FLIGHT FARE PREDICTION SYSTEM

A Project Report Submitted
In Partial Fulfillment
for award of Bachelor of Technology

in
B.Tech (CS)

by

Rahul Kumar Giri (2201330120093)
Priyansh Kisan (2201330120088)
Ayush Tayal (2201330120036)

Under the Supervision of
Prof. Dr. Deepak Uprety
Professor,Computer Science

COMPUTER SCIENCE
SCHOOL OF COMPUTER SCIENCE AND EMERGING TECHNOLOGIES
NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY, GREATER NOIDA
(An Autonomous Institute)
Affiliated to
DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW

May,2024

DECLARATION

We hereby declare that the work presented in this report entitled "**FLIGHT FARE PREDICTION SYSTEM**", was carried out by us. We have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. We have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. We have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

We affirm that no portion of our work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, we shall be fully responsible and answerable.

Name : Rahul Kumar Giri
Roll Number : 2201330120093

Name : Priyansh Kisan
Roll Number : 2201330120088

Name : Ayush Tayal
Roll Number : 2201330120036

CERTIFICATE

Certified that **Rahul Kumar Giri , Priyansh Kisan and Ayush Tayal** (enrolment numbers: 2201330120088) have carried out the research work presented in this Project Report entitled "**Flight Fare Prediction System"** for the award of **Bachelor of Technology**, **Computer Science** from Dr. APJ Abdul Kalam Technical University, Lucknow under our supervision. The Project Report embodies results of original work, and studies are carried out by the students herself/himself. The contents of the Project Report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Signature

Dr. Deepak Uprety

Professor
Computer Science
NIET Greater Noida

Date: 11/05/2024

ACKNOWLEDGEMENT

We would like to express my gratitude towards Guide name and Co guide name for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

Our thanks and appreciations to respected HOD for their motivation and support throughout.

ABSTRACT

The Flight Fare Prediction System offers a comprehensive solution to the challenge of accurately forecasting flight ticket prices amidst the dynamic landscape of the airline industry. With the industry's continuous expansion and evolving fare structures, predicting flight fares has become increasingly complex. This system harnesses the power of machine learning algorithms and extensive historical flight data to deliver precise fare predictions. Drawing from a vast dataset encompassing various factors such as travel dates, destinations, airlines, departure times, and other pertinent variables, the system employs advanced machine learning techniques to discern patterns and relationships, thereby enabling reliable predictions of future flight fares. Utilizing a blend of regression algorithms and ensemble methods, the Flight Fare Prediction System ensures high accuracy in its predictions. It takes into account a multitude of factors influencing ticket prices, including seasonality, market demand, fuel costs, competition, and other dynamic variables. By integrating real-time data updates, the system ensures that predictions remain current and reflective of the latest market trends.

TABLE OF CONTENTS

(Signature of the Candidate)

Name: ……………………..

Enrollment No:…………….

(Signature of the Candidate)

Name: ……………………..

Enrollment No:…………….

(Signature of the Candidate)

Name: ……………………..

Enrollment No:…………….

(Signature of the Candidate)

Name: ……………………..

Enrollment No:…………….

CHAPTER 1

INTRODUCTION

DOMAIN INTRODUCTION

Machine learning is an area of artificial intelligence (AI) that focuses oncreating statistical models and algorithms that let computers learn from data without beingexplicitly explained and make predictions. It contains a variety of methods and methodsthat let computers spot trends, understand situations better, and get better over time.Many industries, including finance, healthcare, marketing, transportation, andtechnology, among others, now depend heavily on machine learning. It has the ability toalter how companies and organizations run their operations as well as how we usetechnology in our daily lives.

There are many different types of machine learning, such as supervised, semi-supervised, unsupervised, and reinforcement learning. Under supervision, an algorithm is trained on correctly extracted, labelled data and then learns predictions based on tagged cases. In contrast, unsupervised learning involves training an algorithm on unlabeled data, with the goal of the program being to find patterns or links in the unlabeled data. Machine learning is widely applied to a variety of tasks, such as recommendation systems, natural language processing, fraud detection, anomaly detection, predictive analytics, and photo and speech recognition. To do this, data must be gathered and prepared, models must be developed and tested, and models must be successful.

However, there are moral questions raised by machine learning regarding data in algorithms, privacy, and the impact on society and industry. It is essential to apply machine learning responsibly and ethically in order to guarantee that the benefits of this technology are realized while minimizing potential risks.

To sum up, machine learning is a rapidly developing field that is changing business and the course of technology. This allows computers to learn from data and make predictions or decisions, which could change the way people interact with technology in many areas of our lives.

PROJECT INTRODUCTION

The Flight Fare Prediction System represents a machine learning initiative dedicated to estimating aircraft ticket costs using relevant features and historical data. This approach serves travellers, travel firms, and airlines, aiding them in projecting trip expenses for planning, budgeting, and making informed choices.

The project's objective lies in constructing a dependable machine-learning model for forecasting flight expenses by encompassing various factors such as travel class, airline, departure and arrival destinations, travel dates, and other relevant details. To train the algorithm effectively, an extensive dataset comprising historical flight data, including ticket pricing and related attributes, will be utilized.

Users of the Flight Fare Prediction System will gain access to a user-friendly interface where they can input their travel information and receive an estimated flight fare. The system will meticulously analyse input data and generate precise predictions through feature engineering, data preprocessing, and machine learning methodologies. Rigorous evaluation criteria will be applied to ensure the accuracy and reliability of the model and its associated attributes.

While the accuracy of predictions heavily relies on the quality of the training and prediction data, the project places a premium on data quality and integrity. Data preprocessing techniques such as data cleaning, handling missing values, and feature scaling will be employed to validate the legitimacy and trustworthiness of the data utilized for training and prediction purposes.

The Flight Fare Prediction System holds the potential to assist consumers in planning their travel budgets, aid travel agencies in offering competitive pricing to their clientele, and support airlines in devising effective pricing strategies and revenue management tactics. By leveraging machine learning, the system aims to offer valuable insights and advantages to the travel industry, precisely estimating airline fares and enhancing decision-making processes.

The project's ultimate aim is to establish a dependable and accurate flight fare prediction system that delivers flight rates based on key parameters. Evaluation metrics such as forecast accuracy, model performance, and usability will be leveraged to assess systems thoroughly. Ethical considerations, including the handling of personal data and ensuring fairness in assumptions, are carefully incorporated into the project framework.

In conclusion, the Flight Fare Prediction System signifies a machine learning endeavor aimed at developing a system capable of accurately predicting trip expenses based on historical data and relevant attributes. This technology harbors the potential to enhance decision-making within the travel industry and offer valuable insights and benefits to travelers, travel agencies, and airlines alike.

