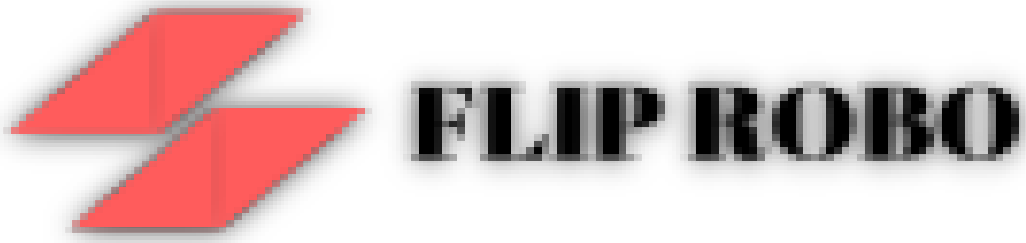


Flight Price Prediction

Submitted by:

Utsav Rastogi

ACKNOWLEDGMENT

This project was made possible with the help of my supervisor Swati Mahaseth. Under her guidance, I was able to complete the project on time and ample learning possibilities this project provided me with.

This project gave me the glimpse of how real-world datasets are mined and what is actually need to be provided to the client in terms of solutions. This was another experience with a dataset that I crafted for the client who required current prices of flights to and from major cities.

Data analysis and preprocessing was done keeping in mind how to make the code run efficiently and quickly with utilizing minimal memory. I had to refer to various articles on Medium, Analytics Vidhya, Kaggle and stack exchange to look for potential debugging solutions and visualization plots.

INTRODUCTION

Business Problem Framing

This project tries to shed light on an efficient way to predict the price of air itineraries across India. This project meant to help understanding the various factors at play when predicting the price of flights from and to major cities. The predictive model can help the company optimally price their air journeys to attract customers, and enhance their marketing strategy to outcompete other players from the market.

Conceptual Background of the Domain Problem

Pricing flights can be optimally done using statistical and predictive modelling. Such prices if set properly can easily generate higher revenue which can benefit the consumers and the company equally.

Review of Literature

This dataset was mined from "Yatra.com". All the itineraries begin on 1st march, 2022. This data was further analysed and cleaned with standard pre-processing steps which will be subsequently used to create a predictive model. The model predicts the price of the flight journeys using several independent features. The model is based on regression algorithms which are usually used to predict a continuous target variable.

The model with least difference in r2 score and cross validation score is finalized and saved in a pickle file. This model can be further integrated by webapps to check price of any flight journey in India.

Motivation for the Problem Undertaken

The objective behind this project was to provide the client with an overview of how flight prices can be marked efficiently with a statistical model. The growth of commercial aviation sector is well complemented by the growth of incomes and the demand for domestic air travel as well.

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

Statistical analysis	Zscore was used to determine the outliers in the data and Correlation helped understand the relation between features and target. Skewness was also used to check the distribution and reduce it in the features which weren't gaussian.
Visual analysis	Libraries used: Matplotlib and Seaborn
Algorithms	Different Regression algorithms such as XGB Regressor & Lasso Regression were used which are all provided by sklearn library.

Data Sources and their formats

1. Data is sourced from:

   Data has been mined from Yatra.com

2. Data has been stored in the CSV format
3. Data Features:
   - Data consists of both numerical and categorical columns.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2139 entries, 0 to 2138
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Name            2139 non-null   object
 1   Duration        2139 non-null   object
 2   Layovers        2139 non-null   object
 3   Departure_time  2139 non-null   object
 4   Arrival_time    2139 non-null   object
 5   Source          2139 non-null   object
 6   Destination     2139 non-null   object
 7   Details         2139 non-null   object
 8   Price           2139 non-null   object
dtypes: object(9)
memory usage: 150.5+ KB
```

4. Null values are absent.

Data Pre-processing Done

1. Dropped ‘Unnamed: 0' which was unique for each data point.
2. Label encoding the categorical features with dtype = ‘object’
3. Converted ‘Price’ to int32 format.
4. Dropped 75 duplicates from the dataset.
5. Converted Arrival_time and Departure_time to datetime object.
6. Extracted hours and minutes from both and also from Duration.
7. Dropped Duration, Arrival_time and Departure_time.
8. Label encoded categorical columns.
9. Removed outliers and reduced skewness in “Dur_hours”.
10. Scaled the training dataset excluding the target column “Price”.

Data Inputs- Logic- Output Relationships

Highest relationship is between Layovers and Dur_hours of the journey.

Hardware and Software Requirements and Tools Used

Hardware: Windows 11

Software: Jupyter notebook

Libraries: seaborn, matplotlib, statsmodels, numpy, pandas, sklearn, scipy, pickle

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods)

The dataset was analyzed using the standard procedure where we indulge in statistical and graphical analysis of the dataset. Statistical methods like Zscore were used but couldn't be implemented as the loss in data was huge around 25% which is not feasible as data is expensive .

Best random state is found and is used to split the data to obtain an optimum score.

Regression algorithms were used to create a model as the target variable was a continuous variable.

Testing of Identified Approaches (Algorithms)

1. Lasso Regression
2. Decision Tree Regressor
3. Random Forest Regressor
4. XGB regressor

Run and evaluate selected models

