches its destination on the next day.

3. The Duration is in a string format, which we will need to convert to integer type.

4. We observe that we have all the columns as ‘object’ data types, and only ‘Price’ column (the output) is of integer type. Since we know what our columns signify, we know which columns we need to treat!

5.We have 1 missing value in Route column, and 1 missing value in Total stops column. We will meaningfully replace the missing values going further.

We further proceed to explore the dataset.

We observe in categorical that –

1. 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'.

2. Source column has 5 unique values – ‘Bangalore’, ‘Kolkata’, ‘Chennai’, ‘Delhi’ and ‘Mumbai’.

3. Destination column has 6 unique values - 'New Delhi', 'Banglore', 'Cochin', 'Kolkata', 'Delhi' , 'Hyderabad'.

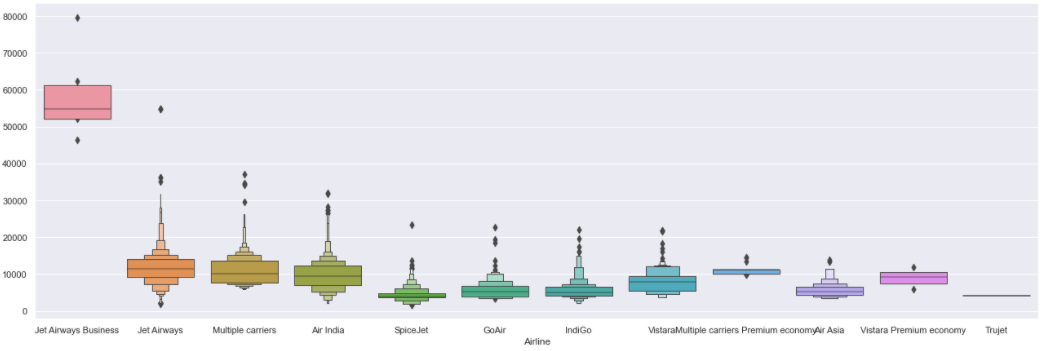
4. Additional info column has 10 unique values - 'No info', 'In-flight meal not included', 'No check-in baggage included', '1 Short layover' , 'No Info', '1 Long layover', 'Change airports' , 'Business class', 'Red-eye flight' , '2 Long layover'.

We split the Dep\_time column and arrival\_time, and create separate columns for departure hours and minutes and arrival hours and minutes

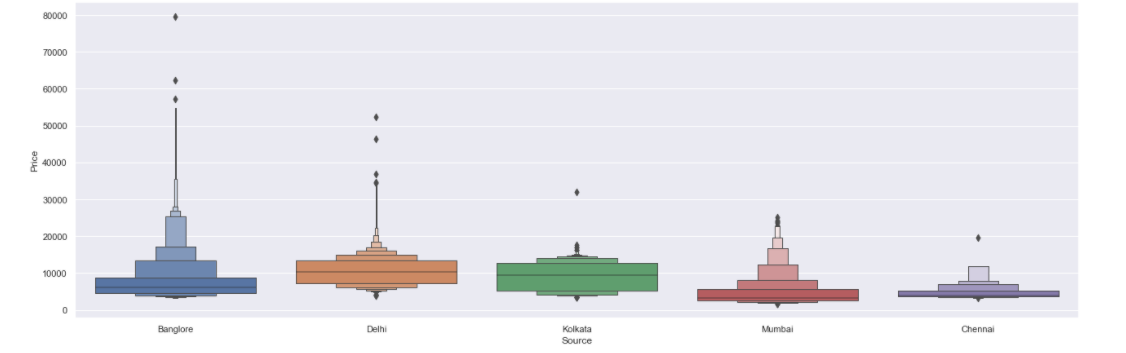
Next, we divide the ‘Duration’ column to ‘Duration\_hours’ and ‘ Duration\_mins

