A PROJECT REPORT

on

"FLIGHT FARE PREDICTION"

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE AND ENGINEERING

BY

SAKSHI SHREYA	1905632
MIKITA MAJUMDAR	1905327
RACHEET PRADHAN	1905112
ABHIJIT ROUT	1905074

UNDER THE GUIDANCE OF
AMIYA RANJAN PANDA

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA - 751024

May 2023

A PROJECT REPORT

on

"FLIGHT FARE PREDICTION"

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN COMPUTER SCIENCE AND ENGINEERING

BY

SAKSHI SHREYA	1905632
MIKITA MAJUMDAR	1905327
RACHEET PRADHAN	1905112
ABHIJIT ROUT	1905074

UNDER THE GUIDANCE OF

AMIYA RANJAN PANDA

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA -751024

April 2023

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

"FLIGHT FARE PREDICTION"

submitted by

SAKSHI SHREYA	1905632
MIKITA MAJUMDAR	1905327
RACHEET PRADHAN	1905112
ABHIJIT ROUT	1905074

is a record of bonafide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Sci-ence & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during the year 2022-2023, under our guidance.

Date: 03 /05 /2023

AMIYA RANJAN PANDA
Project Guide

Acknowledgements

We are profoundly grateful to **AMIYA RANJAN PANDA** for his expert guidance and continuous encouragement throughout to see that this project meets its target since its commencement to its completion.

We owe our deepest sense of gratitude to KIIT Deemed to be University for granting us this opportunity to gain industrial training and exposure from some of the best minds in their respective fields.

This project would have been an uphill task had it not been for the support of the resources provided and guidance extended by the team members.
Lastly, we would like to express our sincere gratitude to the contributors of all the resources we have referred to during our project work.

SAKSHI SHREYA

MIKITA MAJUMDAR

RACHEET PRADHAN

ABHIJIT ROUT

ABSTRACT

Currently, everyone loves to travel by flights. Going along with the study, the charge of traveling through a plane changes now and then which also includes the day and night time. Additionally, it changes with special times of the year or celebration seasons. There are a few unique elements upon which the cost of air transport depends. The salesperson has data regarding each of the variables, however, buyers can get confined information which is not sufficient to foresee the airfare costs. Considering the provisions, for example, time of the day, the number of days remaining and the time of take-off this will provide the perfect time to purchase the plane ticket. The motivation behind this paper is to concentrate on every component that impacts the variations in the costs of this means of transport and how these are connected with the diversity in the airfare. Subsequently, at that point, utilizing this data, construct a framework that can help purchasers when to purchase a ticket. Machine Learning algorithms prove to be the best solution for the above-discussed problems. In this project, there is an implementation of LR (Linear Regression), DT (Decision Tree), RF (Random Forest) , KNN and XGBoost Regression.

Keywords:
Machine Learning Algorithms,Python, airfare, supervised learning, predictions, flight, Linear Regression, Random Forest, Decision Tree ,KNN , XGBoost Regression.

Contents

LIST OF FIGURES

Chapter 1

Introduction

Our project is Flight fare price prediction using machine learning which predicts the price of different flights on the basis of different parameters provided. Our project takes different inputs like departure date, arrival date, source, destination, stoppage and the airline we want to travel to, hence it predicts the flight fare.

Our project is mainly written in Python using NumPy and Pandas libraries for building the machine learning model and flask is used for the interface.

The data is cleaned and the model is trained and tested using two datasets i.e, train data and test data. The model was then integrated with the user interface, where any user can enter the input parameters and get their flight fare predicted.

Chapter 2

Basic Concepts

Data classification is the process of organizing data into categories that make it easy to retrieve, sort, and store for future use. A well-planned data classification system makes essential data easy to find and retrieve. This can be of particular importance for risk management, legal discovery, and regulatory compliance. Once a data classification scheme is created, security standards should be identified that specify appropriate handling practices for each category. Storage standards that define the data's lifecycle requirements must be addressed, as well.

2.1 Data

In computing, data is information that has been translated into a form that is efficient for movement or processing. Relative to today's computers and transmission media, data is information converted into binary digital form. It is acceptable for data to be used as a singular subject or a plural subject.

2.2 Classification

2.2.1 Unstructured Data

Characteristics of unstructured data include:

- No defined data models
- Difficult to search
- Text, pdf, images, or video-based
- Shows why

It primarily resides in Applications, data warehouses, and lakes. It is stored in various forms. A few examples are - documents, emails and messages, conversation transcripts, image files, and open-ended survey answers.

2.2.2 Structured Data

Characteristics of structured data include:

- Pre-defined data models
- Easy to search
- Text-based
- Shows what’s happening

It primarily resides in Relational databases, data warehouses, etc. It is stored in the form of rows and columns. A few examples are - dates, phone numbers, social security numbers, customer names, transaction info, etc.

2.2.3 Semi-structured Data

Characteristics of unstructured data include:

- Loosely organized
- Meta-level structure that can contain unstructured data
- HTML, XML, JSON

It primarily resides in Relational databases, tagged-text format, etc. It is stored in the form of abstracts and figures. A few examples are - Server logs, tweets organized by hashtags, and emails sorted by folders (inbox; sent; draft).

2.3 Types of Data Classification

In the most simple terms, data can be recognized and categorized in three approaches. These are

- *Content-based classification*: In this classification type, the contents of each file is the basis for categorization.

- *User-based classification*: User-based classification relies on the user's knowledge of creation, editing, reviewing, or dissemination to label sensitive documents. These individuals can specify how sensitive each document is.

- *Context-based classification*: Context-based classification focuses on the context of the data, such as the location, application, and creator, as well as other variables that affect the data.

2.4 Purpose of Data Classification

Systematic classification of data helps organizations manipulate, track and analyze individual pieces of data. Data professionals often have a specific goal when categorizing data. The goal affects the approach they take and the classification levels they use.

Some common business goals for these projects include the following:

➔ *Confidentiality*: A classification system safeguards highly sensitive data, such as customers' personally identifiable information (PII), including credit card numbers, Social Security numbers, and other vulnerable data types. Establishing a classification system helps an organization focus on confidentiality and security policy requirements, such as user permissions and encryption.

➔ *Data integrity* : A system that focuses on data integrity will require more storage, user permissions, and proper channels of access.

➔ *Data availability*: Addressing and ensuring information security and integrity makes it easier to know what data can be shared with specific users.

Chapter 3

Problem Statement / Requirement Specifications

3.1 Project Planning

We are proposing a system that helps the user to predict the price of an airline ticket with optimum accuracy. Firstly, the user needs to fill the required input fields provided on the web page. The input fields include the information about the date of the journey i.e., the date of departure and the departure time suitable for the user to start his flight. Up next, the user needs to select the arrival time Source and destination are to be chosen by the user from the drop down menu linked to the input field. Later, he/she has to select the number of halts in the journey which will impact the cost of the ticket. Lastly, the most important factor is the choice of the airline company that the users choose to travel with. A drop down menu is attached for the same. Upon providing all the input fields. and clicking the 'Submit" button, the system enables the user to predict the price of the airline ticket.

