Flight Ticket Price Prediction

Submitted By,
Poovarasi Vijayan

Acknowledgement:

I wish to express my sincere thanks to the following companies, without whom I would have not got opportunity to work on this project; Data Trained Institute and Flip Robo Technology-Bangalore.

Introduction:

Business Problem Facing:

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.

Anyone who has booked a flight ticket knows how unexpectedly the prices vary. The cheapest available ticket on a given flight gets more and less expensive over time. This usually happens as an attempt to maximize revenue based on –

1. Time of purchase patterns (making sure last-minute purchases are expensive).
2. Keeping the flight as full as they want it (raising prices on a flight which is filling up in order to reduce sales and hold back inventory for those expensive last-minute expensive purchases).

Here we are trying to help the buyers to understand the price of the flight tickets by deploying machine learning models. These models would help the sellers/buyers to understand the flight ticket prices in market and accordingly they would be able to book their tickets.

Concept Background of the Domain Problem:

The main aim of this project is to predict the price of flight tickets based on various features. The purpose of this project is to study the factors which influence the fluctuations in the airfare prices and how they are related to the change in the prices. Then using this information, build a system that can help buyers whether to buy a ticket or not. So, we will deploy a Machine Learning model for flight ticket price prediction and analysis. This model will provide the approximate selling price for the flight tickets based on different features.

Review of Literature:

As per the requirement of our client, data is scraped from www.easemytrip.com, which is one of the platforms to book flight tickets. This project was built with a aim of gaining insights about various factors affecting the price of flight. This project is more about data exploration, feature engineering and pre-processing of data.

The goal of this project is to build an application which can predict the price of flight tickets with the help of the features available. In long term this would allow people to better explain and review their purchase in this digital world.

Motivation for Problem Undertaken:

This problem is taken based on the requirement of the client and also, with a curiosity to know how the flight of price changes frequently and on what basis it changes. And also, to get hands on experience and to know how it works on real bases.

Analytical Problem Framing:

Mathematical/Analytical Modelling of the Problem:

Effective Machine Learning model needs to be created which predicts the price of flight tickets. So 'price' is the target variable which is continuoes, so it is Regression problem where regression algorithm is used to predict the price of flight tickets.
Web Scraping is done to collect data from www.easemytrip.com.
First, the analysis is started with importing the data, this dataset contains Null values in 'price' column as it is target variable all null values are dropped.
Once data is cleaned outliers and skewness are checked, if present they are removed, then Data Pre-processing, Standard Scaler is used to standardize the data and VIF is checked for multicollinearity in dataset.
Once this is all done then data is ready for modelling. As the target variable contains continuous data, Regression is used. 4 regressors – Decision Tree Regressor, Support Vector Regressor, KNeighbors Regressor, Linear Regression, 4 ensemble -Random Forest Regressor, AdaBoost Regressor, Gradient Boosting Regressor, XGB Regressor, 3 metrics – r2_score, mean_squared_error, mean_absolute_error and 3 regularization – Lasso, Ridge, ElasticNet techinques are used in this project to build Regression model. From this the best model is identified and using Cross Validation technique is checked for overfitting and underfitting and Hypertuning is done to increase accuracy. Then, Finally the best model is saved.

Data Source and their Format:

The data is scrapped using selenium webscraping which is stored in xlsx format.

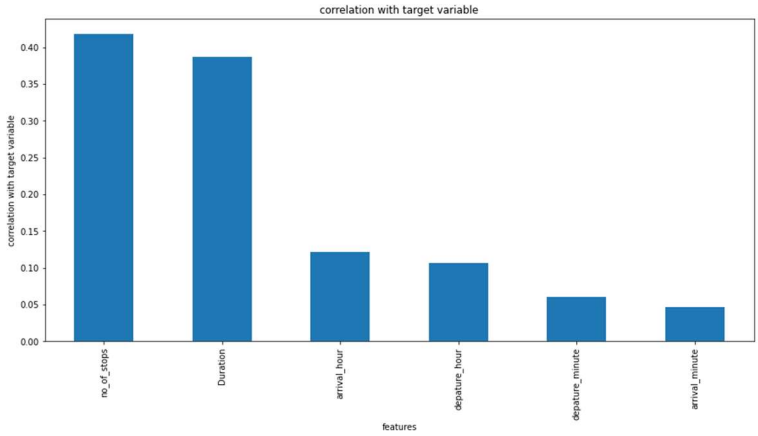
- Data contains 1829 entries each having 9 variables.
- Here the dataset is divided into independent and target variable.

Data Pre-Processing:

In Machine Learning, data pre-processing refers to the process of cleaning and organising raw data in order to make it appropriate for creating and training Machine Learning models. In other words, anytime data is received from various sources, it is collected in raw format, which makes analysis impossible. Data pre-processing is a crucial stage in Machine Learning since the quality of data and the relevant information that can be gleaned from it has a direct impact on our model's capacity to learn; consequently, we must pre-process our data before feeding it into our model. As a result, it is the first and most important stage in developing a machine learning model.
Some of the techniques used in this project are listed below:

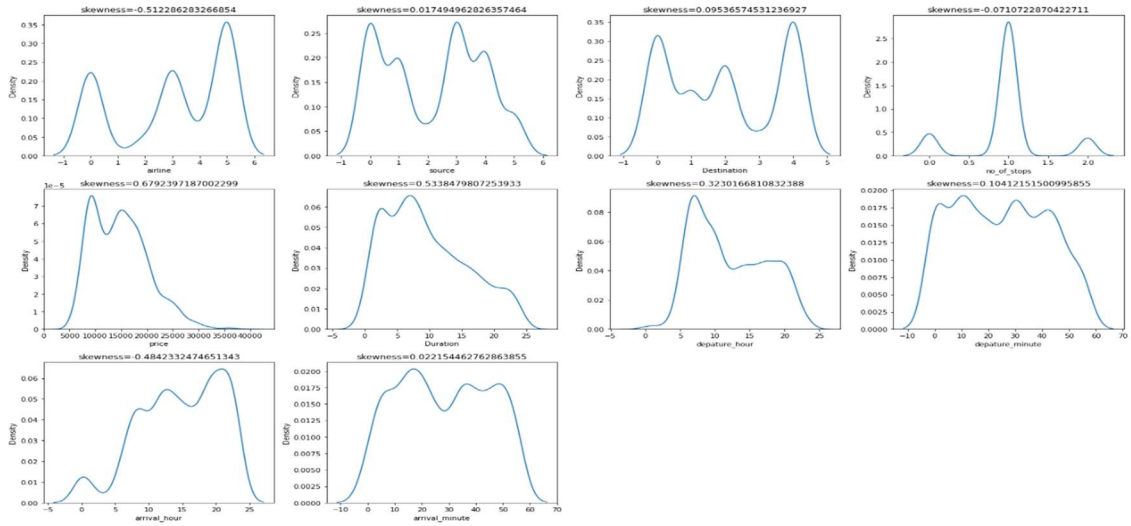
1. Started with web scraping using selenium from www.easemytrip.com website. Different flights are scrapped using different locations. Then the scraped data is stored in xlsx format.
2. Then the required libraries are imported and then the dataset which is in xlsx format.
3. Dataset contains 1829 rows and 9 variables. Then the information of the columns is observed carefully.
4. Few columns like 'Unnamed: 0' is dropped as it does not contribute much in model building.

5. Then Null values is checked, as there are few null values in target column, null values are dropped.
6. Numeric data and Categorical data are then separated so that visualization is done with distplot for numeric data and countplot for categorical data.
7. Then all the categorical data is converted to numeric by using LabelEncoder so that analysis can be made in better way.
8. Data Description is made followed by correlation where positive and negative correlated data is checked.

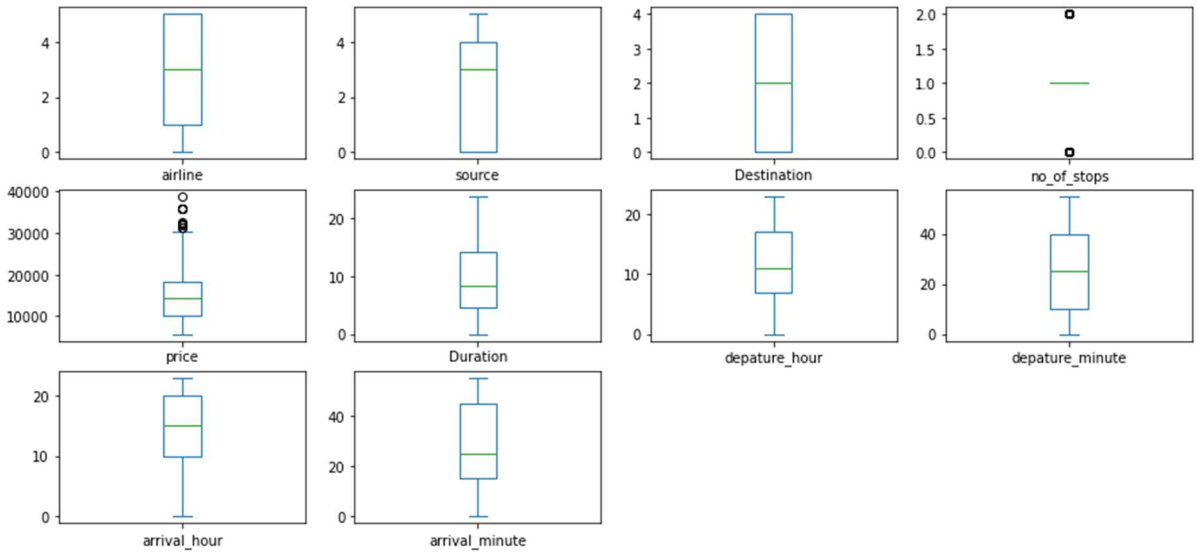


- 'no_of_stops','duration' is highly correlated with price which means if number of stops and duration decreases flight price increases.
- All other variables arrival_hour, depature_hour, depature_minute, arrival_minute are positively correlated with target variable.

9. Outliers and Skewness is checked and removed in order to avoid bias while model building.



Keeping +/- 0.5 as skew value, all columns have less than .5.



- Outliers are removed by using z-score method.

