ADVANCED ACADEMIC CENTER

A CENTER FOR INTER-DISCIPLINARY RESEARCH
2022-23

FLIGHT FARE PREDICTION WITH DEPLOYMENT

GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
AUTONOMOUS

Advanced Academic Center

(A Center for Inter-Disciplinary Research)

FLIGHT FARE PREDICTION

is a bonafede work carried out by the following students in partial fulfillment of the requirements for Advanced Academic Centre intern, submitted to the chair, AAC during the academic year 2020-21.

NAME	ROLL NO.	BRANCH
ABHINAV PENDELA	20241A04S4	ECE
SAI AKSHITHA KORIVI	20241A04R3	ECE
GANGULAKURTHI CHANDRA LEKHA	20241A04Q0	ECE

This work was not submitted or published earlier for any study

Dr/Ms./Mr.

Project Supervisor

Dr.B.R.K. Reddy
Program Coordinator

Dr.Ramamurthy Suri
Associate Dean, AAC

ADVANCED ACADEMIC CENTER

ACKNOWLEDGEMENTS

We express our deep sense of gratitude to our respected Director, Gokaraju Rangaraju Institute of Engineering and Technology, for the valuable guidance and for permitting us to carry out this project.

With immense pleasure, we extend our appreciation to our respected Principal, for permitting us to carry out this project.

We are thankful to the Associate Dean, Advanced Academic Centre, for providing us an appropriate environment required for the project completion.

We are grateful to our project supervisor who spared valuable time to influence us with their novel insights.

We are indebted to all the above-mentioned people without whom we would not have concluded the project.

CONTENTS **PAGE NO**

1. ABSTRACT

The Flight ticket prices increase or decrease every now and then depending on various factors like timing of the flights, destination, duration of flights. In the proposed system a predictive model will be created by applying machine learning algorithms to the collected historical data of flights. Optimal timing for airline ticket purchasing from the consumer's perspective is challenging principally because buyers have insufficient information for reasoning about future price movements. In this project we majorly targeted to uncover underlying trends of flight prices in India using historical data and also to suggest the best time to buy a flight ticket. The project implements the validations or contradictions towards myths regarding the airline industry, a comparison study among various models in predicting the optimal time to buy the flight ticket and the amount that can be saved if done so. Remarkably, the trends of the prices are highly sensitive to the route, month of departure, day of departure, time of departure, whether the day of departure is a holiday and airline carrier.

Highly competitive routes like most business routes (tier 1 to tier 1 cities like Mumbai-Delhi) had a non-decreasing trend where prices increased as days to departure decreased, however other routes (tier 1 to tier 2 cities like Delhi - Guwahati) had a specific time frame where the prices are minimum. Moreover, the data also uncovered two basic categories of airline carriers operating in India – the economical group and the luxurious group, and in most cases, the minimum priced flight was a member of the economical group. The data also validated the fact that, there are certain time-periods of the day where the prices are expected to be maximum. The scope of the project can be extensively extended across the various routes to make significant savings on the purchase of flight prices across the Indian Domestic Airline market.

2.INTRODUCTION

Pricing in the airline industry is often compared to a brain game between carriers and passengers where each party pursues the best rates. Carriers love selling tickets at the highest price possible , while still not losing consumers to competitors. Passengers are crazy about buying flights at the lowest cost available , while not missing the chance to get on board. All this makes flight prices fluctuant and hard to predict. But nothing is impossible for people armed with intellect and algorithms.

There are two main use cases of flight price prediction in the travel industry. OTAs and other travel platforms integrate this feature to attract more visitorslooking for the best rates. Airlines employ the technology to forecast rates of competitors and adjust their pricing strategies accordingly. The task is quite challenging because numerous internal and external factors influence airfares.

Internal factors include

- purchase and departure dates,
- seasonality,
- holidays,
- the number of available airlines and flights,
- fare class,
- the current market demand, and
- flight distance.

External factors embrace events going on in the arrival or departure cities like

- Concerts,
- sports competitions,
- festivals,
- terrorist attacks,
- natural disasters,
- political gatherings,

3.MACHINE LEARNING

3.1 Introduction to Machine Learning:

Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks. Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so.

It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step.

3.2 Types of Data in ML:

Training Data: The part of data we use to train our model. This is the data which your model actually sees(both input and output) and learn from.

Validation Data: The part of data which is used to do a frequent evaluation of model, fit on training dataset along with improving involved hyperparameters (initially set parameters before the model begins learning). This data plays it’s part when the model is actually training.

Testing Data: Once our model is completely trained, testing data provides the unbiased evaluation. When we feed in the inputs of Testing data, our model will predict some values(without seeing actual output). After prediction, we evaluate our model by comparing it with actual output present in the testing data. This is how we evaluate and see how much our model has learned from the experiences feed in as training data, set at the time of training.

3.3 CLASSIFICATION IN ML:

Classification is the task of “classifying things” into sub-categories. Classification is of two types:
1. Binary Classification : When we have to categorize given data into 2 distinct classes. Example – On the basis of given health conditions of a person, we have to determine whether the person has a certain disease or not.
2. Multi class Classification : The number of classes is more than 2. For Example – On the basis of data about different species of flowers, we have to determine which specie does our observation belong to.

3.3.1 Types of Classifiers (Algorithms):

There are various types of classifiers. Some of them are :

1. Linear Classifiers : Logistic Regression
2. Tree Based Classifiers : Decision Tree Classifier
3. Support Vector Machines
4. Artificial Neural Networks
5. Bayesian Regression
6. Gaussian Naive Bayes Classifiers
7. Stochastic Gradient Descent (SGD) Classifier
8. Ensemble Methods : Random Forests, AdaBoost, Bagging Classifier, Voting
9. Classifier, ExtraTrees Classifier

3.3.2 Practical Applications of Classifiers:

Here are a few practical applications of classifiers.

