INSTITUTE FOR ADVANCE COMPUTING AND SOFTWARE DEVELOPMENT AKURDI, PUNE.

Documentation On,

"AIRLINE TICKET PRICE PREDICATION"

e-DBDA MAY 2021

Submitted By:

Group No: 15

PATIL ROHIT SUBHASH **(1338)**

PRATIK MUKUNDRAO GAJBHIYE **(1342)**

Mr. Prashant Karhale
Centre Coordinator

Mr. Akshay Tilekar
Project Guide

A
PROJECT REPORT ON

"AIRLINE TICKET PRICE PREDICATION"

Submitted by

Group No: 15
PATIL ROHIT SUBHASH **(1338)**
PRATIK MUKUNDRAO GAJBHIYE **(1342)**

Centre Coordinator
Mr. PRASHANT KARHALE

Under the Guidance of
Mr. AKSHAY TILEKAR

In the fulfillment of e – Diploma in Big Data Analytics course from Institute for Advance Computing and Software Development Akurdi, Pune in the academic year

May 2021 – Sept.2021

e-DBDA
May 2021

Institute for Advance Computing and Software Development
Akurdi, Pune. 411044

ACKNOWLEDGEMENT

It gives us immense pleasure to present our report for project on **"Airline Ticket Price Predication"** The able guidance of all teaching staff of this department made the study possible. They have been a constant source of encouragement throughout this project. We would like to express our grateful thanks to **Mr. Prashant Karhale Sir, Mr. Akshay Tilekar Sir** who guided us properly for this project. We would also like to express our sincere thanks to Institute for Advance Computing and Software Development Akurdi, Pune for giving us an opportunity to explore the subject and use our knowledge by conducting this project.

Group No: 15

Patil Rohit Subhash (1338)

Pratik Mukundrao Gajbhiye (1342)

e- DBDA, May 2021

Institute for Advance Computing and Software Development, Akurdi

PROJECT OVERVIEW

Travelling through airline has become an integral part of today's lifestyle as more and more people are opting for faster travelling options. The airline ticket prices increase or decrease every now and then depending on various factors like timing of the flights, destination, duration of airline s. various occasions such as vacations or festive season. Therefore, having some basic idea of the airline fares before planning the trip will surely help many people save money and time. In the proposed system a predictive model will be created by applying machine learning algorithms to the collected historical data of airlines. This system will give people the idea about the trends that prices follow and also provide a predicted price value which they can refer to before booking their flight tickets to save money. This kind of system or service can be provided to the customers by flight booking companies which will help the customers to book their tickets accordingly. In this we did visualize large data and then implemented Random Forest Regressor, Extra Trees Regressor, XGB Regressor, Linear Regression machine learning with data visualization. This project aims to develop machine learning model for airline ticket price predication with visualization.

CONTENTS

Chapter - 1

INTRODUCTION

Now a days, the airline corporations are using complex strategies and methods to assign airfare prices in a dynamic fashion. These strategies are taking into account several financial, marketing, commercial and social factors closely connected with the final airfare prices. Due to the high complexity of the pricing models applied by the airlines, it is very difficult a customer to purchase an air ticket in the lowest price, since the price changes dynamically. For this reason, several techniques able to provide the right time to the buyer to purchase an air ticket by predicting the airfare price, have been proposed recently. The majority of these methods are making use of sophisticated prediction models from the computational intelligence research field known as Machine Learning (ML).

GLOSSARY

Training data: The data that are used to train classification models.

Validation data: The data that are used to test the performance of the classification model during the training process. Validation data are used to fine tune parameters in the classification model and the data should not be part of the training data.

