FLIP ROBO

FLIGHT PRICE PREDICTION PROJECT

Submitted by:

SUDHANSU MANDAL

INTRODUCTION

➢ Business Problem Framing

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)

➢ Conceptual Background of the Domain Problem

As we know covid 19 has impacted the market and we have seen changes in customer demands and behaviour. Since covid has impacted the economy badly so have seen significant changes in airline industry. Machine learning model will help customer to identify the airline fare pattern to gain maximum saving while booking the airline ticket.

➢ Review of Literature

This project observed those features, attribute or factor which impact on predict the airline ticket fare. To complete the project, we have followed below process:

1. **Understand the problem:** Before getting the data, we need to understand the problem statement and the objective of the project. We need to understand the problem we are trying to solve with the help of data science.
2. **Collect Data:** In this step, we have collected the data from My trip site, we will see that the collected data is unorganized and we need to profile the data for the analysis purpose.
3. **Data Pre-Processing:** Here we will clean our dataset based on our model requirement. We will check whether the null value is available and if so we will impute them with most effective technique. Also, we need to profile the dataset for our analysis purpose and identifying the datatype of features so that we can do feature engineering accordingly
4. **Data Exploration:** In this section, we will need to deep dive into the dataset and tries to understand the nature of variables, so that they can be treated properly. This step will involve to creates lots of charts, graphs (Univariate, Bivariate and Multivariate analysis) understand the behaviours of features.
5. **Feature Engineering:** In this step we need to impute with various imputation technique so model can predict the airline ticket fare with high performance. We will also look into the best features which contribute in predicting airline ticket price.
6. **Model Training:** Using various algorithm which is suitable for the dataset and we train the model on the given training dataset.
7. **Model Evaluation:** This is very important steps to know that how the model has been trained in provided dataset. We will evaluate the model performance based on various technique.
8. **Finding Key Factor:** In this section we have found answer of some possible question or query through deep analysis of dataset.

➢ Motivation for the Problem Undertaken

Airline fare always fluctuate depends on various factors and demand. For Customer it seems very difficult to predict the price of airlines ticket. For Airlines company fixing the air ticket prices is one of the key factors for company growth and revenue. To become successful in this industry the companies must analyse the various factors which can predict the fare. Even for customer machine learning algorithm can provide deep insight about the airline industry various factors which resulted in increasing or decreasing the fare ticket.
Here Data Science can help the customer to know the fact about the airlines ticket price and also for the company to know better the industry and demands to align their strategies and marketing mix based on the attributes which are highly correlated with sale price.

ANALYTICAL PROBLEM FRAMING

➢ Mathematical/ Analytical Modeling of the Problem

In this project we have scrap the data from MakeMyTrip site for airlines booking details. There are 83273 observation and 9 features columns including our target label.
To understand our data, we have done basic data exploration.

- **Statistic Summary**

Here we have tried to understand the Statistic of data with the help of describe () and Info () function. Mention below are the observation made from this summary:

1. **Data Type:** As we have seen that Dataset having 9 columns all are in object datatype.
2. **Null Values:** We have also checked whether null values present in the dataset.
3. We have observed that the target variable i.e. Price is continuous data type so we must apply Regression technique.
4. We have seen that lot of features we need to profiling of these features to get insight in it.

- **Feature Selection**

We have seen that 9 features in dataset, many of which have analysed that they are not in proper formant and we need to fetch the required information from those columns. We have to do deep dive in required features and find the required information which could be useful to predict the flight ticket price.

- **Data Visualization**

While doing EDA, we have found mentioned below important points for our dataset:

a. Target Variable is not normally distributed.
b. Outliers in some numerical columns which we need tackle.

c. Most Significate Numerical and Categorical features on predicting ticket Price.
d. Categorical features having some outliers in respect of sale price.
e. We need to apply datetime technique to deal with date and time columns.
f. Have mentioned the key finding from the dataset about the airlines industry and attempted to answer some of the important question about the flight timing, stoppage, journey hour, increase or decrease in price over a period of time.

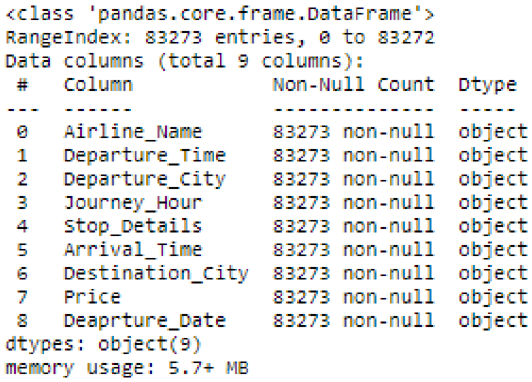
- **Data Normalization**

  Since the data was not normal and having right skewed so we have normalized all the features including our target variable.

➢ Data Sources and their formats

The data sources is in .csv format and. We have imported and converted into Pandas dataframe for analysing purpose. The dataset columns having all float even the price columns as shown in Fig 1.

Fig 1:



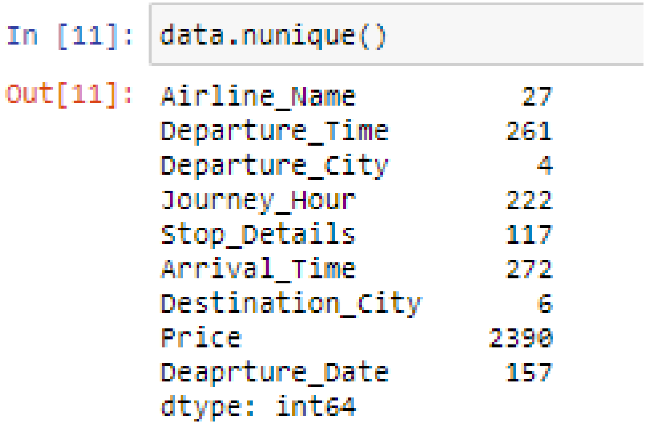
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 83273 entries, 0 to 83272
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Airline_Name      83273 non-null  object
 1   Departure_Time    83273 non-null  object
 2   Departure_City    83273 non-null  object
 3   Journey_Hour      83273 non-null  object
 4   Stop_Details      83273 non-null  object
 5   Arrival_Time      83273 non-null  object
 6   Destination_City  83273 non-null  object
 7   Price             83273 non-null  object
 8   Deaprture_Date    83273 non-null  object
dtypes: object(9)
memory usage: 5.7+ MB
```

➢ Data Pre-processing

As we know Pre-Processing of our data is an important step for our model to perform better and get valuable information from the dataset. Below is the step for data Preprocessing done:

- **Checking unique value of dataset.**
  As shown in Fig 2, we have checked the unique value in dataset, which helps us to know the categorical features level in the particular columns and we will get some idea how to perform feature engineering and which technique can be useful.

Fig: 2



```
In [11]: data.nunique()

Out[11]: Airline_Name          27
         Departure_Time       261
         Departure_City         4
         Journey_Hour         222
         Stop_Details         117
         Arrival_Time         272
         Destination_City       6
         Price               2390
         Deaprture_Date       157
         dtype: int64
```

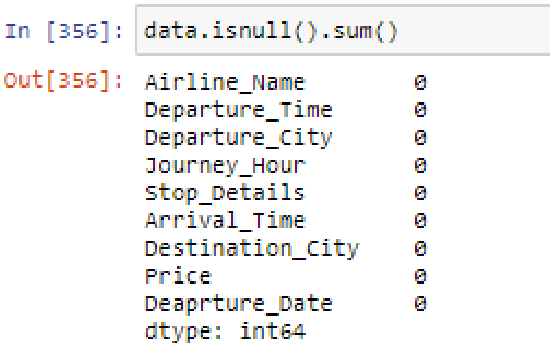
- **Checking for null value**

In real world data, sometimes we would be missing value in the dataset which occur due to various reason like corrupt data, failure to load the information or incomplete extraction, and sometime due to human error. Handling these missing values is one of the most important challenges faced by analysts because it plays very important role in making the right decision on how to handle it, which helps in to make robust data models.

As Shown in Fig 3, we don’t have any Null values in the dataset .

Fig 3:

Null Value Dataset:

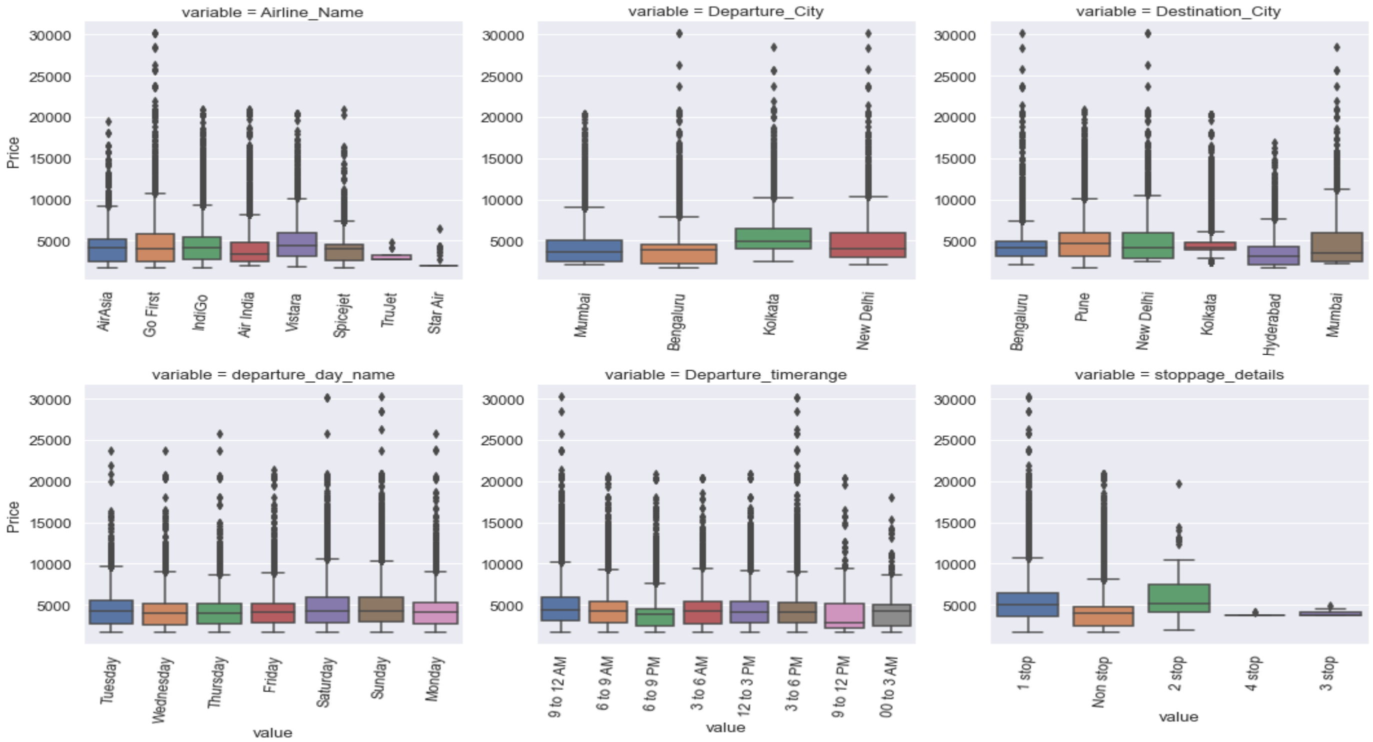