- Google’s self -driving car uses deep learning enabled classification techniques which enables it to detect and classify obstacles.
- Spam E-mail filtering is one of the most widespread and well recognized uses of Classification techniques.
- Detecting Health Problems, Facial Recognition, Speech Recognition,Object Detection, Sentiment Analysis all use Classification at their core.

3.3.3 More about Random Forest:

Every decision tree has high variance, but when we combine all of them together in parallel then the resultant variance is low as each decision tree gets perfectlytrained on that particular sample data and hence the output doesn’t depend on one decision tree but multiple decision trees. In the case of a classification problem, the final output is taken by using the majority voting classifier. In the case of a regression problem, the final output is the mean of all the outputs. This part is Aggregation.

Step 1 : Import the required libraries.

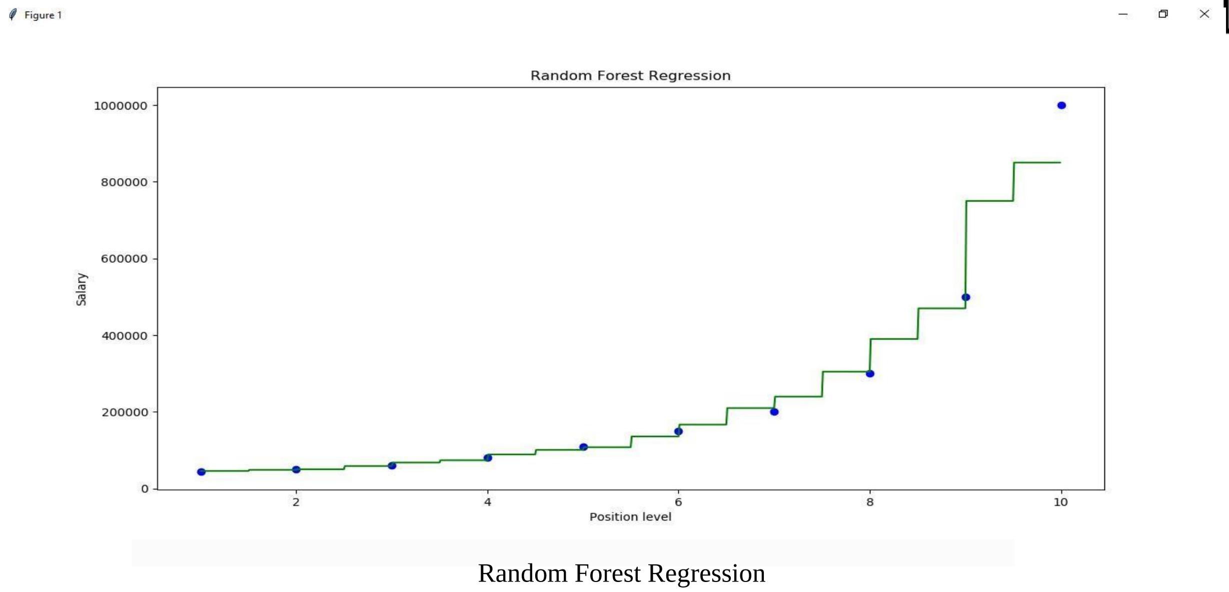
Step 2 : Import and print the dataset

Step 3 : Select all rows and column 1 from dataset to x and all rows and column 2 as y

Step 4 : Fit Random forest regressor to the dataset

Step 5 : Predicting a new result

Step 6 : Visualising the result

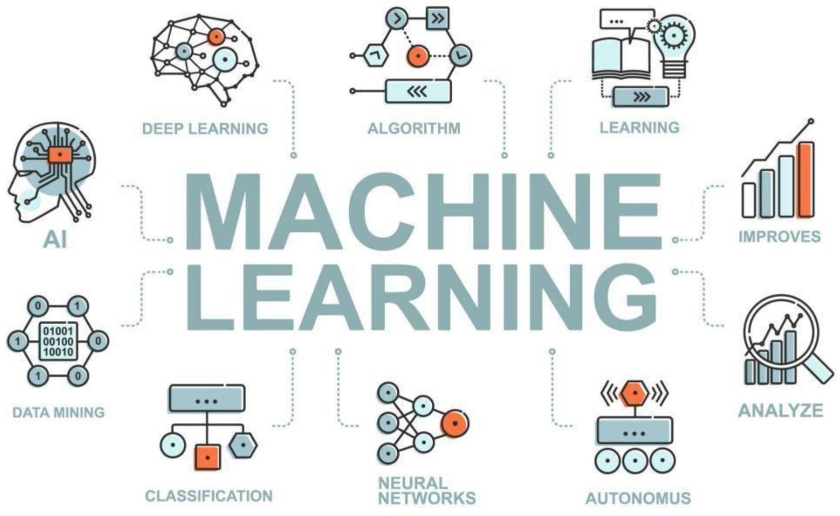


Random Forest Regression

3.4 History of ML

Tom M. Mitchell provided a widely quoted, more formal definition of the algorithms studied in the machine learning field: **"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."** This definition of the tasks in which machine learning is concerned offers a fundamentally operational definition rather than defining the field in cognitive terms.

Modern day machine learning has two objectives, one is to classify data based on models which have been developed, the other purpose is to make predictions for future outcomes based on these models. A hypothetical algorithm specific to classifying data may use computer vision of moles coupled with supervised learning in order to train it to classify the cancerous moles. Where as, a machine learning algorithm for stock trading may inform the trader of future potential predictions.



3.5 Future of ML:

Machine Learning has been one of the hottest topics of discussion among the C-suite. With its incredible potential to compute and analyze huge amounts of data, advanced ML techniques are being used in businesses to perform complex tasks quicker and more efficiently.

The machine learning market is expected to grow from USD 1.03 Billion in 2016 to USD 8.81 Billion by 2022, at a Compound Annual Growth Rate (CAGR) of 44.1% during the forecast period.