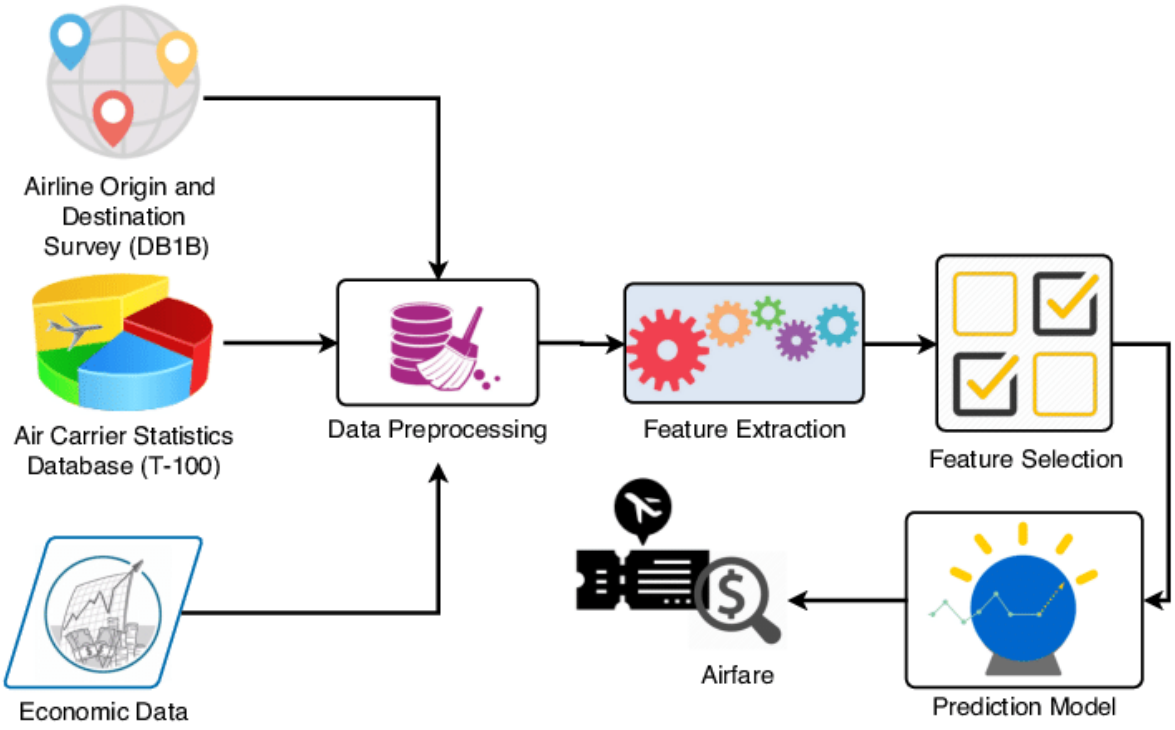
Testing data: The data that are used to test the performance of the trained Regressor model (after training process). Testing data are used to evaluate the final performance of the Regressor model. Testing data should not be part of training data or validation data. No fine tune should be made based on the result of testing data.

Overfitting: The Regression model performs consistently better on training data than on validation data and testing data.

OBJECTIVE

- The objective of this project is to predict airline ticket prices given the various parameters.
- Data used in this project is publicly available at Kaggle.
- This will be a regression problem since the target or dependent variable is the price (continuous numeric value).