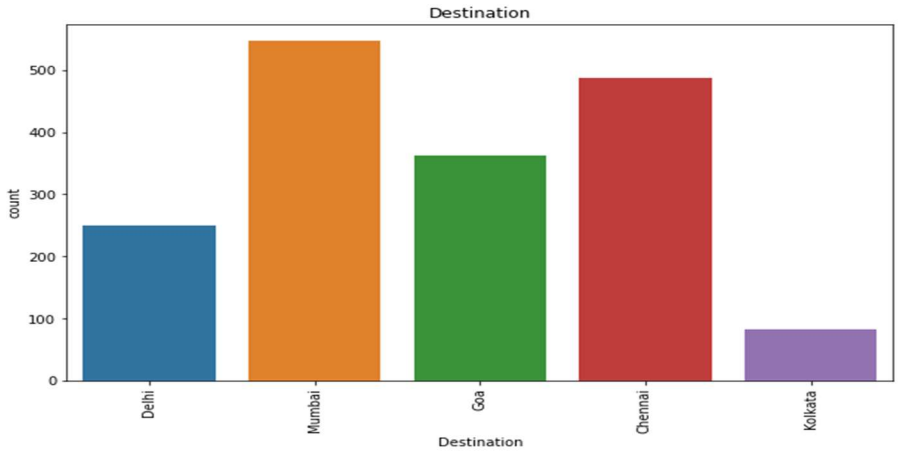
10. Standard Scaler is used for data standardization and VIF is used to check Multicolinearity is checked to find if any data variable is correlated with each other and it is removed.
11. Once all these processes are done then data is ready for model building where various Machine Learning models is used to check the accuracy of data.

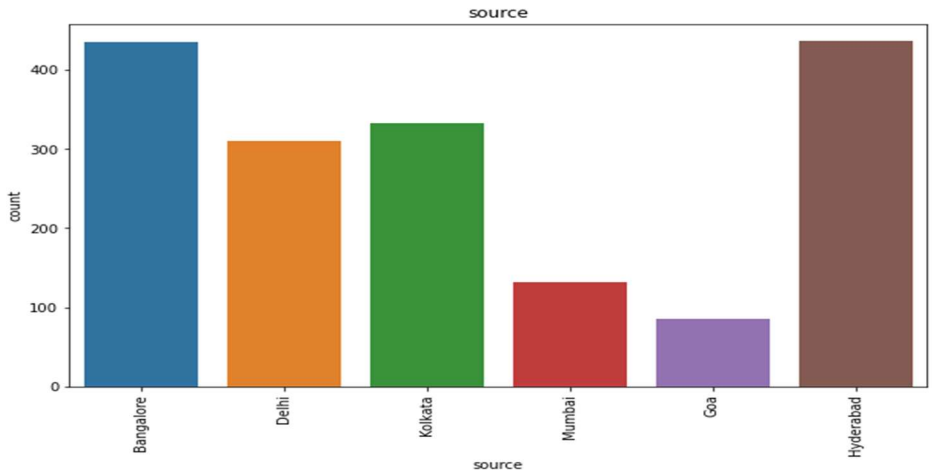
Data Input Logic Output Relations:

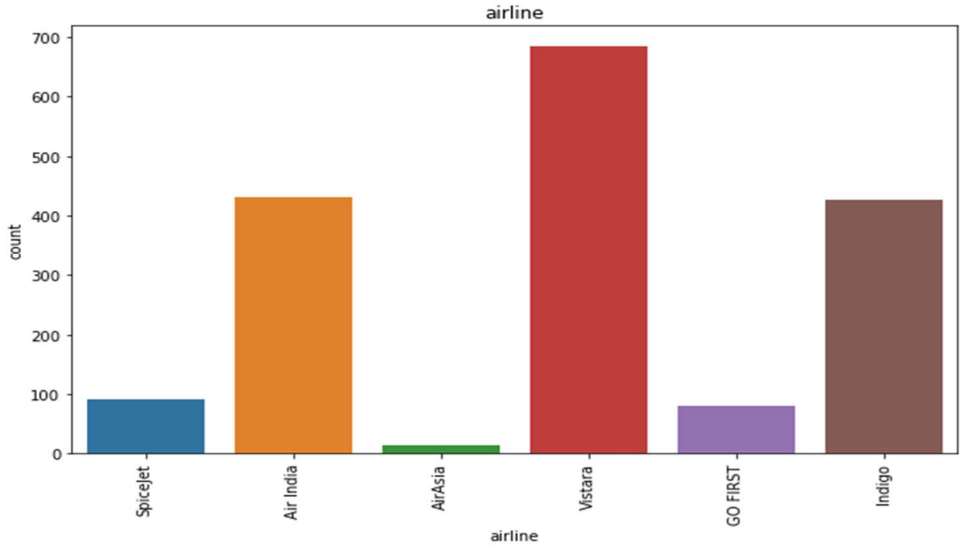
Data visualization is used to find the relation between input and the output variable.