```
ls = Lasso(alpha=10,random_state=0)
ls.fit(x_train,y_train)
ls.score(x_train,y_train)
pred_ls = ls.predict(x_test)
```

```
lss = r2_score(y_test,pred_ls)

for k in range(2,10):
    lsscore=cross_val_score(ls,x,y,cv=k)
    lrcv=lsscore.mean()
    print("At cv= ",k)
    print("Cross Val score : ",lrcv*100)
    print("r2 score is : ",lss*100)
    print("\n")
```

```
dtr = DecisionTreeRegressor(criterion='friedman_mse',
                            max_depth= 8,
                            max_leaf_nodes= 15,
                            min_samples_leaf= 40,
                            min_samples_split= 10)
dtr.fit(x_train, y_train)
pred_dtr= dtr.predict(x_test)
```

```
dtrr2 = r2_score(y_test,pred_dtr)

for k in range(2,10):
    dtrscore=cross_val_score(dtr,xs,y,cv=k)
    dtrcv=dtrscore.mean()
    print("At cv= ",k)
    print("Cross Val score : ",dtrcv*100)
    print("r2 score is : ",dtrr2*100)
    print("\n")
```

```
rfr = RandomForestRegressor(criterion='poisson',
                            max_depth= 10,
                            max_features="sqrt",
                            n_estimators=200)
rfr.fit(x_train, y_train)
pred_rfr= rfr.predict(x_test)
```

```
rfrr2 = r2_score(y_test,pred_dtr)

for k in range(2,10):
    rfrscore=cross_val_score(rfr,xs,y,cv=k)
    rfrcv=rfrscore.mean()
    print("At cv= ",k)
    print("Cross Val score : ",rfrcv*100)
    print("r2 score is : ",rfrr2*100)
    print("\n")
```

```
xgb = XGBRegressor(n_estimators= 200,
                   importance_type= 'gain',
                   eta=0.1, booster='gbtree')
xgb.fit(x_train,y_train)
xgb.score(x_train,y_train)
pred_xgb = xgb.predict(x_test)
```

```
xgbs = r2_score(y_test,pred_xgb)

for k in range(2,9):
    xgbscore=cross_val_score(xgb,xs,y,cv=k)
    xgbcv=xgbscore.mean()
    print("At cv= ",k)
    print("Cross Val score : ",xgbcv*100)
    print("r2 score is : ",xgbs*100)
    print("\n")
```

r2 Score:

Model	Score
Decision Tree Regressor	44.676337881779304
Random Forest Regressor	44.676337881779304
Lasso Regression	38.361184528699674
XGB Regressor	70.65154124703409

Cross Validation Score:

Model	Score
Decision Tree Regressor	5.694688746255702
Random Forest Regressor	-43.097194856804045
Lasso Regression	8.440088154115239
XGB Regressor	-32.231021461641305

Key Metrics for success in solving problem under consideration

The least difference between r2score and cv score was seen in gradient boosting regressor hence proving least fitting problem. Hence it is considered for modelling the data under consideration (test data).

The r2 score was used.

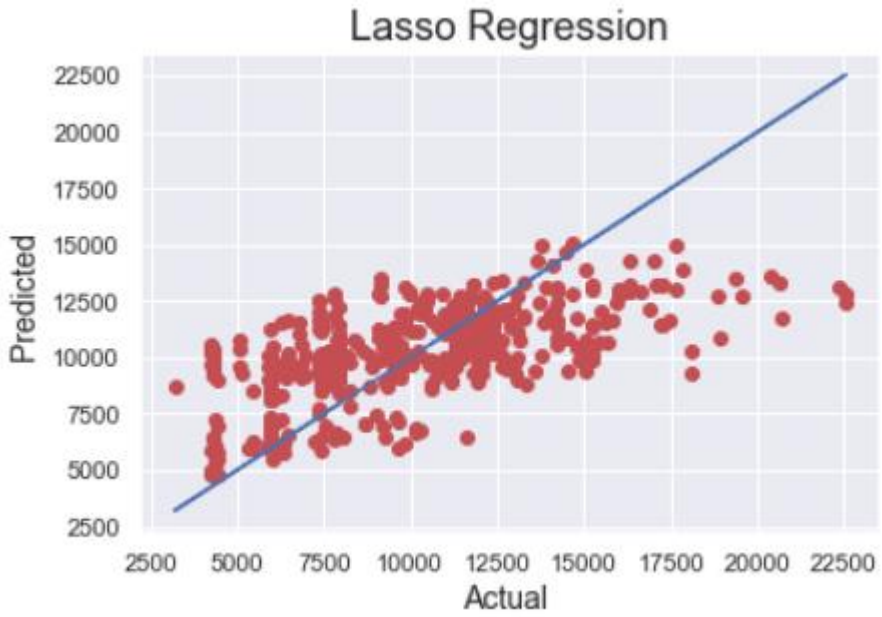
R-squared is a metric of correlation. Correlation is measured by “r” and it tells us how strongly two variables can be related.

A correlation closer to +1 means a strong relationship in the positive direction, while -1 means a stronger relationship in the opposite direction.

A value closer to 0 means that there is not much of a relationship between the variables. R-squared is closely related to correlation

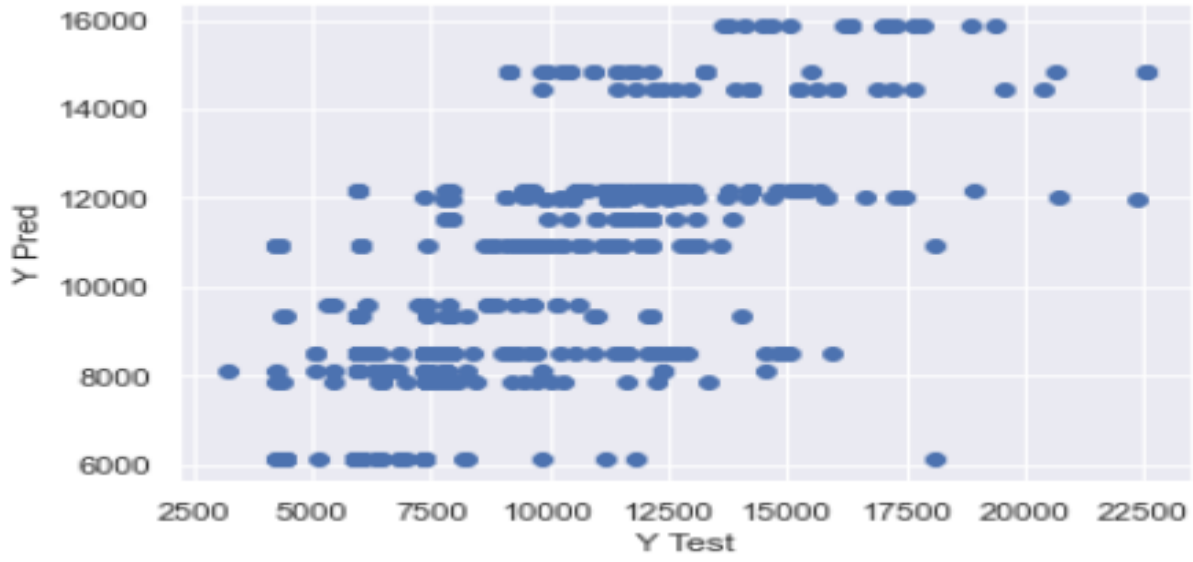
1. Lasso Regression