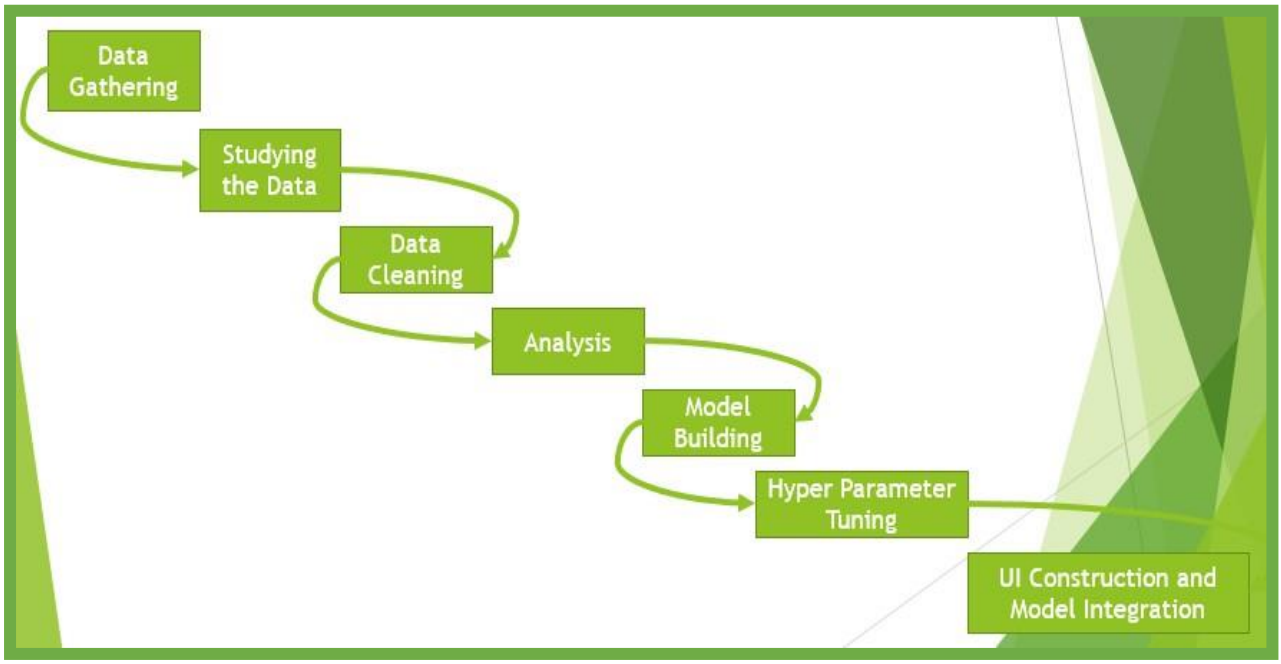
BLOCK DIAGRAM



Chapter – 2

OVERALL DESCRIPTION

Workflow of Project:



Workflow Diagram

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, and it will be a different story.

To solve this problem, we have been provided with prices of flight tickets for various airlines between the months of March and June of 2019 and between various cities, using which we aim to build a model which predicts the prices of the flights using various input features.

Data Gathering:

Link for the dataset — https://www.kaggle.com/vinayshaw/airfare-price-prediction/data

We have 2 datasets here — training set and test set.

The training set contains the features, along with the prices of the flights. It contains 10683 records, 10 input features and 1 output column — 'Price'.

The test set contains 2671 records and 10 input features. The output 'Price' column needs to be predicted in this set. We will use Regression techniques here, since the predicted output will be a continuous value.

Following is the description of features available in the dataset –

1. **Airline**: The name of the airline.

2. **Date_of_Journey**: The date of the journey

3. **Source**: The source from which the service begins.

4. **Destination**: The destination where the service ends.

5. **Route**: The route taken by the flight to reach the destination.

6. **Dep_Time**: The time when the journey starts from the source.

7. **Arrival_Time**: Time of arrival at the destination.

8. **Duration**: Total duration of the flight.

9. **Total_Stops**: Total stops between the source and destination.

10. **Additional_Info**: Additional information about the airline

11. **Price**: The price of the ticket

Study on Data:

This article explains the complete process to build a machine learning model. Below mentioned are the various phases that we will go through, throughout the project –

1. Exploratory data analysis and Data modeling

2. Outlier detection and skewness treatment

3. Encoding the data — Label Encoder

4. Scaling the data — Standard scaler

5. Fitting the machine learning models

6. Cross-validation of the selected model

7. Model hyper-tuning

8. Saving the final model and prediction using saved model

So let’s begin exploring our data set and start building a prediction model

Data Cleaning:

All the collected data needed a lot of work so after the collection of data, it is needed to be clean and prepare according to the model requirements. All the unnecessary data is removed like duplicates and null values.

