# FLIGHT TICKET PRICE PREDICTION

Mini Project Report

Submitted by

**Rinu Anna Philip**

*Submitted in partial fulfillment of the requirements for the award of the degree of*

***Master of Computer Applications***
***Of***
***A P J Abdul Kalam Technological University***


**FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®**
**ANGAMALY-683577, ERNAKULAM(DIST)**
**MARCH 2022**

# DECLARATION

I, **Rinu Anna Philip**, hereby declare that the report of this project work, submitted to the Department of Computer Applications, Federal Institute of Science and Technology (**FISAT**), Angamaly in partial fulfillment of the award of the degree of Master of Computer Application is an authentic record of our original work.

The report has not been submitted for the award of any degree of this university or any other university.

**Date : 04-03-2022**

**Place: Angamaly**

# FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®

**ANGAMALY, ERNAKULAM-683577**

**DEPARTMENT OF COMPUTER APPLICATIONS**


## CERTIFICATE

This is to certify that the project report titled **"Flight Ticket Price Prediction"** submitted by **Rinu Anna Philip** towards partial fulfillment of the requirements for the award of the degree of Master of Computer Applications is a record of bonafide work carried out by them during the year 2022.

**Project Guide** **Head of the Department**

Submitted for the viva-voice held on ..... at .....

**Examiner1 :**

**Examiner2 :**

# ACKNOWLEDGEMENT

Gratitude is a feeling which is more eloquent than words, more silent than silence. To complete this project work I needed the direction, assistance and co-operation of various individuals, which is received in abundance with the grace of God.

I hereby express our deep sense of gratitude to **Dr. Manoge George**, Principal of FISAT and **Dr. C Sheela**, Vice principal of FISAT, for allowing us to utilize all the facilities of the college.

My sincere thanks to **Dr. Deepa Mary Mathew**, Head of the department of Computer Applications FISAT and scrum master and our Internal guide for this project **Ms.Shidha and Ms.Joice T** for giving valuable guidance, constructive suggestions and comment during my project work. I also express my boundless gratitude to all the lab faculty members for their guidance.

Finally I wish to express a whole heart-ed thanks to my parents, friends and well-wishers who extended their help in one way or other in preparation of my project. Besides all, I thank GOD for everything.

# ABSTRACT

People who frequently travel through flight will have better knowledge on best discount and right time to buy the ticket. For the business purpose many airline companies change prices according to the seasons or time duration. They will increase the price when people travel more. Estimating the highest prices of the airlines data for the route is collected with features such as Duration, Source, Destination, Arrival, Departure. Features are taken from chosen dataset and in this paper, we have used machine learning techniques and regression strategies for prediction of the price wherein the airline price ticket costs vary overtime. I have implemented flight price prediction for users by using random forest algorithms. Random Forest Regression shows the best accuracy for predicting the flight price. Also, we have done correlation tests and used ExtraTreesRegressor for selecting important feature.

# Contents

# Chapter 1

# INTRODUCTION

The flight ticket buying system is to purchase a ticket many days prior to flight takeoff so as to stay away from the effect of the most extreme charge. Mostly, aviation routes don't agree this procedure. Plane organizations may diminish the cost at the time, they need to build the market and at the time when the tickets are less accessible. They may maximize the costs. So the cost may rely upon different factors.

To foresee the costs this venture uses AI to exhibit the ways of flight tickets after some time. All organizations have the privilege and opportunity to change it's ticket costs at anytime. Explorer can set aside cash by booking a ticket at the least costs. People who had travelled by flight frequently are aware of price fluctuations. The airlines use complex policies of Revenue Management for execution of distinctive evaluating systems.

The evaluating system as a result changes the charge depending on time, season, and festive days to change the header or footer on successive pages. The ultimate aim of the airways is to earn profit whereas the customer searches for the minimum rate. Customers usually try to buy the ticket well in advance of departure date so as to avoid hike in airfare as date comes closer. But actually this is not the fact. The customer may wind up by giving more than they ought to for the same seat.