Machine learning-driven solutions are being leveraged by organizations to improve customer experience, ROI, and to gain a competitive edge in business. Big players in the field like Google, IBM, Microsoft, Apple, and Salesforce are already leveraging ML benefits using Machine Learning.

3.6 Applications of ML:

a. Machine Learning in Education

Advances in AI are enabling teachers to realize a far better understanding of how their students are progressing with learning. AI will make big and positive changes in education helping students to enjoy the training process and have a far better understanding with their teachers. Students will not feel apprehensive towards their teachers and be frightened of being judged.

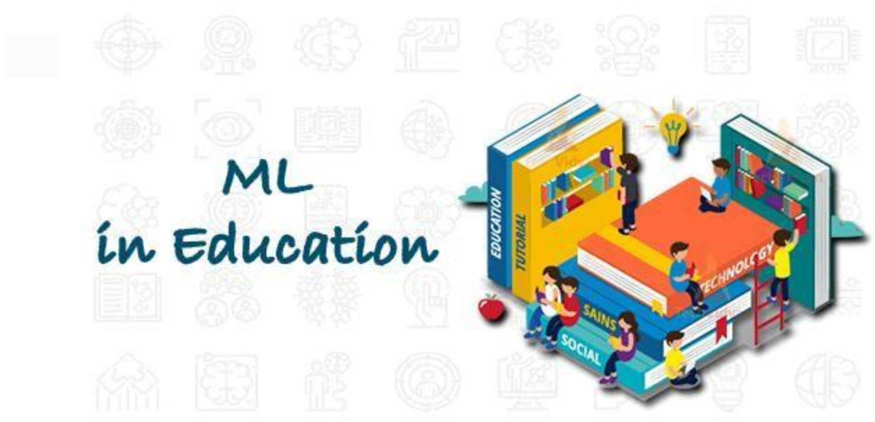


Fig : ML in Education

b. Machine learning in Search Engine

Search engines rely on machine learning to improve their services is no secret today. Implementing these Google has introduced some amazing services. Such as voice recognition, image search and many more. Google services like its image search and translation tools use sophisticated machine learning which permit computers to ascertain , listen and speak in much an equivalent way as human do.

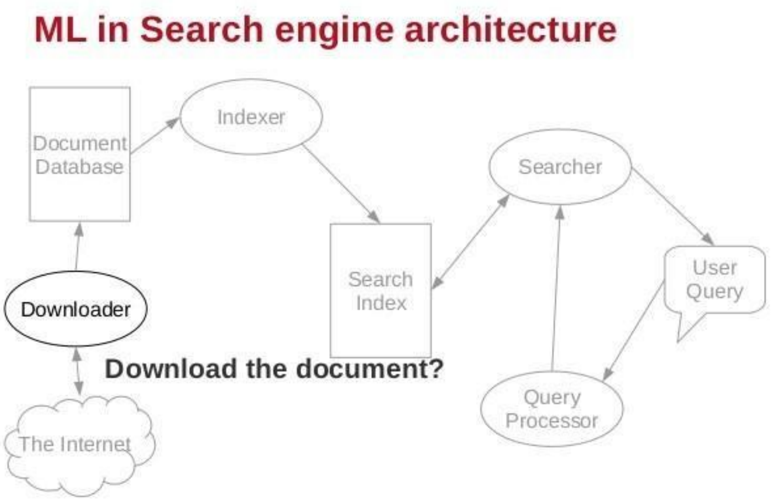


Fig : ML in Search Engine

c. Machine Learning in Digital Marketing

This is where machine learning can help significantly. Machine Learning is being implemented in digital marketing departments round the globe It allows a more relevant personalization. Thus, companies can interact and engage with the customer.

As consumer expectations grow for more personalized, relevant, and assistive experiences, machine learning is becoming a useful tool to assist meet those demands. Sophisticated segmentation focus on the appropriate customer at the right time. Also, with the right message.

SEO
VIDEO
SOCIAL MEDIA
DIGITAL MARKETING
MOBILE & TABLET
CONTENT
ROI
PAY PER CLICK
SEM

Fig : ML in Digital Marketing

d. Machine Learning in Health Care

Machine learning, simply put, may be a sort of AI when computers are programmed to find out information without human intervention.

The foremost common healthcare use cases for machine learning are automating medical billing, clinical decision support and therefore the development of clinical care guidelines. More importantly, scientists and researchers are using machine learning (ML) to churn out variety of smart solutions which will ultimately help in diagnosing and treating an illness.

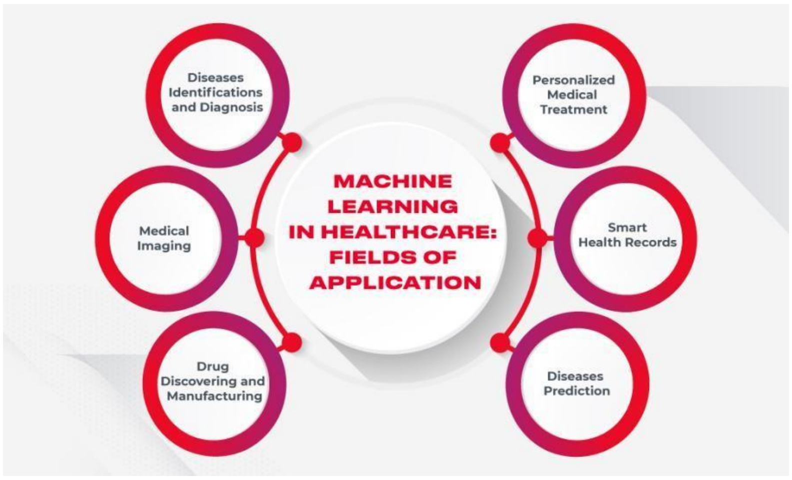

Diseases Identifications and Diagnosis
Personalized Medical Treatment
MACHINE LEARNING IN HEALTHCARE: FIELDS OF APPLICATION
Medical Imaging
Smart Health Records
Drug Discovering and Manufacturing
Diseases Prediction

Fig : ML in Health Care

4.PROJECT WORKFLOW

4.1 Introduction

Machine learning has given the computer systems the abilities to automatically learn without being explicitly programmed. But how does a machine learning system work? So, it can be described using the life cycle of machine learning. Machine learning life cycle is a cyclic process to build an efficient machine learning project. The main purpose of the life cycle is to find a solution to the problem or project.