3.2 Project Analysis

Preparation of data is trailed by breaking down the information, revealing the concealed patterns and afterward applying different Al models. Likewise, a few features can be determined from the current features. Flight days can be issued by computing the difference of the flight date and the date on which information is collected. This can be observed for 45 days. Additionally, flight date is important, whether it is on a festive day or a weekday or weekend. Instinctively the flights planned during the weekends cost more than the flights on weekdays. Additionally, time plays an important role. So the time is considered in classes as: Morning, evening and night.

From the data collected and through exploratory data analysis, we can determine the following:

• The trend of flight prices vary over various months and across the holiday.

• There are two groups of airlines: the economical group and the luxurious group. Spicejet, AirAsia, IndiGo, Go Air are in the economical class, whereas Jet

Airways and Air India in the other.

•The airfare varies depending on the time of departure, making the time slot used in analysis an important parameter.

• The airfare increases during the holiday season. In our time period, during Diwali the fare remained high for all the values of days to departure. We have considered the holiday season as a parameter which helped in increasing the accuracy.

• Airfare varies according to the day of the week of travel. It is higher for weekends and Monday and slightly lower for other days.

• There are a few times when an offer is run by an airline because of which the prices drop suddenly. These are difficult to incorporate in our mathematical models, and hence lead to error.

• Along the business routes, we find that the price of flights increases or remains constant as the days to departure decreases. This is because of the high frequency of the flights, high demand and also could be due to heavy competition.

3.3 System Design

3.3.1 Design Constraints

•Airline: So this column will have all the types of airlines like Indigo, Jet Airways, Air India, and many more.
•Date_of _Journey: This column will let us know about the date on which the passenger's journey will start.
•Source: This column holds the name of the place from where the passenger's journey will start.
•Destination: This column holds the name of the place to where passengers wanted to travel.
•Route: Here we can know about what is the route through which passengers have opted to travel from his/her source to their destination.
•Arrival_Time: Arrival time is when the passenger will reach his /her destination.
•Duration: Duration is the whole period that a flight will take to complete its journey from source to destination.

•Total_Stops: This will let us know in how many places flights will stop there for the flight in the whole journey.
•Price: Price of the flight for a complete journey including all the expenses before on-boarding.
When designing a flight price prediction project, there are several design constraints that we should consider. Here are a few examples:
Data Availability: One of the main constraints for a flight price prediction project is the availability and quality of data. Some airlines may not make their pricing data publicly available or may charge high fees for access to the data. Additionally, the quality of the data may vary depending on the source, which can affect the accuracy of the predictive model.

Data Volume: The volume of data available for flight price prediction may also be a constraint. Depending on the number of flights and routes that need to be tracked, the amount of data collected could be very large. This can impact the speed of data processing and the storage requirements for the data.

Computational Resources: Flight price prediction models often require significant computational resources to train and run. Depending on the complexity of the model and the volume of data being used, this may require expensive hardware or cloud computing resources.

Accuracy Requirements: The level of accuracy required for the flight price prediction model may also be a constraint. Depending on the use case, the model may need to predict prices with a high degree of accuracy, which may require a more complex or sophisticated model.

Time Sensitivity: Flight prices can change rapidly, which means that prediction models need to be updated frequently to provide accurate predictions. This can place a constraint on the speed at which data can be collected, processed, and analyzed.

Overall, when designing a flight price prediction project, it is important to consider these and other constraints to ensure that the model is accurate, reliable, and feasible to implement.

3.3.2 System Architecture OR Block Diagram

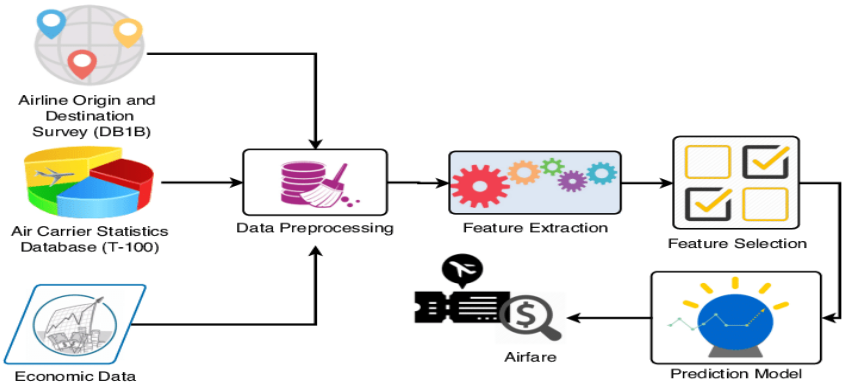


Fig 1.1 : System Architecture

Collecting Dataset: Collecting a data set for flight prediction can be time consuming and require technical expertise. Our data set involves Airline, Date of journey, source, destination, Arrival time, Departure time, Duration, price.

Data Preprocessing: Clean and preprocess the data to remove outliers.In Data Analysis we will need to find the missing values in the dataset,all the numerical variables and their distribution , categorical Variables,outliers and relationship between an independent and dependent feature(*price*).

Modeling: To build a model that can accurately predict the price of a given flight.it predicts fares of flight for a particular date based on various parameters like source, destination, stops and airline.

Deployment: It involves a user friendly application that allows users to input flight details and predict prices.

Chapter 4

Implementation

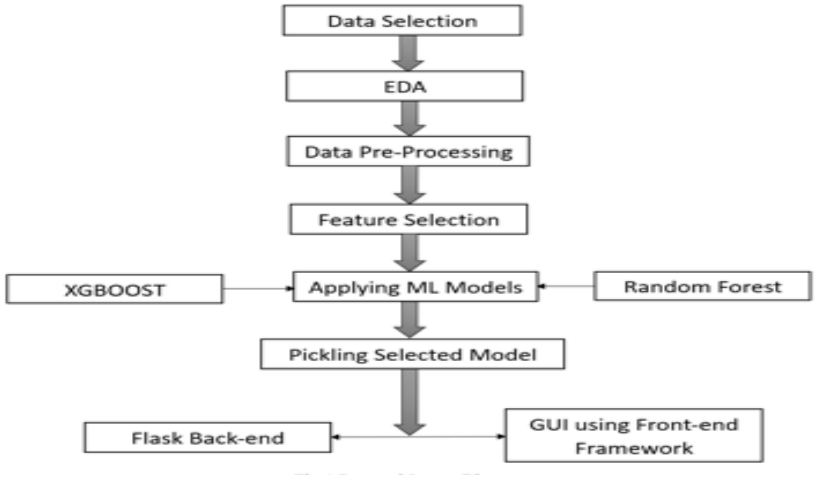


Fig 4.1 : Implementation of project

In this project, we followed these steps:

- Data Description- to get a first overview of our data.
- Feature Selection- to choose which feature to keep or not
- Cleaning of data- after feature selection data is cleaned accordingly
- Encoding - converting categorical values to numerical
- Deployment- pickle the object in a file and deploy it using flask.