# Chapter 2

# PROOF OF CONCEPT

## 2.1 Existing System

Early work also considered using classification models to predict the trends of the itineraries. Ren et al. proposedusing LR, Naive Bayes, Softmax regression, and SVMs to build a prediction model and classify the ticket price into five bins to compare the relative values with the overall average price.More than nine thousand data points, including six features(e.g., the departure week begin, price quote date, the numberof stops in the itinerary, etc.), were used to build the models.The authors reported the best training error rate close to 22.9 using LR model. Their SVM regression model failed to produce a satisfying result.

## 2.2 Proposed System

Through Regression Analysis the visualization and forecasting are performed for the presented model. Blending of technologies, processing is called Conceptually the Intelligence, that is machine learning and virtualization etc. ML is in trend to build our skills and it is one of the highest growth field in computer science and health care informatics.As the time passes by the algorithm should be learnt is the main goal in Machine. Also, used for predicting algorithm that makes the com-

munication with agent and makes easier for learning. In this paper, random forest algorithms is used to find solutions for flight price problems in machine learning tasks.The data collecting is performed followed by data pre-processing. Before data modelling is done, data must be split into train and test dataset to ignore the data leakage. Based on the various attributes in the dataset for example departure and arrival features play the important role for predicting the price. Running the random forest grouping the maximum price of airlines. Next performing the feature engineering and calculating the accuracy.

## 2.3 Objectives

To evaluate the Minimum Flight price, a dataset is built and studied a trend of price variation for the period of limited days. Machine Learning algorithms are applied on the dataset to predict the dynamic fare of flights.

This gives the predicted values of flight fare to get a flight ticket at minimum cost.

Data is collected from the websites which sell the flight tickets so only limited information can be accessed. The algorithm give the accuracy of the model.

# Chapter 3

# SCRUM MEETINGS

**On 24-11-2021**

On this day I started searching the miniproject topic based on the new technology such as deep learning,IoT,machine learning,classification,prediction etc".

**On 29-11-2021**

The topic was selected and did the detail study of the topic,the required dataset was selected.The dataset was searched from the different site such as kaggle,dataset etc.

**On 06-12-2021**

This day I submitted the synopsis and research paper to guide for the topic approval.

**On 15-12-2021**

After getting approval from the guide, the algorithm and model for the project were structured.Then the algorithm were choosen.

**On 18-12-2021**

On this day mam took a detailed class on how to do the project,what IDEs to use,what paper are refered,what steps are follow to do the project and so on

**On 06-01-2022**

According to the project the required IDE such as Visual Studio Code,Colab are choosen.Even checked whether the system was efficient to train the model.Here colab to code the project,then started to deploying the model using the algorithm.Python language is used to code the project.

**On 10-01-2022**

After the project first review according to Mam's opinion added two new Algorithm to the project to find which algorithm is having more accuracy rate.

**On 13-01-2022**

Used different algorithm/data model then choose the maximum accuracy one. The algorithm used are:-
Random Forest Regression
XGBoost
Bagging Regression

**On 19-01-2022**

Started to do project coding.Firstly study the dataset and download the dataset from kaggle.The dataset is about different airlines in India and their details.

**On 25-01-2022**

Testing the data application

**On 28-01-2022**

The training done in three different data model then choose the maximum accuracy with regression for predicting the price airline ticket.Random Forest Regression model is used for prediction.

**On 02-02-2022**

Created the git repository.

**On 07-02-2022**

Used flask for connection.

# Chapter 4

# IMPLEMENTATION

This project aims to develop a website for Flight Ticket Price Prediction. The price is predicted using the data like source,destination,date of journey and airline.A Machine learning algorithms is used here for predicting the ticket price. The algorithm used here is Random Forest Regression algorithm. This is an algorithm which ensembles the less predictive model to produce better predictive models. It aggregates the base model to create a large model. The features are sampled and passed to trees without replacement to obtain the highly uncorrelated decision trees. To select the best split it is required to have less correlation between the trees. The main concept that makes random forest different from the decision tree is aggregated uncorrelated trees.