```
In [356]: data.isnull().sum()

Out[356]: Airline_Name        0
          Departure_Time      0
          Departure_City      0
          Journey_Hour        0
          Stop_Details        0
          Arrival_Time        0
          Destination_City    0
          Price               0
          Deaprture_Date      0
          dtype: int64
```

- **Checking for outlier’s presence in Dataset.**

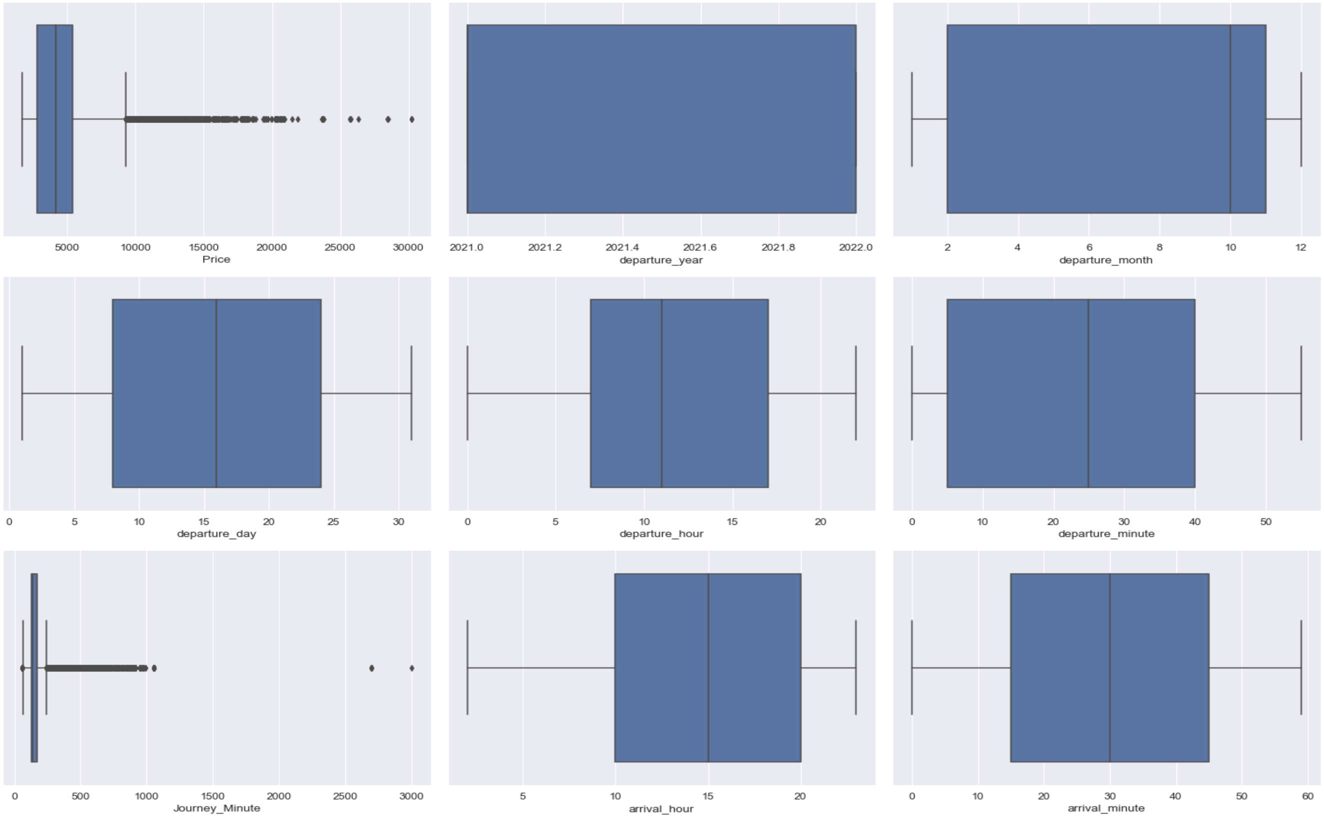
In EDA we have visualized that as shown in Fig 4, that there are lot of Categorical Columns where we have outliers with respect of Sale Price. Also, some numerical columns where we have outliers.

Fig: 4



Also, as shown in Fig 5, Outliers present in Numerical features too

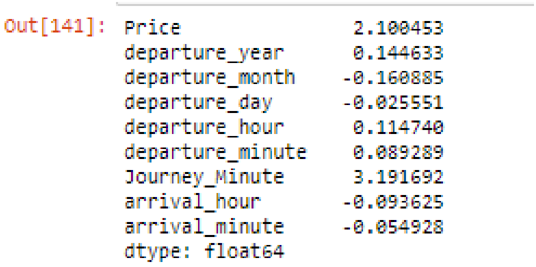
Fig : 5



- **Dealing with Skewed Data**

As shown in Fig 6, we have calculated the skewness of numerical features, also have seen that (Fig 7 and Fig 8) target variable and Journey Minute columns having right skewed data. To Avoid any loss of information we have removed data skewedness with log transformation technique.

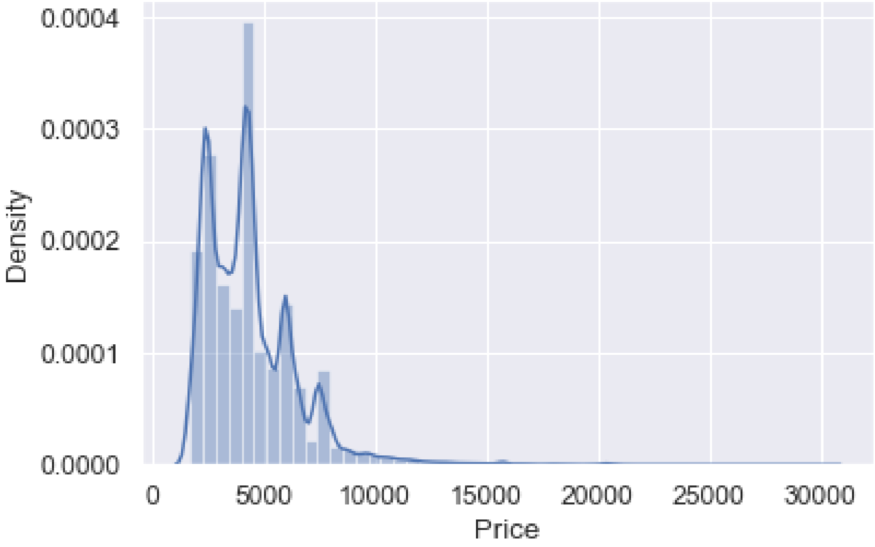
Fig: 6