Machine learning life cycle involves seven major steps, which are given below:

1. Gathering Data
2. Data preparation
3. Data Wrangling
4. Analyse Data
5. Train the model
6. Test the model
7. Deployment

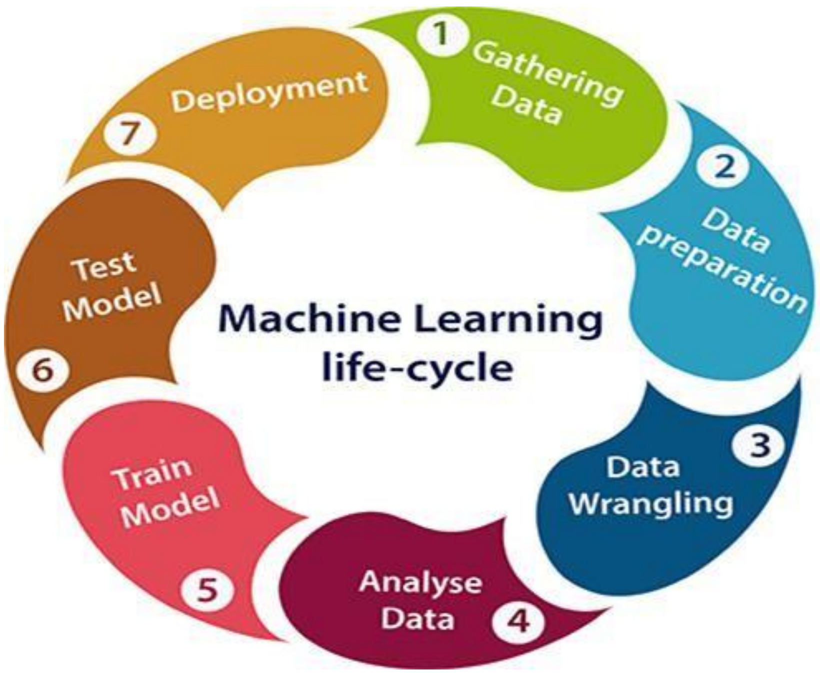


Fig : Steps in Machine Learning

4.2 Gathering Data

Data Gathering is the first step of our project. The goal of this step is to identify and obtain all data-related problems. In this step, we need to identify the different data sources, as data can be collected from various sources such as files, database, internet, or mobile devices. It is one of the most important steps. The quantity and quality of the collected data will determine the efficiency of the output. The more will be the data , the more accurate will be the prediction.

This step includes the below tasks:

- Identify various data sources
- Collect data
- Integrate the data obtained from different sources

By performing the above task, we get a coherent set of data, also called as a dataset.

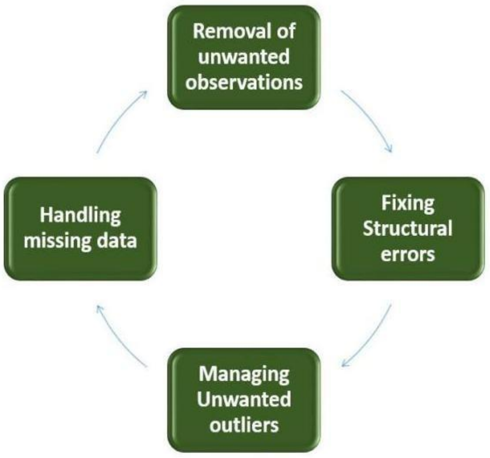
4.3 Data Wrangling

Data wrangling is the process of cleaning and converting raw data into a useable format. It is the process of cleaning the data, selecting the variable to use, and transforming the data in a proper format to make it more suitable for analysis in the next step. It is one of the most important steps of the complete process. Cleaning of data is required to address the quality issues. It is not necessary that data we have collected is always of our use as some of the data may not be useful. In real-world applications, collected data may have various issues, including:

- Missing Values
- Duplicate data
- Invalid data
- Noise

So, we use various filtering techniques to clean the data.

It is mandatory to detect and remove the above issues because it can negatively affect the quality of the outcome.

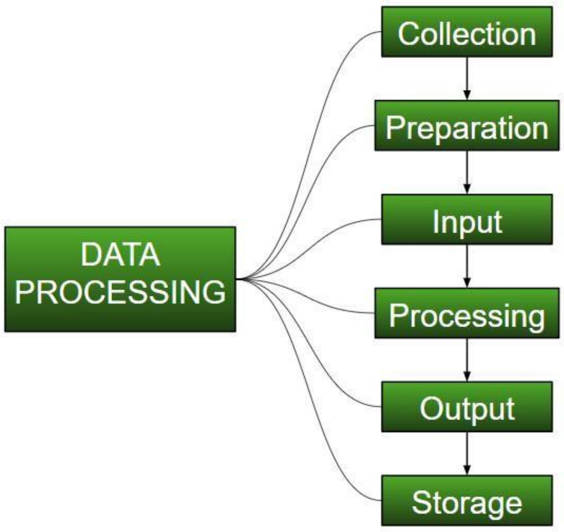


4.4 Data Preparation

After collecting the data, we need to prepare it for further steps. Data preparation is a step where we put our data into a suitable place and prepare it to use in our machine learning training. In this step, first, we put all data together, and then randomize the ordering of data. This step can be further divided into two processes:

Data exploration: It is used to understand the nature of data that we have to work with. We need to understand the characteristics, format, and quality of data. A better understanding of data leads to an effective outcome. In this, we find Correlations, general trends, and outliers.