```
MAE: 2201.2955460122794
MSE: 7870581.382909249
RMSE: 2805.45564622028
```



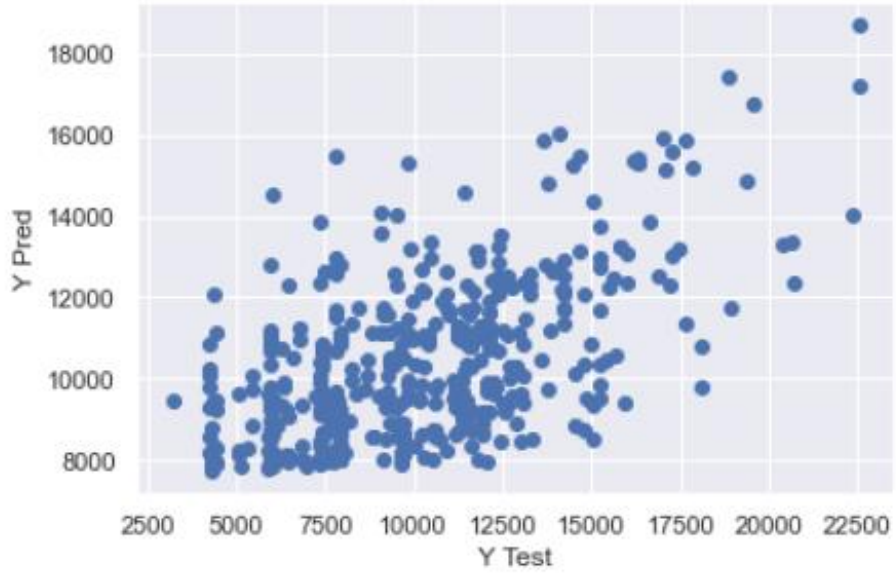
2. Decision Tree Regressor

```
MAE: 1936.9859644962555
MSE: 7064207.541508805
RMSE: 2657.8576977537386
```



3. Random Forest Regressor