PROBLEM STATEMENT

Everyone knows that holidays always call for a much-needed vacation and planning the travel itinerary becomes a time-consuming task. The commercial aviation business has grown tremendously and has become a regulated marketplace as a result of the worldwide growth of the Internet and E-commerce. Hence, for Airline revenue management, different strategies like customer profiling, financial marketing, and social factors are used for setting ticket fairs. When tickets are booked months in advance, airfares are often reasonable, but when tickets are booked in a hurry, they are often higher. But, the number of days/hours until departure isn’t the only factor that decides flight fare, there are numerous other factors as well. Customers find it quite difficult to obtain a perfect and lowest ticket deal due to the aviation industry's complex pricing methodology. This project aims to address the need for a dependable and accurate flight fare prediction system that provides travellers, travel agencies, and airlines with reliable estimates of flight fares.

CHAPTER 2

LITERATURE SURVEY

1. "Airline Ticket Price Prediction: A Machine Learning Approach" by M. L. Ahirrao, et al. (2018): This research paper proposes a flight fare prediction model using machine learning techniques such as regression algorithms and time-series analysis. The study explores various factors influencing ticket prices and compares the performance of different algorithms in predicting fare trends.

2. Flight Fare Prediction using Historical Data and Machine Learning Techniques" by A. Kumar, et al. (2019): The paper presents a flight fare prediction system that combines historical flight data and machine learning algorithms to forecast ticket prices. It analyzes factors such as departure time, travel duration, and airline popularity to generate accurate fare predictions. The study compares the performance of different algorithms and discusses the potential for improving prediction accuracy.

3. "Airline Fare Prediction Using Machine Learning" by A. L. Rodrigues, et al. (2020): This work focuses on predicting airline fares using machine learning techniques. The study considers various parameters, including airline popularity, route distance, and historical fare data, to train a predictive model. The authors explore the performance of different algorithms and discuss the implications of their findings for fare prediction accuracy.

4. "Predicting Airfare Using Machine Learning Techniques" by S. Aruna, et al. (2020): The paper presents a comparative analysis of different machine learning algorithms for predicting airfare. The study considers factors such as seasonality, time of booking, and flight class to develop a prediction model. The authors evaluate the performance of regression algorithms, including linear regression, support vector regression, and random forest regression.

5. "Flight Fare Prediction Using Ensemble Learning Techniques" by M.Sharma, et al. (2021): This research focuses on the application of ensemble learning techniques for flight fare prediction. The study combines multiple machine learning models, including

decision trees, random forests, and gradient boosting, to improve prediction accuracy. The authors compare the performance of individual models and ensemble methods to identify the most effective approach.

6. "Flight Fare Prediction using Machine Learning Techniques" by K. Kumar and team (2017). This study compares the performance of various machine learning techniques, including decision trees, support vector machines, k-nearest neighbours, and random forests for flight fare prediction. This study also employs features engineering techniques to extract relevant features from flight data and evaluates the models using metrics such as mean squared error (MSE) and R-squared.

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

• A software programme called a "flight fare prediction system" forecasts the cost of airline tickets using machine learning methods. This method may forecast future pricing for a specific route or destination by examining historical data, current market patterns, and other pertinent criteria.

• A flight fare prediction system's main objective is to aid travellers in making more informed travel plans by giving them precise and trustworthy information about the price of flying. With the help of this method, travellers can decide with certainty when to book their flights, which airlines to pick, and which routes to take in order to get the greatest deal.

• A flight fare prediction system can be helpful for travel agencies, airlines, and other travel-related organisations in addition to benefiting individual travellers. These companies may improve their pricing strategies, boost income, and better serve the demands of their clients by giving them information into price trends and patterns.

• A flight pricing prediction system is, all things considered, a useful tool for anyone trying to cut costs on air travel or enhance their commercial operations in the travel sector.

3.1 CHARACTERISTICS

The following are some of the main components of the airfare forecasting system:

Machine Learning Algorithms: The system makes precise projections of future flight rates by analyzing historical data and current market patterns using machine learning algorithms.

Information Aggregation: In order to give comprehensive information regarding flight prices, the system gathers data from a variety of sources, including airline websites, travel agents, and other internet sources.

Real-time updates: To make sure that forecast prices are as current as possible, the system offers real-time updates.

Multiple Airlines and Routes: The system is able to estimate costs for a number of airlines and routes, giving customers a wide range of options.

User-friendly interface: The system's user-friendly interface makes it simple and quick for users to enter their travel information and obtain estimates.

Accuracy: Based on historical data, market trends, and other important variables, the system is intended to offer accurate projections.

Customizations: The system can be modified to match the unique requirements of particular travelers or associated businesses.

Mobility: The system is mobile-friendly, making it simple for travelers to access price estimates while on the go.

WHERE IT IS USED?

Several situations call for the use of a flight fare prediction system, including:

Travel Agencies: Travel agencies can utilize airline fare prediction systems to accurately advise their clients about airfares and assist them in making decisions about booking tickets.

Airlines: By offering customers competitive tickets, airlines can use flight fare prediction systems to optimize their pricing strategies, boost revenue, and enhance customer happiness.

Online travel agencies: Online travel agencies can utilize flight fare prediction algorithms to give clients real-time pricing details and assist them in locating the cheapest flight offers.

Travel Management Companies: Flight pricing prediction algorithms can be used by travel management organizations to assist their clients in lowering travel expenses and enhancing adherence to travel regulations.

Individual Travelers: To obtain the greatest airfare prices and make their trip preparations more efficient, individual travelers can use flight fare prediction tools.

In general, everyone trying to cut costs on air travel or enhance their company operations in the travel industry uses aircraft fare prediction tools.

SYSTEM DESIGN

ARCHITECTURAL DIAGRAM

An architectural diagram is a graphic representation of a set of concepts that are part of architecture, including principles, elements, and components. Application architecture diagram, system architecture diagram, application architecture diagram, security architecture diagram etc. There are many types of architecture diagrams such as System architecture, or system architecture, is a conceptual model that describes structure, behavior, and more. system. An architecture statement is a formal description and description of a system that is constructed in a way that supports reasoning about the system's structure and behavior.