Data pre-processing: Now the next step is preprocessing of data for its analysis.



4.5 Data Analysis

Now the cleaned and prepared data is passed on to the analysis step. This step involves:

- Selection of analytical techniques
- Building models
- Review the result

The aim of this step is to build a machine learning model to analyze the data using various analytical techniques and review the outcome. It starts with the determination of the type of the problems, where we select the machine learning techniques such as Classification, Regression, Cluster analysis, Association, etc. then build the model using prepared data, and evaluate the model. Hence, in this step, we take the data and use machine learning algorithms to build the model.

4.6 Train Model

Now the next step is to train the model, in this step we train our model to improve its performance for better outcome of the problem.

We use datasets to train the model using various machine learning algorithms.Training a model is required so that it can understand the various patterns, rules, and,features.

4.7 Test Model

Once our machine learning model has been trained on a given dataset, then we test the model. In this step, we check for the accuracy of our model by providing a test dataset to it. Testing the model determines the percentage accuracy of the model as per the requirement of project or problem.

4.8 Deployment

The last step of our project workflow is deployment, where we deploy the model in the real-world system. If the above-prepared model is producing an accurate result as per our requirement with acceptable speed, then we deploy the model in the real system. But before deploying the project, we will check whether it is improving its performance using available data or not. The deployment phase is similar to making the final report for a project.

5.CODE

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

sns.set()
train_data =
pd.read_excel(r"E:\MachineLearning\EDA\Flight_Price\Data_Train.
xlsx")
pd.set_option('display.max_columns', None)
train_data.head()
train_data.info()
train_data["Duration"].value_counts()
train_data.dropna(inplace = True)
train_data.isnull().sum()

#  From description we can see that Date_of_Journey is a object
data type,Therefore, we have to convert this datatype into
timestamp so as to use this column properly for predictionFor
this we require pandas to_datetime to convert object data type
to datetime dtype.

train_data["Journey_day"] =
pd.to_datetime(train_data.Date_of_Journey,
format="%d/%m/%Y").dt.day
train_data["Journey_month"] =
pd.to_datetime(train_data["Date_of_Journey"], format =
"%d/%m/%Y").dt.month
train_data.head()

# Since we have converted Date_of_Journey column into integers,
Now we can drop as it is of no use.

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

```
# Departure time is when a plane leaves the gate.
# Similar to Date_of_Journey we can extract values from
Dep_Time
# Extracting Hours

train_data["Dep_hour"] =
pd.to_datetime(train_data["Dep_Time"]).dt.hour

# Extracting Minutes

train_data["Dep_min"] =
pd.to_datetime(train_data["Dep_Time"]).dt.minute

# Now we can drop Dep_Time as it is of no use

train_data.drop(["Dep_Time"], axis = 1, inplace = True)
train_data.head()

# Arrival time is when the plane pulls up to the gate.
# Similar to Date_of_Journey we can extract values from
Arrival_Time
# Extracting Hours

train_data["Arrival_hour"] =
pd.to_datetime(train_data.Arrival_Time).dt.hour

# Extracting Minutes

train_data["Arrival_min"] =
pd.to_datetime(train_data.Arrival_Time).dt.minute

# Now we can drop Arrival_Time as it is of no use

train_data.drop(["Arrival_Time"], axis = 1, inplace = True)
train_data.head()

# 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:
        if "h" in duration[i]:
            duration[i] = duration[i].strip() + " 0m"
        else:
            duration[i] = "0h " + duration[i]
```

```
sns.catplot(y = "Price", x = "Source", data =
train_data.sort_values("Price", ascending = False),
kind="boxen", height = 4, aspect = 3)
plt.show()
duration_hours = []
duration_mins = []
for i in range(len(duration)):
    duration_hours.append(int(duration[i].split(sep = "h")[0]))
    duration_mins.append(int(duration[i].split(sep =
"m")[0].split()[-1]))
train_data["Duration_hours"] = duration_hours
train_data["Duration_mins"] = duration_mins
train_data.drop(["Duration"], axis = 1, inplace = True)
train_data.head()
train_data["Airline"].value_counts()

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

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

# As Airline is Nominal Categorical data we will perform
OneHotEncoding

Airline = train_data[["Airline"]]
Airline = pd.get_dummies(Airline, drop_first= True)
Airline.head()
train_data["Source"].value_counts()
sns.catplot(y = "Price", x = "Source", data =
train_data.sort_values("Price", ascending = False),
kind="boxen", height = 4, aspect = 3)
plt.show()
```

```
Source = train_data[["Source"]]
Source = pd.get_dummies(Source, drop_first= True)
Source.head()
train_data["Destination"].value_counts()
Destination = train_data[["Destination"]]
Destination = pd.get_dummies(Destination, drop_first = True)
Destination.head()
train_data["Route"]

# Additional_Info contains almost 80% no_info
# Route and Total_Stops are related to each other

train_data.drop(["Route", "Additional_Info"], axis = 1, inplace
= True)
train_data["Total_Stops"].value_counts()

# As this is case of Ordinal Categorical type we perform
LabelEncoder
# Here Values are assigned with corresponding keys

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

# Concatenate dataframe --> train_data + Airline + Source +
Destination

data_train = pd.concat([train_data, Airline, Source,
Destination], axis = 1)
data_train.head()
data_train.drop(["Airline", "Source", "Destination"], axis = 1,
inplace = True)
data_train.head()
data_train.shape
test_data =
pd.read_excel(r"E:\MachineLearning\EDA\Flight_Price\Test_set.xl
sx")
test_data.head()
```

```
# Preprocessing

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

# EDA
# Date_of_Journey

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

# Dep_Time

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

# Arrival_Time

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

# Duration
duration = list(test_data["Duration"])
for i in range(len(duration)):
    if len(duration[i].split()) != 2:
        if "h" in duration[i]:
            duration[i] = duration[i].strip() + " 0m"
        else:
            duration[i] = "0h " + duration[i]
duration_hours = []
duration_mins = []
```

```
for i in range(len(duration)):
    duration_hours.append(int(duration[i].split(sep = "h")[0]))
    duration_mins.append(int(duration[i].split(sep =
"m")[0].split()[-1]))
test_data["Duration_hours"] = duration_hours
test_data["Duration_mins"] = duration_mins
test_data.drop(["Duration"], axis = 1, inplace = True)

# Categorical Data

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

# Additional_Info contains almost 80% no_info
# Route and Total_Stops are related to each other

test_data.drop(["Route", "Additional_Info"], axis = 1, inplace =
True)

# Replacing Total_Stops

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

# Concatenate dataframe --> test_data + Airline + Source +
Destination

data_test = pd.concat([test_data, Airline, Source, Destination],
axis = 1)
data_test.drop(["Airline", "Source", "Destination"], axis = 1,
inplace = True)
print()
print()
print("Shape of test data : ", data_test.shape)
print()
print()
print("Shape of test data : ", data_test.shape)
data_test.head()
data_train.shape
data_train.columns
X = data_train.loc[:, ['Total_Stops', 'Journey_day',
'Journey_month', 'Dep_hour',
       'Dep_min', 'Arrival_hour', 'Arrival_min',
'Duration_hours',
       'Duration_mins', '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()
y = data_train.iloc[:, 1]
y.head()

#Finds correlation between Independent and dependent attributes

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

# Important feature using ExtraTreesRegressor

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