```
Out[141]: Price                2.100453
          departure_year       0.144633
          departure_month     -0.160885
          departure_day       -0.025551
          departure_hour       0.114740
          departure_minute     0.089289
          Journey_Minute       3.191692
          arrival_hour        -0.093625
          arrival_minute      -0.054928
          dtype: float64
```

Fig: 7

Before treating skewedness.



After Treating Skewedness:

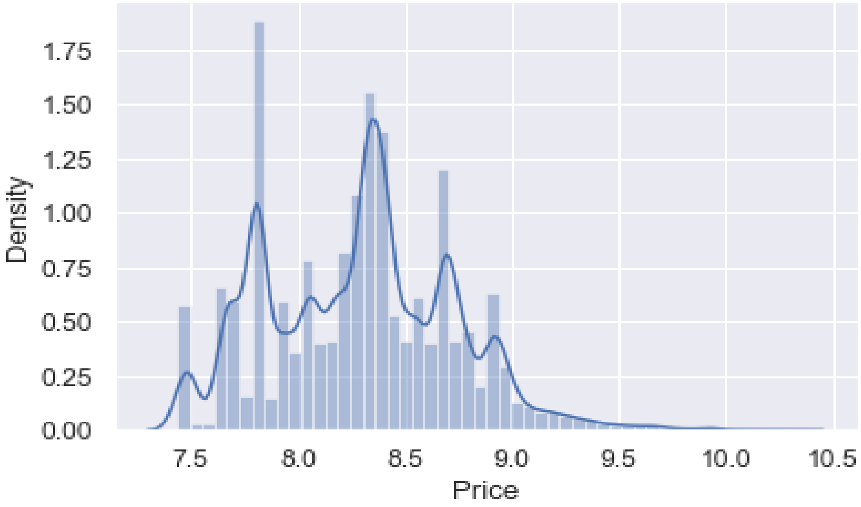
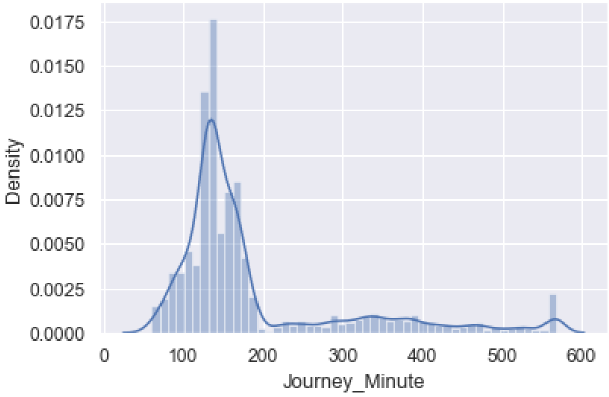
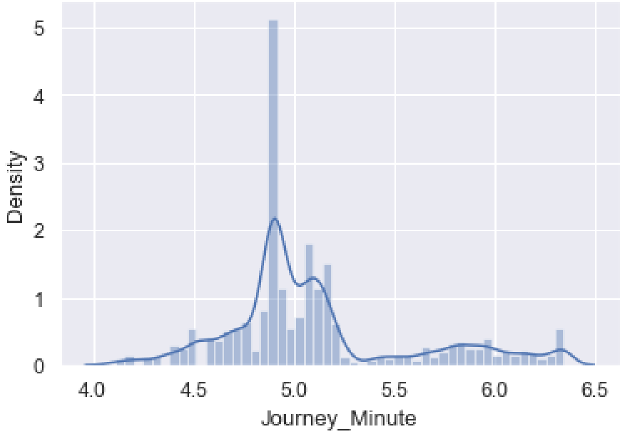


Fig 8:

Before treating skewedness.

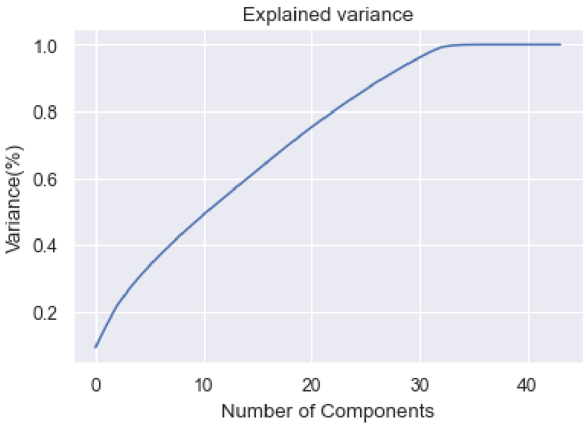


After treating skewedness



➢ Data Inputs- Logic- Output Relationships

In this dataset, our input variable having all object data type. Our target variable also object data type, so we need have converted into integer data type.
We have seen there are 8 features and 1 target variable in dataset but after doing Feature Engineering & Transformation our features have increased and now having 44 features.
We have also applied Dimension reduction technique through Principal Component Analysis (PCA) where we have found that 30 number of components explained almost more than 90 % variance, as shown in Fig 9. But we will see our model performance without applying PCA First and based on performance we will decide whether to apply PCA data as input for model or not.
Fig: 9



➢ Hardware and Software Requirements and Tools Used

In this project the mention below hardware, software and tools used to complete this project:

✓ Hardware:

Processor	Intel(R) Pentium(R) CPU 3825U @ 1.90GHz 1.90 GHz
Installed RAM	4.00 GB
System type	64-bit operating system, x64-based processor

✓ Software:

Edition	Windows 10 Home Single Language
Anaconda	

✓ Library & Tools:

Jupyter Notebook
Pandas
NumPy
Matplotlib
Seaborn

Model/s Development and Evaluation

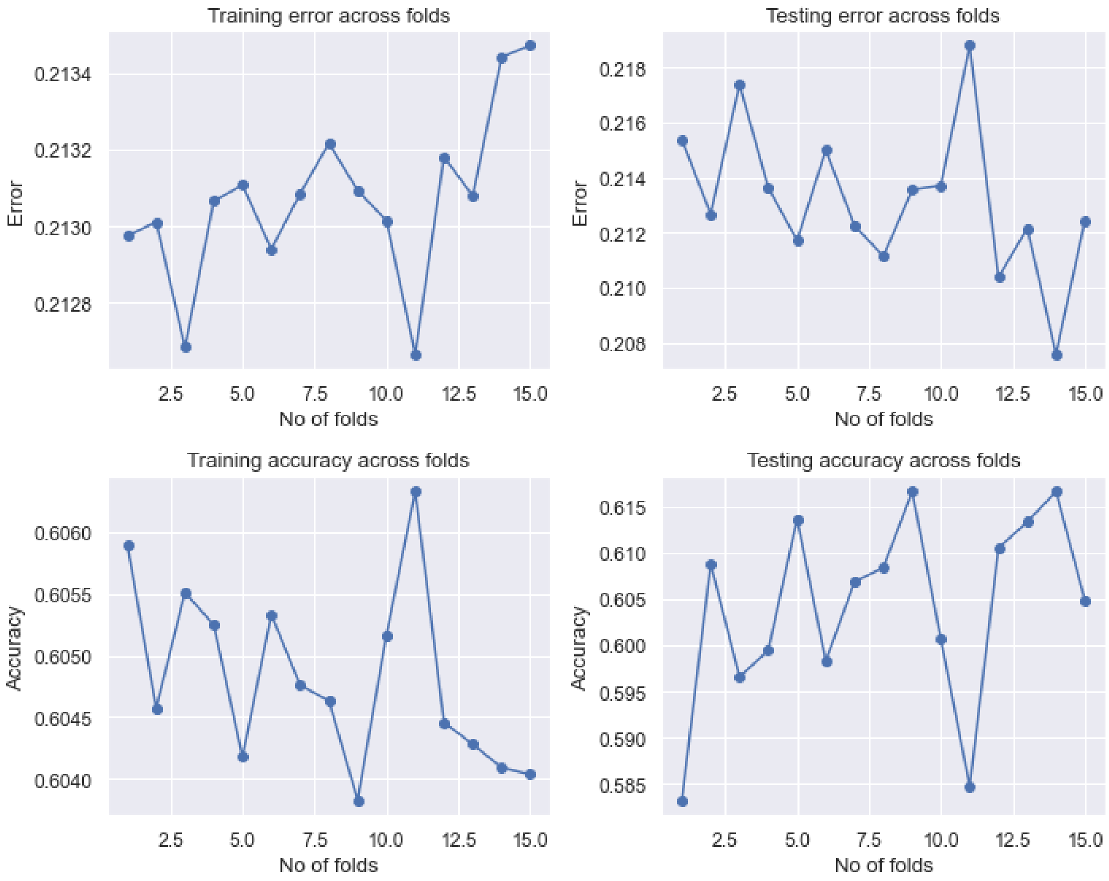
➢ Identification of possible problem-solving approaches (methods)

As we have seen that our target variable is price and Continuous type, so we need to apply Regression task to the dataset.
We have seen that our categorical data having outliers and dealing with them it will take long time for each variable of categorical features, so we have normalized the features columns with **Standard Scaler** techniques.
Also, we have done PCA **(Principal component Analysis)** for dimension reduction of our feature's dataset, as shown in fig 9, we can choose 23 components for our model, but we will see the performance of model without removing any features and then will decide whether need to use dimension reduction technique or not.
We have seen that without dimension reduction technique model is performing better.

As shown in Fig 10, we have tested model Overfitting and Underfitting for all model and evaluate the model performance.

Fig 10:

Linear Regression Model Performance on Training and Testing across different Kfolds.



➢ Testing of Identified Approaches (Algorithms)

As we know our target variable having continuous type of data, so we need apply regression algorithm for model building. Since we have outliers in dataset, so we have applied linear model, tree-based model and ensemble model algorithm to the dataset and have observed their performance with different Cross Validation.
Below are the Regression algorithms have applied to the model. Here as we will see that Linear Regression is not performing better due to non-linearity between dependant and independent features. So We will evaluate our model by using Trees based and ensemble based algorithm.

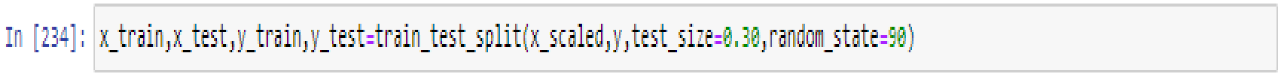
1. Linear Regression
2. Random Forest Regression
3. Decision Tree Regressor
4. Gradient Boosting Regressor

We have applied mentioned above model in our training and test dataset. Here in project we have split training 70% and testing 30% of data.

➢ Run and Evaluate selected models

As shown in Fig 11, we have separated the training and test data set into 75% and 25%.

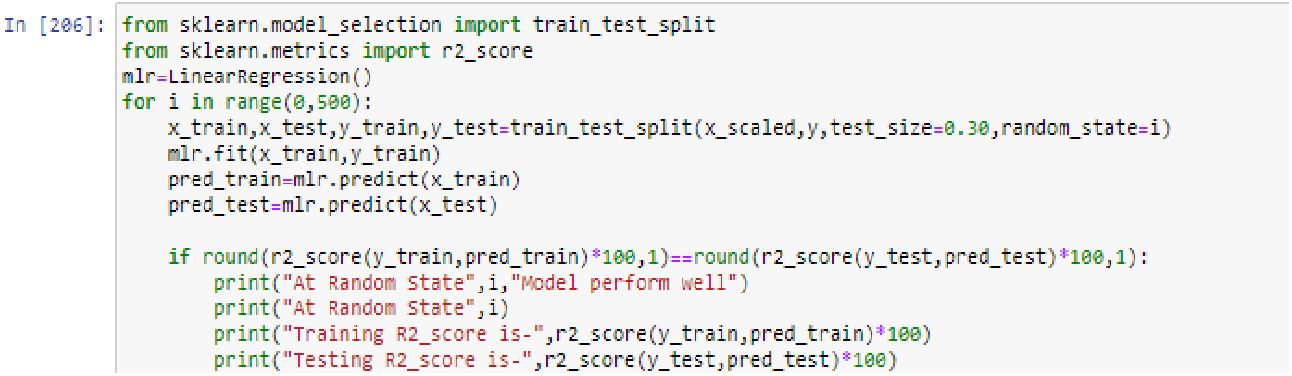
Fig: 11



```
In [234]: x_train,x_test,y_train,y_test=train_test_split(x_scaled,y,test_size=0.30,random_state=90)
```

Here we have chosen Best Random State 90 for the model as shown in Fig 12.

Fig: 12



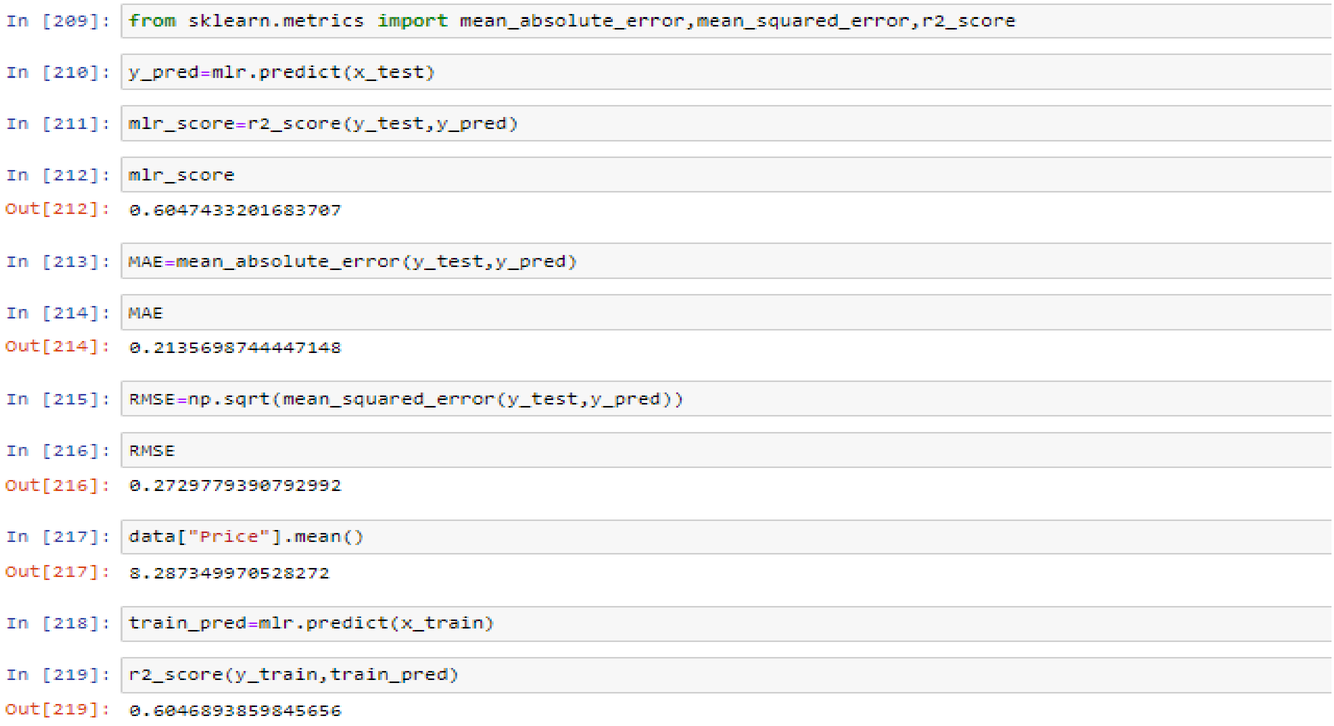
```
In [206]: from sklearn.model_selection import train_test_split
          from sklearn.metrics import r2_score
          mlr=LinearRegression()
          for i in range(0,500):
              x_train,x_test,y_train,y_test=train_test_split(x_scaled,y,test_size=0.30,random_state=i)
              mlr.fit(x_train,y_train)
              pred_train=mlr.predict(x_train)
              pred_test=mlr.predict(x_test)