4.1 Methodology OR Proposal

4.1.1 Importing the dataset and importing the libraries.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

sns.set()
```

Fig 4.2 : Dependencies

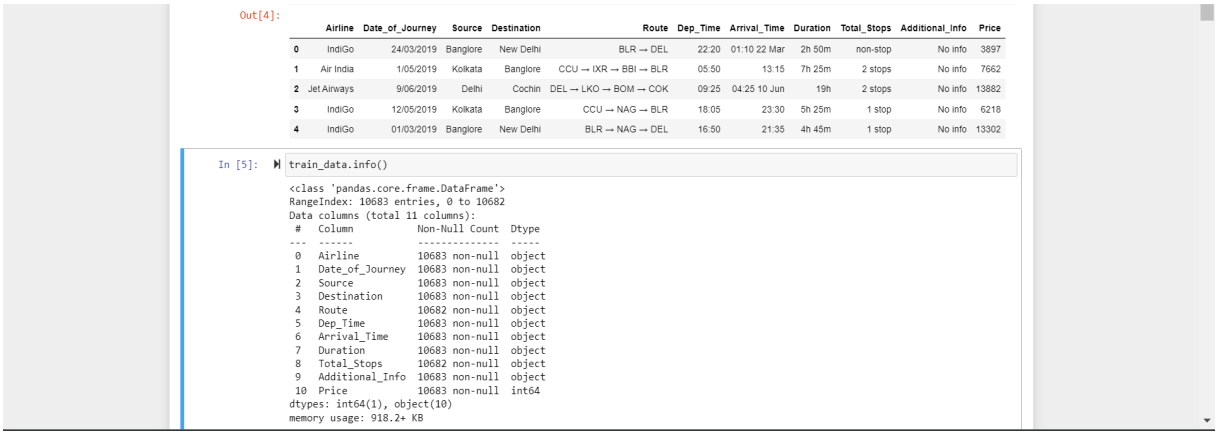
- First up I've imported the python packages/libraries that I've used for this app.
- Since data is in the form of an excel file we have to use pandas read_excel to load the data.

4.1.2 Data Preprocessing

- Before modeling our dataset, we will pre-process it to understand the data, make it suitable for the modeling, and make sure that we are extracting the most information from it.

- In Data Analysis we will need to find the missing values in the dataset,all the numerical variables and their distribution , categorical Variables,outliers and relationship between an independent and dependent feature(*price*).

- After dropping the columns of no use , we will get the final cleaned data.



Out[4]:

	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

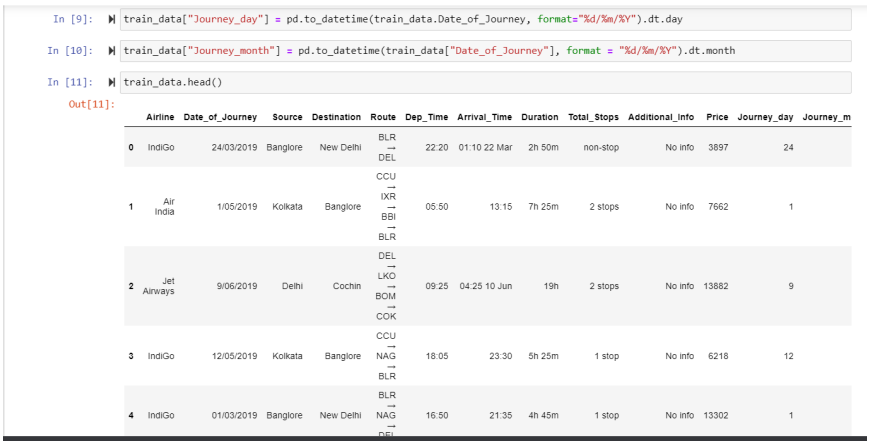
In [5]:
```
train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   Airline          10683 non-null  object
 1   Date_of_Journey  10683 non-null  object
 2   Source           10683 non-null  object
 3   Destination      10683 non-null  object
 4   Route            10682 non-null  object
 5   Dep_Time         10683 non-null  object
 6   Arrival_Time     10683 non-null  object
 7   Duration         10683 non-null  object
 8   Total_Stops      10682 non-null  object
 9   Additional_Info  10683 non-null  object
 10  Price            10683 non-null  int64 
dtypes: int64(1), object(10)
memory usage: 918.2+ KB
```

Fig 4.3 : Dataset(Cleaned)

4.1.3 Encoding and EDA

- From the description we can see that Date_of_Journey is an object data type,Therefore, we have to convert this data type into timestamp so as to use this column properly for prediction. For this we require pandas ``to_datetime" to convert object data type to datetime dtype.



In [9]:
```
train_data["Journey_day"] = pd.to_datetime(train_data.Date_of_Journey, format="%d/%m/%Y").dt.day
```

In [10]:
```
train_data["Journey_month"] = pd.to_datetime(train_data["Date_of_Journey"], format = "%d/%m/%Y").dt.month
```

In [11]:
```
train_data.head()
```

Out[11]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price	Journey_day	Journey_m
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897	24	
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662	1	
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882	9	
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218	12	
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302	1	

- Calculating the time taken by the plane to reach the destination.

```
# Time taken by plane to reach destination is called Duration
# It is the differnce betwwen Departure Time and Arrival time


# Assigning and converting Duration column into list
duration = list(train_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_hours and duration_mins list to train_data dataframe

train_data["Duration_hours"] = duration_hours
train_data["Duration_mins"] = duration_mins
```

```
train_data.drop(["Duration"], axis = 1, inplace = True)
```

Fig 4.4 : Encoding

4.1.4 Handling the categorical data

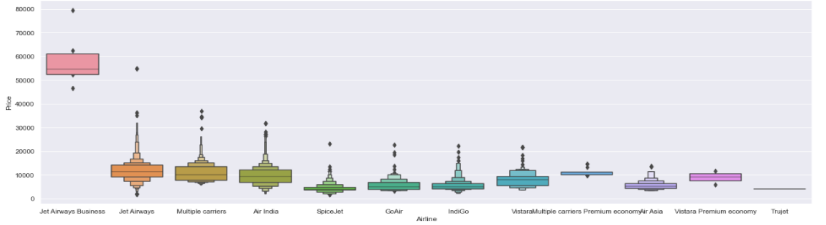
Some of the categorical data are:

1. Nominal data : Data is not in the order.
2. Ordinal data : Data are in order.

Since “Airline” column has 12 unique values — ‘IndiGo’ , ‘Air India’, ‘Jet Airways’ , ‘SpiceJet’ , ‘Multiple carriers’ , ‘GoAir’, ‘Vistara’, ‘Air Asia’, ‘Vistara Premium economy’ , ‘Jet Airways Business’, ‘Multiple carriers Premium economy’, ‘Trujet’.Hence Airline column is nominal categorical data and there is less cardinality, so we will perform One Hot Encoding.