- Feature Selection: Finding out the best feature which will contribute and have good realtion with target variable.
- Extra Tree Regressor: Finding the important.
- Hyperparameter Tuning:Hyperparameter tuning is choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a model argument whose value is set before the learning process begins.

## ALGORITHM

### Random Forest Regression

This is an algorithm which ensembles the less predictive model to produce better predictive models. It aggregates the base model to create a large model. The features are sampled and passed to trees without replacement to obtain the highly uncorrelated decision trees. To select the best split it is required to have less correlation between the trees. The main concept that makes random forest different from the decision tree is aggregated uncorrelated trees.

### XGBoost Regression

XGBoost is one of the most popular machine learning algorithms these days. XGBoost stands for eXtreme Gradient Boosting. Regardless of the type of prediction task at hand; regression or classification. XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.

### Bagging Regression

Bagging Regressor is an ensemble estimator which fits base estimator on each random subset of the Train dataset and then aggregates their individual predictions to form a final prediction using voting or averaging method. Here the base estimator is Decision Trees.

```
results
```

| | ML Model | Train Accuracy | Test Accuracy | Train RMSE | Test RMSE |
|---|---|---|---|---|---|
| 0 | XGBoost Regression | 0.860 | 0.836 | 1723.017 | 1879.706 |
| 1 | Bagging Regression | 0.886 | 0.704 | 1553.304 | 2524.835 |
| 2 | Random Forest Regression | 0.953 | 0.797 | 0.000 | 0.000 |

## 4.1 System Architecture

The use case diagram that describes the operation of the system .

## 4.2 Dataset

The data requirements is very high for the project.We get a dataset that contains the component like:-

• Date of journey
• Time of Departure
• Place of Departure
• Time of Arrival
• Place of Destination/Arrival
• Airway company
• Total Fare

## 4.3 Modules

### 4.3.1 Data Preprocessing

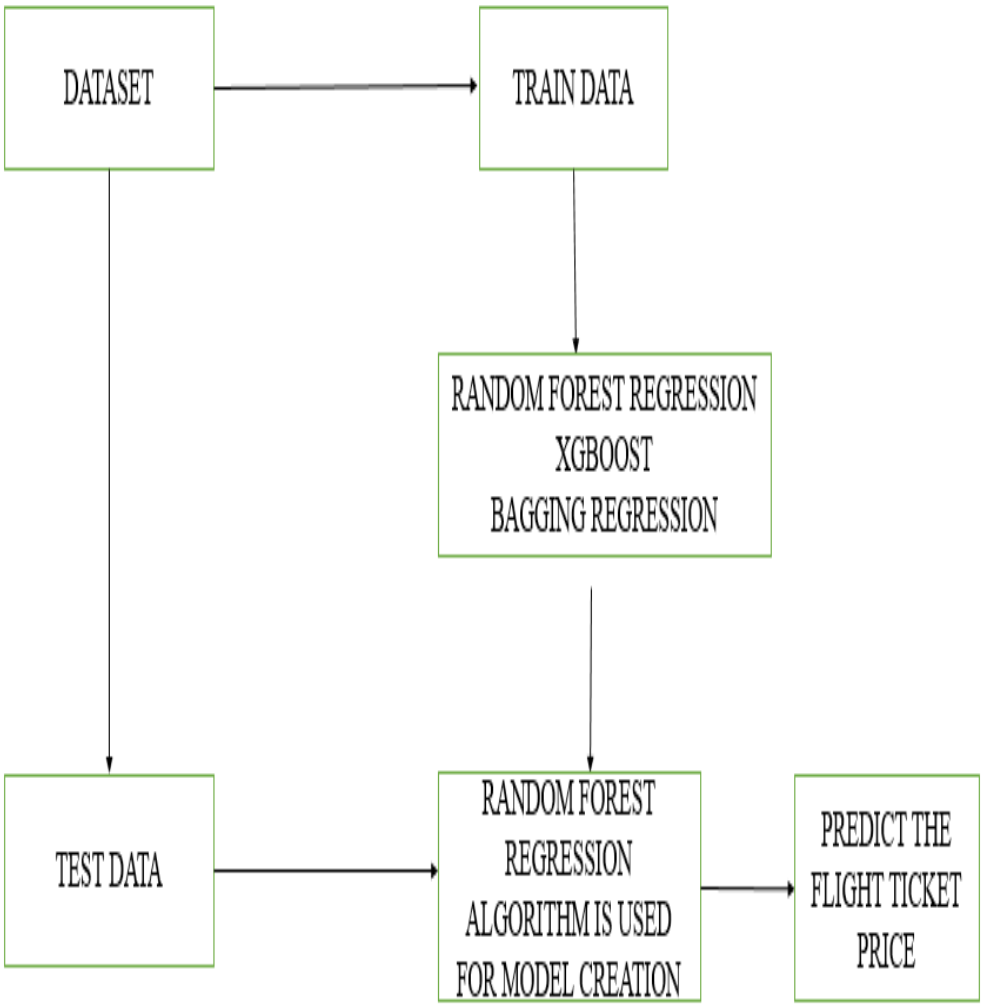
Explore the dataset and analyse it.The attribute $Date_of_journey, Dep_Time, Arrival_Timeareobjectdatatype,$
$DataslikeAirline, Source, Destination, Totalstopsarecategoricaldata, sousingOneHotEncoderandLabel$

