A

Mini Project

On

AIRFARE PRICES PREDICTION USING MACHINE LEARNING TECHNIQUES

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

Yashwanth Kumaar Kakani (217R1A05R8)

U. Vinod Kumar (227R5A0523)

G. Manoj Reddy (217R1A05M7)

Under the Guidance of

Dr. B. LAXMAIAH

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

2021-25

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**AIRFARE PRICES PREDICTION USING MACHINE LEARNING TECHNIQUES**" being submitted **YASHWANTH KUMAAR KAKANI (217R1A05R8), U. VINOD KUMAR (227R5A0523)** and **G. MANOJ REDDY (217R1A05M7)** in partial fulfillment of the requirements for the award of the degree of B. Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2024-25.

The results embodied in this project have not been submitted to any other University or Institute for the award of any degree or diploma.

Dr. B. Laxmaiah
(Associate Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. Nuthanakanti Bhaskar
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on ________________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to our guide **Dr. B. Laxmaiah**, Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) Coordinators: **Dr. J. Narasimha Rao, Mr. K. Ranjith Reddy, Dr. K. Maheshwari, Mrs. K. Shilpa** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. Nuthanakanti Bhaskar,** Head of the Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We would like to express our sincere gratitude to **Sri. Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

K. YASHWANTH KUMAAR (217R1A05R8)

U. VINOD KUMAR (227R5A0523)

G. MANOJ REDDY (217R1A05M7)

ABSTRACT

Airfare price prediction using machine learning techniques is an emerging area that aims to forecast flight prices accurately, helping travelers and businesses make informed decisions. This task involves analyzing historical flight data and extracting meaningful patterns to predict future price trends. Machine learning models such as regression analysis, decision trees, support vector machines, and deep learning techniques are commonly applied to capture the complex relationships between various factors like booking time, seasonality, demand, airline pricing strategies, and route characteristics. By leveraging large datasets and advanced algorithms, these models can provide dynamic pricing forecasts with high accuracy, assisting users in identifying the best time to purchase tickets and optimizing travel costs. Moreover, the integration of real-time data and advanced techniques like decision tree, random forest algorithm and Regression tree can further enhance the precision of airfare predictions, making it a valuable tool in the travel industry.

Key points: Airfare, Machine learning models, Regression analysis, support vector machine, Decision tree, Regression tree.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

This project aims to predict airfare prices by applying machine learning techniques to large datasets of historical flight data. It will forecast price trends to help travelers and businesses optimize travel costs and ticket purchasing strategies. In this the models will identify patterns in pricing based on factors like booking time, demand, and route characteristics. The scope also includes integrating real-time data sources and offering ersonalized recommendations based on user preferences.

1.2 PROJECT PURPOSE

The purpose of the project is to build a system that can accurately predict airfare prices, allowing users to make informed decisions on when to book flights. This system will help travelers find the best times to purchase tickets and avoid overpaying. Airlines and travel agencies can also benefit by optimizing their pricing strategies based on the forecasted data.

1.3 PROJECT FEATURES

The features of this Airfare Price Prediction project focus on leveraging advanced machine learning techniques to provide accurate, real-time airfare forecasts. This uses Advanced ML algorithms to analyze historical flight data and identify patterns that influence pricing, including booking time, demand, and airline competition. It offers personalized predictions tailored to user preferences, such as favorite airlines or travel dates, improving the relevance of the forecast. Additionally, the system integrates real-time data, ensuring that predictions reflect current market conditions. Users are provided with transparent predictions, complete with explanatory insights into why prices may rise or fall, giving them greater confidence in making informed booking decisions. The system's adaptability allows it to quickly respond to sudden events like promotions or price fluctuations, ensuring accuracy and user satisfaction.

2. SYSTEM ANALYSIS

1. SYSTEM ANALYSIS

2.1 SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.2 PROBLEM DEFINITION

The problem definition for this Airfare Price Prediction project stems from the inherent complexity and unpredictability of airfare pricing in the airline industry. Airline ticket prices are influenced by a multitude of dynamic factors, including demand, seasonality, airline competition, booking time, fuel costs, economic conditions, and even geopolitical events. Traditional methods, such as relying on historical price trends or expert opinions, fail to capture the complex, non-linear relationships between these factors, often leading to inaccurate or incomplete forecasts. Additionally, existing price prediction tools provide broad recommendations without offering detailed reasoning behind their forecasts. They struggle to adapt to sudden market shifts, such as promotions or unexpected global events, which can drastically alter prices. This project aims to address these challenges by developing a machine learning-based system capable of overcoming the limitations of existing methods.