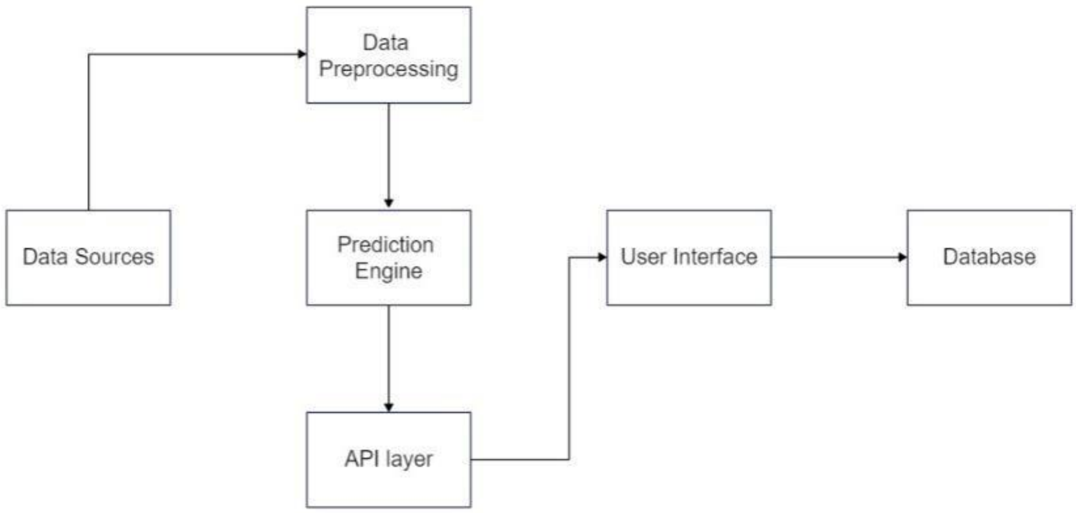


Fig 1. Architecture Diagram

UML DIAGRAM

Unified Modeling Language (UML) is a general purpose, development, modeling language in the field of software engineering that aims to provide a standardized way to visualize system design.

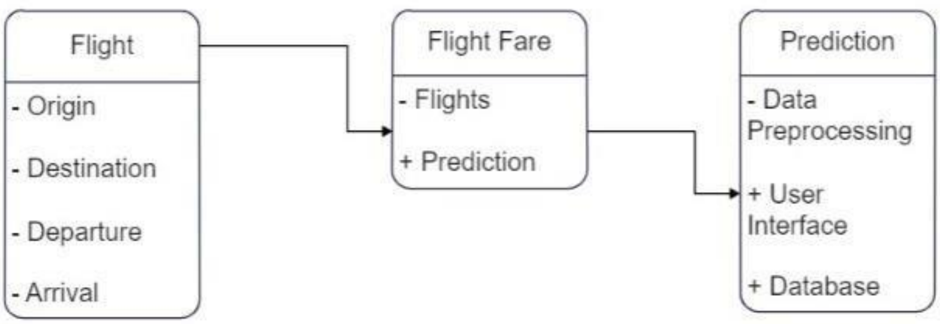


Fig. 2 UML Diagram

CHAPTER 4

SYSTEM SPECIFICATIONS

A system specification is a structured collection of information that contains system requirements. A system specification describes the functional and non-functional requirements embedded in a system element (system, enabling system, or segment). Requirements for developing system specifications will be derived from high-level system element specifications or general system specifications.

REQUIREMENT ANALYSIS

Requirements analysis is the process of analysing, documenting, validating, and managing software in systems engineering and software engineering while taking into account the potentially conflicting requirements of various stakeholders in order to satisfy the needs or requirements of a new or modified product or project. concentrate on the issue that created the circumstance. system specifications.

A system or software project's ability to succeed or fail depends on the results of the requirements analysis. Requirements must be specified in detail enough for system design, be documented, actionable, quantifiable, tested, and tied to clearly defined business needs or capabilities.

HARDWARE SPECIFICATIONS

Hardware Requirements The most common requirements defined by any operating system or software are physical computer resources, also known as hardware. The Hardware Requirements List is often accompanied by a Hardware Compatibility List (HCL) in the case of an operating system.

- Processor-Intel
- Ram-4GB
- Hard disk-260GB
- Keyboard
- Mouse

SOFTWARE SPECIFICATIONS

Software requirements are concerned with determining the software requirements and conditions that must be installed on the computer in order for the software to function properly. These terms or conditions are usually not included in the software installation package and must be installed separately before the software is installed.

Specifications facilitate the systematic and organized storage of requirements knowledge and effective communication and change management. Use cases, user stories, functional requirements, and visual analysis models are popular choices for defining requirements.

- HTML
- CSS
- Python

HTML

HTML stands for Hyper Text Markup Language. A standard markup language for creating web pages. It allows you to create and structure sections, paragraphs, and links using HTML elements (web page structural elements) such as tags and attributes.

HTML has many use cases, namely:

Web development - Developers use HTML code to design how browsers display web page elements such as text, hyperlinks, and media files.

Internet Navigation - Users can easily navigate and link between related pages and websites because HTML is widely used to display hyperlinks.

Web document - HTML allows you to organize and format documents similar to Microsoft Word.

The fact that HTML cannot develop dynamic functionality means that it is not regarded as a programming language. It is now regarded as the accepted web standard. The HTML specification is updated frequently by the World Wide Web Consortium (W3C).

CSS

CSS stands for Cascading Style Sheets. It is a language style sheet used to describe the look and feel of a document written in a markup language. It provides additional features for HTML. It is commonly used with HTML to style web pages and user interfaces. It also works with any XML document, including plain XML, SVG, and XUL.

CSS is used in most web applications along with HTML and JavaScript to create user interfaces for web pages and user interfaces for many mobile applications.

What does CSS do?

• You can add new views to your old HTML documents.

• With just a few changes to the CSS code, you can change the look of your website.

Why use CSS?

There are three major benefits of CSS and are as followed,

1. Solves big problems - Before CSS, specifications such as font, color, background style, element alignment, border, and size had to be repeated on every web page. It's a long process. For example: If you are developing a large website where font and color information is added to each page, this will be a long and expensive process. CSS was designed to solve this problem. It is a W3C recommendation.

2. Saves a lot of time - CSS style definitions are stored in external CSS files, so the entire web page can be changed by changing just one file.

3. Provides more attributes - CSS provides more detailed properties than simple HTML to define the look and feel of a web page.

PYTHON

Python, a high-level, interpreted programming language, was first available in 1991. Developed by Guido van Rossum, it emphasizes code simplicity and readability and is loved by both novice and experienced programmers. Python's straightforward syntax simplifies learning and allows developers to write code quickly and efficiently. It is also famous for having pre-built modules and a large standard library that can be used to perform various tasks.

Features of Python:

• Python has a straightforward syntax that is simple to learn even for complete beginners.

• Python programming is cross-platform, meaning it can be used with Linux, Mac OS X, and Windows.

• Python supports object-oriented programming, allowing programmers to create reusable, modular programs.

• Python does not require compilation before execution; instead, each line of the code is read and then executed. Testing and debugging code is now simpler and quicker as a result.

• Python includes a sizable standard library that gives programmers access to a variety of modules and functions.

• Python is dynamically typed, which enables variables to hold values of any data type and allows for runtime type changes.

CHAPTER 5

MODULE EXPLANATION

5.1 LIST OF MODULES

Creating an index page 5.1.2 Training the data 5.1.3 Obtaining the pickle file 5.1.4 Building the final application

5.1.1 CREATING AN INDEX PAGE

We will be creating an index webpage using HTML and CSS since we're creating a web application.

5.1.2 TRAINING THE DATA

After creating a webpage to run the web application now we will be focusing on training the existing data to train our model. We'll be using sever machine learning algorithms like Decision Tree Regressor, Random Forest, and Linear Regression algorithm.