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.

```
1 train_data.isnull().sum()
```

```
Airline            0
Date_of_Journey    0
Source             0
Destination        0
Route              1
Dep_Time           0
Arrival_Time       0
Duration           0
Total_Stops        1
Additional_Info    0
Price              0
dtype: int64
```

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.

In all machine learning this technology, this is the most important and time consuming step. Various statistical techniques and logic built in python are used to clean and prepare the data. For example, the price was character type, not an integer.

Further, we split the Route column to create multiple columns with cities that the flight travels through. We check the maximum number of stops that a flight has, to confirm what should be the maximum number of cities in the longest route –

```
[6] 1 #replace the values in key values
    2 train_data.replace({'non-stop':0,'1 stop':1,'2 stops':2,'3 stops':3,'4 stops':4},inplace=True)
```

We now start exploring the columns available in our dataset. The first thing we do is to create a list of categorical columns, and check the unique values present in these columns –

```
1 train_data['Airline'].unique()
```

```
array(['IndiGo', 'Air India', 'Jet Airways', 'SpiceJet',
       'Multiple carriers', 'GoAir', 'Vistara', 'Air Asia',
       'Vistara Premium economy', 'Jet Airways Business',
       'Multiple carriers Premium economy', 'Trujet'], dtype=object)
```

Analysis:

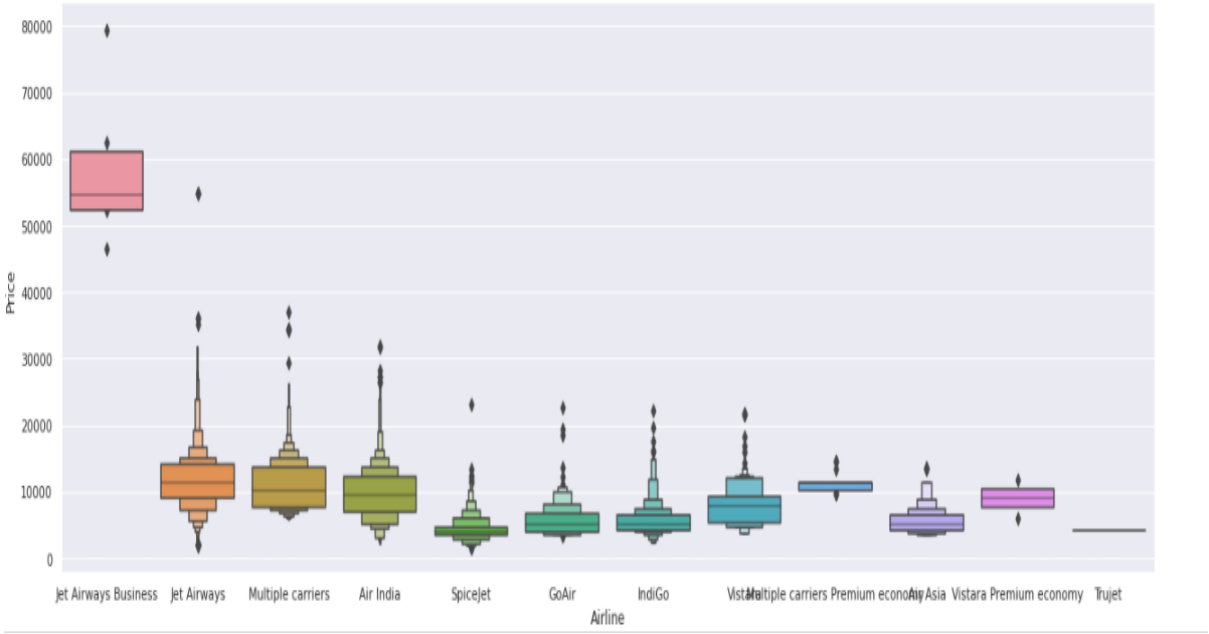
Data preparation is followed by analysing the data, uncovering the hidden trends and then applying various machine learning models. Also, some features can be calculated.

```
[231]  1 train_data.head(10)
```