              if round(r2_score(y_train,pred_train)*100,1)==round(r2_score(y_test,pred_test)*100,1):
                  print("At Random State",i,"Model perform well")
                  print("At Random State",i)
                  print("Training R2_score is-",r2_score(y_train,pred_train)*100)
                  print("Testing R2_score is-",r2_score(y_test,pred_test)*100)
```

Now we have evaluated our model, based on R2_Score, MAE and RMSE.

1. **Linear Regression**

Evaluation Matrix:

Fig: 13



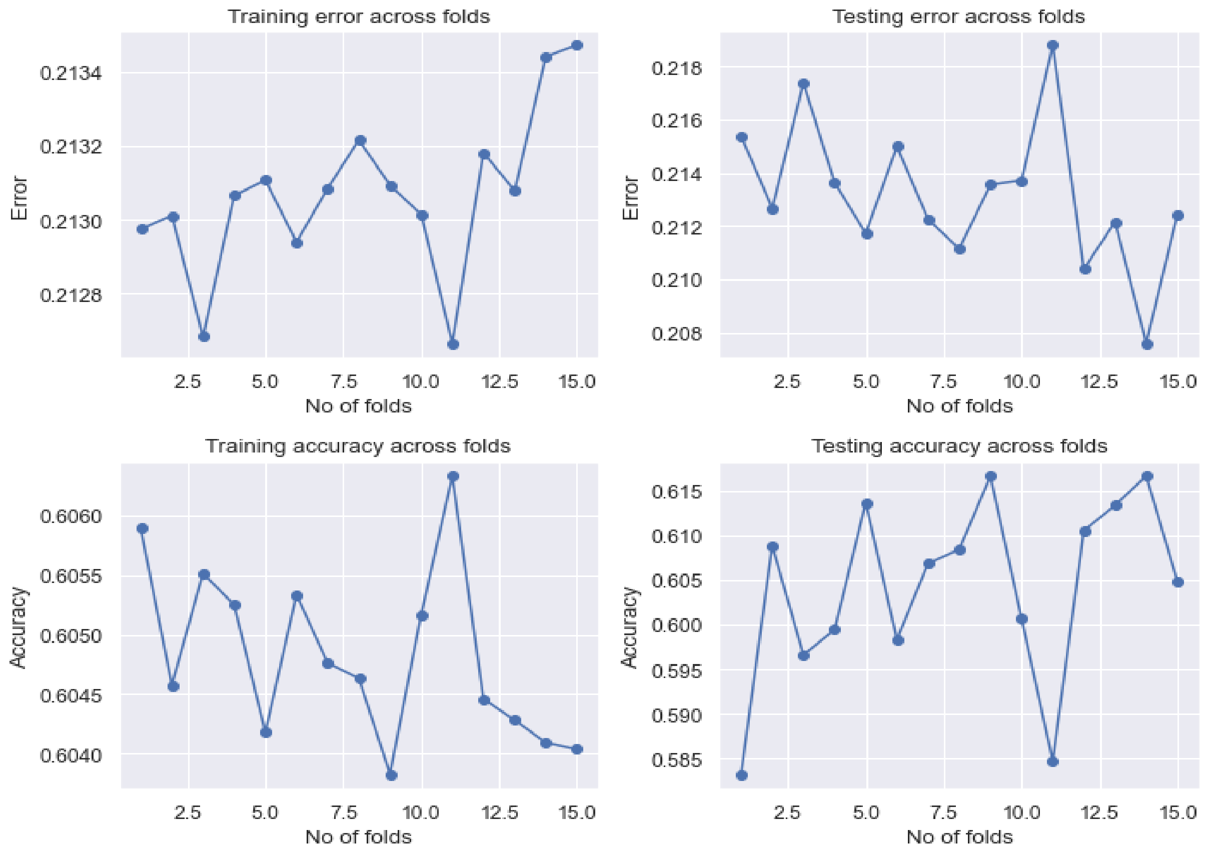
```
In [209]: from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score

In [210]: y_pred=mlr.predict(x_test)

In [211]: mlr_score=r2_score(y_test,y_pred)

In [212]: mlr_score
Out[212]: 0.6047433201683707

In [213]: MAE=mean_absolute_error(y_test,y_pred)

In [214]: MAE
Out[214]: 0.2135698744447148

In [215]: RMSE=np.sqrt(mean_squared_error(y_test,y_pred))

In [216]: RMSE
Out[216]: 0.2729779390792992

In [217]: data["Price"].mean()
Out[217]: 8.287349970528272

In [218]: train_pred=mlr.predict(x_train)

In [219]: r2_score(y_train,train_pred)
Out[219]: 0.6046893859845656
```

Here we can see the Linear Regression R2 Score is 60%, which is not good performing model due to non-linearity of dataset between different variable.

Also, we have calculated Cross Validation Score to understand our model performance and have seen that our Cross-Validation Score is also almost same 60%. Hence, it indicates that model don’t have issue with overfitting or underfitting problems.

To understand this, we have plotted training and testing performance of Linear Regression model in different folds and observed the performance of model and error as shown in Fig 14.

Fig 14:

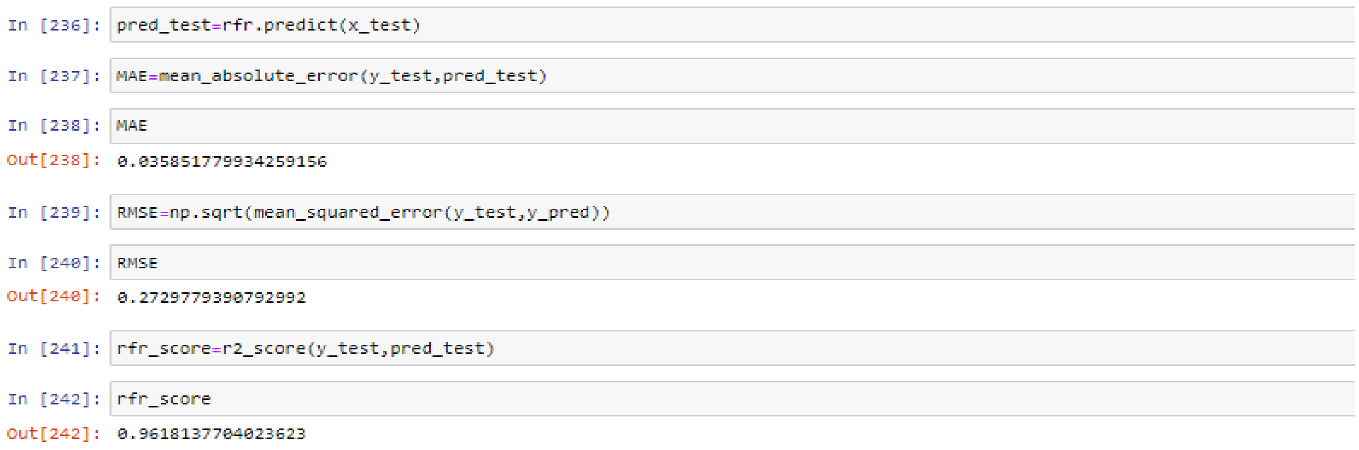


2. Random Forest Regressor

Random Forest Regressor is non-parametric algorithm, So in this dataset this model might perform better than Linear Regression model

Evaluation Matrix:

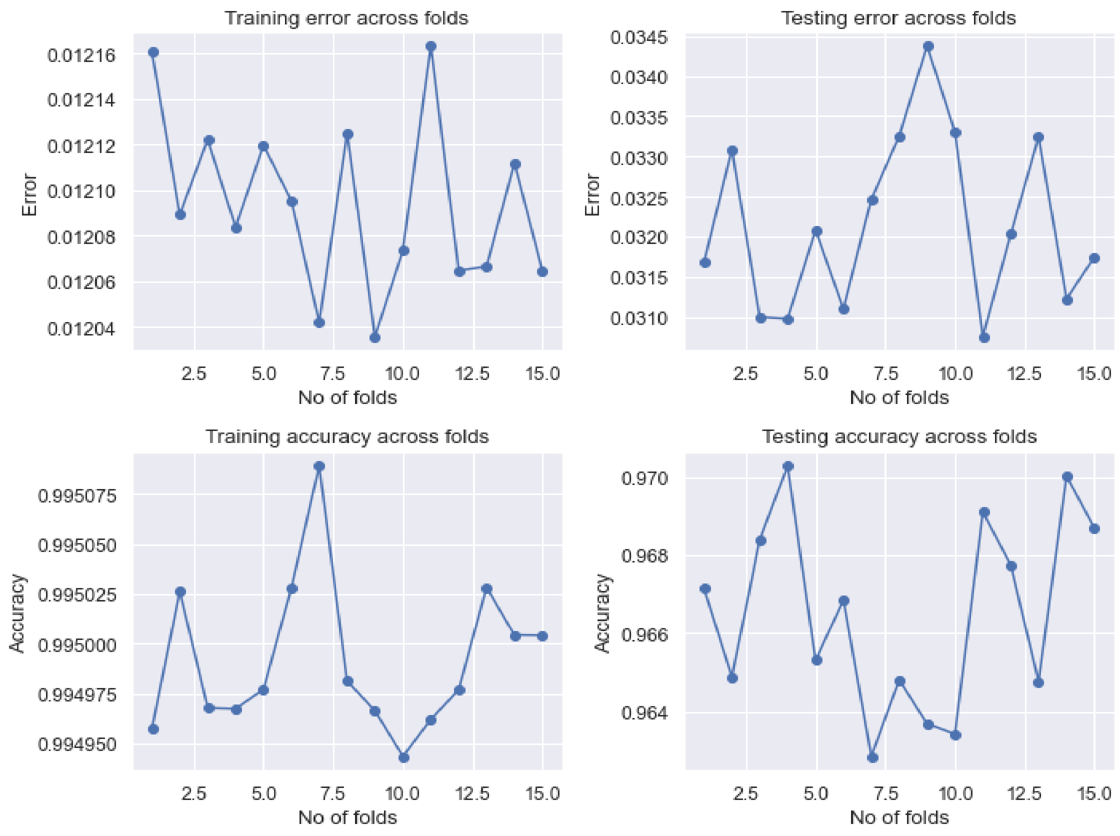
Fig: 15



```
In [236]: pred_test=rfr.predict(x_test)

In [237]: MAE=mean_absolute_error(y_test,pred_test)

In [238]: MAE
Out[238]: 0.035851779934259156

In [239]: RMSE=np.sqrt(mean_squared_error(y_test,y_pred))

In [240]: RMSE
Out[240]: 0.2729779390792992

In [241]: rfr_score=r2_score(y_test,pred_test)