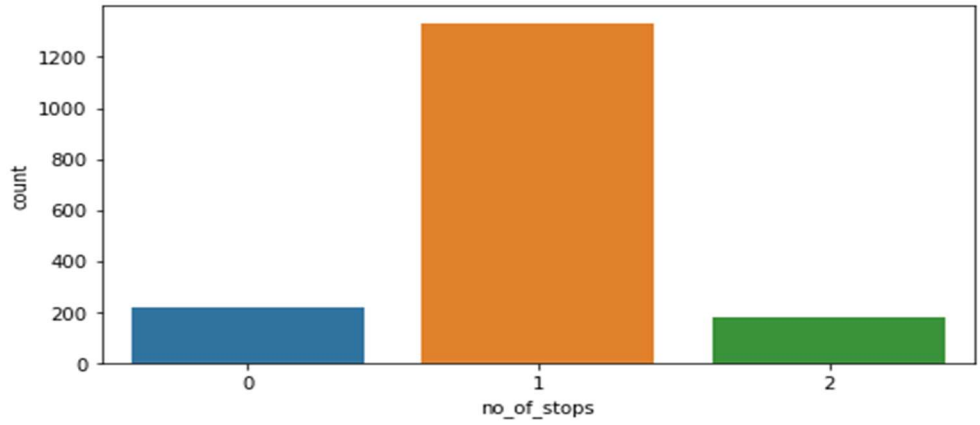
Data Visualization:

Visualization of Categorical Variables:





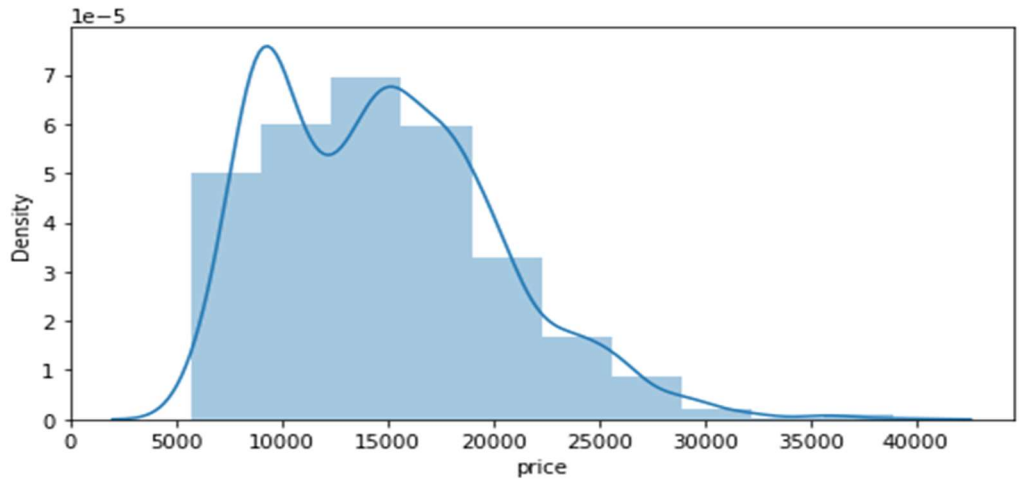




Key Observations:

1. Many people travel from Mumbai and Chennai followed by Goa.
2. Destination is mostly in Bangalore followed by Hyderabad.
3. Vistara is mostly prefered airlines followed by AirIndia and Indigo
4. Our dataset contains many one stop airplanes compared to non-stop.

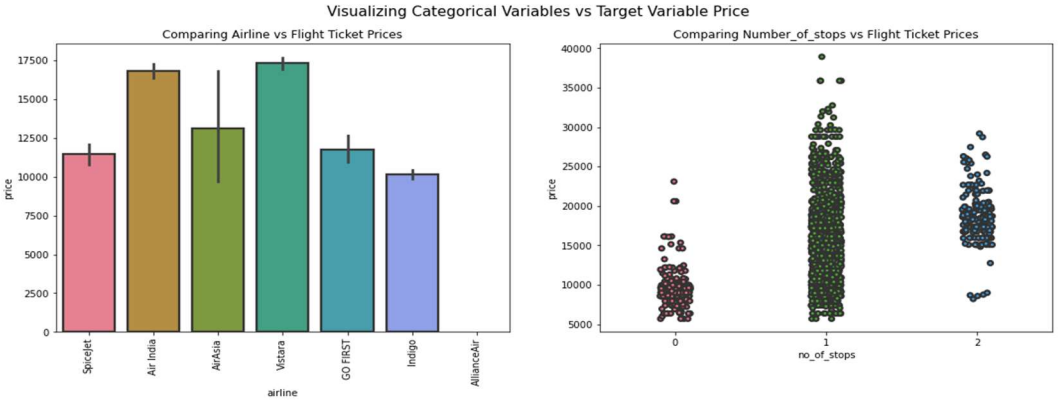
Visualization of Numerical columns:



Key Observations:

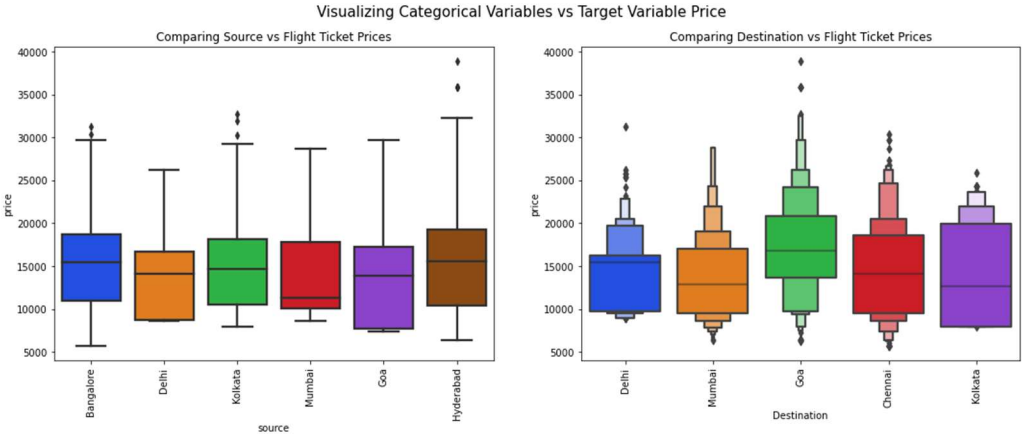
1. Mostly the price ranges from 5000 to 20000 and it extends upto 40000