### 4.3.2 TEXT ENGINEERING

Feature Selection: Finding out the best feature which will contribute and have good realtion with target variable.

1. Extra Tree Regressor: Finding the important.

2. Hyperparameter Tuning:Hyperparameter tuning is choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a model argument whose value is set before the learning process begins.


DATASET
TRAIN DATA
RANDOM FOREST REGRESSION
XGBOOST
BAGGING REGRESSION
TEST DATA
RANDOM FOREST
REGRESSION
ALGORITHM IS USED
FOR MODEL CREATION
PREDICT THE
FLIGHT TICKET
PRICE

# Chapter 5

# RESULT ANALYSIS

The Flight Ticket Price Prediction helps us to get the price of airline based on the date of journey,source,destination and airline.The predicted price is displayed.

# Chapter 6

# CONCLUSION AND FUTURE SCOPE

## 6.1 Conclusion

To evaluate the conventional algorithm, a dataset is built and studied a trend of price variation for the period of limited days. Machine Learning algorithms are applied on the dataset to predict the dynamic fare of flights.This gives the predicted values of flight fare to get a flight ticket at minimum cost. Data is collected from the websites which sell the flight tickets so only limited information can be accessed. The values of R-squared obtained from the algorithm give the accuracy of the model.

## 6.2 Future Scope

In the future, if more data could be accessed such as the current availability of seats, the predicted results will be more accurate.

# Chapter 7

# SOURCE CODE

```
[6] train_data
```

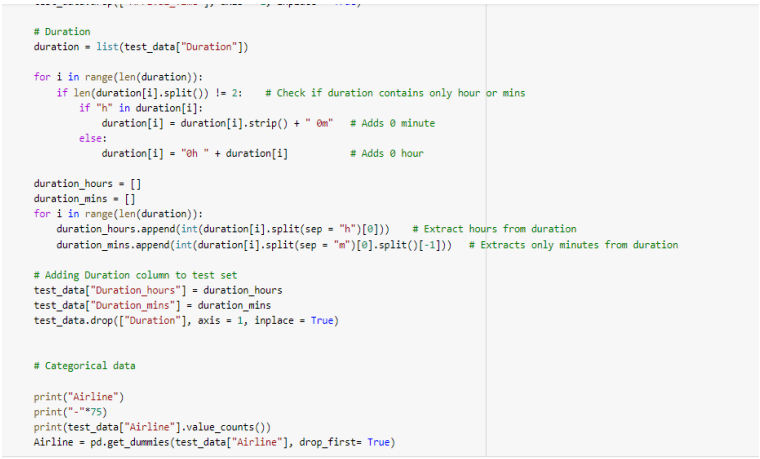
| | Airline | Date_of_Journey | Source | Destination | Route | Dep_Time | Arrival_Time | Duration | Total_Stops | Additional_Info | Price |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | IndiGo | 24/03/2019 | Banglore | New Delhi | BLR → DEL | 22:20 | 01:10 22 Mar | 2h 50m | non-stop | No info | 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 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 10678 | Air Asia | 9/04/2019 | Kolkata | Banglore | CCU → BLR | 19:55 | 22:25 | 2h 30m | non-stop | No info | 4107 |
| 10679 | Air India | 27/04/2019 | Kolkata | Banglore | CCU → BLR | 20:45 | 23:20 | 2h 35m | non-stop | No info | 4145 |
| 10680 | Jet Airways | 27/04/2019 | Banglore | Delhi | BLR → DEL | 08:20 | 11:20 | 3h | non-stop | No info | 7229 |
| 10681 | Vistara | 01/03/2019 | Banglore | New Delhi | BLR → DEL | 11:30 | 14:10 | 2h 40m | non-stop | No info | 12648 |
| 10682 | Air India | 9/05/2019 | Delhi | Cochin | DEL → GOI → BOM → COK | 10:55 | 19:15 | 8h 20m | 2 stops | No info | 11753 |