In [242]: rfr_score
Out[242]: 0.9618137704023623
```

As shown in Fig 15, we can see the Random Forest Regressor R2 Score is 96.18%.
MAE is 0.035
RMSE is 0.272
Also, we have calculated Cross Validation Score to understand our model performance and have seen that our Cross-Validation Score also 96%.
To understand better, we have plotted training and testing performance of Random Forest Regressor model in different folds and observed the performance of model and error as shown in Fig 16.

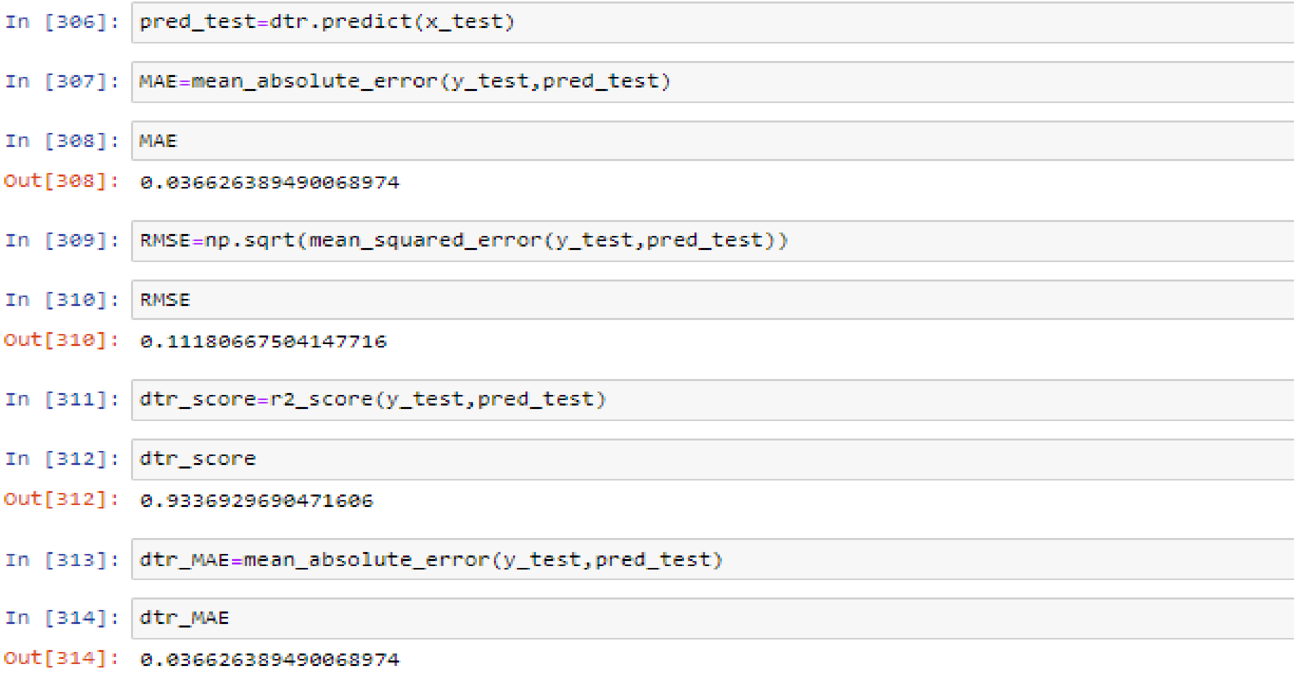
Fig 16:



As seen in Above Fig 16, The model performance is better than Linear Regression and doesn’t have overfitting or underfitting issue.

3. Decision Tree Regressor

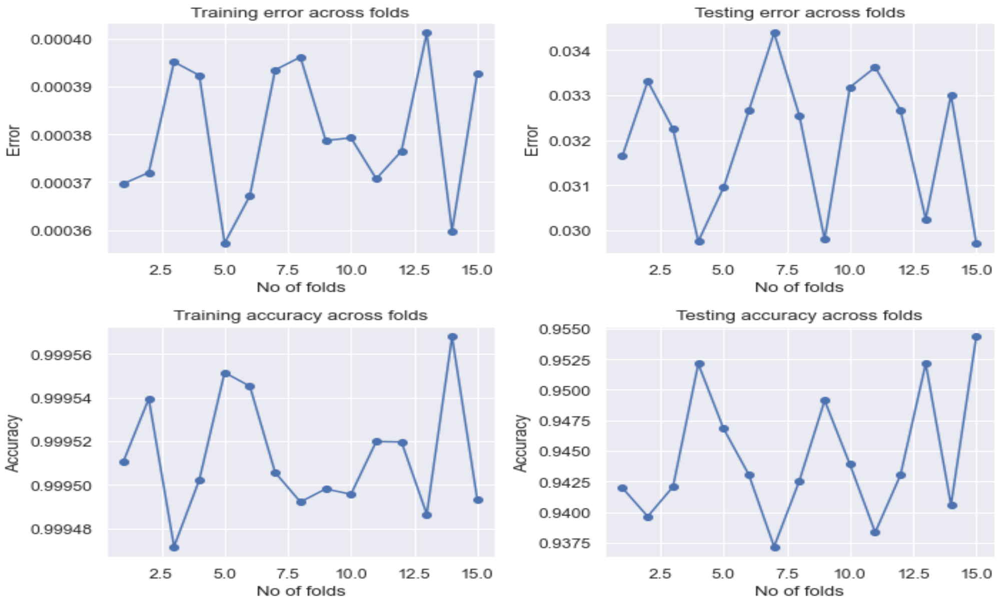
Evaluation Matrix:
Fig: 17



```
In [306]: pred_test=dtr.predict(x_test)

In [307]: MAE=mean_absolute_error(y_test,pred_test)

In [308]: MAE
Out[308]: 0.036626389490068974

In [309]: RMSE=np.sqrt(mean_squared_error(y_test,pred_test))

In [310]: RMSE
Out[310]: 0.11180667504147716

In [311]: dtr_score=r2_score(y_test,pred_test)

In [312]: dtr_score
Out[312]: 0.9336929690471606

In [313]: dtr_MAE=mean_absolute_error(y_test,pred_test)

In [314]: dtr_MAE
Out[314]: 0.036626389490068974
```

As shown in Fig 17, we can see the Decision Tree Regressor R2 Score is 93.37%.
MAE is 0.036
RMSE is 0.111
Also, we have calculated Cross Validation Score to understand our model performance and have seen that our Cross-Validation Score is 93.09%.
To understand better, we have plotted training and testing performance of Decision Tree Regressor model in different folds and observed the performance of model and error as shown in Fig 18.

Fig 18:

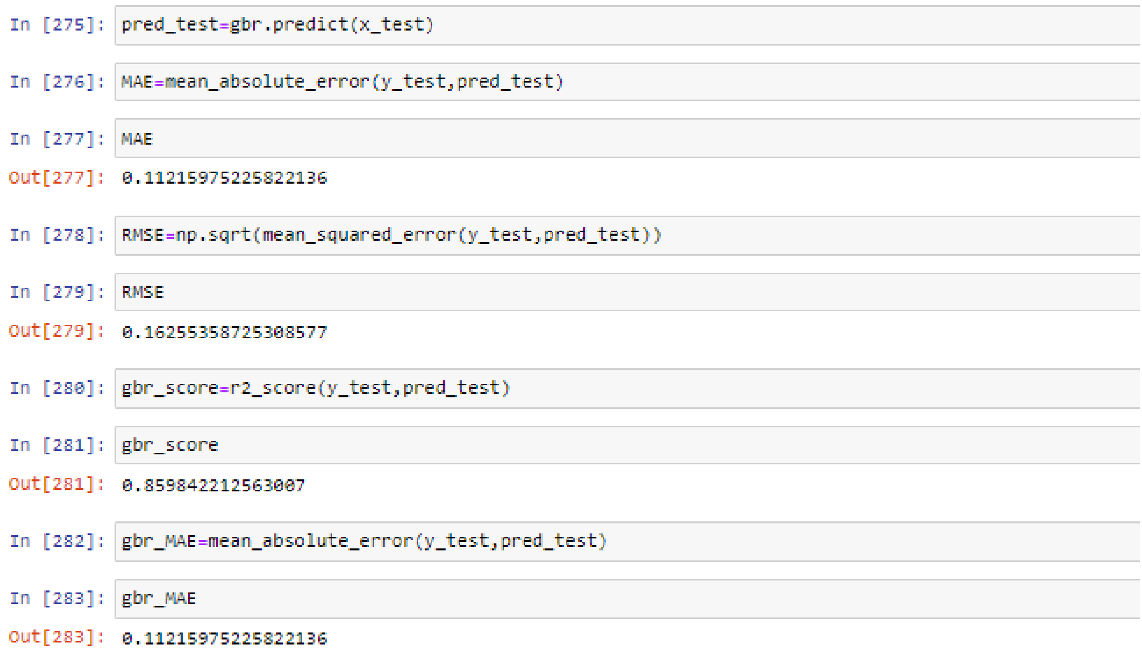


As seen in Above Fig 18, The model performance has almost similar error rate in both training dataset compare to testing. Here the model is not performing good as random forest regressor.

4. Gradient Boosting Regressor:

Evaluation Matrix:

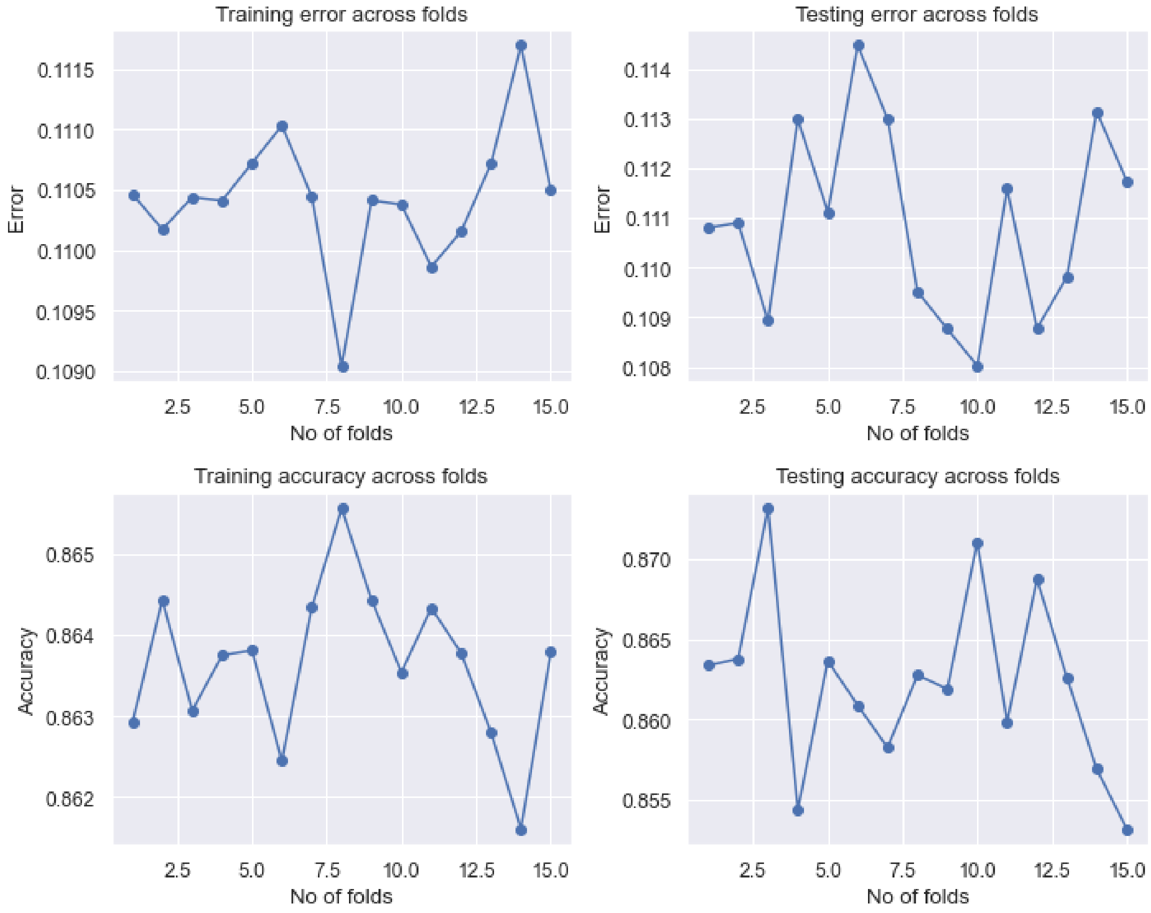
Fig 19