Bivarient Analysis:



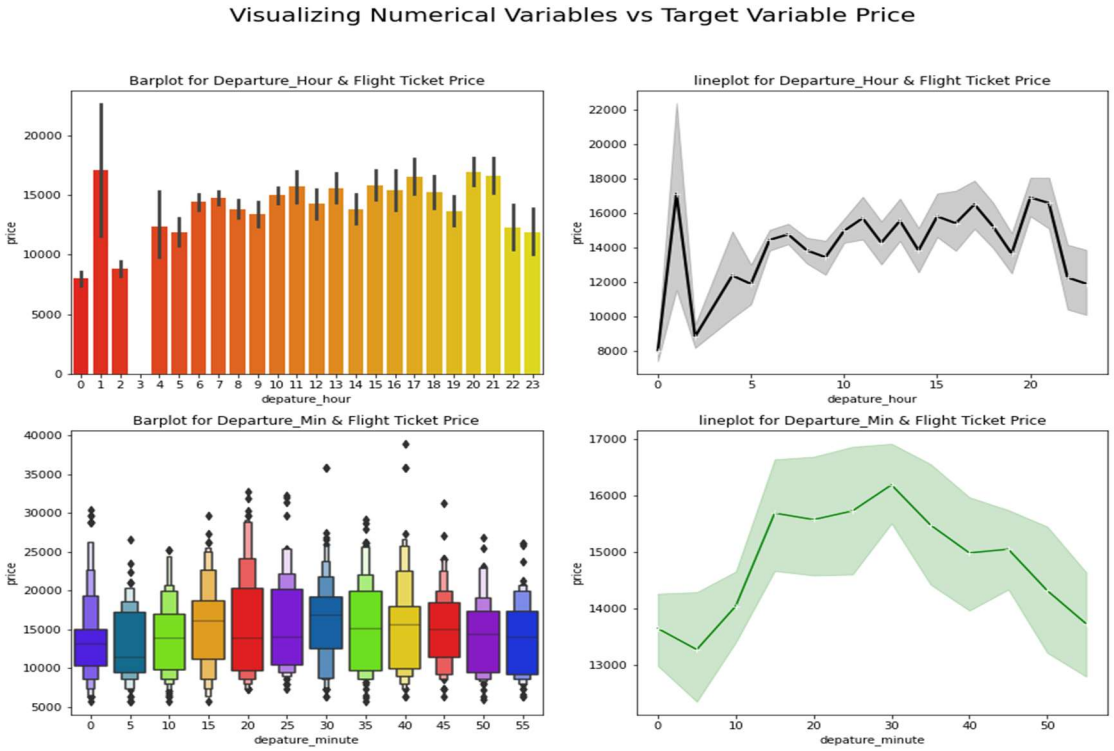
Key Observations:

2. Airline vs Price: From the bar plot we can notice "Vistara" and "Air India" airlines have highest ticket prices compared to other airlines.
3. Number_of_stops vs Price: From the strip plot we can notice the flights which have 1 stop between source and destination have highest ticket prices compared to others.



Key Observations:

1. Source vs Price : From the box plot we can observe the flights from Hyderabad are having somewhat higher prices compared to other sources.
2. Destination vs Price: From the boxen plot we can notice that the flights travelling to Goa have higher flight ticket prices.



Key Observations:

1. Departure_Hour vs Price: From the bar plot and line plot we can see that there are some flights departing in the early morning 1 AM having most expensive ticket prices compared to late morning flights. We can also observe the flight ticket prices are higher during afternoon (may fluctuate) and it decreases in the evening.
2. Departure_Min vs Price: The boxen plot and line plot gives there is no significant difference between price and departure min.

Tools Used:

1. Python 3.8
2. Numpy
3. Pandas
4. Matplotlib
5. Seaborn
6. Data Science
7. Machine Learning

Model/s Development and Evaluation:

Identification of possible problem solving approach:

In this project, both Statistical and Analytical methods are used, in which Data Pre-processing, Exploratory Data Analysis is used after ensuring that data is cleaned. Here Target column is price, as it is continuous data, Regression algorithm is used. This project consists of 4 regressors – Decision Tree Regressor, Support Vector Regressor, KNeighbors Regressor, Linear Regression, 3 ensemble -Random Forest Regressor, AdaBoost Regressor, Gradient Boosting Regressor, XGB Regressor, 3 metrics – r2_score, mean_squared_error, mean_absolute_error and 3 regularization – Lasso, Ridge, ElasticNet techniques. Out of which Random Forest Regressor gave high accuracy which is also chosen as best model in which there is least difference between r2 score and cv. Then with this Hypertuning is done in which accuracy is increased to 95%. Once all this is done, best model is saved using joblib. Then by using this saved model the output is determined which predicted the flight ticket price.