5.1.3 OBTAINING THE PICKLE FILE

After training the existing data with machine learning algorithms, we should write a set of codes for the trained model to save to a pickle file so that we could easily import and use it in the main application.

5.1.4 BUILDING THE FINAL APPLICATION

As we have created an index home page using HTML and CSS and trained the existing data using machine learning algorithms and obtained the trained model as a pickle file, now we must build our main application to import and use the trained data and to run the whole application on the web.

5.2 USER INTERFACE MODULE

As we only focused on making the model work proficiently, we didn't give importance to the user experience even though we used HTML and CSS we just used them to a certain required prototype point. But as a future enhancement, we can work on this project by improving the user interface and by adding several other features.

5.3 OUTPUT MODULE

Finally, our proposed system is successfully executed by using machine learning algorithms and our result came out with 99% accuracy.

CHAPTER 6

RESULTS AND DISCUSSION

Flight Fare Prediction system is something for travelers who are looking to cut costs on their airfare, flight fare prediction systems can be a useful tool. When making a flight reservation, it's crucial to take into account a variety of aspects, including the timing of the booking, the airline's track record for on-time arrivals, and the overall convenience of the flight schedule.

The accuracy of such systems can vary depending on the complexity of the algorithms used, the quality of the data, and the specific factors being considered. Some systems may be more accurate than others, and it's important to keep in mind that predictions are not always guaranteed to be accurate.

CHAPTER 7

CONCLUSION

In conclusion, flight fare prediction systems can be a useful tool for travelers to estimate the cost of a particular flight, based on various factors such as time of year, destination, and airline. These systems use historical data and algorithms to generate predictions that can help travelers plan their trips and potentially save money on airfare.

However, it's important to keep in mind that the accuracy of these predictions can vary depending on the quality of the data and algorithms used. Therefore, it's always a good idea to compare multiple prediction systems and also consider other factors when booking a flight, such as the reputation of the airline, the convenience of the flight schedule, and the timing of the booking.

Overall, flight fare prediction systems can be a valuable tool in the travel industry, providing travelers with useful information to make informed decisions about their trips.

DATA TRAINING FILE

```python
#importing the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
pd.set_option('display.max_columns', None)
```
Python

```python
#data is in excel format so, read data as 'read_excel'
train = pd.read_excel('Data_Train.xlsx')
train.head()
```
Python

```python
import pandas as pd
test = pd.read_excel('Test_set.xlsx')
test.head()
```
Python

```python
print('Training dataset shape:', train.shape)
print('Test dataset shape:', test.shape)
```
Python

```python
train.isnull().sum()
```
Python

```python
train[train['Total_Stops'].isnull()]
```
Python

```python
train[train['Route'].isnull()]
```
Python

```python
train=train.dropna(axis=0, how='any')
#since, there is only one missing value in Total_Stops and Route and both coincidentally are from same record, we can just drop that record/row
```
Python

```python
train.shape
```
Python

```python
train.dtypes
```
Python

```python
test.isnull().sum()
```
Python

```python
#First we consider Duration column
train["Duration"].value_counts()
```
Python

```python
# Converting 'Duration' column into a list
duration_train = list(train["Duration"])
duration_train
```
Python

```python
#apply loop to separate hours from minutes
#.split() method splits a string into a list
#also, hour is represented by 'h' and minute by'm'
#.strip() method returns a copy of the string by removing both the leading and the trailing characters

for i in range(len(duration_train)):
    if len(duration_train[i].split()) != 2:
        if "h" in duration_train[i]:
            duration_train[i] = duration_train[i].strip() + ' 0m'  # add 0 minute
        else:
            duration_train[i] = '0h '+ duration_train[i]           # add 0 hour
```
Python

```python
duration_train
```
Python

```python
duration_hours = []
duration_mins = []
for i in range(len(duration_train)):
    duration_hours.append(int(duration_train[i].split(sep = "h")[0]))
    duration_mins.append(int(duration_train[i].split(sep = "m")[0].split()[-1]))
```
Python

```python
train['Duration_hrs'] = duration_hours
train['Duration_hrs']
```
Python

```python
train['Duration_mins'] = duration_mins
train['Duration_mins']
```
Python

```python
train.drop('Duration', axis=1, inplace=True)
```
Python

```python
#first we consider 'Date_of_Journey'
train['Day_of_Journey']=pd.to_datetime(train['Date_of_Journey'], format='%d/%m/%Y').dt.day
train['Month_of_Journey']=pd.to_datetime(train['Date_of_Journey'], format='%d/%m/%Y').dt.month
train.drop('Date_of_Journey', axis = 1, inplace = True)
```
Python

```python
# Now, we need to take care of Dep_Time
train['Dep_hr'] = pd.to_datetime(train['Dep_Time']).dt.hour
train['Dep_min'] = pd.to_datetime(train['Dep_Time']).dt.minute
train.drop('Dep_Time', axis = 1, inplace = True)
```
Python

```python
#Now, we take care of Arrival_Time
train['Arrival_hr'] = pd.to_datetime(train['Arrival_Time']).dt.hour
train['Arrival_min'] = pd.to_datetime(train['Arrival_Time']).dt.minute
train.drop('Arrival_Time', axis = 1, inplace = True)
```
Python

```python
print('Train dataset shape:', train.shape)
```
Python

```python
train.head()
```
Python

```python
#Taking care of Airline column
print(train['Airline'].unique())
#print(train['Airline'].nunique())
print(train['Airline'].value_counts())
sns.catplot(y = 'Price', x = 'Airline', data = train.sort_values('Price', ascending = False),
            kind='box', height = 4, aspect = 3, orient='v')
plt.show()
```
Python

```python
print(train.shape)
train.head()
```

Python

```python
#select categorical variables from then dataset, and then implement categorical encoding for nominal variables
Airline=train[['Airline']]
Airline=pd.get_dummies(Airline, drop_first=True)
Airline.head()
```

Python

```python
Source=train[['Source']]
Source=pd.get_dummies(Source, drop_first= True)
Source.head()
```

Python

```python
Destination=train[['Destination']]
Destination=pd.get_dummies(Destination, drop_first= True)
Destination.head()
```

Python

```python
# Concatenate dataset with Airline, Source, Destination, Additional_Info

train = pd.concat([train, Airline, Source, Destination], axis = 1)

#Dropping the non-encoded Airline, Source, Destination variables
train.drop(['Airline', 'Source', 'Destination', 'Additional_Info', 'Route'], axis = 1, inplace = True)
#dropping route column as we have a stop column which basically covers the entire zest of it
```

Python

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

Python

```python
print(train.shape)
train.head()
```

Python

```python
#First we consider Duration column
test["Duration"].value_counts()
```

Python

```python
# Converting 'Duration' column into a list
duration_test = list(test["Duration"])
duration_test
```