```
#plot graph of feature importances for better visualization

plt.figure(figsize = (12,8))
feat_importances = pd.Series(selection.feature_importances_,
index=X.columns)
feat_importances.nlargest(20).plot(kind='barh')
plt.show()
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)
from sklearn.ensemble import RandomForestRegressor
reg_rf = RandomForestRegressor()
reg_rf.fit(X_train, y_train)
y_pred = reg_rf.predict(X_test)
reg_rf.score(X_train, y_train)
reg_rf.score(X_test, y_test)
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()
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)))

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

2090.5509/(max(y)-min(y))
metrics.r2_score(y_test, y_pred)
from sklearn.model_selection import RandomizedSearchCV
n_estimators = [int(x) for x in np.linspace(start = 100, stop =
1200, num = 12)]
max_features = ['auto', 'sqrt']
max_depth = [int(x) for x in np.linspace(5, 30, num = 6)]
min_samples_split = [2, 5, 10, 15, 100]
min_samples_leaf = [1, 2, 5, 10]
```

```
# Create the random grid

random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf}
rf_random = RandomizedSearchCV(estimator = reg_rf,
param_distributions =
random_grid,scoring='neg_mean_squared_error', n_iter = 10, cv =
5, verbose=2,
rf_random.fit(X_train,y_train)
rf_random.best_params_
prediction = rf_random.predict(X_test)
plt.figure(figsize = (8,8))
sns.distplot(y_test-prediction)
plt.show()
plt.figure(figsize = (8,8))
plt.scatter(y_test, prediction, alpha = 0.5)
plt.xlabel("y_test")
plt.ylabel("y_pred")
plt.show()
plt.figure(figsize = (8,8))
plt.scatter(y_test, prediction, alpha = 0.5)
plt.xlabel("y_test")
plt.ylabel("y_pred")
plt.show()