Test of Identified Approaches:

The approaches followed in this project are

1. Find the best random state of a model.
2. Use all the other models to find accuracy score, mean squared error, mean absolute error and r2 score.
3. Then find Cross Validation of all models to find the best accuracy, which is the least difference between r2 score and cv score.
4. With the best model accuracy, we need to do hyper tuning using GridSearchCV.
5. Then finally saving the best model and by keeping this we need to predict the price of flight.

Run and evaluate of selected model:

```
score = []
mean_squared_err = []
mean_absolute_err = []
r2 = []

for m in model:
    m.fit(x_train,y_train)
    m.score(x_test,y_test)
    predm = m.predict(x_test)
    print("Accuracy Score of ",m," is ",m.score(x_train,y_train))
    score.append(m.score(x_train,y_train))

    print("Mean Squared Error is ",mean_squared_error(y_test,predm))
    mean_squared_err.append(mean_squared_error(y_test,predm))
    print("Mean Absolute Error is ",mean_absolute_error(y_test,predm))
    mean_absolute_err.append(mean_absolute_error(y_test,predm))
    print("R2 Score is ",r2_score(y_test,predm))
    r2.append(r2_score(y_test,predm))
    print("\n\n")
```

```
Accuracy Score of  LinearRegression()  is  0.2478018282595661
Mean Squared Error is  18098610.298981544
Mean Absolute error is  3363.9737999209638
R2 Score is  0.3406470604982038
```

Linear Regression gives 2.4% accuracy and r2_score 3.40

```
Accuracy Score of  Lasso()  is  0.2478015262606016
Mean Squared Error is  18097946.634757105
Mean Absolute error is  3363.960088377232
R2 Score is  0.3406712385400532
```

Lasso gives 2.4% accuracy and r2_score 3.40

```
Accuracy Score of  Ridge()  is  0.24780169103046246
Mean Squared Error is  18098513.56820031
Mean Absolute error is  3364.046641415179
R2 Score is  0.340650584510484
```

Ridge give 2.4% accuracy score and r2_score 3.40

```
Accuracy Score of  ElasticNet()  is  0.22494808376240016
Mean Squared Error is  18956931.86093775
Mean Absolute error is  3513.0211959580474
R2 Score is  0.30937743064460976
```

ElasticNet gives 2.2% accuracy score and r2_score 3.09

```
Accuracy Score of  KNeighborsRegressor()  is  0.663947310996818
3
Mean Squared Error is  12809053.962851638
Mean Absolute error is  2629.48978805395
R2 Score is  0.5333516085973397
```

KNeighborsRegressor gives 6.63% accuracy and r2 score 5.33

```
Accuracy Score of  DecisionTreeRegressor()  is  0.9935322394461
178
Mean Squared Error is  17429729.118550632
Mean Absolute error is  2724.6843288375076
R2 Score is  0.3650151619827343
```

DecisionTreeRegressor gives 99.3% accuracy and r2_score 3.65

```
Accuracy Score of  SVR()  is  -0.0023407849688064086
Mean Squared Error is  27321254.745828442
Mean Absolute error is  4321.621828450882
R2 Score is  0.00465564317095124
```

SVR gives 2.03% accuracy and 0.046 r2_score

```
Accuracy Score of  RandomForestRegressor()  is  0.9528707009221
111
Mean Squared Error is  9092817.944409715
Mean Absolute error is  2192.528767677463
R2 Score is  0.6687383096845507
```

RandomForestRegressor gives 95% accuracy and 66.8 r2_score

```
Accuracy Score of  AdaBoostRegressor()  is  0.48265752110617066
Mean Squared Error is  17833748.814202636
Mean Absolute error is  3576.464622921608
R2 Score is  0.35029626536337566
```

AdaBoostRegressor gives 48% accuracy score and 35.02 r2_score

```
Accuracy Score of  GradientBoostingRegressor()  is  0.7323200431878931
Mean Squared Error is  10810315.192867458
Mean Absolute error is  2516.70355181926
R2 Score is  0.606167933249594
```

Gradient Boosting Regressor gives 73% accuracy and 60.61 r2_score

```
Accuracy Score of  XGBRegressor(base_score=0.5, booster='gbtree', callbacks=None,
             colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
             early_stopping_rounds=None, enable_categorical=False,
             eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
             importance_type=None, interaction_constraints='',
             learning_rate=0.300000012, max_bin=256, max_cat_to_onehot=4,
             max_delta_step=0, max_depth=6, max_leaves=0, min_child_weight=1,
             missing=nan, monotone_constraints='()', n_estimators=100, n_jobs=0,
             num_parallel_tree=1, predictor='auto', random_state=0, reg_alpha=0,
             reg_lambda=1, ...)  is  0.9892864963468904
Mean Squared Error is  8863217.417970328
Mean Absolute error is  2128.3484211298774
R2 Score is  0.6771029177687133
```

XGB Regressor gives 98.9% accuracy score and 67.7 r2_score

Cross Validation:

```
Score of  LinearRegression()  is  [0.03955413 0.23023852 0.26881082 0.19296977 0.23354596]
Mean Score is  0.19302384019645955
Standard Deviation is  0.08040347660107508
```

CV Score of LinearRegressor is 0.08040

```
Score of  Lasso()  is  [0.03995973 0.23012071 0.26899633 0.19315835 0.23358129]
Mean Score is  0.19316328226636814
Standard Deviation is  0.08027633194549011
```