Python

```python
#apply loop to separate hours from minutes
#.split() method splits a string into a list
#also, hour is represented by 'h' and minute by'm'
#.strip() method returns a copy of the string by removing both the leading and the trailing characters
for i in range(len(duration_test)):
    if len(duration_test[i].split()) != 2:
        if "h" in duration_test[i]:
            duration_test[i] = duration_test[i].strip() + ' 0m'  # add 0 minute
        else:
            duration_test[i] = '0h '+ duration_test[i]           # add 0 hour

duration_hours = []
duration_mins = []
for i in range(len(duration_test)):
    duration_hours.append(int(duration_test[i].split(sep = "h")[0]))
    duration_mins.append(int(duration_test[i].split(sep = "m")[0].split()[-1]))
```

Python

```python
duration_test
```
Python

```python
test['Duration_hrs'] = duration_hours
test['Duration_hrs']
```
Python

```python
test['Duration_mins'] = duration_mins
test['Duration_mins']
```
Python

```python
test.drop('Duration', axis=1, inplace=True)
```
Python

```python
test['Day_of_Journey']=pd.to_datetime(test['Date_of_Journey'], format='%d/%m/%Y').dt.day
test['Month_of_Journey']=pd.to_datetime(test['Date_of_Journey'], format='%d/%m/%Y').dt.month
test.drop('Date_of_Journey', axis = 1, inplace = True)

test['Dep_hr'] = pd.to_datetime(test['Dep_Time']).dt.hour
test['Dep_min'] = pd.to_datetime(test['Dep_Time']).dt.minute
test.drop('Dep_Time', axis = 1, inplace = True)

test['Arrival_hr'] = pd.to_datetime(test['Arrival_Time']).dt.hour
test['Arrival_min'] = pd.to_datetime(test['Arrival_Time']).dt.minute
test.drop('Arrival_Time', axis = 1, inplace = True)
```
Python

```python
print('Test dataset shape:', test.shape)
```
Python

```python
#select categorical variables from then dataset, and then implement categorical encoding for nominal variables
Airline=test[['Airline']]
Airline=pd.get_dummies(Airline, drop_first=True)

Source=test[['Source']]
Source=pd.get_dummies(Source, drop_first= True)

Destination=test[['Destination']]
Destination=pd.get_dummies(Destination, drop_first= True)


# Concatenate dataset with Airline, Source, Destination, Additional_Info
test= pd.concat([test, Airline, Source, Destination], axis = 1)

#Dropping the non-encoded Airline, Source, Destination variables
test.drop(['Airline', 'Source', 'Destination', 'Additional_Info', 'Route'], axis = 1, inplace = True)
#dropping route column as we have a stop column which basically covers the entire zest of it


#Let's take care of Total_Stops
test.replace({"non-stop": 0, "1 stop": 1, "2 stops": 2, "3 stops": 3, "4 stops": 4}, inplace = True)

from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
test['Total_Stops'] = encoder.fit_transform(test['Total_Stops'])

print(test.shape)
test.head()
```
Python

```python
train.columns
```
Python

```python
test.columns
```
Python

```python
#train.drop(labels=['Airline_Trujet'], axis=1, inplace=True)
```
Python

```python
train.dtypes
```

Python

```python
price=train.Price
train.drop('Price', axis=1, inplace=True)
train=train.join(price)
train.head()
```

Python

```python
X = train.loc[:, ['Total_Stops', 'Duration_hrs', 'Duration_mins',
       'Day_of_Journey', 'Month_of_Journey', 'Dep_hr', 'Dep_min', 'Arrival_hr',
       'Arrival_min', 'Airline_Air India', 'Airline_GoAir', 'Airline_IndiGo',
       'Airline_Jet Airways', 'Airline_Jet Airways Business',
       'Airline_Multiple carriers',
       'Airline_Multiple carriers Premium economy', 'Airline_SpiceJet',
       'Airline_Trujet', 'Airline_Vistara', 'Airline_Vistara Premium economy',
       'Source_Chennai', 'Source_Delhi', 'Source_Kolkata', 'Source_Mumbai',
       'Destination_Cochin', 'Destination_Delhi', 'Destination_Hyderabad',
       'Destination_Kolkata', 'Destination_New Delhi']]
X.head()
```

Python

```python
y = train.iloc[:, -1]
y
```

Python

```python
# Important feature using ExtraTreesRegressor
from sklearn.ensemble import ExtraTreesRegressor
selection = ExtraTreesRegressor()
selection.fit(X, y)

#bar graph of feature importances
plt.figure(figsize = (10,8))
feat_importances = pd.Series(selection.feature_importances_, index=X.columns)
feat_importances.nlargest(20).plot(kind='barh')
plt.show()
```

Python

```python
from sklearn.model_selection import cross_val_score
from sklearn import metrics
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import RandomizedSearchCV
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import StratifiedKFold
kfold = StratifiedKFold(n_splits=20)
```

Python

```python
#Train Test Split
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=0)
```

Python

```python
#Linear Regression
from sklearn.linear_model import LinearRegression
lin_reg=LinearRegression()
lin_reg.fit(X_train, y_train)
y_pred=lin_reg.predict(X_test)

print("Linear Regression Score on Training set is",lin_reg.score(X_train, y_train))#Training Accuracy
print("Linear Regression Score on Test Set is",lin_reg.score(X_test, y_test))#Testing Accuracy

accuracies = cross_val_score(lin_reg, X_train, y_train, cv = kfold)
print(accuracies)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2f} %".format(accuracies.std()*100))

mae=mean_absolute_error(y_pred, y_test)
print("Mean Absolute Error:" , mae)

mse=mean_squared_error(y_test, y_pred)
print("Mean Squared Error:" , mse)

print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

print('The r2_score is', metrics.r2_score(y_test, y_pred))
```

Python

```python
#Decision Tree Regressor
from sklearn.tree import DecisionTreeRegressor
dt_reg = DecisionTreeRegressor(random_state = 0)
dt_reg.fit(X_train, y_train)
y_pred=dt_reg.predict(X_test)

print("Decision Tree Score on Training set is",dt_reg.score(X_train, y_train))#Training Accuracy
print("Decision Tree Score on Test Set is",dt_reg.score(X_test, y_test))#Testing Accuracy

accuracies = cross_val_score(dt_reg, X_train, y_train, cv = kfold)
print(accuracies)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2f} %".format(accuracies.std()*100))

mae=mean_absolute_error(y_pred, y_test)
print("Mean Absolute Error:" , mae)

mse=mean_squared_error(y_test, y_pred)
print("Mean Squared Error:" , mse)

print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

print('The r2_score is', metrics.r2_score(y_test, y_pred))
```