```
MAE: 2305.5656981114385
MSE: 8332299.444731821
RMSE: 2886.5722656347652
```

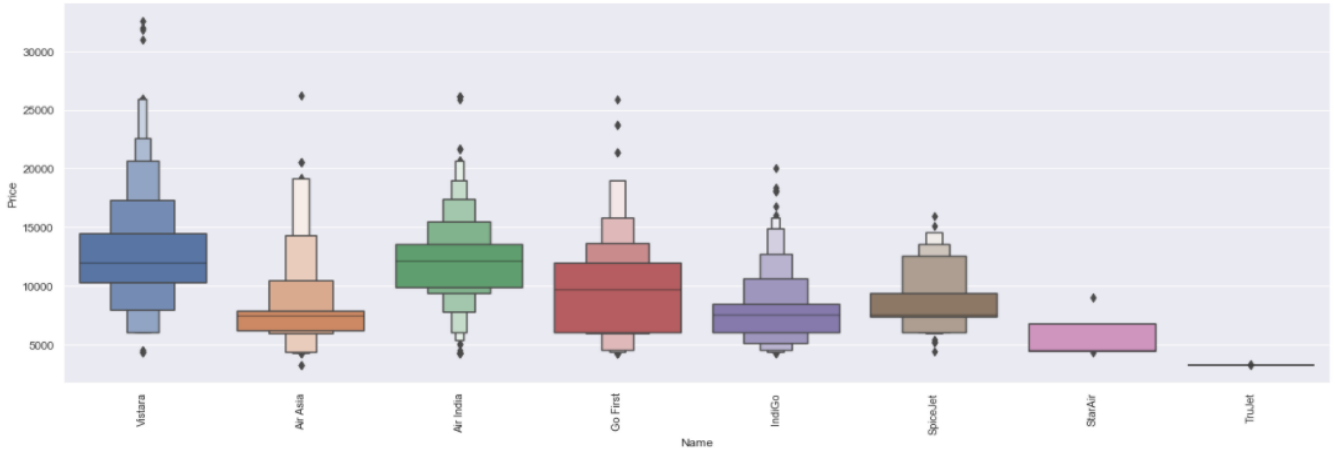


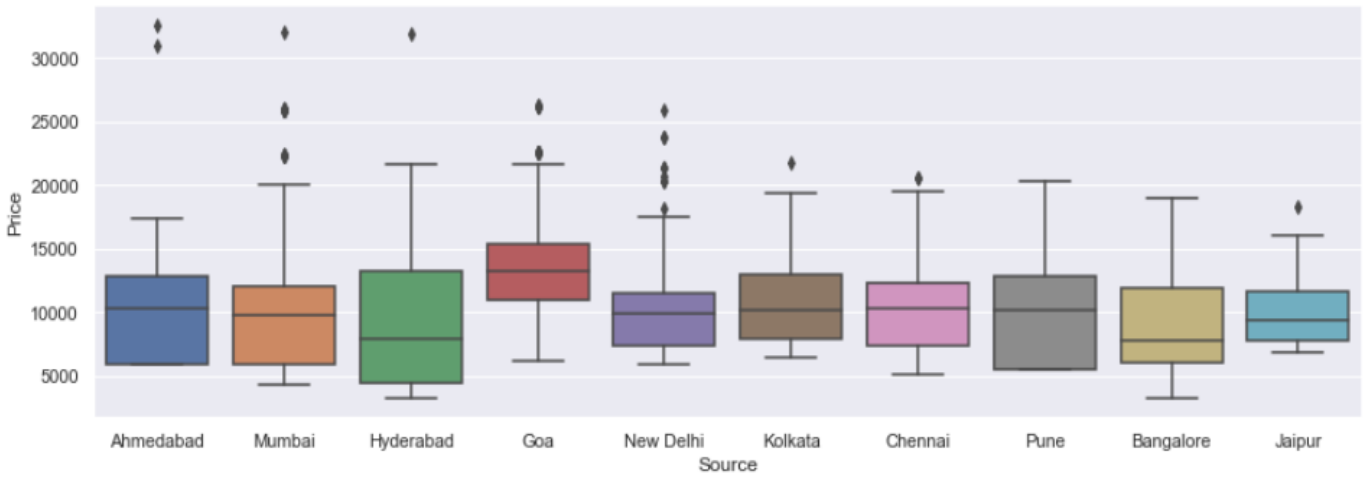
4. XGB Regressor

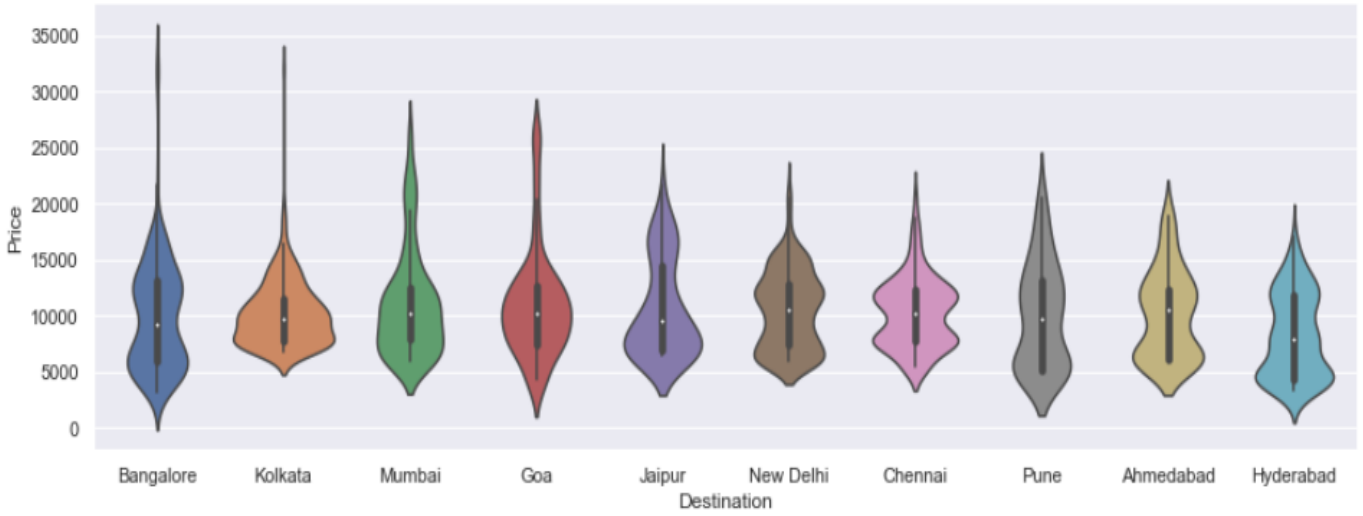
```
MAE: 1346.0690571289063
MSE: 3747467.100267762
RMSE: 1935.8375707346322
```

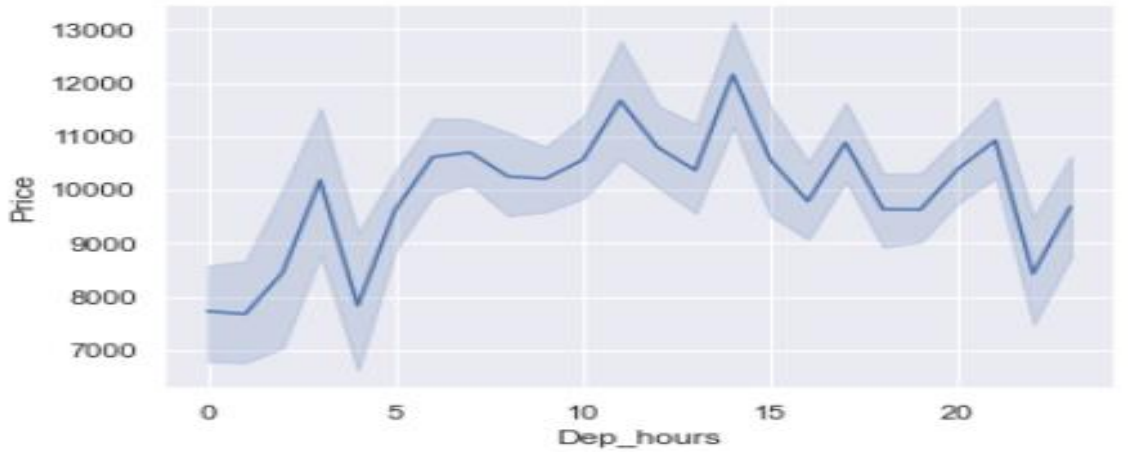
Visualizations

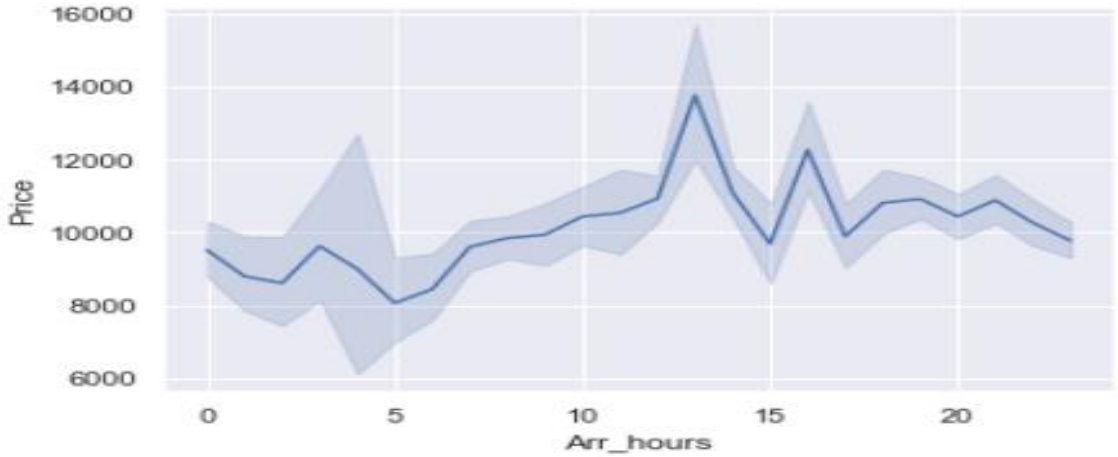
- Features plotted against target column



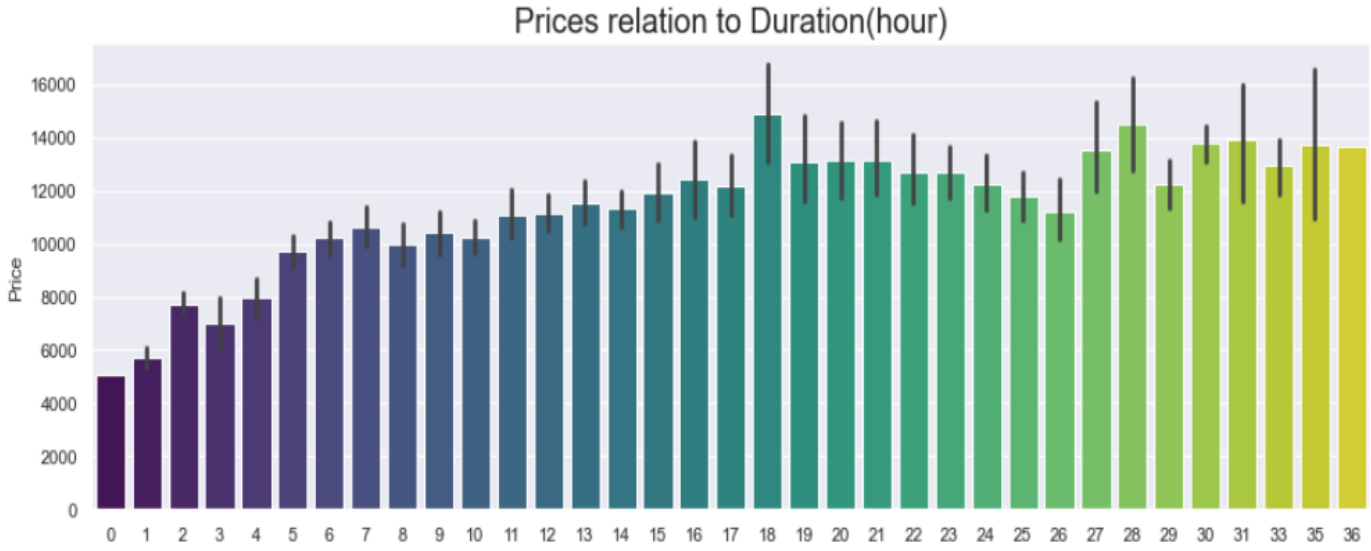


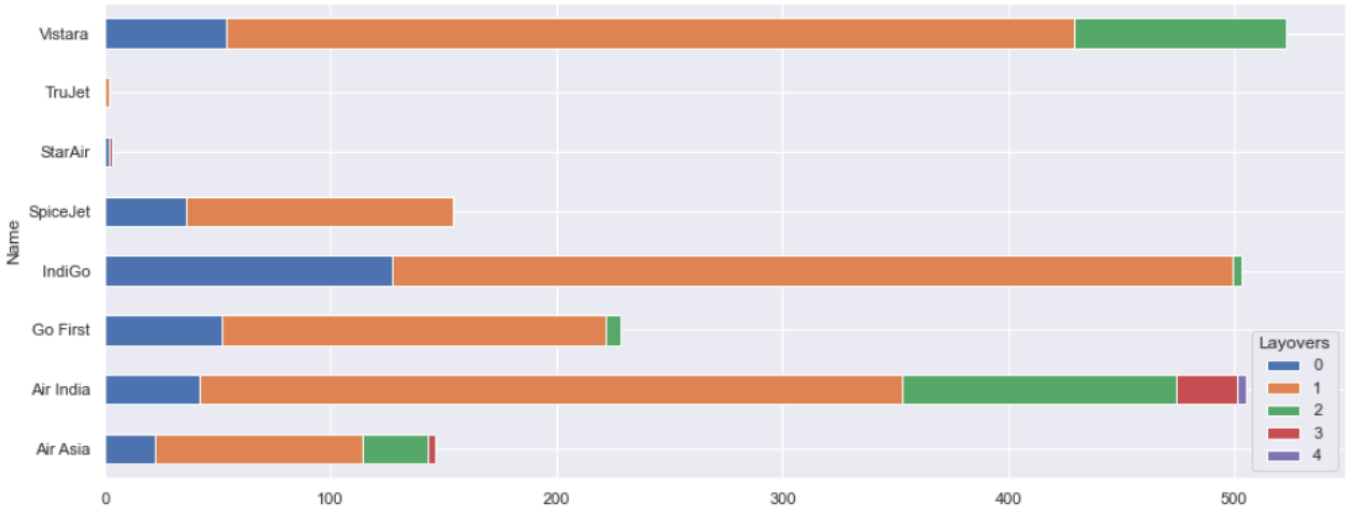




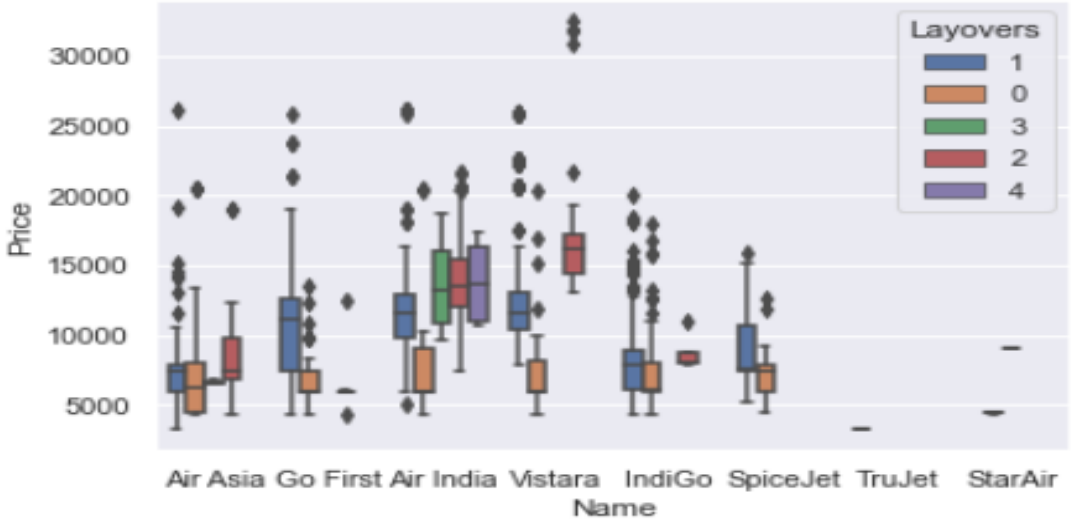


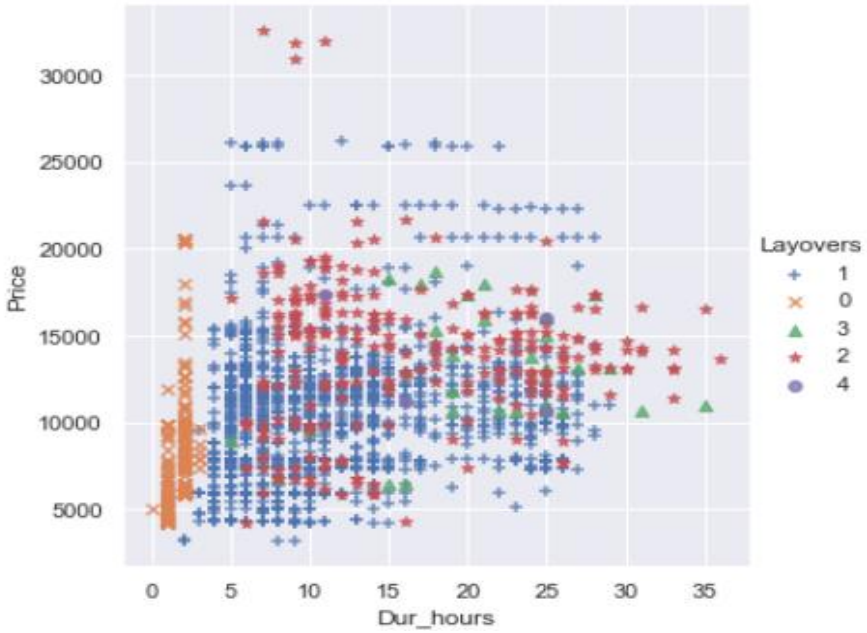
- Bivariate Analysis:



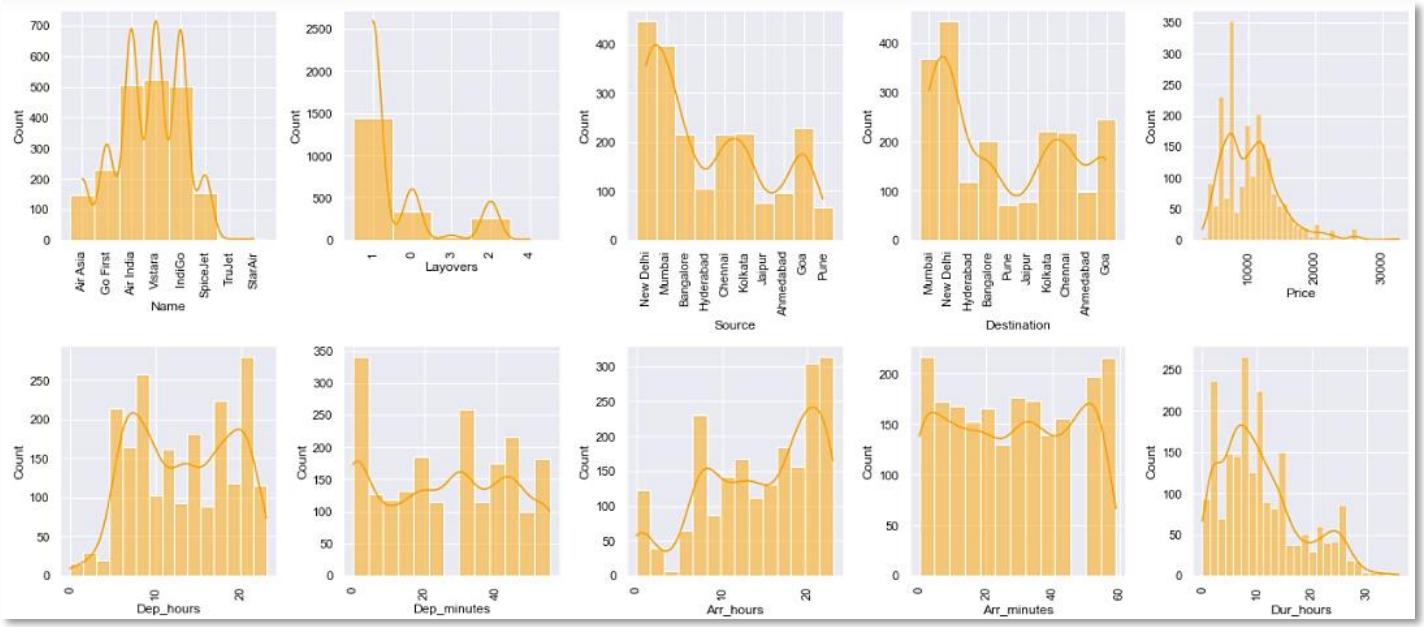


- Multivariate plots:





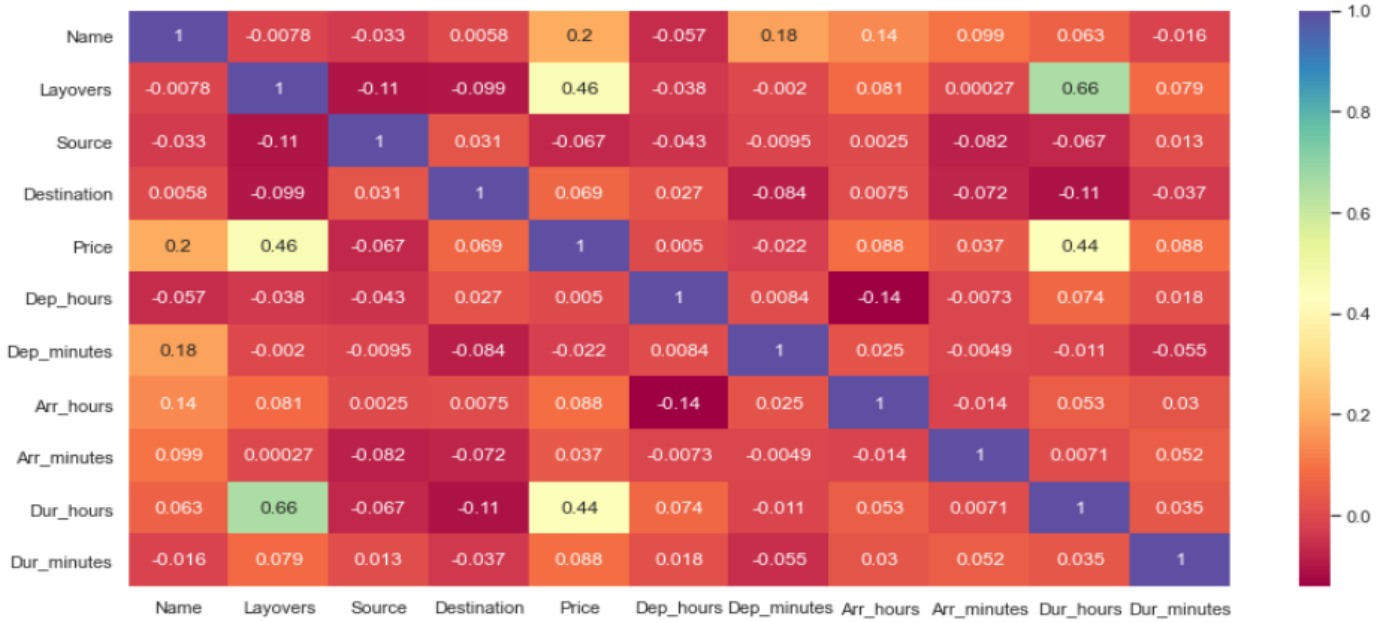
- Feature distributions:



- VIF (to check for multicollinearity)

	variables	VIF
0	Name	1.079866
1	Layovers	1.817606
2	Source	1.022604
3	Destination	1.029826
4	Dep_hours	1.047489
5	Dep_minutes	1.047734
6	Arr_hours	1.046857
7	Arr_minutes	1.026171
8	Dur_hours	1.810618
9	Dur_minutes	1.015986

- df correlation heatmap



Key Findings and Conclusions of the Study

- Vistara, Air India and Indigo offers the flights to majority of the destinations.
- New Delhi and Mumbai are top destinations searched.
- Majority of the flights originate from New Delhi and Mumbai.
- Almost 75% of the flights in the dataset have 1 stop during journey.
- Majority of the flights cost less than 15k rupees.
- Less number of flights depart early morning (before 5am).
- Almost half the flights reached in the late evening and night (after 6pm).
- Most of the flights have a duration of less than 12 hours.
- Vistara have the most expensive flights with air India at the second spot.
- Air Asia is the cheapest flight in the dataset.
- Vistara and Indigo have the most “1 stop “layovers.
- Indigo also operated the largest non-stop flights in India.
- Air India has the maximum number of “2 stop” layover flights with Vistara coming in at second place.
- Air India is also the only carrier to operate flights having 4 stops and is also the major operator of 3 stop flights.
- Hyderabad as the destination have the cheapest flights.
- Bangalore and Kolkata as destinations have the most expensive flights.
- Flights from Goa are usually more expensive than other cities.
- Flights from Hyderabad are the cheapest among the group.