```
In [21]: train_data["Airline"].value_counts()

Out[21]: Jet Airways                          3849
         IndiGo                               2053
         Air India                            1751
         Multiple carriers                    1196
         SpiceJet                              818
         Vistara                               479
         Air Asia                              319
         GoAir                                 194
         Multiple carriers Premium economy      13
         Jet Airways Business                    6
         Vistara Premium economy                 3
         Trujet                                  1
         Name: Airline, dtype: int64
```

```
In [22]: # From graph we can see that Jet Airways Business have the highest Price.
         # Apart from the first Airline almost all are having similar median

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



From the above graph we can see that Jet Airways has the highest price.

- In the similar way , one hand coding of source and destination column is done.

```
# As Source is Nominal Categorical data we will perform OneHotEncoding

Source = train_data[["Source"]]

Source = pd.get_dummies(Source, drop_first= True)

Source.head()
```

	Source_Chennai	Source_Delhi	Source_Kolkata	Source_Mumbai
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	0

```
# As Destination is Nominal Categorical data we will perform OneHotEncoding

Destination = train_data[["Destination"]]

Destination = pd.get_dummies(Destination, drop_first = True)

Destination.head()
```

	Destination_Cochin	Destination_Delhi	Destination_Hyderabad	Destination_Kolkata	Destination_New Delhi
0	0	0	0	0	1
1	0	0	0	0	0
2	1	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	1

- Total_Stops has 5 unique values — ‘1 stop’ , ‘non-stop’ , ‘2 stops’ , ‘3 stops’ , ‘4 stops’. In this case of Ordinal Categorical type we perform Label Encoder.

```
train_data.replace({"non-stop": 0, "1 stop": 1, "2 stops": 2, "3 stops": 3, "4 stops": 4}, inplace = True)
```

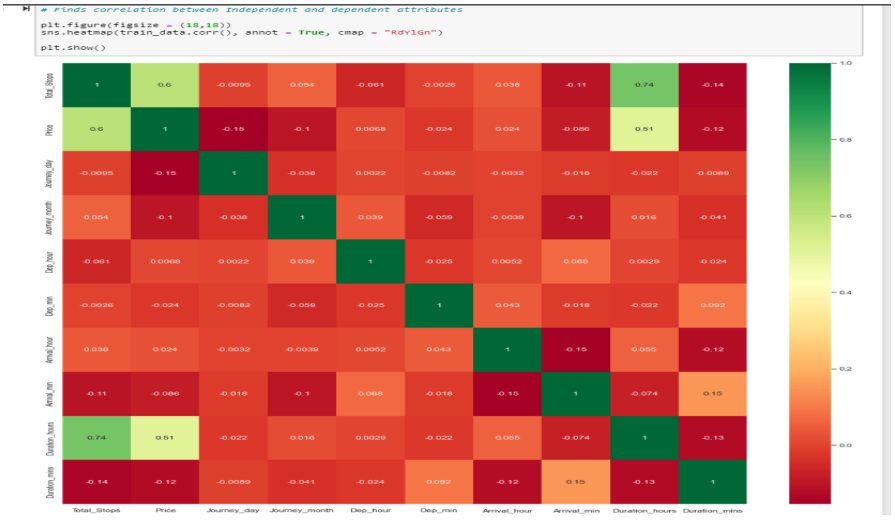
```
train_data.head()
```

	Airline	Source	Destination	Total_Stops	Price	Journey_day	Journey_month	Dep_hour	Dep_min	Arrival_hour	Arrival_min	Duration_hours	Duratic
0	IndiGo	Banglore	New Delhi	0	3897	24	3	22	20	1	10	2	
1	Air India	Kolkata	Banglore	2	7662	1	5	5	50	13	15	7	
2	Jet Airways	Delhi	Cochin	2	13882	9	6	9	25	4	25	19	
3	IndiGo	Kolkata	Banglore	1	6218	12	5	18	5	23	30	5	
4	IndiGo	Banglore	New Delhi	1	13302	1	3	16	50	21	35	4	

- All the above preprocessing and feature engineering steps are performed with a test data set.

- Finding the correlation between independent and dependent attributes.

```
# Finds Correlation between Independent and dependent attributes
plt.figure(figsize = (18,18))
sns.heatmap(train_data.corr(), annot = True, cmap = "RdYlGn")
plt.show()
```



4.1.5 ML Models:

- We have used five different models in this project:

a. Linear Regression:

Linear regression is a type of statistical analysis used to predict the relationship between two variables. It assumes a linear relationship between the independent variable and the dependent variable, and aims to find the best-fitting line that describes the relationship.

Advantages:

•It is used to find the nature of the relationship among the variables.

•Able This model is easier to implement, interpret and very efficient to train.

Disadvantages:

•It is often quite prone to noise and overfitting.

•Before applying this model, multicollinearity should be removed as it assumes that there is no relationship among independent variables.

b. Decision Tree:

Decision Tree may be a supervised classification and regression formula that maybe sculptural within the sort of tree structure normally utilized in data processing and machine learning. It may be drawn in 2 ways that like a classification tree during which target variables square measure separate in nature.

Advantages:

•Able to handle each numerical and categorical information. but scikit-learn implementation doesn't support categorical variables for now.

•Performs well albeit its assumptions square measure somewhat profaned by the truth model from that the information was generated.

Disadvantages:

•Decision-tree learners will produce over-complex trees that don't generalize the information well. This is often referred to as overfitting.

•It is unstable as a result of tiny variations within the knowledge may end in a very completely different tree being generated. This downside is lessened by mistreatment call trees inside associate degree ensemble.

c. Random Forest :

Random Forest is an ensemble of randomized decision tree classifiers used for classifi- cation and regression kind of problems in machine learning. Random forest consists of many classification trees. This algorithm has some of the advantages such as 1. high classification a random forest.

Advantages:

•It reduces overfitting in call trees and helps to enhance the accuracy.

•It works well with each and continuous values .

Disadvantages :

•It needs a lot of machine power similarly as resources because it builds varied trees to mix their outputs.

•It conjointly suffers interpretability and fails to see the importance of every variable.

d. KNN:

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. It stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suited category by using K- NN algorithm.

Advantages :

•It is robust to the noisy training data.

•It is simple to implement and can be more effective if the training data is large.