#Saving the model to reuse it

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(reg_rf, file)
model = open('flight_price_rf.pkl','rb')
forest = pickle.load(model)
y_prediction = forest.predict(X_test)
metrics.r2_score(y_test, y_prediction)
```

6. GRAPHICAL ANALYSIS

Fig 1: Dependency of Parameters on each other

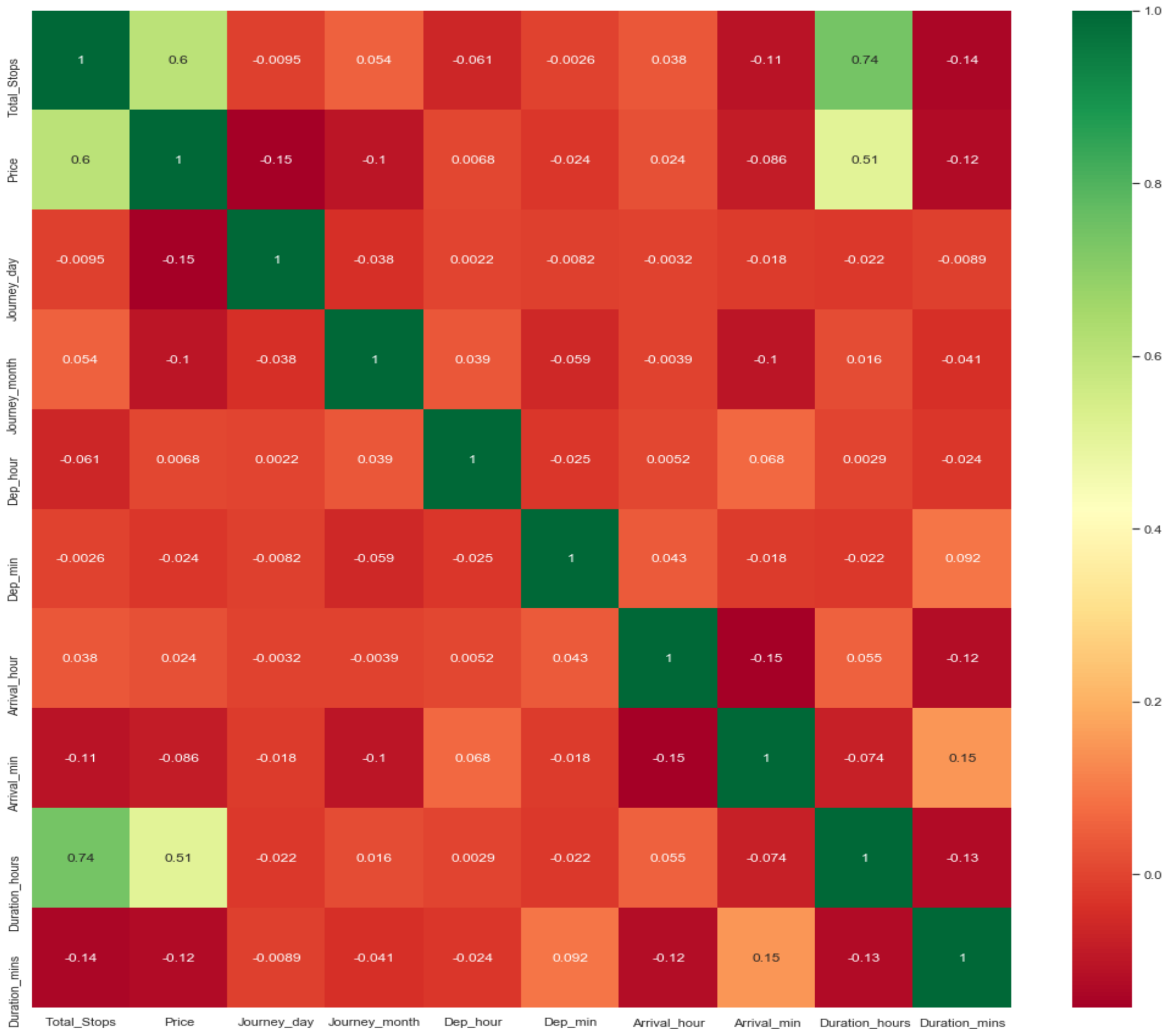
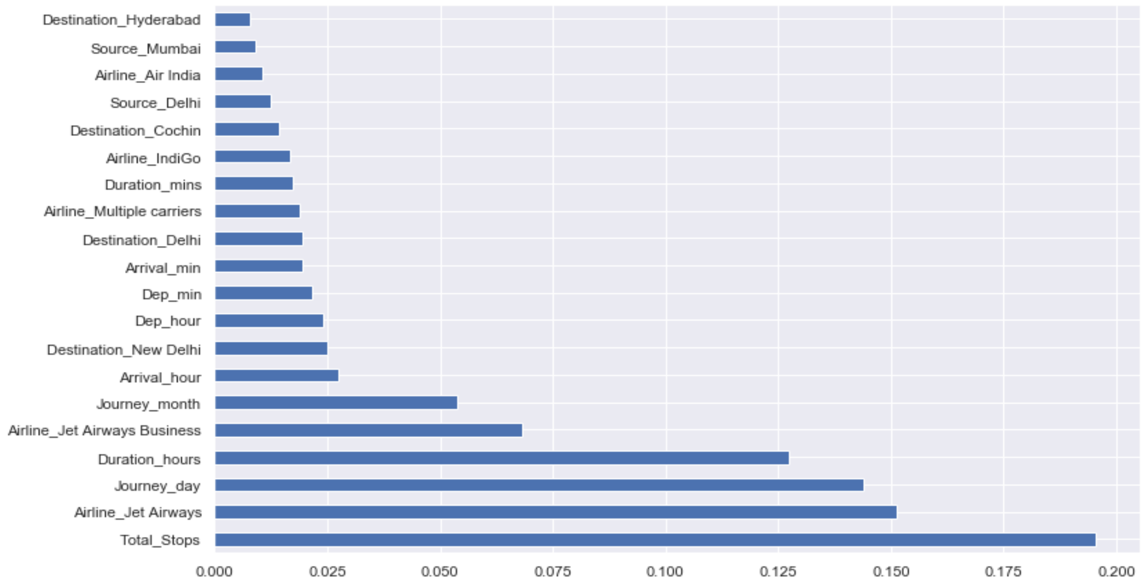
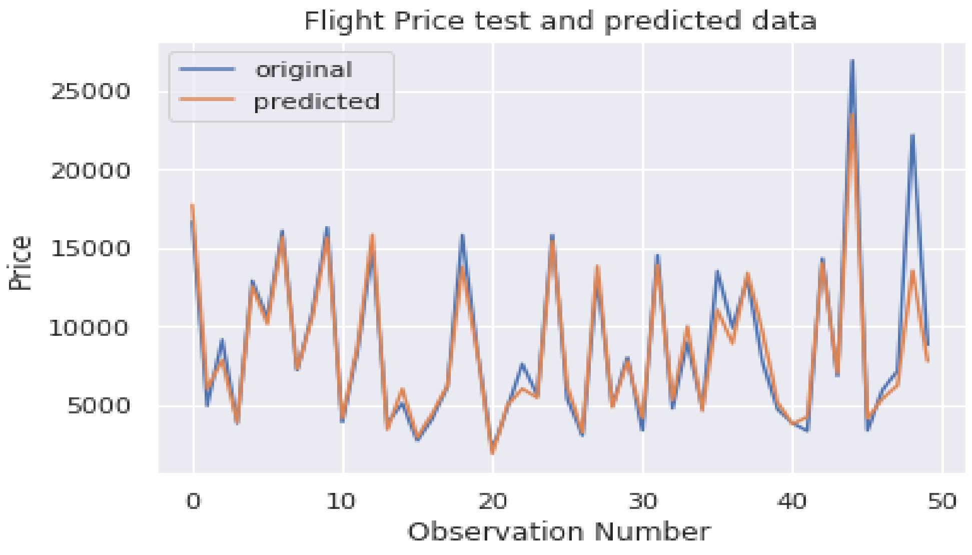


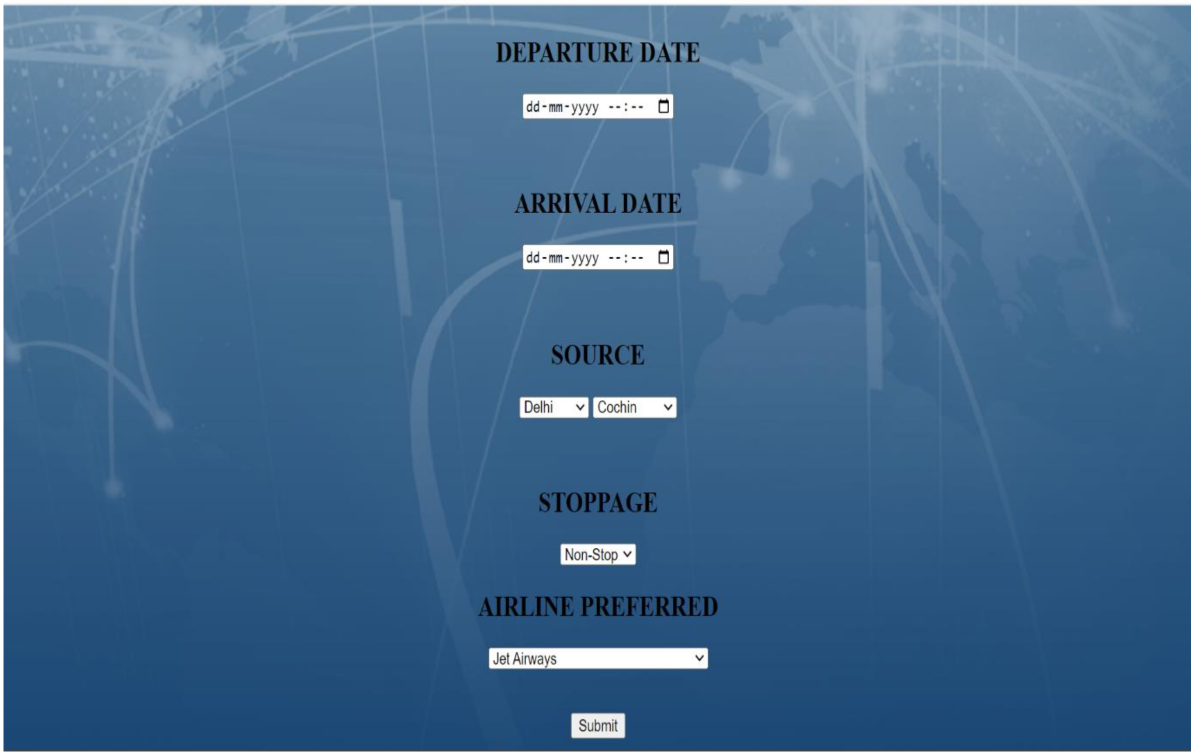
Fig 2: Graph of feature importances





7.PROTOTYPE

Fig 1: FLIGHT PRICE PREDICTION WEBPAGE



DEPARTURE DATE

dd-mm-yyyy --:--

ARRIVAL DATE

dd-mm-yyyy --:--

SOURCE

Delhi Cochin

STOPPAGE

Non-Stop

AIRLINE PREFERRED

Jet Airways

8.FUTURE DEVELOPMENTS

According to the outcome of this paper reported on a basic analysis on “flight fare price prediction”. We gathered transportation information from a selected Indian airline from the online and showed that it is possible to predict costs for flights supported historical fare information. We have known the compliments and complaints of shoppers, variations in sentiment over an amount of your time. This analysis provides a general opinion of passengers towards airlines. XG-boost gives better results compared to other machine learning models. The dataset we have collected from the Kaggle consisting of more than 2000 Indian airlines flight data and deprived that it is easy to forecast the prices of airlines based on previous price data. The outcome of the experiments derives from the fact that machine learning models satisfies the need of forecasting the airfare costs. Other vital parts in airfare prediction are the feature selection and data collection from which we have derived some helpful conclusions. We have derived some features which affect the airfare forecasting at most using experiments. Apart from the options elect, there are different options that would improve the price forecasting accuracy. In the future, this work could be extended to predict the airfare prices for the entire flight map of the airline. We need to test these machine learning algorithms on the huge airline