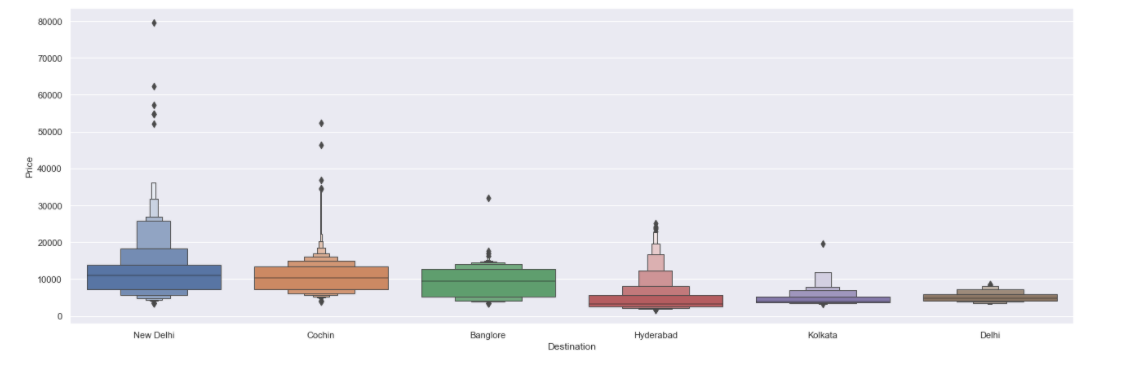
**EDA.**

**Airlines Affecting Prices.**

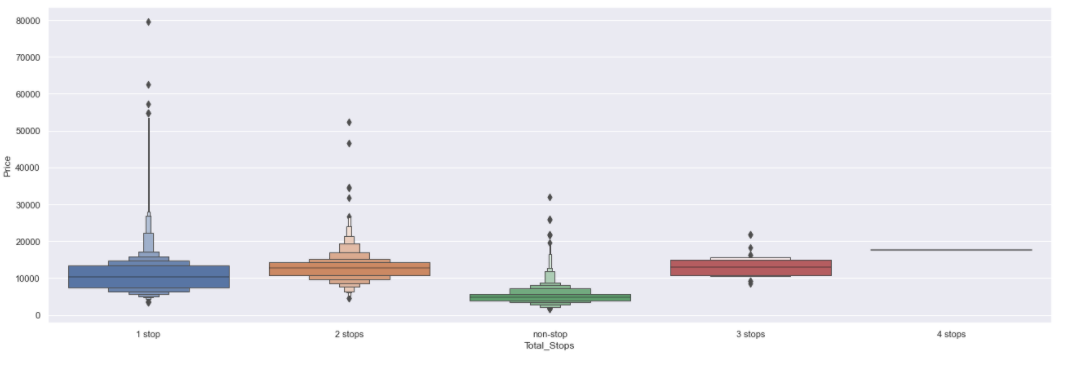


**Source and destination affecting Prices**

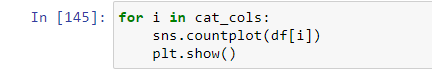




**Total stops affecting prices**



**Plotting countplots for categorical data**

****

Airlines

* Jet Airways is the most preferred airline with the highest row count, followed by Indigo and AirIndia.
* Count for Vistara Premium economy, Trujet, Multiple carries premium economy and Jet airways business is quite low.

Source

* Majority of the flights take off from Delhi
* Chennai has the minimum count of flight take-offs

Destination

* Maximum flights land in Cochin
* Kolkata has the lowest count of receiving the flights

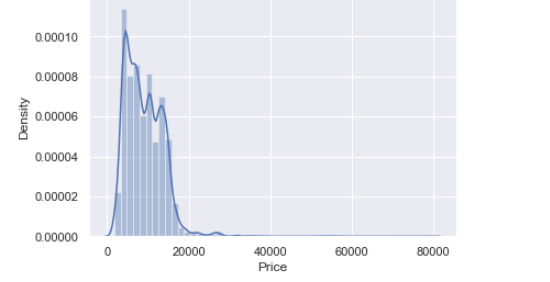
Total stops

* Majority of the flights have stops as 1, flights with 3 and 4 stops are quite low

Additional Info

* Maximum rows have No info as the value.
* We need to check how this column impacts the prices

Distribution of Price



The price column contains the minimum value as 1759 and maximum value as 79512. Majority of the flights have price range between 1759–20k, and number of flights having prices greater than 20k are quite less. Price range is skewed towards right.