```
In [275]: pred_test=gbr.predict(x_test)

In [276]: MAE=mean_absolute_error(y_test,pred_test)

In [277]: MAE
Out[277]: 0.11215975225822136

In [278]: RMSE=np.sqrt(mean_squared_error(y_test,pred_test))

In [279]: RMSE
Out[279]: 0.16255358725308577

In [280]: gbr_score=r2_score(y_test,pred_test)

In [281]: gbr_score
Out[281]: 0.859842212563007

In [282]: gbr_MAE=mean_absolute_error(y_test,pred_test)

In [283]: gbr_MAE
Out[283]: 0.11215975225822136
```

As shown in Fig 19, we can see the Gradient Boosting Regressor R2 Score is 85.98%.
MAE is 0.11
RMSE is 0.16
Also, we have calculated Cross Validation Score to understand our model performance and have seen that our Cross-Validation Score is 86.29%.
To understand better, we have plotted training and testing performance of Gradient Boosting Regressor model in different folds and observed the performance of model and error as shown in Fig 20.

Fig 20:



As seen in Above Fig 20, The model performance has almost similar error rate in both training dataset compare to testing and R2 score also almost same in both training and testing.
But this model is not performing as good as Random forest regressor and Decision tree regressor in terms of error rate and R2 score.

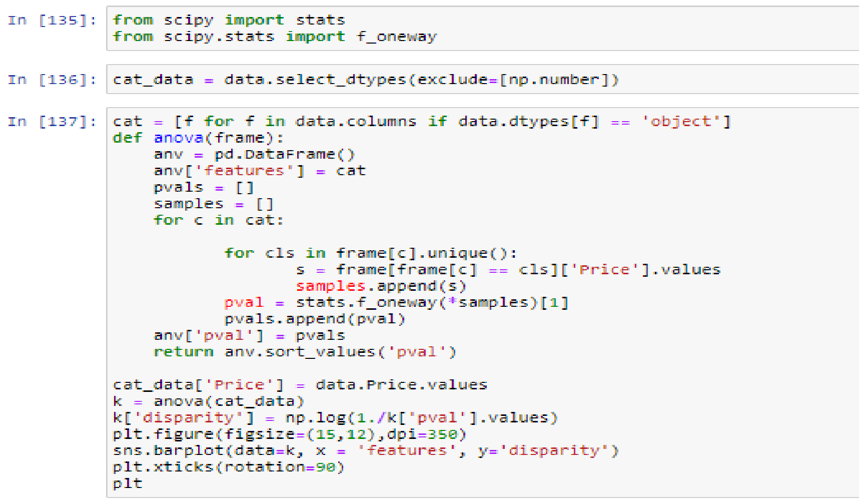
➢ Visualizations

In this Section we have visualized the features with various graph and plot to understand or analyse our features for our model.

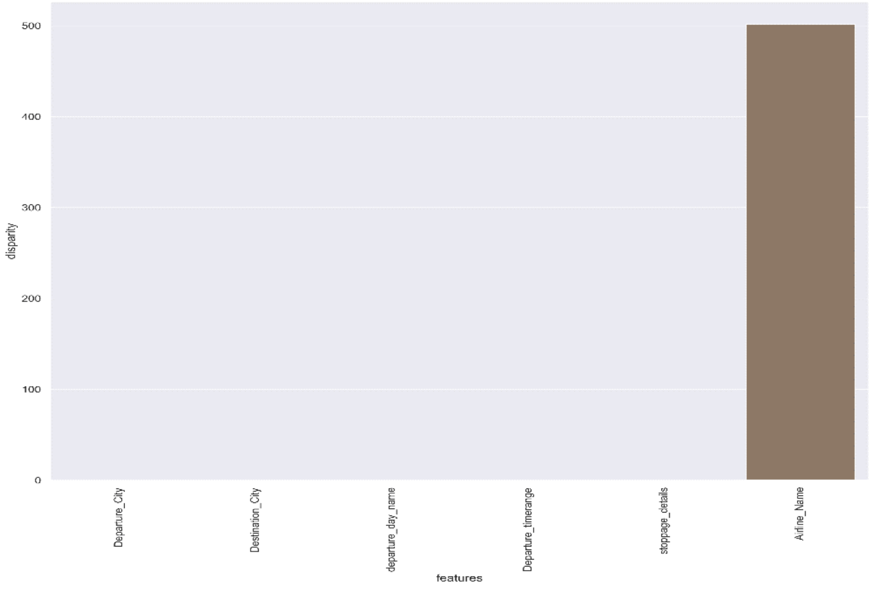
- **Categorical Feature:**

As shown in Fig 21, we have tried to find most important categorical variable with ANOVA technique.

Fig 21:

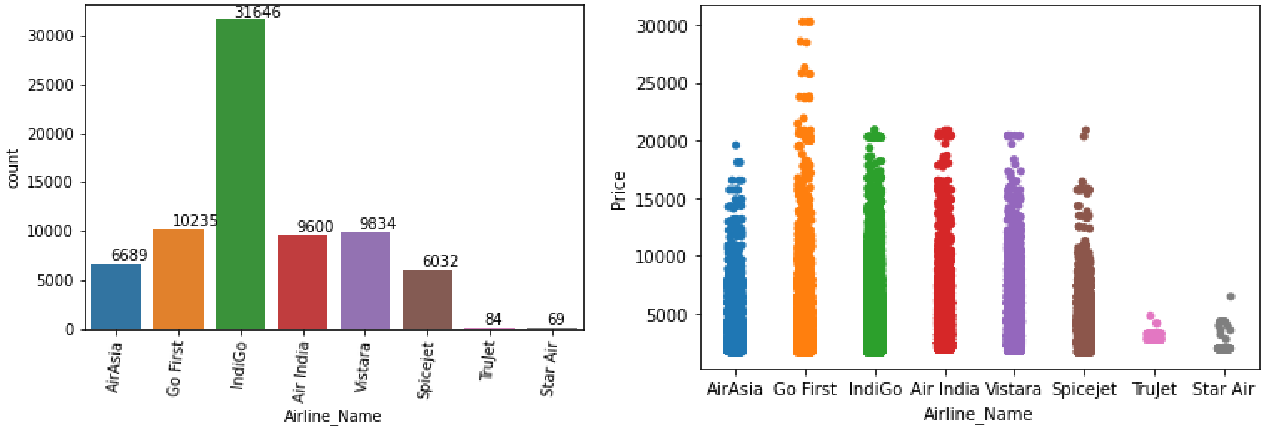


```
In [135]: from scipy import stats
          from scipy.stats import f_oneway

In [136]: cat_data = data.select_dtypes(exclude=[np.number])

In [137]: cat = [f for f in data.columns if data.dtypes[f] == 'object']
          def anova(frame):
              anv = pd.DataFrame()
              anv['features'] = cat
              pvals = []
              samples = []
              for c in cat:

                     for cls in frame[c].unique():
                            s = frame[frame[c] == cls]['Price'].values
                            samples.append(s)
                     pval = stats.f_oneway(*samples)[1]
                     pvals.append(pval)
              anv['pval'] = pvals
              return anv.sort_values('pval')