2.3 EXISTING SYSTEM

Existing systems for airfare price prediction largely rely on traditional data analysis methods combined with some basic machine learning models. Popular platforms like Google Flights, Hopper, and Skyscanner offer users historical price data and price alerts, notifying travelers when prices are likely to rise or fall. These systems utilize large datasets that include historical pricing trends, seasonality, and demand patterns, alongside simple predictive algorithms to estimate future prices. Some apps, like Hopper, use machine learning models to forecast price trends, offering advice on the best time to book a flight, whether to buy now or wait. However, many of these systems still have limitations in terms of accuracy, as they struggle to account for sudden, unpredictable events such as changes in airline policies, fuel prices, or geopolitical factors. Additionally, existing systems often provide limited insight into the specific factors driving price fluctuations, leaving users with broad recommendations that lack detailed explanations or confidence levels. As machine learning techniques continue to evolve, there is significant room for improvement in the precision and reliability of airfare prediction models in these systems.

2.3.1 LIMITATIONS OF EXISTING SYSTEM

- Existing systems cannot fully account for sudden events such as fuel price hikes or airline promotions.
- Broad recommendations (e.g., “buy now” or “wait”) are given without clear explanations for users.
- Limited incorporation of real-time data, leading to less accurate predictions.
- The systems lack personalized predictions based on individual user preferences like preferred airlines or frequent flyer programs.
- Existing tools are slow to adapt to new data, which can result in missed opportunities to secure the best prices.

2.4 PROPOSED SYSTEM

The proposed system for Airfare Price Prediction improves accuracy and adaptability by utilizing machine learning algorithms such as SVM, regression trees, random forest, linear regression, ELM, and MLP. It integrates real-time data like airline pricing feeds and global events to provide dynamic, up-to-date predictions. Unlike existing systems, it offers personalized forecasts based on user preferences (e.g., preferred airlines, loyalty programs) and includes transparent explanations of why prices are expected to rise or fall. The system is also highly adaptive, quickly responding to market shifts, making it more reliable and relevant for travelers seeking to book at the optimal time.

2.4.1 ADVANTAGES OF THE PROPOSED SYSTEM

- **Accuracy is improved** by using advanced machine learning techniques
- **Integrates Real-Time Data** such as airline pricing, global events, and weather conditions, for dynamic and timely predictions.
- **Personalized Recommendations** Can be offered which makes user satisfy.
- **Adaptability** Adjusts to sudden market changes.
- **Consumer & Business Benefits**

2.5 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis are .

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.5.1 ECONOMIC FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased. Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

2.5.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.5.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it.

2.6 HARDWARE & SOFTWARE REQUIREMENTS

2.6.1 HARDWARE REQUIREMENTS:

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- Processor : Intel Dual Core@ CPU 2.90GHz.
- Hard disk : 256 GB and Above.
- Memory : 4GB RAM and Above.

2.6.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

- Operating system : Windows 8 and Above.
- Languages : Python

3.ARCHITECTURE

2. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for Airfare prices prediction using machine learning, starting from input to final output.