10683 rows × 11 columns

Figure 7.1: Train Data

```
[ ] # Preprocessing

    print("Test data Info")
    print("-"*75)
    print(test_data.info())

    print()
    print()

    print("Null values :")
    print("-"*75)
    test_data.dropna(inplace = True)
    print(test_data.isnull().sum())

    # EDA

    # Date_of_Journey
    test_data["Journey_day"] = pd.to_datetime(test_data.Date_of_Journey, format="%d/%m/%Y").dt.day
    test_data["Journey_month"] = pd.to_datetime(test_data["Date_of_Journey"], format = "%d/%m/%Y").dt.month
    test_data.drop(["Date_of_Journey"], axis = 1, inplace = True)

    # Dep_Time
    test_data["Dep_hour"] = pd.to_datetime(test_data["Dep_Time"]).dt.hour
    test_data["Dep_min"] = pd.to_datetime(test_data["Dep_Time"]).dt.minute
    test_data.drop(["Dep_Time"], axis = 1, inplace = True)

    # Arrival_Time
    test_data["Arrival_hour"] = pd.to_datetime(test_data.Arrival_Time).dt.hour
    test_data["Arrival_min"] = pd.to_datetime(test_data.Arrival_Time).dt.minute
    test_data.drop(["Arrival_Time"], axis = 1, inplace = True)
```

Figure 7.2: Data Preprocessing



```
# Duration
duration = list(test_data["Duration"])

for i in range(len(duration)):
    if len(duration[i].split()) != 2:    # Check if duration contains only hour or mins
        if "h" in duration[i]:
            duration[i] = duration[i].strip() + " 0m"   # Adds 0 minute
        else:
            duration[i] = "0h " + duration[i]           # Adds 0 hour

duration_hours = []
duration_mins = []
for i in range(len(duration)):
    duration_hours.append(int(duration[i].split(sep = "h")[0]))    # Extract hours from duration
    duration_mins.append(int(duration[i].split(sep = "m")[0].split()[-1]))   # Extracts only minutes from duration

# Adding Duration column to test set
test_data["Duration_hours"] = duration_hours
test_data["Duration_mins"] = duration_mins
test_data.drop(["Duration"], axis = 1, inplace = True)


# Categorical data

print("Airline")
print("-"*75)
print(test_data["Airline"].value_counts())
Airline = pd.get_dummies(test_data["Airline"], drop_first= True)
```

Figure 7.3: Data Preprocessing

```
print("Source")
print("-"*75)
print(test_data["Source"].value_counts())
Source = pd.get_dummies(test_data["Source"], drop_first= False)

print()

print("Destination")
print("-"*75)
print(test_data["Destination"].value_counts())
Destination = pd.get_dummies(test_data["Destination"], drop_first = False)

# Additional_Info contains almost 80% no_info
# Route and Total_Stops are related to each other
test_data.drop(["Route", "Additional_Info"], axis = 1, inplace = True)

# Replacing Total_Stops
test_data.replace({"non-stop": 0, "1 stop": 1, "2 stops": 2, "3 stops": 3, "4 stops": 4}, inplace = True)

# Concatenate dataframe --> test_data + Airline + Source + Destination
data_test = pd.concat([test_data, Airline, Source, Destination], axis = 1)

data_test.drop(["Airline", "Source", "Destination"], axis = 1, inplace = True)

print()
print()

print("Shape of test data : ", data_test.shape)
```