datasets, but the preliminary studies remark on the capabilities of ML Models to help the end users to give an idea when to buy the tickets in what period so they will be in profit. Features took from external factors such as social media data and search engine query are not taken. Therefore, we will introduce and discuss the concept of using social media data for ticket/demand prediction (i.e., twitter sentiment analysis). In today's date, social media sentiment analysis has become an enough library of knowledge for varied data processing models. For example, social media data has been used for event prediction, price prediction and tourist traffic flow prediction. A similar approach might be followed to extract helpful social media info associated with various external factors poignant airline rider demand and price tag value. For example, Analysis of various twitter hash tags might provide valuable information concerning the presence of an occurrence at an origin/destination city, competitors’ promotions, volume of traveler traffic flow, weather condition, economic activity etc. This in turns may allow us to predict the amendment in price tag price/demand.

It is expected that a data processing model that utilizes info ensuing from social media information would provide higher results than existing ad forecasting route demand and our price ticket worth. But there are no facilities or studies available on the Web that use social media data to forecast the demand of path or cost value. There is room for improvements in several areas including predicting exact value of ticket prices/demand, dataset issues, limited number of features, lacking generality, better prediction techniques and performance and complexity issues.

9.REFERENCES

[1] Prediction/blob/master/flight_price.ipynb
[2] https://www.kaggle.com/datasets/nikhilmittal/flight-fare-prediction-mh
[3] https://flight-price-prediction-api.herokuapp.com/
[4] https://www.datascience2000.in/2021/12/flight-fare-prediction-using-machine.html
[5]https://www.researchgate.net/publication/335936877_A_Framework_for_Airfare_Price_Prediction_A_Machine_Learning_Approach