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302
5	SpiceJet	24/06/2019	Kolkata	Banglore	CCU → BLR	09:00	11:25	2h 25m	non-stop	No info	3873
6	Jet Airways	12/03/2019	Banglore	New Delhi	BLR → BOM → DEL	18:55	10:25 13 Mar	15h 30m	1 stop	In-flight meal not included	11087
7	Jet Airways	01/03/2019	Banglore	New Delhi	BLR → BOM → DEL	08:00	05:05 02 Mar	21h 5m	1 stop	No info	22270
8	Jet Airways	12/03/2019	Banglore	New Delhi	BLR → BOM → DEL	08:55	10:25 13 Mar	25h 30m	1 stop	In-flight meal not included	11087
9	Multiple carriers	27/05/2019	Delhi	Cochin	DEL → BOM → COK	11:25	19:15	7h 50m	1 stop	No info	8625

From the existing feature. Days to departure can be obtained by calculating the difference between the departure date and the date on which data is taken. This parameter is considered to be within 45 days. Also, the day of departure plays an important role in whether it is holiday or weekday. Intuitively the flights scheduled during weekends have a more price compared to the flights on Wednesday or Thursday. Similarly, time also seems to play an important factor. So the time is been divided into four categories: Morning, afternoon, evening, night.

```
sns.catplot(y = "Price", x = "Airline", data = train_data.sort_values("Price", ascending = False), kind="boxen", height = 6, aspect = 3)
plt.show()
```



Model Building:

Train/Test split:

One important aspect of all machine learning models is to determine their accuracy.Now, in order to determine their accuracy, one can train the model using the given dataset and then predict the response values for the same dataset using that modeland hence, find the accuracy of the model. A better option is to split our data into two parts: first one for training our machine learning model, and second one for testing our model.

- Split the dataset into two pieces: a training set and a testing set.
- Train the model on the training set.
- Test the model on the testing set, and evaluate how well our model did.

Advantages of train/test split:

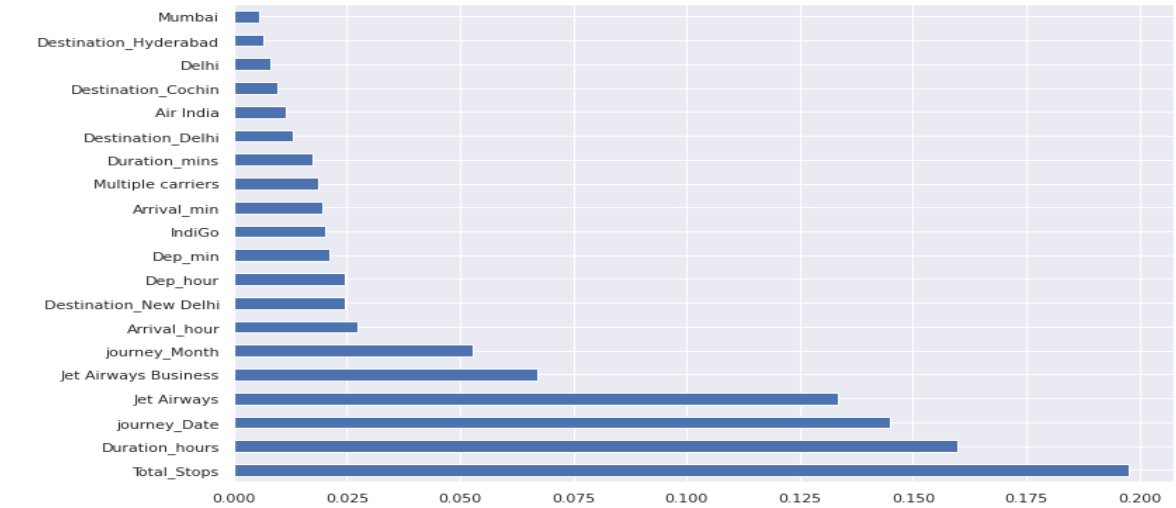
- Model can be trained and tested on different data than the one used for training.
- Response values are known for the test dataset, hence predictions can be evaluated
- Testing accuracy is a better estimate than training accuracy of out-of sampleperformance.