Disadvantages:

•In this model, we need to determine the value of K which may be complex some time.

•The computation cost is high because of calculating the distance between the data points for all the training samples.

e. XGBoost Regression:

It stands for Extreme Gradient Boosting. This model is a scalable, distributed gradient boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems.

Advantages :

•This model can be used for a variety of machine-learning tasks, including classification, regression, and ranking.

•This model includes regularization techniques that help to prevent overfitting, which is a common problem in machine learning.

Disadvantages:

•This model can be memory-intensive, especially for large datasets.

•Overfitting is possible if parameters are not tuned properly.

STEPS:

- First split the dataset into train and test set in order to prediction w.r.t X_test, then import different model .

```
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)
```

- Fit the data in models and then predict the accuracy.
- Fitting the parametric distributions using distplot() and Scatter plot. This will visually evaluate how closely it corresponds to the observed data.
- MSE and RSME score is calculated.

a. Linear Regression:

-importing model and finding accuracy

```
from sklearn.linear_model import LinearRegression
reg_rf = LinearRegression()
reg_rf.fit(X_train, y_train)
y_pred= reg_rf.predict(X_test)
```

```
reg_rf.score(X_train, y_train)
```

```
0.6240840020468166
```

```
reg_rf.score(X_test, y_test)
```

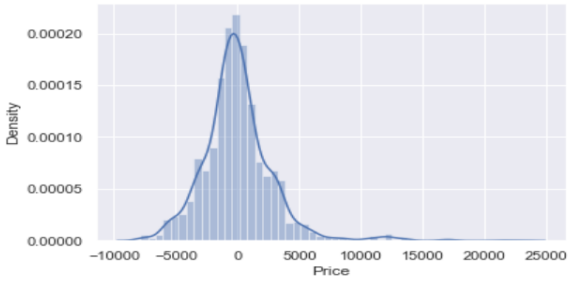
```
0.6195943729070101
```

-graph

(i) distplot

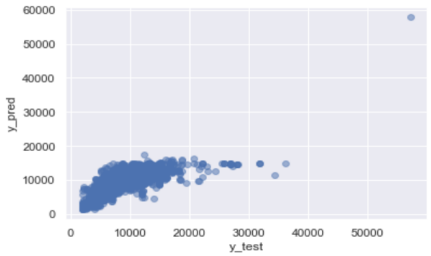
```
sns.distplot(y_test-y_pred)
plt.show()
```

```
C:\Users\shrey\anaconda3\lib\site-packages\seaborn\distr
and will be removed in a future version. Please adapt yo
r flexibility) or `histplot` (an axes-level function for
  warnings.warn(msg, FutureWarning)
```



(ii). Scatter plot

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



-MAE,MSE,RMSE

```
from sklearn import metrics
```

```
print('MAE:', metrics.mean_absolute_error(y_test, y_pred))
print('MSE:', metrics.mean_squared_error(y_test, y_pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
MAE: 1972.9372855148047
MSE: 8202327.557407132
RMSE: 2863.9705929717807
```

```
# RMSE/(max(DV)-min(DV))

2863.9705/(max(y)-min(y))
```

```
: 0.036834212184738854
```

b. Decision Tree:

-importing model and finding accuracy

```
from sklearn.tree import DecisionTreeRegressor
reg_rf = DecisionTreeRegressor()
reg_rf.fit(X_train, y_train)
y_pred= reg_rf.predict(X_test)
```

```
reg_rf.score(X_train, y_train)
```

```
0.9692484150527355
```

```
reg_rf.score(X_test, y_test)
```

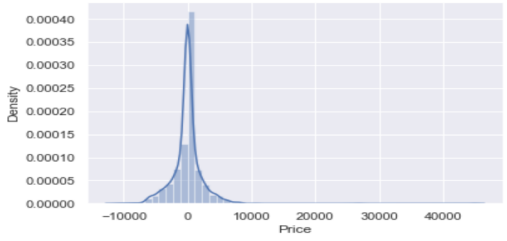
```
0.7282502570426728
```

-graph

(i) distplot

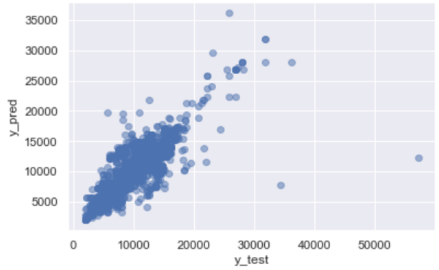
```
sns.distplot(y_test-y_pred)
plt.show()
```

```
C:\Users\shrey\anaconda3\lib\site-packages\seaborn\dist
and will be removed in a future version. Please adapt y
r flexibility) or `histplot` (an axes-level function fo
  warnings.warn(msg, FutureWarning)
```



(ii) Scatter plot

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



-MAE,MSE,RMSE

```
from sklearn import metrics
```

```
print('MAE:', metrics.mean_absolute_error(y_test, y_pred))
print('MSE:', metrics.mean_squared_error(y_test, y_pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
MAE: 1332.4896661987211
MSE: 5950610.796478579
RMSE: 2439.3873813887326
```

```
# RMSE/(max(DV)-min(DV))

2439.3873/(max(y)-min(y))
```

```
: 0.03137354571527787
```

c. Random Forest:

-importing model and finding accuracy

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

```
RandomForestRegressor()
```

```
y_pred = reg_rf.predict(X_test)
```

```
reg_rf.score(X_train, y_train)
```

```
0.9535703004744431
```

```
reg_rf.score(X_test, y_test)
```

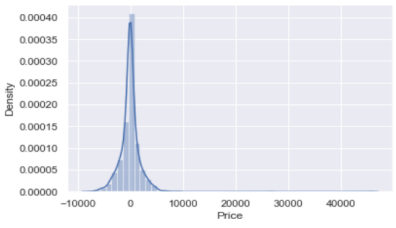
```
0.7975519262537325
```

-Graph

(i). distplot

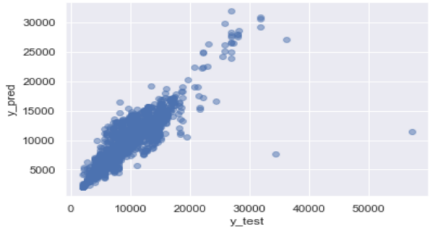
```
sns.distplot(y_test-y_pred)
plt.show()
```

```
C:\Users\shrey\anaconda3\lib\site-packages\seaborn\distribu
n and will be removed in a future version. Please adapt yo
ilar flexibility) or `histplot` (an axes-level function fo
  warnings.warn(msg, FutureWarning)
```