- Flights departing from 10am to 3pm have a high price tag as it’s the most convenient time to leave.
- Flight arriving in the timeframe from 12pm to 4 pm are the costliest as this is also a good time to arrive in broad daylight.

- Longer the duration, more expensive the flight is except flights are most expensive when the duration is around 18 hours.
- Higher number of stops do not translate to a higher price.
- 1 stop layover flights disregarding the duration are always the most expensive. Some outlier 2 stop flights with lesser duration have the highest price tag.
- Non-stop flights are cheaper for every airline.
- Vistara's 2 stop layover flights have a higher price compared to air India in the same category.
- Go First airline charges the most for 1 stop flights among the group.
- Air India's 4 stop airline is still cheaper than Vistara's 2 stop flight.
- IndiGo has the cheapest 1 stop flight journey among its rivals.

Learning Outcomes of the Study in respect of Data Science

This dataset sheds light on the power of visualization and EDA. As the data is largely limited which makes it expectedly inconclusive to perform exploratory data analysis and reach any fruitful conclusions. I learnt different ways to manipulate code to create efficient graphs which can reveal more information significantly.

Limitations of this work and Scope for Future Work

The dataset is severely limited and can be expanded extensively with data mining. The number of datapoints is very low. To have an accurate model, more data needs to be mined and cleaned. Data can become difficult to interpret without a context.