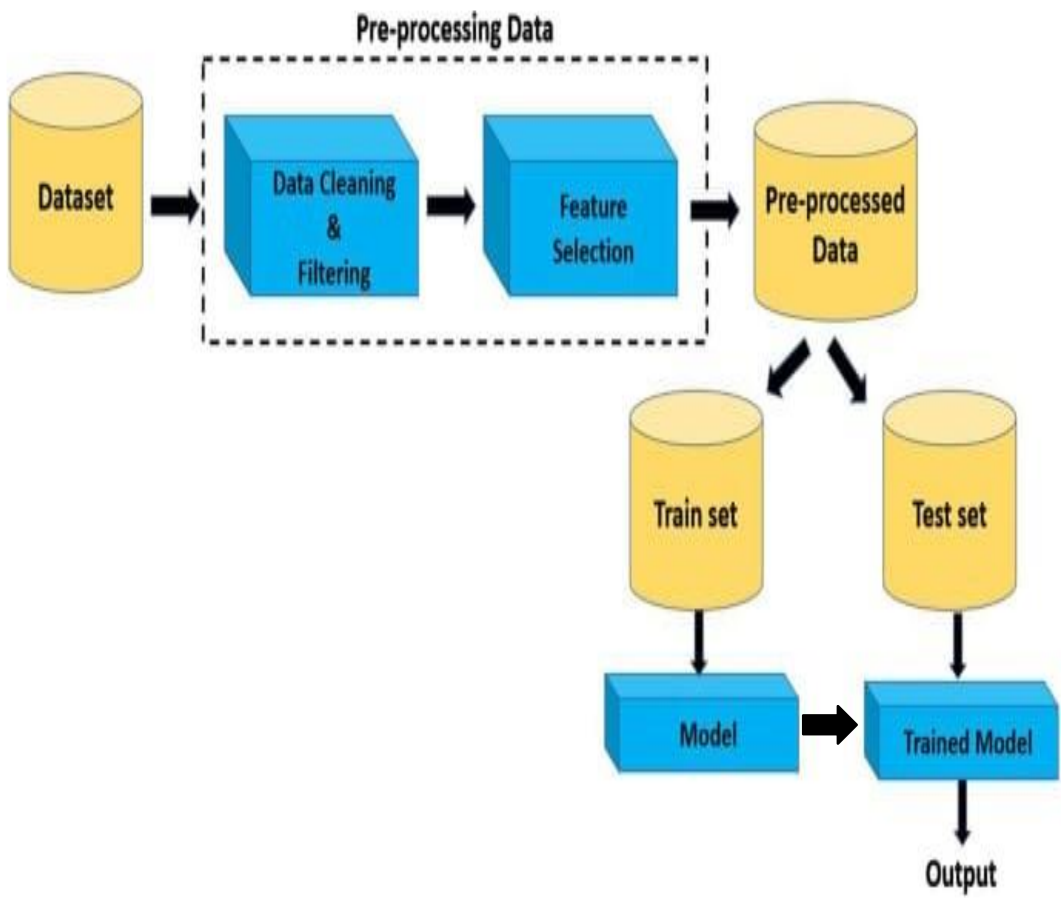


Figure 3.1: Project Architecture of Airfare Prices Prediction Using Machine Learning Techniques

3.2 DESCRIPTION

1. **Objective**: To accurately predict airfare prices using machine learning algorithms, helping users identify the best times to book flights.
2. **Machine Learning Algorithms**: Utilizes algorithms such as Support Vector Machines (SVM), regression trees, random forest, linear regression, Extreme Learning Machine (ELM), and Multi-Layer Perceptron (MLP) for prediction.
3. **Data Analysis**: Analyzes historical flight data to identify patterns in pricing, influenced by factors like booking time, demand, and airline competition.
4. **Real-Time Data Integration**: Incorporates real-time data, such as current airline pricing trends, to provide dynamic and timely predictions.
5. **Personalization**: Offers personalized predictions tailored to user preferences (e.g., favorite airlines, travel dates, loyalty programs), making the forecasts more relevant.
6. **Transparency**: Provides clear explanations for price changes, offering insights into why prices are expected to rise or fall.
7. **Use Case**: Aids both travelers in saving on airfare and airlines in optimizing pricing strategies, providing a data-driven tool for the travel industry.
8. **Adaptability**: The system can quickly adapt to market changes, ensuring that users receive the most accurate and up-to-date forecasts.

3.3 USE CASE DIAGRAM

In the use case diagram we have basically two actors who are the user and the system. The user will upload the airline dataset. Whereas the system has to preprocess the data and then extracts the necessary features.

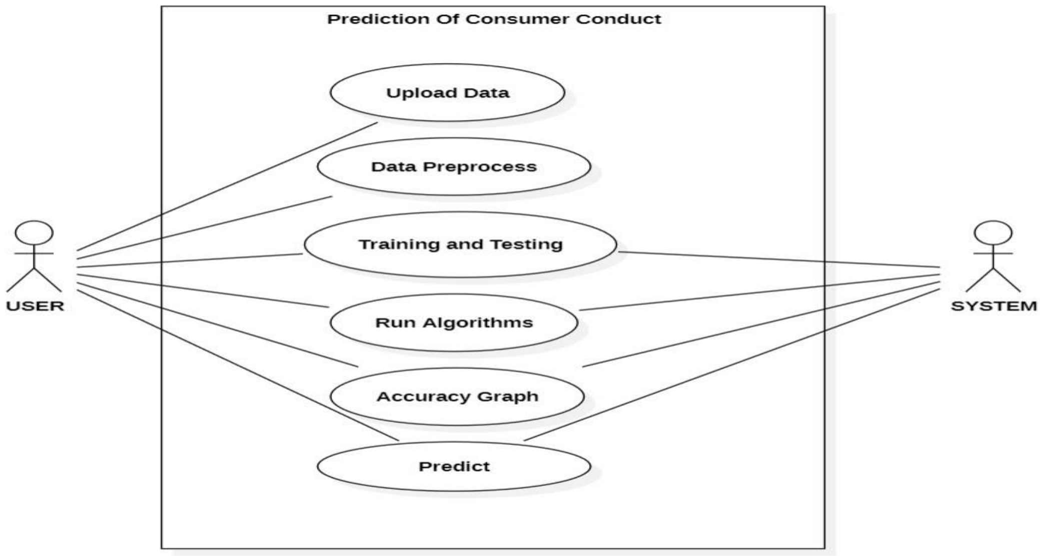


Figure 3.3: Use Case Diagram for User and System for Airfare Prices prediction Using Machine Learning Techniques

3.4 CLASS DIAGRAM

Class Diagram is a collection of classes and objects. In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes.