(ii). Scatter plot

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



-MAE , MSE , RMSE

```
print('MAE:', metrics.mean_absolute_error(y_test, y_pred))
print('MSE:', metrics.mean_squared_error(y_test, y_pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
MAE: 1169.71792808255
MSE: 4317481.233453424
RMSE: 2077.8549596767875
```

```
# RMSE/(max(DV)-min(DV))

2097.6158/(max(y)-min(y))
```

```
: 0.026977940401013468
```

d. KNN

-importing model and finding accuracy

```
from sklearn.neighbors import KNeighborsClassifier
reg_rf = KNeighborsClassifier()
reg_rf.fit(X_train, y_train)
y_pred = reg_rf.predict(X_test)
```

```
reg_rf.score(X_train, y_train)
```

```
: 0.45043885313048565
```

```
reg_rf.score(X_test, y_test)*100
```

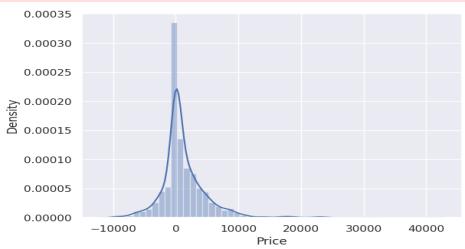
```
: 21.993448759943846
```

-graph

(i) distplot

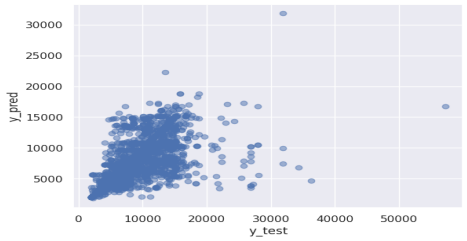
```
sns.distplot(y_test-y_pred)
plt.show()
```

```
<ipython-input-100-75adb1dd5983>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(y_test-y_pred)
```



(ii) Scatter plot

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



-MAE,MSE,RMSE

```
print('MAE:', metrics.mean_absolute_error(y_test, y_pred))
print('MSE:', metrics.mean_squared_error(y_test, y_pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
MAE: 2332.3762283575106
MSE: 15972504.086102013
RMSE: 3996.5615328807353
```

```
# RMSE/(max(DV)-min(DV))

4644.4629/(max(y)-min(y))
```

```
: 0.05973355240312272
```

e. XGBoost Regression

-importing model and finding accuracy

```
import xgboost as xgb
reg_rf = xgb.XGBRegressor()
reg_rf.fit(X_train, y_train)
y_pred = reg_rf.predict(X_test)
```

```
reg_rf.score(X_train, y_train)
```

```
0.9353790824683148
```

```
reg_rf.score(X_test, y_test)
```

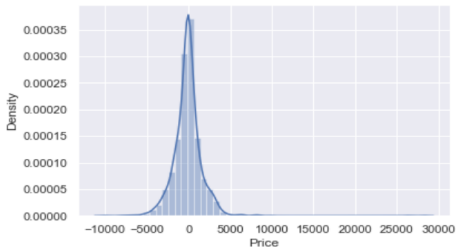
```
0.8463321179731759
```

-graph

(i) distplot

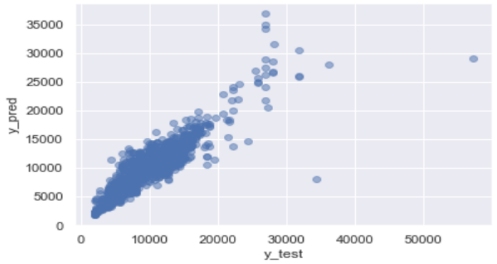
```
sns.distplot(y_test-y_pred)
plt.show()
```

```
C:\Users\shrey\anaconda3\lib\site-packages\seaborn\dist
and will be removed in a future version. Please adapt y
r flexibility) or `histplot` (an axes-level function fo
  warnings.warn(msg, FutureWarning)
```



(ii) Scatter plot

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



-MAE,MSE,RMSE

```
from sklearn import metrics
```

```
print('MAE:', metrics.mean_absolute_error(y_test, y_pred))
print('MSE:', metrics.mean_squared_error(y_test, y_pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
MAE: 1135.7739136142043
MSE: 3313395.5274770306
RMSE: 1820.2734760131596
```

```
# RMSE/(max(DV)-min(DV))

1820.2734/(max(y)-min(y))
```

```
0.023410973210036913
```

4.1.6 Comparison of Accuracy b/w different Models

Model Name	Accuracy	MSE	RMSE
Random Forest	79.88240106162188	4337768.0141620645	2082.7309029641983
Linear Regression	61.95943729070101	8202327.557407132	2863.9705929717807
Decision Tree	72.70657903543061	5885022.801171034	2425.9065936616425
KNN	21.993448759943846	15972504.086102013	3996.5615328807353
XGBoost Regression	84.63321179731759	3313395.5274770306	1820.2734760131596

Fig 4.5 : Comparison of accuracy

After thorough analysis of the models, we conclude that **XGBoost Regression** is the most accurate model for our project. This model has the best combination of prediction performance and processing time compared to others.

4.1.7 Saving the model to reuse again

```
import pickle
# open a file, where you ant to store the data
file = open('flight_rf.pkl', 'wb')

# dump information to that file
pickle.dump(rf_random, file)
```

```
model = open('flight_rf.pkl','rb')
forest = pickle.load(model)
```

```
y_prediction = forest.predict(X_test)
```

```
metrics.r2_score(y_test, y_prediction)
```

```
: 0.8117071697231877
```

4.2 Model Deployment

- Model Deployment is one of the last stages of any machine learning project.
- Here, we will design a user interface using flask to make an HTML file for flight price prediction.
- This will take the input value for each feature and calculate the price for a flight as shown in the image below.

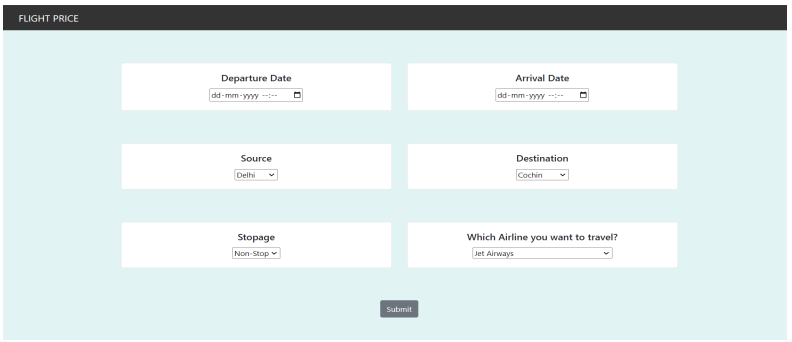
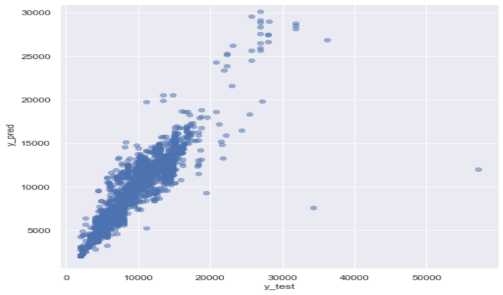