CV Score of Lasso is 0.0802

```
Score of  Ridge()  is  [0.03972816 0.23020053 0.26890221 0.19304976 0.23353812]
Mean Score is  0.19308375617520834
Standard Deviation is  0.08034996753989346
```

CV Score of Ridge is 0.08034

```
Score of  ElasticNet()  is  [0.07988779 0.20075965 0.26790247 0.19973931 0.20853496]
Mean Score is  0.19136483527010054
Standard Deviation is  0.06121821406302606
```

CV Score of ElasticNet is 0.06121

```
Score of  KNeighborsRegressor()  is  [-0.21882474  0.11193382  0.00377547  0.19918414  0.36683787]
Mean Score is  0.09258131101406561
Standard Deviation is  0.19586502747400014
```

CV Score of KNeighborsRegressor is 0.09258

```
Score of  DecisionTreeRegressor()  is  [-0.73972331 -0.07530968 -0.38932338  0.16550756  0.32647432]
Mean Score is  -0.14247489721456857
Standard Deviation is  0.38390560457467326
```

CV Score of DecisionTreeRegressor is 0.1424

```
Score of  SVR()  is  [-0.10007352 -0.00478313 -0.00697583 -0.01794598 -0.03407567]
Mean Score is  -0.03277082628915777
Standard Deviation is  0.0352131334248613
```

CV Score of SVR is 0.03277

```
Score of  RandomForestRegressor()  is  [0.17446547 0.27118609 0.2587541  0.43310471 0.53261232]
Mean Score is  0.33402453681037747
Standard Deviation is  0.12986715907725485
```

CV Score of RandomForestRegressor is 0.334

```
Score of  AdaBoostRegressor()  is  [-0.06201881  0.25181128  0.43104413  0.28996233  0.37805066]
Mean Score is  0.2577699178848142
Standard Deviation is  0.1719465612938745
```

CV Score of AdaBoostRegressor is 0.25770

```
Score of  GradientBoostingRegressor()  is  [0.22536917 0.29477609 0.23470192 0.41401069 0.49512211]
Mean Score is  0.3327959968797824
Standard Deviation is  0.10540785247147368


Score of  XGBRegressor(base_score=0.5, booster='gbtree', callbacks=None,
             colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
             early_stopping_rounds=None, enable_categorical=False,
             eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
             importance_type=None, interaction_constraints='',
             learning_rate=0.300000012, max_bin=256, max_cat_to_onehot=4,
             max_delta_step=0, max_depth=6, max_leaves=0, min_child_weight=1,
             missing=nan, monotone_constraints='()', n_estimators=100, n_jobs=0,
             num_parallel_tree=1, predictor='auto', random_state=0, reg_alpha=0,
             reg_lambda=1, ...)  is  [0.07278462 0.31683332 0.00983676 0.40306714 0.56678952]
Mean Score is  0.2738622729803226
Standard Deviation is  0.20711758621631196
```

CV Score of GradientBoostingRegressor is 0.332
CV Score of XGBRegressor is 0.273

HyperTuning:

```
from sklearn.model_selection import GridSearchCV
```

```
param_grid = [{"n_estimators":[10,20,30,55,100],
               "max_features":["auto","sqrt","log2"]}]
rf = RandomForestRegressor(random_state=42)
```

```
reg_grid = GridSearchCV(rf,param_grid,n_jobs=-1,cv=5)
```

```
reg_grid.fit(x_train,y_train)
```

```
]: GridSearchCV(cv=5, estimator=RandomForestRegressor(random_state=42), n_jobs=-1,
             param_grid=[{'max_features': ['auto', 'sqrt', 'log2'],
                          'n_estimators': [10, 20, 30, 55, 100]}])
```

```
reg_grid.best_score_
```

```
]: 0.6714588702590605
```

```
reg_grid.best_estimator_
```

```
]: RandomForestRegressor(random_state=42)
```

```
reg_final_model = reg_grid.best_estimator_
pred = cross_val_predict(reg_final_model,x_train,y_train,cv=5,n_jobs=-1)
```

```
reg_final_model.fit(x_train,y_train)
```

```
]: RandomForestRegressor(random_state=42)
```

```
reg_final_model.score(x_train,y_train)
```

```
]: 0.9527738653074965
```

After Hypertuning we have 95% accuracy score which is improved.

Saving the best model:

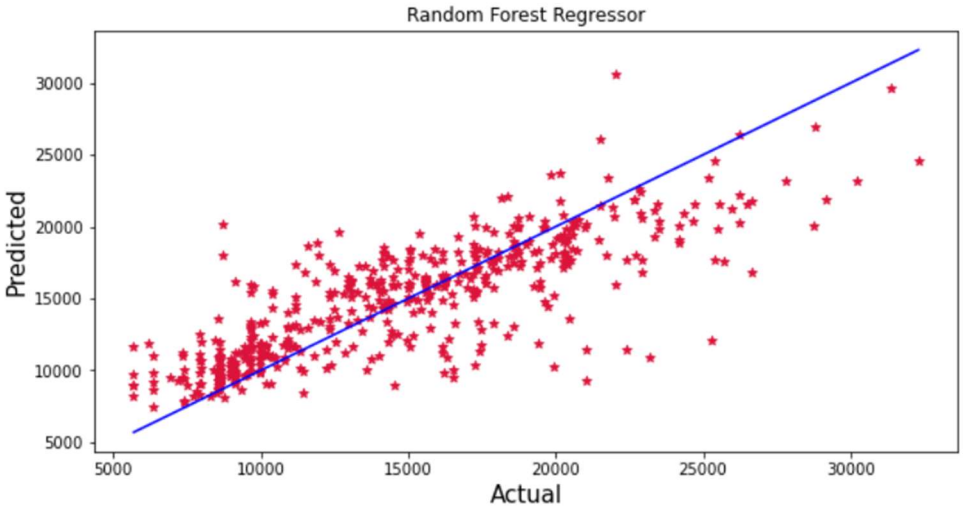
The best model is saved using pickle.

```
import joblib
joblib.dump(reg_final_model,'Flight_Price_Prediction.obj')
```

```
['Flight_Price_Prediction.obj']
```