We now proceed with checking the relation of ‘Price’ column with numerical data –

We observe that –

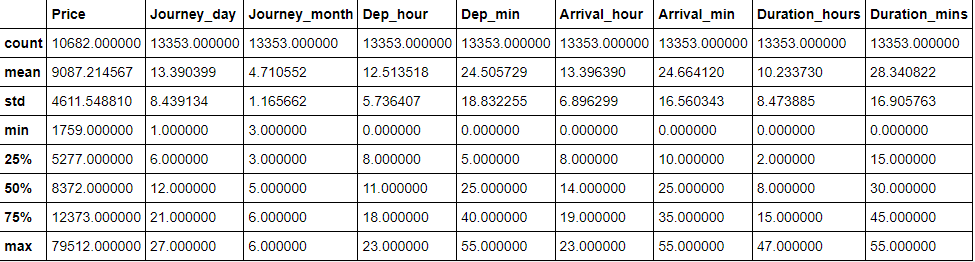
- As number of stops increase, the price range gets decreasing into a smaller price window (10k — 22k)

- High price flights are lesser during end of month

- Prices are higher in the month of March

- With increase in travel hours, price increases, but the number of flights decrease.

Below is the output for df.describe()



**Outlier Detection**

We make the below conclusions

- Outliers are present in Total hours, Total stops and price

- We will not remove outliers from total stops since price is impacted by number of stops

- We will not remove the data with high number of hours, increase in number of hours shows a price pattern in the above graphs plotted for EDA.

**Data was mostly left skewed hence removed**

Additional info in the train data set has values ‘No info’ and test has ‘No Info’ .

Combined the data set together for preprocesssing.

**Pre processing pipelines**

Label Encoded all the data categorical data- We encode the categorical data in this step, to convert it to integer type, since the model does not work on ‘string’ data. We use ‘Label Encoder’ to achieve the desired results

Regularised the data using standard Scaler.- The next step is to bring the data to a common scale, since there are certain columns with very small values and some columns with high values. This process is important as values on a similar scale allow the model to learn better. We use standard scaler for this process

Removed the skewness using power transform-yeo-Johnson- The Yeo–Johnson transformation allows also for zero and negative values in the dataset.

**Building Machine Learning Models**

I tried eleven different models and evaluated them using r2 score.r2 score varies between 0 and 100%. It is closely related to the MSE (see below), but not the same. R2 can be defined as the proportion of the variance in the dependent variable that is predictable from the independent variable(s).

I chose r2 because it is relatively easy to interpret and outliers aren’t particularly bad in for this type of model.

Different models I tried:

| Models | CVS | R2 | diff |
| --- | --- | --- | --- |
| Linear regression | 42.29 | 42.35 | 0.06 |
| Lasso | 42.35 | 42.19 | -0.16 |
| Ridge | 42.45 | 42.35 | -0.10 |
| Elastic Net | 42.25 | 42.35 | 0.10 |
| Decision Tree | 82.56 | 91.48 | 8.92 |
| SVR | 36.86 | 37.57 | 0.71 |
| K Neighbors | 78.71 | 99.60 | 20.89 |
| Random Forest | 88.22 | 98.10 | 9.88 |
| Extra Tree Regressor | 76.72 | 99.51 | 22.79 |
| Ada Boost | 55.77 | 57.81 | 2.04 |
| **Gradient Boost** | **80.13** | **82.35** | **2.22** |
| XGBRegressor | 89.26 | 96.87 | 7.61 |

* #From the above analysis Gradient Boost Regressor has least difference between r2 and cvs

Using hyper parameter tuning on GradientBoostingRegressor further increased the accuracy.

Gradient boosting is a type of machine learning boosting. It relies on the intuition that the best possible next model, when combined with previous models, minimizes the overall prediction error. ... If a small change in the prediction for a case causes no change in error, then next target outcome of the case is zero.

GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage a regression tree is fit on the negative gradient of the given loss function.

# Hypertuning the model

GridSearch CV is a technique used to validate the model with different parameter combinations, by creating a grid of parameters and trying all the combinations to compare which combination gave the best results.

# Cross Validation

We perform the cross validation of our model to check if the model has any overfitting issue, by checking the ability of the model to make predictions on new data, using k-folds. We test the cross validation for all models.

**CONCLUSION**

We further proceed to test the object that we saved using pickle, and create a dataframe of predicted values.



In the proposed paper the overall survey for the dynamic price changes in the flight tickets is presented. This gives the information about the highs and lows in the airfares according to the days, weekend and time of the day that is morning, evening and night. also the machine learning models in the computational intelligence field that are evaluated before on different datasets are studied, their accuracy and performances are evaluated and compared in order to get better result. For the prediction of the ticket prices perfectly different prediction models are tested for the better prediction accuracy. To get result with maximum accuracy regression analysis is used. From the studies, the feature that influences the prices of the ticket were considered. In future the details about number of available seats can improve the performance of the model.