Fig 4.5 : UI of the project

4.3 Result Analysis:

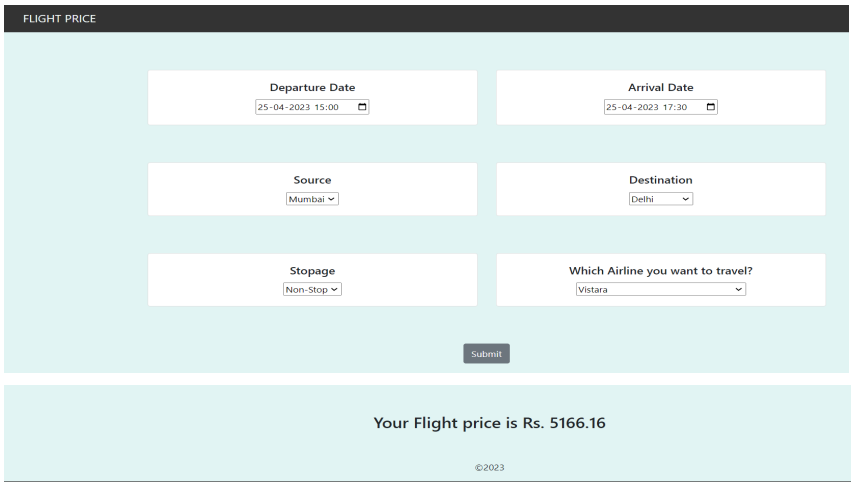
4.3.1 Distplot bw y_pred and y_test



4.3.2 Accuracy

In our project , XGBoost Regression model gives the highest accuracy of 84.6332 and hence this model is used for prediction.

4.3.3 UI



The user input parameters are :
Departure Date : 25.04.2023 15:00
Arrival Date : 25.04.2023 17:30
Source : Mumbai
Destination : Delhi
Stoppage : Non Stop
Airline to travel : Vistara.

Hence the price predicted is Rs. 5166.16

4.4 Quality Assurance

In the working organization, if some department is there to verify the quality of your work, they can produce a certificate or guidelines followed.

Chapter 5

Standards Adopted

5.1 Design Standards

Data Collection: The first step in designing a flight price prediction project is to gather data.

Data Cleaning: After collecting data, you will need to clean it to remove any irrelevant or inaccurate information.

Feature Engineering: Once your data is clean, you will need to create new features that can be used in your predictive model.

Model Selection: You will need to select an appropriate machine learning model that can accurately predict flight prices based on your features.

Model Training: After selecting a model, you will need to train it on your cleaned and engineered data. This involves splitting your data into training and testing sets and using the training set to teach the model how to predict flight prices based on the features you have selected.

Model Evaluation: Once your model has been trained, you will need to evaluate its performance on the testing set. This involves comparing the predicted flight prices to the actual flight prices and measuring the accuracy of the predictions.

Deployment: Finally, you will need to deploy your model in a production environment, where it can be used to make real-time predictions about flight prices.

5.2 Coding Standards

Naming Conventions: Use meaningful names for variables, functions, and classes. Follow a consistent naming convention throughout the codebase.

Commenting: Use comments to explain what the code does, how it works, and why certain decisions were made.

Formatting: Use consistent formatting throughout the codebase. Use whitespace to improve readability.

Error Handling: Include error handling in the code to handle unexpected inputs or failures. Use try/except blocks to catch exceptions, and provide informative error messages to help users troubleshoot issues.

Documentation: Write documentation to explain how to install, configure, and use the code.

5.3 Testing Standards

Unit Testing: Write unit tests for each function or class in the codebase. Unit tests should cover different scenarios and edge cases, including both valid and invalid inputs. Use a testing framework, such as pytest or unittest, to automate the testing process.

Integration Testing: Write integration tests to test the system as a whole. Integration tests should cover end-to-end scenarios, including data retrieval, processing, and prediction. Use mock data or test data to simulate different scenarios.

Performance Testing: Conduct performance testing to measure the speed and resource usage of the predictive model. Use benchmarking tools, such as Apache JMeter, to measure the response time and throughput of the system under different loads.

Regression Testing: Conduct regression testing to ensure that changes to the code do not break existing functionality. Use a continuous integration/continuous deployment (CI/CD) pipeline to automate the testing process and catch issues early in the development cycle.

Data Validation: Validate the accuracy and completeness of the data used to train and test the predictive model. Use statistical analysis and data visualization tools to identify outliers and anomalies in the data.

User Acceptance Testing: Conduct user acceptance testing to gather feedback from users and stakeholders. Use surveys or interviews to collect feedback on the usability and effectiveness of the predictive model.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

Machine Learning algorithms are applied on the dataset to predict the fare of flights. This gives the predicted values of flight fare to get a flight ticket at minimum cost.The values of R-squared obtained from the algorithm give the accuracy of the model.We used Data_Train to train the training data and test_set to test it. These records were used to extract a number of characteristics.Furthermore, we have created a User Interface for the entire process of buying an airline ticket and given a proof of our predictions based on the previous trends with our prediction.To the highest possible standard, much prior studies into flight price prediction using the large dataset depended on standard statistical approaches, which have their own limitations in terms of underlying issue estimates and hypotheses. To our knowledge, no other research has included statistics from holidays, stock market price fluctuations, depression, fuel price, and socioeconomic information to estimate the air transport market sector; nonetheless, there are numerous restrictions. By merging such data, it is feasible to create a more robust and complete daily and even daily flight price forecast model.We have also successfully busted some of the typical myths and misconceptions related to the airline industry and backed them up with data and analysis.Thus leaving it as a battle between ‘The risk appetite of the user’ vs ‘Our understanding of the airline industry’.

6.2 Future Scope

We can use multiple Algorithms for increased efficiency & optimization
More routes can be added and the same analysis can be expanded to major airports and travel routes in India.
The analysis can be done by increasing the data points and increasing the historical data used. That will train the model better giving better accuracy
Developing a more user friendly interface for various routes giving more flexibility to the users.
This framework may be expanded in the future to also include airline tickets payment details, that can offer more detail about each area, such as timestamp of entry and exit, seat placement, covered auxiliary items, etc. the predicted results will be more accurate.