Figure 7.4: Data Preprocessing

```
[59] # Finds correlation between Independent and dependent attributes


    plt.figure(figsize = (18,18))
    sns.heatmap(train_data.corr(), annot = True, cmap = "RdYlGn")

    plt.show()
```

Figure 7.5: Correlation between Independent and Dependent variable

```
[60] # Important feature using ExtraTreesRegressor

    from sklearn.ensemble import ExtraTreesRegressor
    selection = ExtraTreesRegressor()
    selection.fit(X, y)

    ExtraTreesRegressor()
```

Figure 7.6: ExtraTreeRegression

```
[62] #plot graph of feature importances for better visualization

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

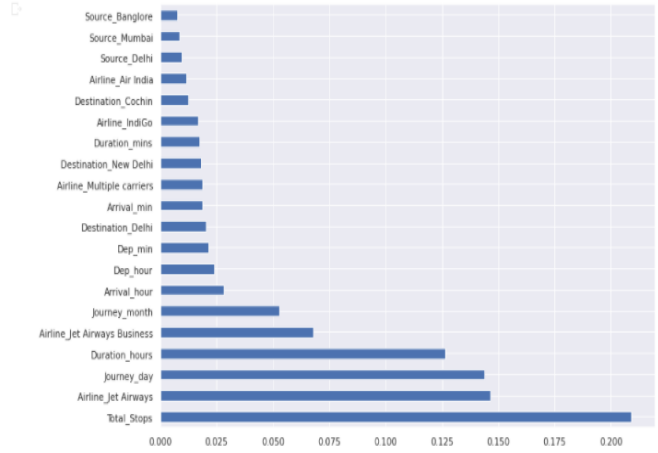



Figure 7.7: Feature importance graph

```
[63] from sklearn.model_selection import train_test_split
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)
```

```
[64] from sklearn.ensemble import RandomForestRegressor
     reg_rf = RandomForestRegressor()
     reg_rf.fit(X_train, y_train)

     RandomForestRegressor()
```

Figure 7.8: Splitting and Fitting data

```
[68] #performance evaluation
     #computing the accuracy of the model performance
     acc_train_reg_rf = reg_rf.score(X_train, y_train)
     acc_test_reg_rf = reg_rf.score(X_test, y_test)
     #computing root mean squared error (RMSE)
     rmse_train_reg_rf = np.sqrt(mean_squared_error(y_train, y_train))
     rmse_test_reg_rf = np.sqrt(mean_squared_error(y_test, y_test))

     print("Random Forest Regression: Accuracy on training Data: {:.3f}".format(acc_train_reg_rf))
     print("Random Forest Regression: Accuracy on test Data: {:.3f}".format(acc_test_reg_rf))
     print('\nRandom Forest Regression: The RMSE of the training set is: ', rmse_train_reg_rf)
     print('Random Forest Regression: The RMSE of the testing set is: ', rmse_test_reg_rf)

     Random Forest Regression: Accuracy on training Data: 0.953
     Random Forest Regression: Accuracy on test Data: 0.797

     Random Forest Regression: The RMSE of the training set is:  0.0
     Random Forest Regression: The RMSE of the testing set is:  0.0
```

Figure 7.9: Random forest Regression

```
[81] plt.figure(figsize = (8,8))
     sns.distplot(y_test-prediction)
     plt.show()