Python

```python
#Random Forest Regression
from sklearn.ensemble import RandomForestRegressor
rf_reg = RandomForestRegressor(n_estimators=400,min_samples_split=15,min_samples_leaf=2,
max_features='auto', max_depth=30)
rf_reg.fit(X_train, y_train)
y_pred=rf_reg.predict(X_test)

print("Random Forest Score on Training set is",rf_reg.score(X_train, y_train))#Training Accuracy
print("Random Forest Score on Test Set is",rf_reg.score(X_test, y_test))#Testing Accuracy

accuracies = cross_val_score(rf_reg, X_train, y_train, cv = kfold)
print(accuracies)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2f} %".format(accuracies.std()*100))

mae=mean_absolute_error(y_pred, y_test)
print("Mean Absolute Error:" , mae)

mse=mean_squared_error(y_test, y_pred)
print("Mean Squared Error:" , mse)

print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

print('The r2_score is', metrics.r2_score(y_test, y_pred))

sns.distplot(y_test-y_pred)
plt.show()

plt.scatter(y_test, y_pred, alpha = 0.5)
plt.xlabel("y_test")
plt.ylabel("y_pred")
plt.show()
```

Python

```python
import pickle
# dump information to the file
pickle.dump(rf_reg, open('rf_reg.pkl', 'wb'))
model = pickle.load(open('rf_reg.pkl', 'rb'))
```

Python

```python
model.predict([[0,2,50,24,3,22,20,1,10,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1]])
```

Python

```
model = pickle.load(open("rf_reg.pkl", "rb"))



@app.route("/")
@cross_origin()
def home():
    return render_template("home.html")




@app.route("/predict", methods = ["GET", "POST"])
@cross_origin()
def predict():
    if request.method == "POST":

        # Date_of_Journey
        date_dep = request.form["Dep_Time"]
        Day_of_Journey = int(pd.to_datetime(date_dep, format="%Y-%m-%dT%H:%M").day)
        Month_of_Journey = int(pd.to_datetime(date_dep, format ="%Y-%m-
%dT%H:%M").month)
        # print("Journey Date : ",Journey_day, Journey_month)

        # Departure
        Dep_hr = int(pd.to_datetime(date_dep, format ="%Y-%m-%dT%H:%M").hour)
        Dep_min = int(pd.to_datetime(date_dep, format ="%Y-%m-%dT%H:%M").minute)
        # print("Departure : ",Dep_hour, Dep_min)

        # Arrival
        date_arr = request.form["Arrival_Time"]
        Arrival_hr = int(pd.to_datetime(date_arr, format ="%Y-%m-%dT%H:%M").hour)
        Arrival_min = int(pd.to_datetime(date_arr, format ="%Y-%m-%dT%H:%M").minute)
        # print("Arrival : ", Arrival_hour, Arrival_min)

        # Duration
        Duration_hrs = abs(Arrival_hr - Dep_hr)
        Duration_mins = abs(Arrival_min - Dep_min)
        # print("Duration : ", dur_hour, dur_min)

        # Total Stops
        Total_stops = int(request.form["Total_stops"])
        # print(Total_stops)

        # Airline
```

```
        Multiple_carriers = 0
        SpiceJet = 0
        Vistara = 0
        GoAir = 0
        Multiple_carriers_Premium_economy = 0
        Jet_Airways_Business = 0
        Vistara_Premium_economy = 0
        Trujet = 0

    elif (Airline=='IndiGo'):
        Jet_Airways = 0
        IndiGo = 1
        Air_India = 0
        Multiple_carriers = 0
        SpiceJet = 0
        Vistara = 0
        GoAir = 0
        Multiple_carriers_Premium_economy = 0
        Jet_Airways_Business = 0
        Vistara_Premium_economy = 0
        Trujet = 0

    elif (Airline=='Air India'):
        Jet_Airways = 0
        IndiGo = 0
        Air_India = 1
        Multiple_carriers = 0
        SpiceJet = 0
        Vistara = 0
        GoAir = 0
        Multiple_carriers_Premium_economy = 0
        Jet_Airways_Business = 0
        Vistara_Premium_economy = 0
        Trujet = 0

    elif (Airline=='Multiple carriers'):
        Jet_Airways = 0
        IndiGo = 0
        Air_India = 0
        Multiple_carriers = 1
        SpiceJet = 0
        Vistara = 0
        GoAir = 0
        Multiple_carriers_Premium_economy = 0
        Jet_Airways_Business = 0
        Vistara_Premium_economy = 0
        Trujet = 0
```

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import time
import pickle
%matplotlib inline
```

[2]:
```
##Source - https://www.kaggle.com/nikhilmittal/flight-fare-prediction-mh
train=pd.read_excel('Data_Train.xlsx')
sample = pd.read_excel('Sample_submission.xlsx')
test = pd.read_excel('Test_set.xlsx')
```

[3]:
```
train.head()
```

[3]:

	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

[4]:
```
test = pd.concat([test,sample],axis=1)
```

[493]:
```
test.head()
```

[493]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	Jet Airways	6/06/2019	Delhi	Cochin	DEL → BOM → COK	17:30	04:25 07 Jun	10h 55m	1 stop	No info	15998

[496]:
```
df.head()
```

[496]:

	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

Feature Engineering

[497]:
```
##Droping columns that does not seem practical to ask to a customer.
df.drop(labels=['Route','Arrival_Time','Duration','Additional_Info'],axis=1,inplace=True)
```

[498]:
```
df.head()
```

[498]:

	Airline	Date_of_Journey	Source	Destination	Dep_Time	Total_Stops	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	22:20	non-stop	3897
1	Air India	1/05/2019	Kolkata	Banglore	05:50	2 stops	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	09:25	2 stops	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	18:05	1 stop	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	16:50	1 stop	13302

[499]:
```python
df['Airline'].value_counts()
```

[499]:
```
Jet Airways                          4746
IndiGo                               2564
Air India                            2192
Multiple carriers                    1543
SpiceJet                             1026
Vistara                               608
Air Asia                              405
GoAir                                 240
Multiple carriers Premium economy      16
Jet Airways Business                    8
Vistara Premium economy                 5
Trujet                                  1
Name: Airline, dtype: int64
```

[500]:
```python
df['Source'].value_counts(),df['Destination'].value_counts()
```

[500]:
```
(Delhi       5682
 Kolkata     3581
 Banglore    2752
 Mumbai       883
 Chennai      456
 Name: Source, dtype: int64,
 Cochin       5682
 Banglore     3581
 Delhi        1582
 New Delhi    1170
 Hyderabad     883
 Kolkata       456
 Name: Destination, dtype: int64)
```

[501]:
```python
df.isnull().sum()
```

[503]:
```
df.head()
```

[503]:

	Airline	Date_of_Journey	Source	Destination	Dep_Time	Total_Stops	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	22:20	non-stop	3897
1	Air India	1/05/2019	Kolkata	Banglore	05:50	2 stops	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	09:25	2 stops	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	18:05	1 stop	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	16:50	1 stop	13302

[504]:
```
df['Day']= df['Date_of_Journey'].str.split('/').str[0]
df['Month']= df['Date_of_Journey'].str.split('/').str[1]
df['Year']= df['Date_of_Journey'].str.split('/').str[2]
```

[505]:
```
df.head()
```

[505]:

	Airline	Date_of_Journey	Source	Destination	Dep_Time	Total_Stops	Price	Day	Month	Year
0	IndiGo	24/03/2019	Banglore	New Delhi	22:20	non-stop	3897	24	03	2019
1	Air India	1/05/2019	Kolkata	Banglore	05:50	2 stops	7662	1	05	2019
2	Jet Airways	9/06/2019	Delhi	Cochin	09:25	2 stops	13882	9	06	2019
3	IndiGo	12/05/2019	Kolkata	Banglore	18:05	1 stop	6218	12	05	2019
4	IndiGo	01/03/2019	Banglore	New Delhi	16:50	1 stop	13302	01	03	2019

[506]:
```
df['Total_Stops']=df['Total_Stops'].str.replace('non-','0 ')
```

[507]:
```
df.head()
```