References

- https://www.kaggle.com/datasets/nikhilmittal/flight-fare-prediction-mh
- https://www.kaggle.com/datasets/nikhilmittal/flight-fare-prediction-mh?select=Test_set.xlsx
- https://medium.com/geekculture/flight-fare-prediction-93da3958eb95
- https://www.techtarget.com/searchenterpriseai/feature/How-to-build-a-machine-learning-model-in-7-steps
- https://www.analyticsvidhya.com/blog/2022/01/flight-fare-prediction-using-machine-learning/
- https://medium.com/geekculture/flight-fare-prediction-93da3958eb95
- https://www.ijraset.com/research-paper/flight-fare-prediction-system-using-ml

SAMPLE INDIVIDUAL CONTRIBUTION REPORT:

FLIGHT FARE PREDICTION USING MACHINE LEARNING

SAKSHI SHREYA 1905632

Abstract: Currently, everyone loves to travel by flights. Going along with the study, the charge of traveling through a plane changes now and then which also includes the day and night time. Additionally, it changes with special times of the year or celebration seasons. There are a few unique elements upon which the cost of air transport depends. The salesperson has data regarding each of the variables, however, buyers can get confined information which is not sufficient to foresee the airfare costs. Considering the provisions, for example, time of the day, the number of days remaining and the time of take-off this will provide the perfect time to purchase the plane ticket. The motivation behind this paper is to concentrate on every component that impacts the variations in the costs of this means of transport and how these are connected with the diversity in the airfare. Subsequently, at that point, utilizing this data, construct a framework that can help purchasers when to purchase a ticket. Machine Learning algorithms prove to be the best solution for the above-discussed problems. In this project, there is an implementation of LR (Linear Regression), DT (Decision Tree), and RF (Random Forest).

Individual contribution and findings: Different blogs were examined on the topic of our project.I have studied many research papers that helped me to get more information about the topic and could easily contribute to the execution and implementation of the project. I have done the data cleaning , data preprocessing without which prediction would not have been possible and then tried different models to find the best one with the highest accuracy.

Individual contribution to project report preparation:
Chapter 2: Basic Concepts
Chapter 4: Implementation - 4.1 Methodology
4.2 ML Models

Individual contribution for project presentation and demonstration: Prerequisites , different ml models used and finding the best suited model on the basis of accuracy and steps involved in implementation.

Full Signature of Supervisor: student:

……. ……..

SAKSHI SHREYA

.

SAMPLE INDIVIDUAL CONTRIBUTION REPORT:

FLIGHT FARE PREDICTION USING MACHINE LEARNING

RACHEET PRADHAN 1905112

Abstract: Currently, everyone loves to travel by flights. Going along with the study, the charge of traveling through a plane changes now and then which also includes the day and night time. Additionally, it changes with special times of the year or celebration seasons. There are a few unique elements upon which the cost of air transport depends. The salesperson has data regarding each of the variables, however, buyers can get confined information which is not sufficient to foresee the airfare costs. Considering the provisions, for example, time of the day, the number of days remaining and the time of take-off this will provide the perfect time to purchase the plane ticket. The motivation behind this paper is to concentrate on every component that impacts the variations in the costs of this means of transport and how these are connected with the diversity in the airfare. Subsequently, at that point, utilizing this data, construct a framework that can help purchasers when to purchase a ticket. Machine Learning algorithms prove to be the best solution for the above-discussed problems. In this project, there is an implementation of LR (Linear Regression), DT (Decision Tree), and RF (Random Forest).

Individual contribution and findings: labored to locate blog posts and articles on our subject for the data that was used in the report. A great deal of study articles that were reviewed and examined while working on our project along with contributing with the execution, conclusion and future scope of the project while maintaining the required standards..

Individual contribution to project report preparation:
Chapter 5: Standard Adopted
Chapter 6 : Conclusion and Future Scope

Individual contribution for project presentation and demonstration:
Presentation & speaking about our project in the result analysis & concluding part while creating the project. Also giving insights for future scope and improvements in our project.

Full Signature of Supervisor: student:

……. ……..

RACHEET PRADHAN

SAMPLE INDIVIDUAL CONTRIBUTION REPORT:

FLIGHT FARE PREDICTION USING MACHINE LEARNING

MIKITA MAJUMDAR 1905327

Abstract: Currently, everyone loves to travel by flights. Going along with the study, the charge of traveling through a plane changes now and then which also includes the day and night time. Additionally, it changes with special times of the year or celebration seasons. There are a few unique elements upon which the cost of air transport depends. The salesperson has data regarding each of the variables, however, buyers can get confined information which is not sufficient to foresee the airfare costs. Considering the provisions, for example, time of the day, the number of days remaining and the time of take-off this will provide the perfect time to purchase the plane ticket. The motivation behind this paper is to concentrate on every component that impacts the variations in the costs of this means of transport and how these are connected with the diversity in the airfare. Subsequently, at that point, utilizing this data, construct a framework that can help purchasers when to purchase a ticket. Machine Learning algorithms prove to be the best solution for the above-discussed problems. In this project, there is an implementation of LR (Linear Regression), DT (Decision Tree), and RF (Random Forest).

Individual contribution and findings: worked on finding blog posts, articles to work on our topic for the information that have been used in the report. and also read and went through many research papers that helped to write and worked on our paper.

Individual contribution to project report preparation:
Chapter 1: Introduction
Chapter 4 : Implementation - 4.3 Result Analysis
4.4 Quality Assurance

Individual contribution for project presentation and demonstration: .presenting and talking about our project in the introduction part. also demonstrating the dataset that have been used in the project.

Full Signature of Supervisor: student:

……. ……..

MIKITA MAJUMDAR

SAMPLE INDIVIDUAL CONTRIBUTION REPORT:

FLIGHT FARE PREDICTION USING MACHINE LEARNING

ABHIJIT ROUT 1905074

Abstract: Currently, everyone loves to travel by flights. Going along with the study, the charge of traveling through a plane changes now and then which also includes the day and night time. Additionally, it changes with special times of the year or celebration seasons. There are a few unique elements upon which the cost of air transport depends. The salesperson has data regarding each of the variables, however, buyers can get confined information which is not sufficient to foresee the airfare costs. Considering the provisions, for example, time of the day, the number of days remaining and the time of take-off this will provide the perfect time to purchase the plane ticket. The motivation behind this paper is to concentrate on every component that impacts the variations in the costs of this means of transport and how these are connected with the diversity in the airfare. Subsequently, at that point, utilizing this data, construct a framework that can help purchasers when to purchase a ticket. Machine Learning algorithms prove to be the best solution for the above-discussed problems. In this project, there is an implementation of LR (Linear Regression), DT (Decision Tree), and RF (Random Forest).

Individual contribution and findings: I have studied many articles on our subject to collect the data that was used in this report. and read many research papers that helped us to complete our project.

Individual contribution to project report preparation:
Chapter 3: Problem statement/Required specifications

Individual contribution for project presentation and demonstration: presenting and talking about our project in the implementation part.

Full Signature of Supervisor: student:

……. ……..

ABHIJIT ROUT

TURNITIN PLAGIARISM REPORT