     /usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated funct
       warnings.warn(msg, FutureWarning)
```

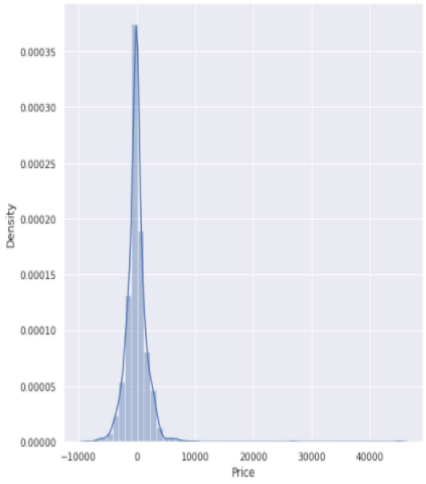



Figure 7.10: Prediction graph

```
[86] #performance evaluation
     #computing the accuracy of the model performance
     acc_train_xgb = xgb.score(X_train, y_train)
     acc_test_xgb = xgb.score(X_test, y_test)

     #computing root mean squared error (RMSE)
     rmse_train_xgb = np.sqrt(mean_squared_error(y_train, y_train_xgb))
     rmse_test_xgb = np.sqrt(mean_squared_error(y_test, y_test_xgb))

     print("XGBoost Regression: Accuracy on training Data: {:.3f}".format(acc_train_xgb))
     print("XGBoost Regression: Accuracy on test Data: {:.3f}".format(acc_test_xgb))
     print('\nXGBoost Regression: The RMSE of the training set is: ', rmse_train_xgb)
     print('XGBoost Regression: The RMSE of the testing set is: ', rmse_test_xgb)

     XGBoost Regression: Accuracy on training Data: 0.860
     XGBoost Regression: Accuracy on test Data: 0.836

     XGBoost Regression: The RMSE of the training set is:  1723.0172992768862
     XGBoost Regression: The RMSE of the testing set is:  1879.7061331550199
```

Figure 7.11: XGBoost Regression

```
[90] #computing the accuracy of the model performance
     acc_train_br = br.score(X_train, y_train)
     acc_test_br = br.score(X_test, y_test)

     #computing root mean squared error (RMSE)
     rmse_train_br = np.sqrt(mean_squared_error(y_train, y_train_br))
     rmse_test_br = np.sqrt(mean_squared_error(y_test, y_test_br))

     print("Bagging Regression: Accuracy on training Data: {:.3f}".format(acc_train_br))
     print("Bagging Regression: Accuracy on test Data: {:.3f}".format(acc_test_br))
     print('\nBagging Regression: The RMSE of the training set is: ', rmse_train_br)
     print('Bagging Regression: The RMSE of the testing set is: ', rmse_test_br)

     Bagging Regression: Accuracy on training Data: 0.886
     Bagging Regression: Accuracy on test Data: 0.704