Graphical representation of actual and predicted values

```
# Visualizing actual and predicted values
plt.figure(figsize=(10,5))
plt.scatter(y_test, prediction, c='crimson',marker="*")
p1 = max(max(prediction), max(y_test))
p2 = min(min(prediction), min(y_test))
plt.plot([p1, p2], [p1, p2], 'b-')
plt.xlabel('Actual', fontsize=15)
plt.ylabel('Predicted', fontsize=15)
plt.title("Random Forest Regressor")
plt.show()
```



The graph shows how our final model is mapping. The plot gives the linear relation between predicted and actual price of the flight tickets. The blue line is the best fitting line which gives the actual values/data and red dots gives the predicted values/data.

Key Metrics for success in solving problem under consideration:

Some of key metrics for the evaluation of this project are

1. R2_score
2. Mean_squared_error
3. Mean_absolute_error
4. Cross_validation
5. Hypertuning

R2 Score:

R-squared is a statistical measure that represents the goodness of fit of a regression model. The ideal value for r-square is 1. Closer the r2_square value to 1 better the model fits.

Mean Squared Error:

The average of the squares of the errors — that is, the average squared difference between the estimated values and what is estimated is measured by the MSE of an estimator (of a process for estimating an unobserved variable). MSE is a risk function that represents the squared error loss's anticipated value. The Root Mean Squared Error is abbreviated as RMSE.

Mean Absolute Error:

MAE is a statistic that assesses the average magnitude of mistakes in a set of forecasts without taking into account their direction. It's the average of the absolute differences between forecast and actual observation over the test sample, where all individual deviations are given equal weight.

Cross Validation:

Cross-validation aids in determining the model's overfitting and underfitting. The model is constructed to run on several subsets of the dataset in cross validation, resulting in numerous measurements of the model. If we fold the data five times, it will be separated into five parts, each representing 20% of the whole dataset. During the Cross-validation, the first part (20%) of the 5 parts will be left out as a holdout set for validation, while the rest of the data will be utilised for training. We'll acquire the initial estimate of the dataset's model quality this way.

Further rounds are produced in the same way for the second 20% of the dataset, which is kept as a holdout set while the remaining four portions are utilised for training data during the process. We'll acquire the second estimate of the dataset's model quality this way. During the cross-validation procedure, these stages are repeated to obtain the remaining estimate of model quality.

Hypertuning:

Hyperparameters in Machine learning are those parameters that are explicitly defined by the user to control the learning process. These hyperparameters are used to improve the learning of the model, and their values are set before starting the learning process of the model.

Interpretation of the Results:

In univariate analysis count plots is used to visualize the counts in categorical variables and distribution plot to visualize the numerical variables. In bivariate analysis bar plots, strip plots, line plots, box plots, and boxen plots are used to check the relation between label and the features, pair plot is used to check the pairwise relation between the features. The heat map and bar plot helped to understand the correlation between dependent and independent features. Detected outliers and skewness with the help of box plots and distribution plots respectively.

After all pre-processing and visualization, it is found that flight ticket price is mainly based on number of stops and duration of flight.

Conclusion:

Key Findings:

The Flight ticket price goes up and down and there is no fixed time. Some flights have early morning and day flights with expensive flight ticket price and mid night ticket price with cheap price. The last-minute flights are expensive. Indigo and SpiceJet have almost same cost of ticket price.

Learning outcomes of study in Data Science:

While working on this project many things like the features of flights, the flight ticket selling web platforms are learnt and got idea about how the machine learning models helps to predict the price of flight tickets.

The challenges faced are while scrapping the real time data from easemytrip.com website, it took so much time to gather data. Finally, our aim was achieved by predicting the flight ticket price and built flight price evaluation model that could help the buyers to understand the future flight ticket prices.

Limitations of this work and scope of Future work:

The main drawback of this study is the low number of records that have been used. In the dataset, data is not properly distributed in some of the columns many of the values in the columns are having string values. Due to some reasons the models may not make the right patterns and the performance of the model also reduces. So that issues need to be taken care.

The greatest shortcoming of this work is the shortage of data. Anyone wishing to expand upon it should seek alternative sources of historical data manually over a period of time. Additionally, more varied set of flights should be explored, since it is entirely possible that airlines vary their pricing strategy according to the characteristics of the flight. Finally, it would be interesting to compare our system's accuracy against that of the commercial systems available today (preferably over a period of time).

References:

A few external references have been used to complete this project successfully. Below are the sources

1. https://scikit-learn.org/
2. https://github.com/
3. https://analyticsvidhya.com/