          cat_data['Price'] = data.Price.values
          k = anova(cat_data)
          k['disparity'] = np.log(1./k['pval'].values)
          plt.figure(figsize=(15,12),dpi=350)
          sns.barplot(data=k, x = 'features', y='disparity')
          plt.xticks(rotation=90)
          plt
```



- **Airline Name**

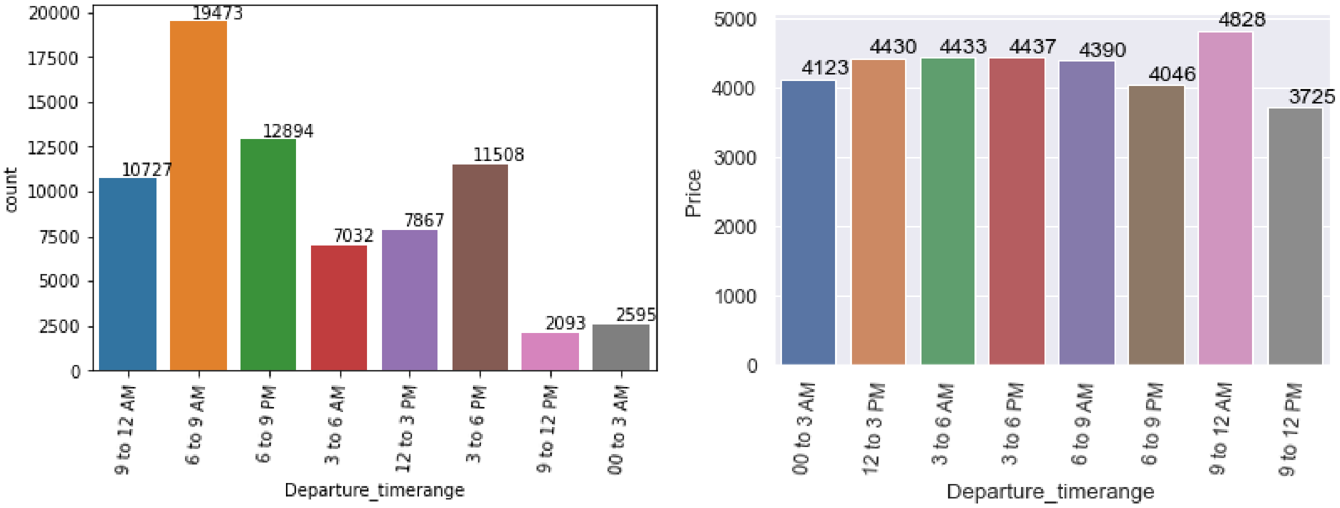
Fig: 22



As we can infer from fig 22 that most of the Flight which running belong to IndiGo followed by Go First, Vastara and Air India. The most expensive flight fare is Go First compared to Indigo, Air India, Vistara.

- **Departure Timerange:**

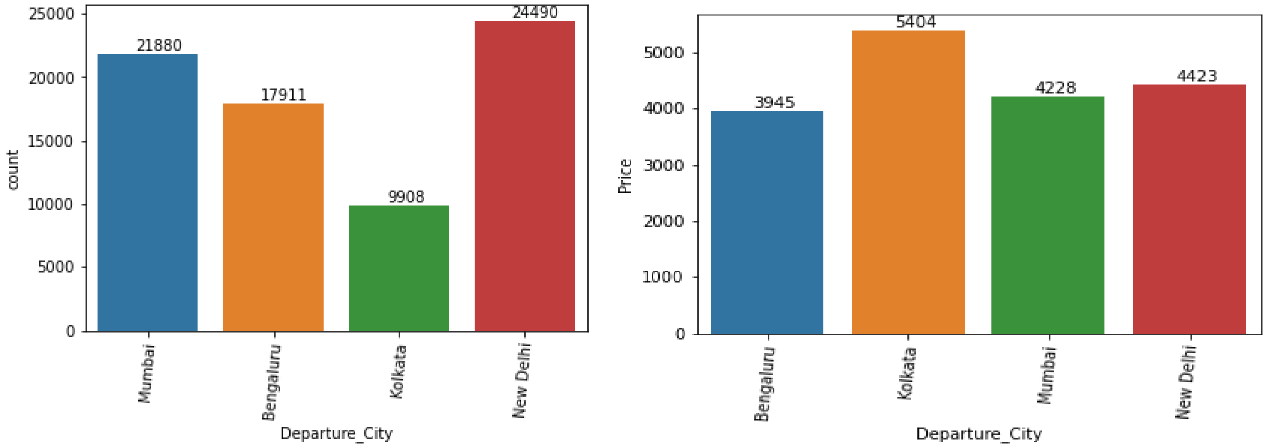
Fig: 23



As we can infer from Fig 23, that maximum number of flight available from 6 to 9 AM and 6 to 9 PM. Also we can infer from the above plot that the average fare of the flight also is high for 9 to 12 AM and lowest average fare of the flight from 9 to 12 PM.

- **Departure City:**

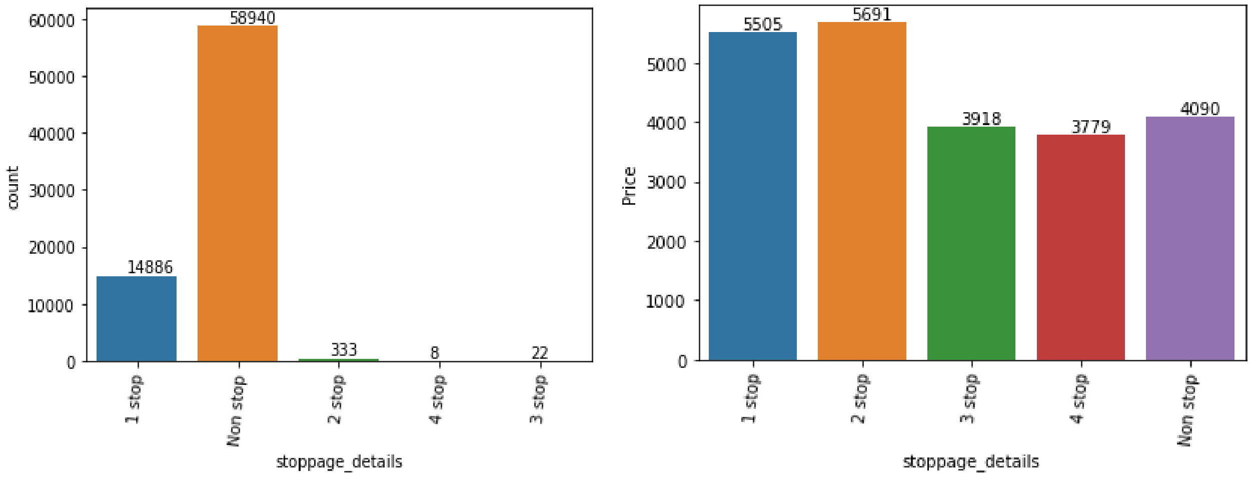
Fig: 24



As we can infer from above fig 24, Maximum number of flight are available from New Delhi, followed by Mumbai, Bengaluru and at last Kolkata.

- **Stoppage Details:**

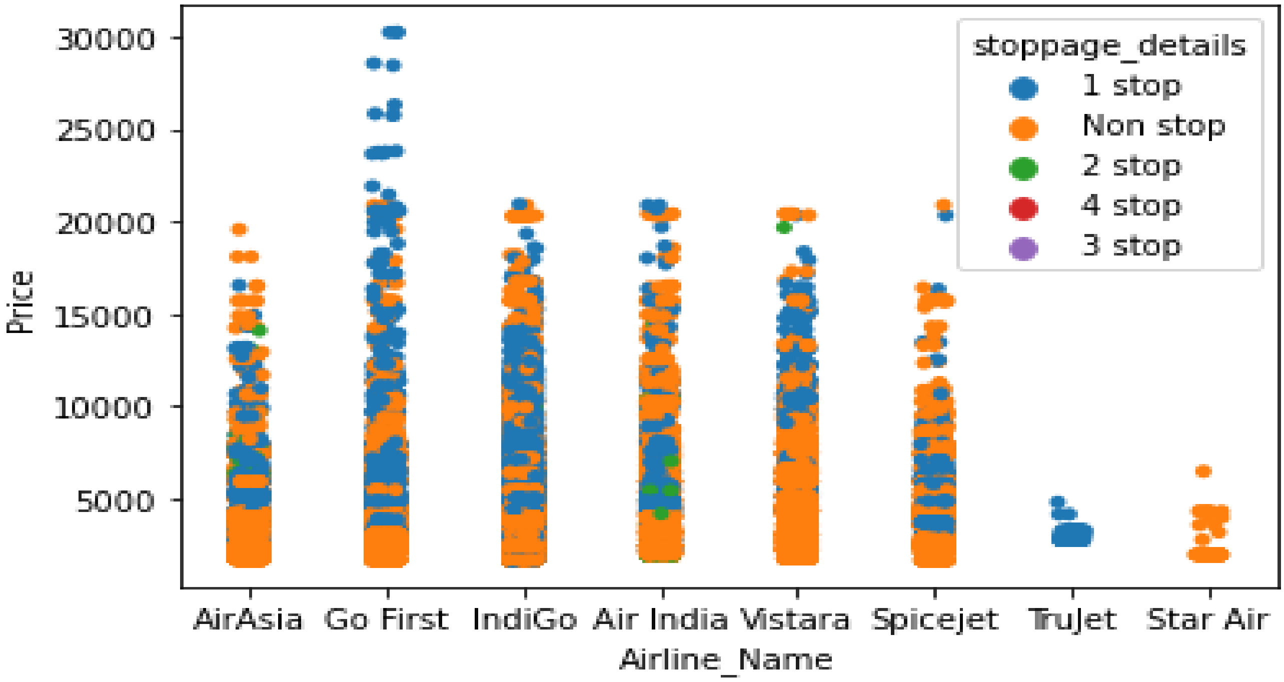
Fig: 25



We can infer from above Fig 25, that maximum car service has done with 0 to 100000 km. and also, we can see some outliers in last service Km features.

- **Airline Vs Price Vs Stoppage:**

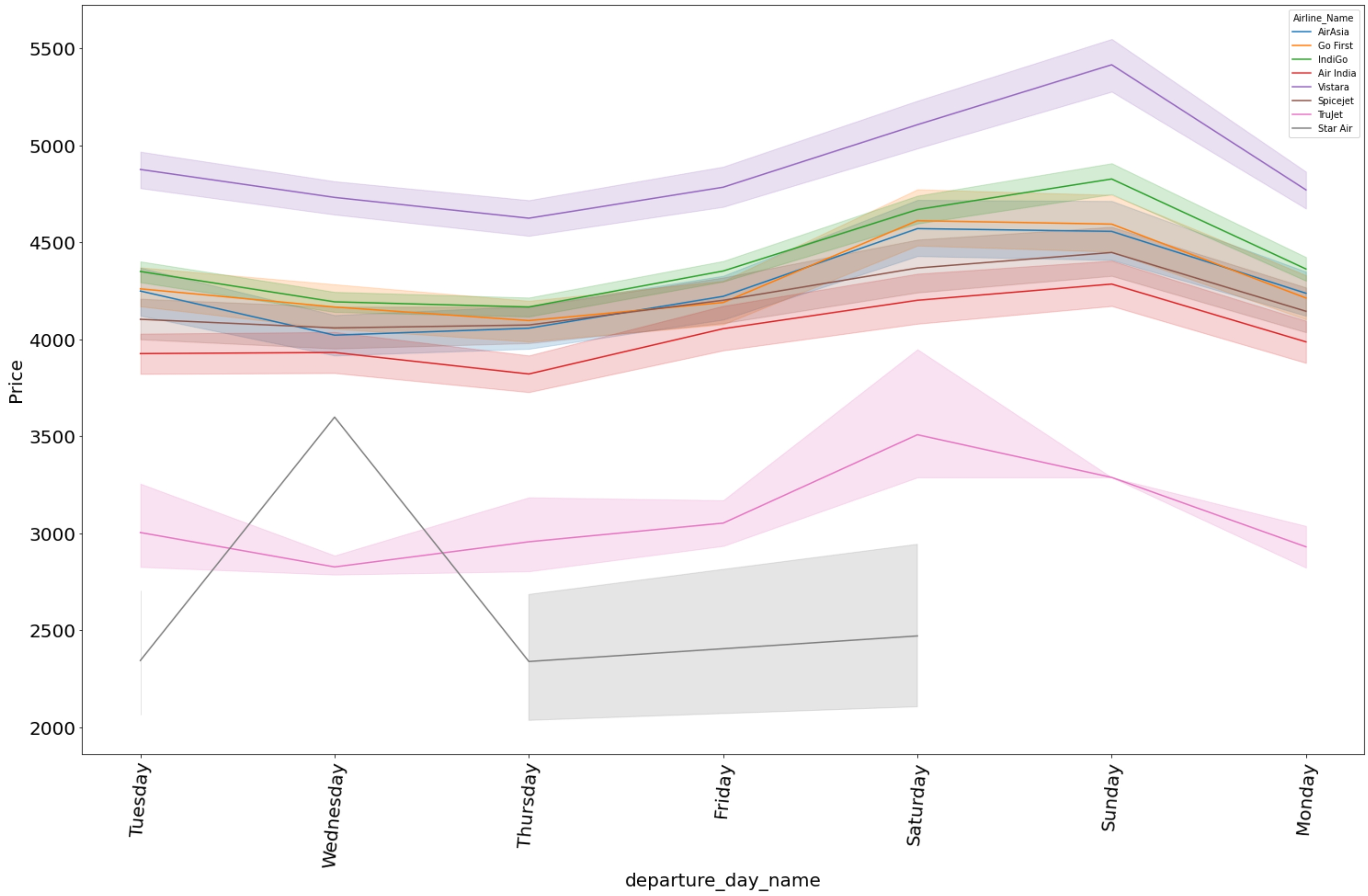
Fig: 26



We can infer from Fig 26 that Vistara airlines providing more Non stop stoppage in lesser fare range, where Go First Airline have more 1 stop flight with high fare compare to other Airlines.

- **Departure day Vs Price Vs Airline:**

  Fig: 27

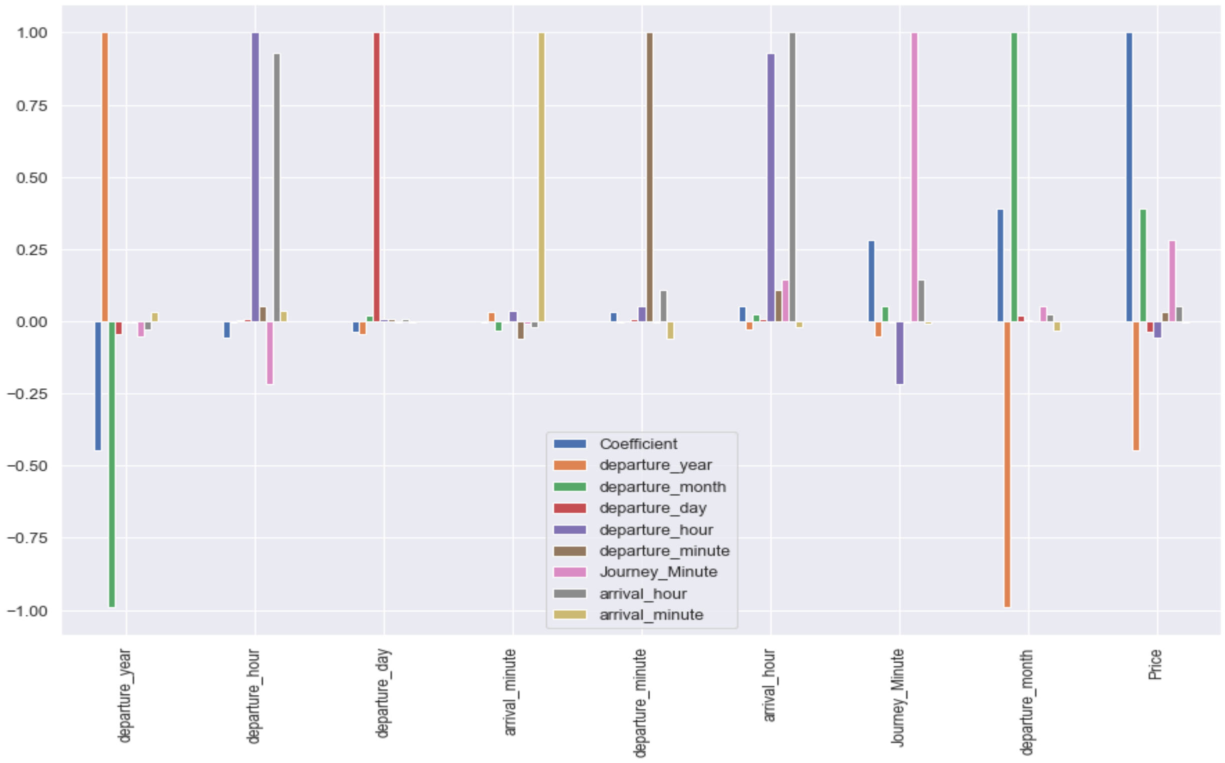


  We can infer from fig 27 that Saturday and Sunday Fight fare is expensive in all the airlines and Monday and Thursday the price is cheaper. Vistara Airlines fare is expensive throughout the week. In weekdays maximum airlines fare is from 3500 to 4500 except some airlines like Vistara, Star Air and True jet.

- Numerical Features Importance:

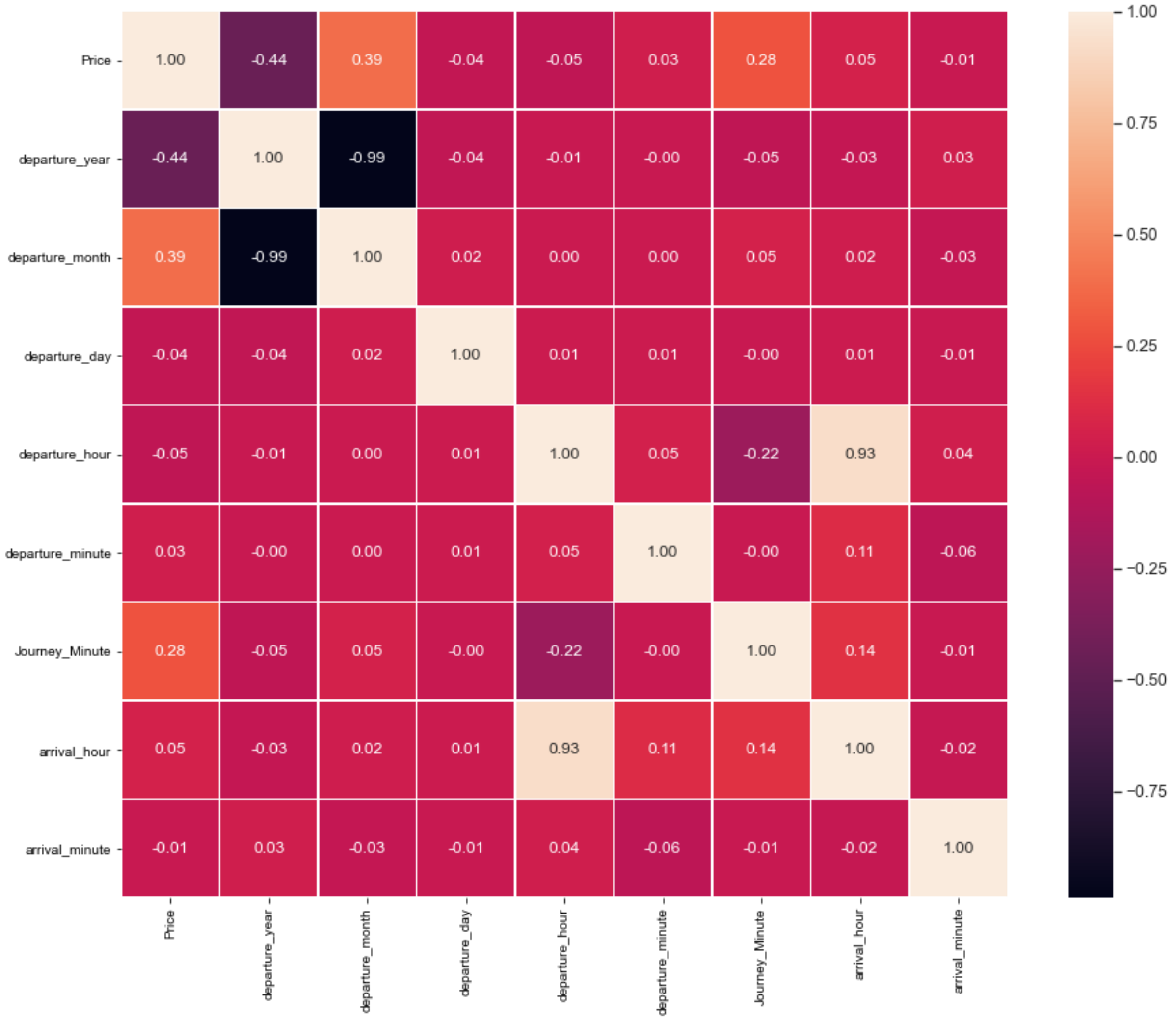
As shown in Fig 29, we can infer the features impact positively and negatively on target variable i.e. price.

Fig 29:



- **Correlation matrix:**

Fig 30



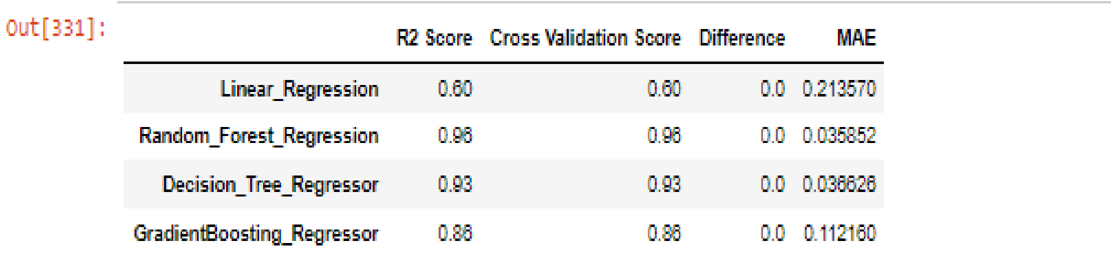
	Price	departure_year	departure_month	departure_day	departure_hour	departure_minute	Journey_Minute	arrival_hour	arrival_minute
Price	1.00	-0.44	0.39	-0.04	-0.05	0.03	0.28	0.05	-0.01
departure_year	-0.44	1.00	-0.99	-0.04	-0.01	-0.00	-0.05	-0.03	0.03
departure_month	0.39	-0.99	1.00	0.02	0.00	0.00	0.05	0.02	-0.03
departure_day	-0.04	-0.04	0.02	1.00	0.01	0.01	-0.00	0.01	-0.01
departure_hour	-0.05	-0.01	0.00	0.01	1.00	0.05	-0.22	0.93	0.04
departure_minute	0.03	-0.00	0.00	0.01	0.05	1.00	-0.00	0.11	-0.06
Journey_Minute	0.28	-0.05	0.05	-0.00	-0.22	-0.00	1.00	0.14	-0.01
arrival_hour	0.05	-0.03	0.02	0.01	0.93	0.11	0.14	1.00	-0.02
arrival_minute	-0.01	0.03	-0.03	-0.01	0.04	-0.06	-0.01	-0.02	1.00

➢ Interpretation of the Results

Below are the interpreted from the visualizations and modelling:

- ✓ We have seen that R2 score with compare to CV Score with different 15 k fold and MAE error is providing good performance Random Forest Regression Model compared to others applied model like Random Forest and Decision Tree model, as shown in Fig 31.