     Bagging Regression: The RMSE of the training set is:  1553.303725248185
     Bagging Regression: The RMSE of the testing set is:  2524.8354197321223
```

Figure 7.12: Bagging Regression

# Chapter 8

# SCREEN SHOTS

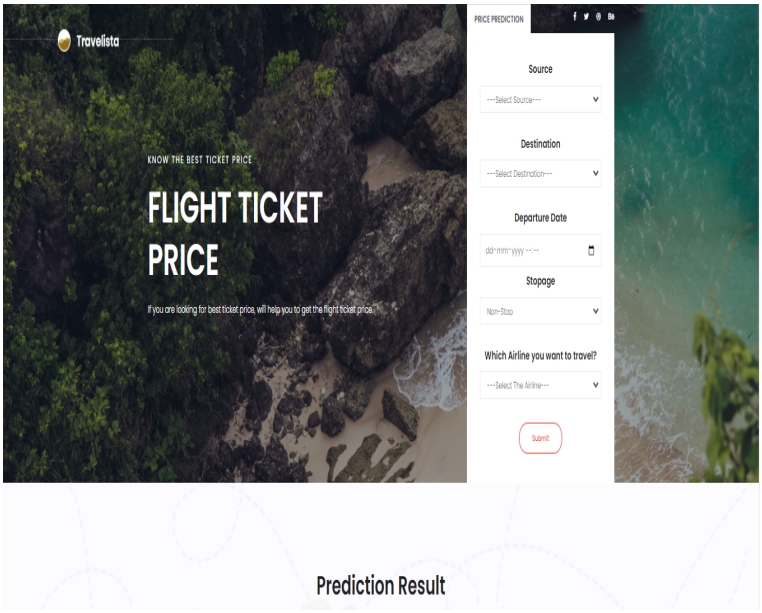



Figure 8.1: Main Page

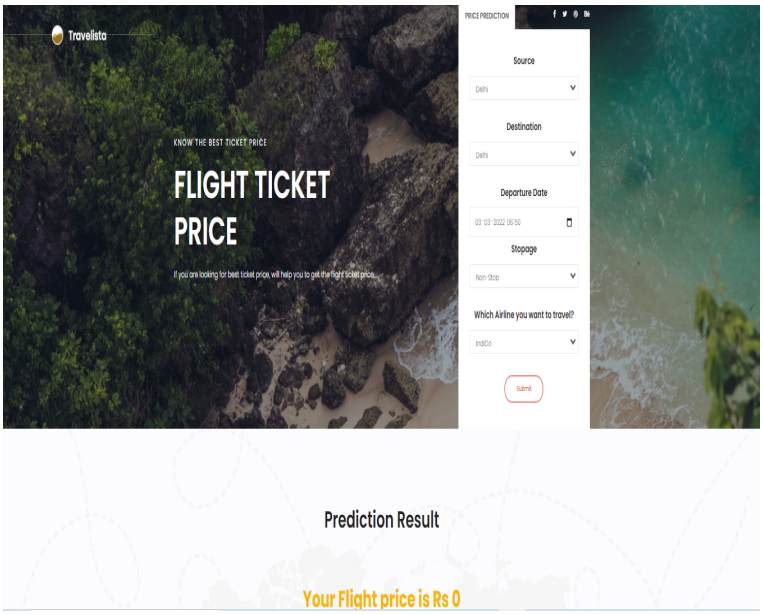



Figure 8.2: Prediction Page

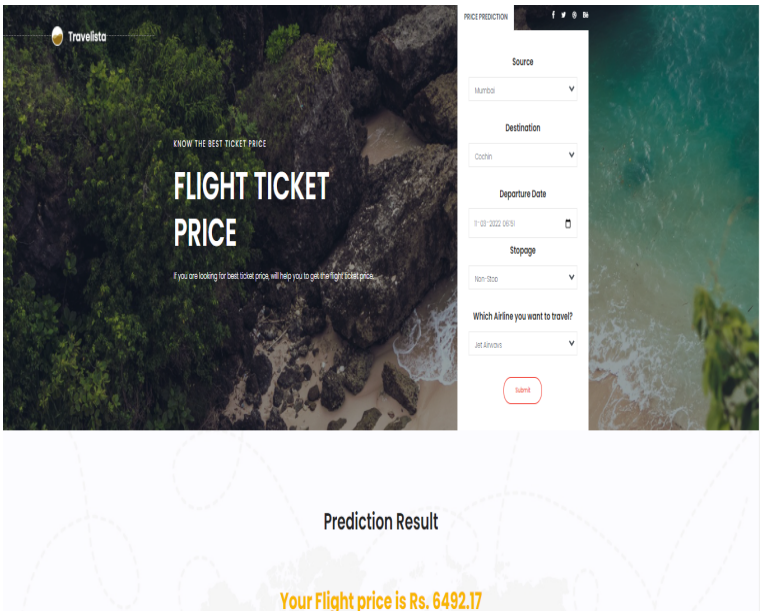



Figure 8.3: Prediction Page

# Chapter 9

# REFERENCES


- www.youtube.com
- www.wikipedia.com

1 http://www.ijstr.org/final-print/dec2019/Predicting-The-Price-Of-A-Flight-Ticket-With-The-Use-Of-Machine-Learning-Algorithms.pdf

[2] B. Smith, J. Leimkuhler, R. Darrow, and Samuels,—Yield management at American airlines, Interfaces, vol.22, pp. 8–31,1992.

[3] https://www.kaggle.com/nikhilmittal/flight-fare-prediction-mh