Machine learning consists of algorithms that can automate analytical model building. Using algorithms that iteratively learn from data, machine learning modelsfacilitate computers to find hidden insights from Big Data without being explicitly programmed where to look.

Extra Trees Regressor:

This class implements a meta estimator that fits a number of randomized decision trees (a.k.a. extra-trees) on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

```
plt.figure(figsize = (12,8))
feat_importances = pd.Series(selection.feature_importances_, index=X.columns)
feat_importances.nlargest(20).plot(kind='barh')
plt.show()
```



Accuracy Score:

```
[ ]   1 print('MAE:', metrics.mean_absolute_error(y_test,y_pred1))
      2 print('MSE:', metrics.mean_squared_error(y_test, y_pred1))
      3 print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred1)))
```

```
MAE: 1302.7215028856651
MSE: 5116533.846116345
RMSE: 2261.9756510883017
```

```
[ ]   1 print('r2_score:',metrics.r2_score(y_test,y_pred1))
```

```
r2_score: 0.7819334402527878
```

Linear Regression:

In simple linear regression there is only one independent and dependent feature but as our dataset consists of many independent features on which the price may depend upon, we will be using multiple linear regression which estimates relationship between two or more independent variables and one dependent variable. The multiple linear regression model is represented by:

$Y = \beta 0x1 + \ldots. + \beta nxn + Ɛ$

Y = the predicted value of the dependent variable

Xn = the independent variables

βn = independent variables coefficients

Ɛ = y-intercept when all other parameters are 0

Accuracy Score:

```
103]  1 print('MAE:', metrics.mean_absolute_error(y_test,y_pred2))
      2 print('MSE:', metrics.mean_squared_error(y_test, y_pred2))
      3 print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred2)))
```

```
MAE: 2048.272626605159
MSE: 9646558.195863314
RMSE: 3105.8908860201955
```

```
104]  1 print('r2_score:',metrics.r2_score(y_test,y_pred2))
```

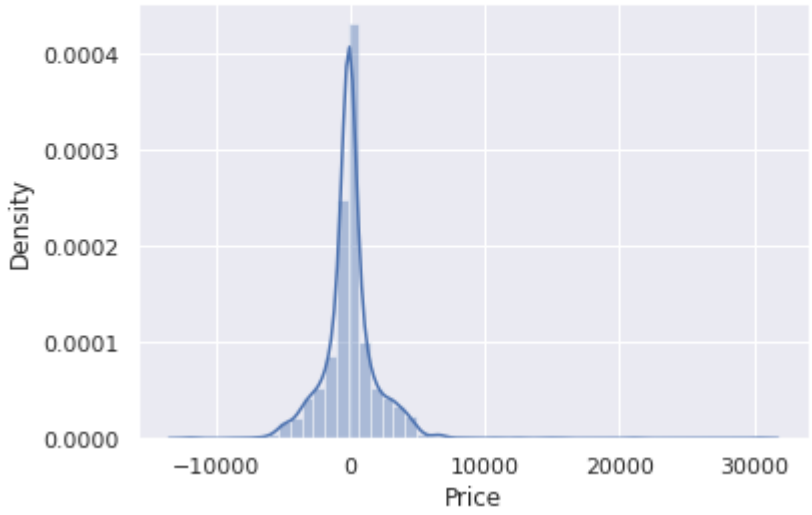
```
r2_score: 0.5888639023134974
```

Random Forest Regressor:

Random Forest is an ensemble learning technique where training model uses multiple learning algorithms and then combine individual results to get a final predicted result. Under ensemble learning random forest falls into bagging category where random number of features and records will average value of the predicted values if considered as the output of the random forest model.

```
2 sns.distplot(y_test-y_pred,kde=True)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7fe10ab5c110>
```

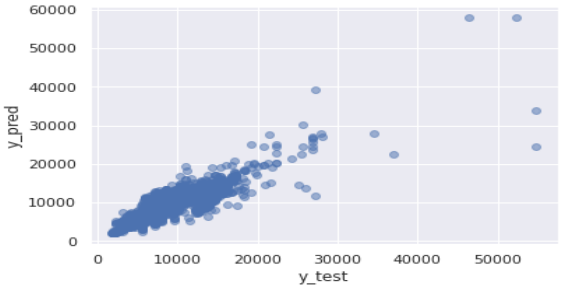


It is a supervised learning algorithm. The benefit of the random forest is, it very well may be utilized for both characterization and relapse issue which structure

most of current machine learning framework. Random forest forms numerous decision trees, what’s more, adds them together to get an increasingly exact and stable expectation. Random Forest has nearly the equivalent parameters as a decision tree or a stowing classifier model. It is very simple to discover the significance of each element on the expectation when contrasted with others in this calculation. The regular component in these techniques is, for the kth tree, a random vector theta k is produced, autonomous of the past random vectors theta 1, …, theta k-1 however with the equivalent distribution, while a tree is developed utilizing the preparation set and bringing about a classifier. X is an information vector. For a period, in stowing the random vector is created as the includes in N boxes where N is the number of models in the preparation set of information. In random split, choice includes various autonomous random whole numbers between 1 to K. The dimensionality and nature of theta rely upon its utilization in the development of a tree. After countless trees are created, they select the most famous class. These methodology are called as random forests.

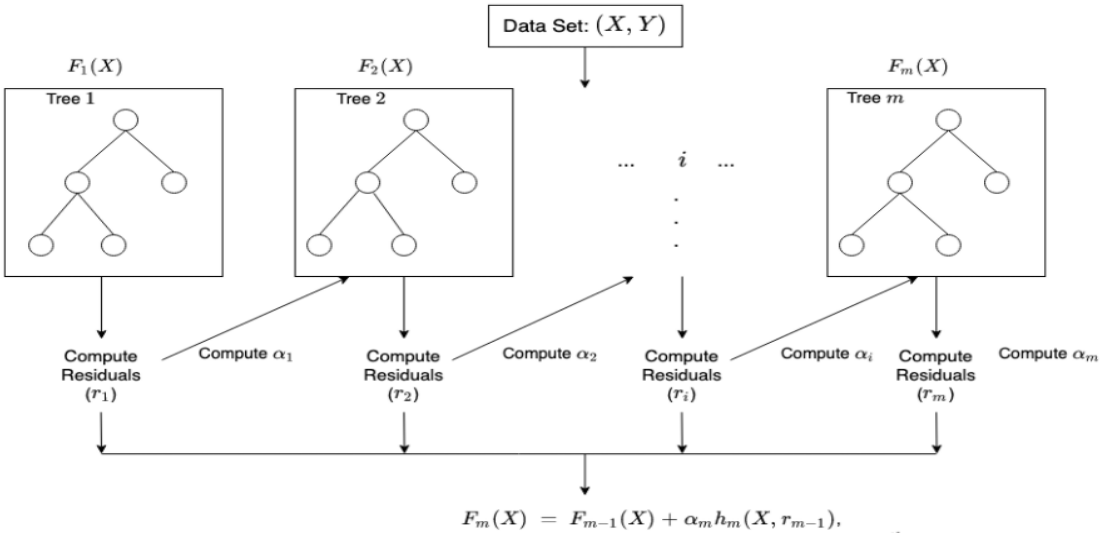
```
#Plotting scatter graph to check linear relations
plt.scatter(y_test,y_pred,alpha=0.5)
plt.xlabel('y_test')
plt.ylabel('y_pred')
```

```
Text(0, 0.5, 'y_pred')
```