```
[510]: df['Departure_Hour'] = df['Dep_Time'].str.split(':').str[0]
       df['Departure_Minute'] = df['Dep_Time'].str.split(':').str[1]
```

```
[511]: df.head()
```

[511]:

	Airline	Date_of_Journey	Source	Destination	Dep_Time	Total_Stops	Price	Day	Month	Year	Stops	Departure_Hour	Departure_Minute
0	IndiGo	24/03/2019	Banglore	New Delhi	22:20	0 stop	3897	24	03	2019	0	22	20
1	Air India	1/05/2019	Kolkata	Banglore	05:50	2 stops	7662	1	05	2019	2	05	50
2	Jet Airways	9/06/2019	Delhi	Cochin	09:25	2 stops	13882	9	06	2019	2	09	25
3	IndiGo	12/05/2019	Kolkata	Banglore	18:05	1 stop	6218	12	05	2019	1	18	05
4	IndiGo	01/03/2019	Banglore	New Delhi	16:50	1 stop	13302	01	03	2019	1	16	50

```
[512]: #Converting the datatype o newly created features
       df['Day'] = df['Day'].astype(int)
       df['Month'] = df['Month'].astype(int)
       df['Year'] = df['Year'].astype(int)
       df['Stops'] = df['Stops'].astype(int)
       df['Departure_Hour'] = df['Departure_Hour'].astype(int)
       df['Departure_Minute'] = df['Departure_Minute'].astype(int)
```

```
[513]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13353 entries, 0 to 2670
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Airline           13353 non-null  object
```

[522]:

	Airline	Source	Destination	Price	Day	Month	Year	Stops	Departure_Hour	Departure_Minute	Airline_Encoded	Source_Encoded	Destination_Encoded
0	IndiGo	Banglore	New Delhi	3897	24	3	2019	0	22	20	3	0	5
1	Air India	Kolkata	Banglore	7662	1	5	2019	2	5	50	1	3	0
2	Jet Airways	Delhi	Cochin	13882	9	6	2019	2	9	25	4	2	1
3	IndiGo	Kolkata	Banglore	6218	12	5	2019	1	18	5	3	3	0
4	IndiGo	Banglore	New Delhi	13302	1	3	2019	1	16	50	3	0	5

[523]:

```
df = df.drop(['Airline','Source','Destination'],axis=1)
df.head()
```

[523]:

	Price	Day	Month	Year	Stops	Departure_Hour	Departure_Minute	Airline_Encoded	Source_Encoded	Destination_Encoded
0	3897	24	3	2019	0	22	20	3	0	5
1	7662	1	5	2019	2	5	50	1	3	0
2	13882	9	6	2019	2	9	25	4	2	1
3	6218	12	5	2019	1	18	5	3	3	0
4	13302	1	3	2019	1	16	50	3	0	5

Feature Selection

[524]:

```
from sklearn.linear_model import Lasso
from sklearn.feature_selection import SelectFromModel
from sklearn.model_selection import train_test_split
```

[514]:

	Airline	Source	Destination	Price	Day	Month	Year	Stops	Departure_Hour	Departure_Minute
0	IndiGo	Banglore	New Delhi	3897	24	3	2019	0	22	20
1	Air India	Kolkata	Banglore	7662	1	5	2019	2	5	50
2	Jet Airways	Delhi	Cochin	13882	9	6	2019	2	9	25
3	IndiGo	Kolkata	Banglore	6218	12	5	2019	1	18	5
4	IndiGo	Banglore	New Delhi	13302	1	3	2019	1	16	50

[515]:
```
df.Airline.value_counts().index
```

[515]:
```
Index(['Jet Airways', 'IndiGo', 'Air India', 'Multiple carriers', 'SpiceJet',
       'Vistara', 'Air Asia', 'GoAir', 'Multiple carriers Premium economy',
       'Jet Airways Business', 'Vistara Premium economy', 'Trujet'],
      dtype='object')
```

[516]:
```
source_dict = {y:x for x,y in enumerate(df.Source.value_counts().index.sort_values())}
source_dict
```

[516]:
```
{'Banglore': 0, 'Chennai': 1, 'Delhi': 2, 'Kolkata': 3, 'Mumbai': 4}
```

[517]:
```
df.Destination.value_counts().index.sort_values()
```

[517]:
```
Index(['Banglore', 'Cochin', 'Delhi', 'Hyderabad', 'Kolkata', 'New Delhi'], dtype='object')
```

[518]:
```
destination_dict = {'Banglore':0,'Cochin':1,'Delhi':2,'Kolkata': 3,'Hyderabad':4,'New Delhi':5}
```

[519]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
df['Airline_Encoded']= le.fit_transform(df['Airline'].values)
```

[532]:
```
(Index(['Day', 'Month', 'Stops', 'Departure_Hour', 'Departure_Minute',
        'Airline_Encoded', 'Source_Encoded', 'Destination_Encoded'],
       dtype='object'),
 (7420, 9),
 8)
```

We see that year feature is not selected so we will eliminate Year feature from our dataset

[533]:
```
X_train = X_train.drop(['Year'],axis=1)
X_test = X_test.drop(['Year'],axis=1)
```

[534]:
```
X_train.head()
```

[534]:

	Day	Month	Stops	Departure_Hour	Departure_Minute	Airline_Encoded	Source_Encoded	Destination_Encoded
4003	1	5	1	16	30	4	3	0
217	1	6	0	1	30	3	0	2
1625	1	6	2	13	15	1	2	1
7636	9	5	0	13	55	1	4	4
4833	6	6	1	9	35	4	3	0

[535]:
```
X_test.head()
```

[535]:

	Day	Month	Stops	Departure_Hour	Departure_Minute	Airline_Encoded	Source_Encoded	Destination_Encoded
5790	12	3	0	14	20	3	3	0
4340	1	6	0	11	40	2	0	2
3028	24	6	0	10	10	3	0	2
3037	18	5	0	10	25	2	1	2