Fig 31:



Out[331]:

	R2 Score	Cross Validation Score	Difference	MAE
Linear_Regression	0.60	0.60	0.0	0.213570
Random_Forest_Regression	0.96	0.96	0.0	0.035852
Decision_Tree_Regressor	0.93	0.93	0.0	0.036626
GradientBoosting_Regressor	0.86	0.86	0.0	0.112160

- ✓ As shown in Fig 21, we have seen the most important Categorical Features is Airline Name and important numeric features are Departure month and Journey minute as shown in Fig 30.

CONCLUSION

➢ Key Findings and Conclusions of the Study

- ✓ We have shown in Fig 31; Random Forest Regressor is performing better compared to Linear Regression, Decision Tree Regressor and Gradient Boosting Regressor.
- ✓ While doing Principal Components Analysis, as shown in Fig 9 we can go with 30 number of components to get more than 90% of variance explained, but we have tried the model performance without PCA applied in model.
- ✓ While doing visual analysis and under EDA we have got like the some of the most important feature for prediction of fight fare are Airline Name, Journey Minute and Journey Month.
- ✓ We have Confirmed that based on R2 Score, CV Score and MAE Random Forest Regressor Model is performing better.
- ✓ At end we have found answer of some important question as mention below:

  - **Which Airline Services Average Fare is high and low?**
    Vistara Airline is high and Star Air is less.
  - **Which Airline having Maximum number of flights running domestically?**
    IndiGo
  - **What is the peak time of travelling by customer in flight?**
    6 to 9 AM and 6 to 9 PM
  - **From Which City the maximum flights are running?**
    New Delhi
  - **Maximum No. of Stoppage of domestic Airlines Services.**
    Non-Stop Type

- **Best time to book ticket for getting lowest price?**
  Before 3 months
- **Which day Airlines ticket are cheaper than other days in week?**
  Wednesday and Thursday
- **Which day Airline ticket are expensive in week?**
  Saturday and Sunday
- **Which City Airlines ticket are most expensive and cheapest?**
  Expensive- Kolkata
  Cheapest- Bangalore
- **How the stoppages impact on Airlines fare?**
  Fare is high for 1 and 2 type Stoppage
  Fare is low for Non-Stoppage type.
- **Which Airlines service having more option of stoppages?**
  Air India

➢ Learning Outcomes of the Study in respect of Data Science

Below are the Learning Outcome of the Study:

- ✓ Collecting the data from website, where after scraping data would be not in proper format.
- ✓ Dataset is huge.
- ✓ Feature data pattern are not normally distributed.
- ✓ Having lots of Outliers in Categorical features, which would be time consuming if dealing separately each variable for features so we have done features transformation with Standard Scaler technique.
- ✓ Target variable are not normally distributed as data was right skewed which we have rectified by doing log transformation.
- ✓ Selecting the Best Model based on R2 Score, CV Score and MAE error.

➢ Limitations of this work and Scope for Future Work

Mention below are the limitations and future scope steps:

- ✓ Due to time consuming we could able to fetch less attribute (i.e. 9 features), we could fetch more attribute to make our model more robust.