XG-Boost Regressor :

XG-Boost is a popular and efficient open-source implementation of the gradient boosted trees algorithm. When using gradient boosting for regression, the weak learners are regression trees, and each regression tree maps an input data point to one of it's leafs that contains a continuous score.



Accuracy Score:

```
[112]  1 print('MAE:', metrics.mean_absolute_error(y_test,y_pred4))
       2 print('MSE:', metrics.mean_squared_error(y_test, y_pred4))
       3 print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred4)))
```

```
MAE: 1586.0642547964437
MSE: 5891792.721275569
RMSE: 2427.30153076942
```

```
[113]  1 print('r2_score:',metrics.r2_score(y_test,y_pred4))
```

```
r2_score: 0.7488919240810953
```

Performance Metrics

Performance metrics are statistical models which will be used to compare the accuracy of the machine learning models trained by different algorithms. The sklearn.metrics module will be used to implement the functions to measure the errors from each model using the regression metrics. Following metrics will be used to check the error measure of each model.

MAE (Mean Absolute Error)

Mean Absolute Error is basically the sum of average of the absolute difference between the predicted and actual values.

$$\mathbf{MAE = 1/n[\sum(y-\acute{y})]}$$

y = actual output values,

ý = predicted output values

n = Total number of data points Lesser the value of MAE the better the performance of your model.

MSE (Mean Square Error)

Mean Square Error squares the difference of actual and predicted output values before summing them all instead of using the absolute value.

$$\mathbf{MSE = 1/n[\sum(y-\acute{y})2\ ]}$$

y=actual output values

ý=predicted output values

n = Total number of data points MSE punishes big errors as we are squaring the errors. Lower the value of MSE the better the performance of the model.

RMSE (Root Mean Square Error)

RMSE is measured by taking the square root of the average of the squared difference between the prediction and the actual value.

$$\mathbf{RMSE = \sqrt{1/n[\sum(y-\acute{y})2\ ]}}$$

y=actual output values

ý=predicted output values

n = Total number of data points RMSE is greater than MAE and lesser the value of RMSE between different model the better the performance of that model.

R 2 (Coefficient of determination)

It helps you to understand how well the independent variable adjusted with the variance in your model.

$$\mathbf{R\ 2 = 1 - \sum (\acute{y}-\bar{y})\ 2\ \sum (y-\bar{y})\ 2}$$

The value of R-square lies between 0 to 1. The closer its value to one, the better your model is when comparing with other model values.

Chapter - 3

SIGNIFICANCE

In 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. As the pricing models of the company are developed in order to maximize the revenue management. So to get result with maximum accuracy regression analysis is used. From the studies, the feature that influences the prices of the ticket are to be considered. In future the details about number of available seats can improve the performance of the model.

Chapter - 4

FUTURE SCOPE

Currently, there are many fields where prediction-based services are used such as stock price predictor tools used by stock brokers and service like Zestimate which gives the estimated value of house prices. Therefore, there is requirement for service like this in the aviation industry which can help the customers in booking tickets. There are many researches works that have been done on this using various techniques and more research is needed to improve the accuracy of the prediction by using different algorithms. More accurate data with better features can be also be used to get more accurate results.

Chapter - 5

CONCLUSION

A proper implementation of this project can result in saving money of inexperienced people by providing them the information related to trends that flight prices follow and also give them a predicted value of the price which they use to decide whether to book ticket now or later. In conclusion this type of service can be implemented with good accuracy of prediction. As the predicted value is not fully accurate there is huge scope for improvement of these kind of service

.