[452]: 2445.2201351591057

[453]:
```
mean_squared_error(y_true=y_test,y_pred=predictions)
```

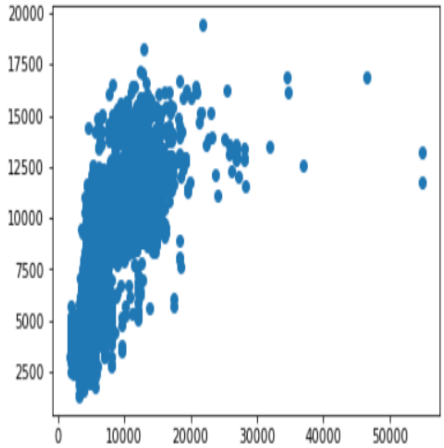
[453]: 11455278.451351777

[454]:
```
lm.coef_
```

[454]:
```
array([-223.98313012,  -98.49583742, 6474.33227294,   96.45466435,
        -37.41287622,  502.70145703, -337.34527856,   96.0680051 ])
```

[455]:
```
plt.scatter(y_test,predictions)
```

[455]: <matplotlib.collections.PathCollection at 0x1d3c4a36b50>



[463]:
```
y_test
```

[562]:
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score,mean_absolute_error,mean_squared_error
```

[563]:
```
lm = LinearRegression()
```

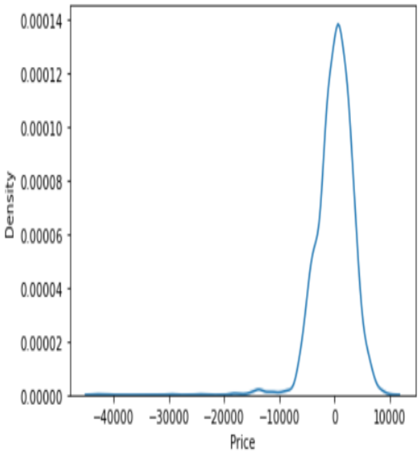
[564]:
```
lm.fit(X_train,y_train)
```

[564]: LinearRegression()

[445]:
```
predictions = lm.predict(X_test)
```

[446]:
```
sns.kdeplot(x=predictions-y_test)
```

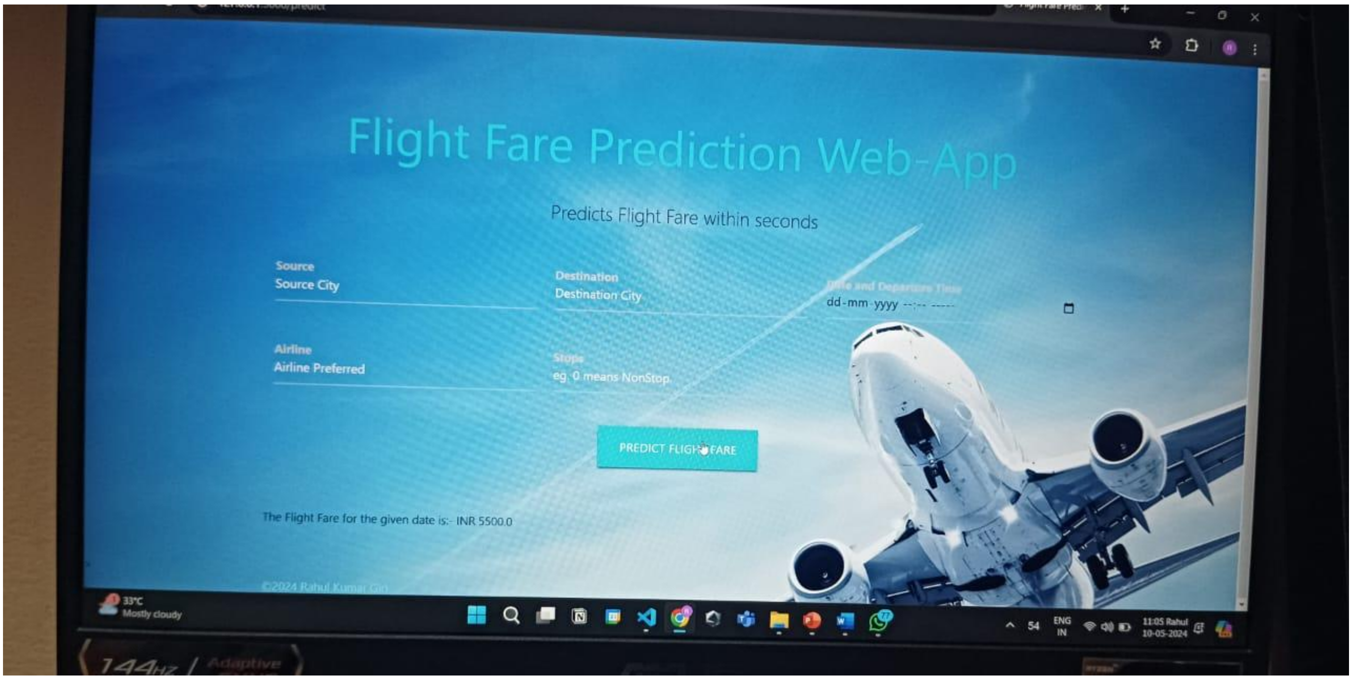
[446]: <AxesSubplot:xlabel='Price', ylabel='Density'>



[468]:
```
r2_score(y_true=y_test,y_pred=predictions)
```

[468]: 0.44393013500989653

OUTPUT



Flight Fare Prediction Web-App
Predicts Flight Fare within seconds
Source
Source City
Destination
Destination City
Airline
Airline Preferred
eg. 0 means NonStop
PREDICT FLIGHT FARE
The Flight Fare for the given date is:- INR 5500.0

REFERENCES

1. "Predicting Airline Ticket Prices Using Historical Data" Authors: R. B. Patterson and R. M. Patel Published in: International Journal of Computer Applications, Vol. 52, No. 5, 2012.

2. Flight Fare Prediction using Historical Data and Machine Learning Techniques" Authors: A. Kumar, et al. Published in: Proceedings of the 3rd International Conference on Computer, Communication, and Signal Processing, 2019.

3. "Flight Fare Prediction Using Machine Learning Techniques" Authors: S. G. Sonawane and A. N. Kadam Published in: International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 7, Issue 7, 2017.

4. "Airline Fare Prediction Using Machine Learning" Authors: A. L. Rodrigues, et al. Published in: Proceedings of the International Conference on Data Engineering and Communication Technology, 2020.

5. "Flight Fare Prediction Using Machine Learning Techniques" Authors: R. N. Sahoo, B. Mishra, and S. P. Dash Published in: Proceedings of the 3rd International Conference on Computational Intelligence in Data Science (ICCIDS), 2019. DOI: 10.1109/ICCIDS.2019.9010314

6. "Predicting Airfare using Machine Learning Algorithms" Authors: K. Gupta and S. Agarwal Published in: International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 9, Issue 3,