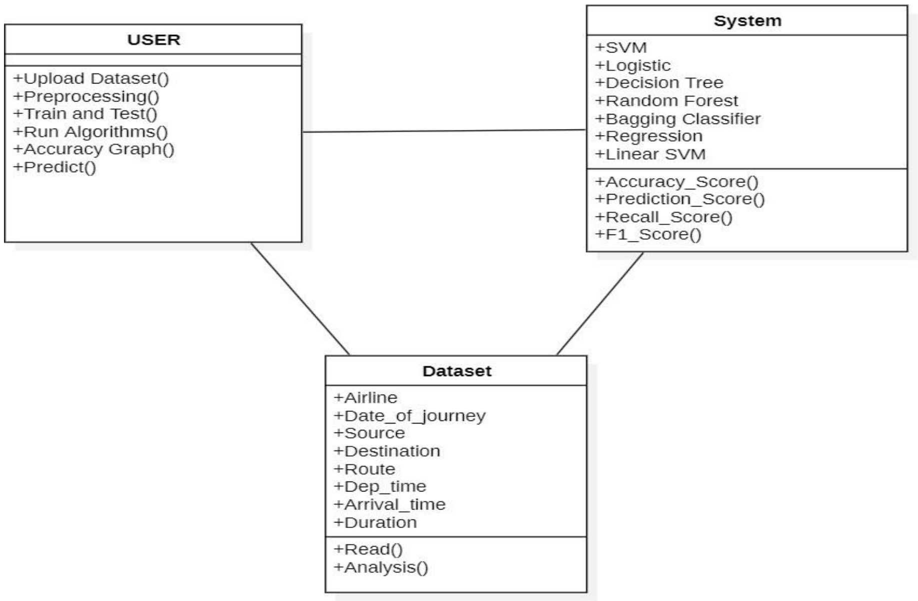


Figure 3.4: Class Diagram for User , Dataset and System for Airfare Prices Prediction Using Machine Learning techniques

3.5 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

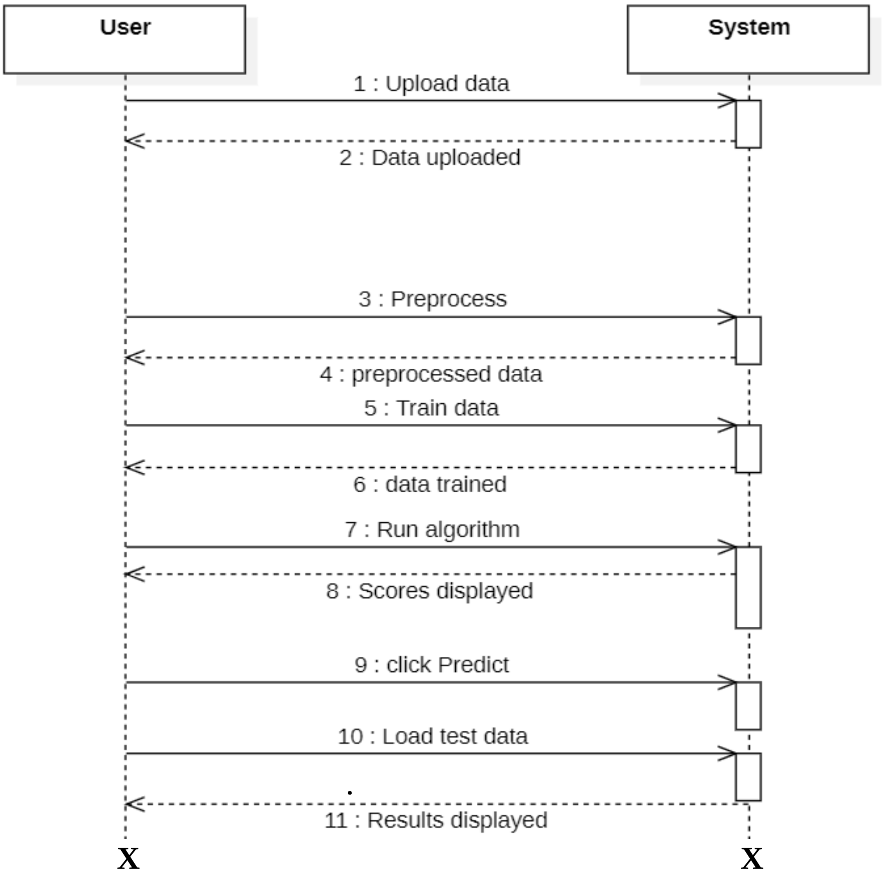


Figure 3.5: Sequence Diagram for User and System for Airfare Prices Prediction Using Machine Learning techniques

3.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and the actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

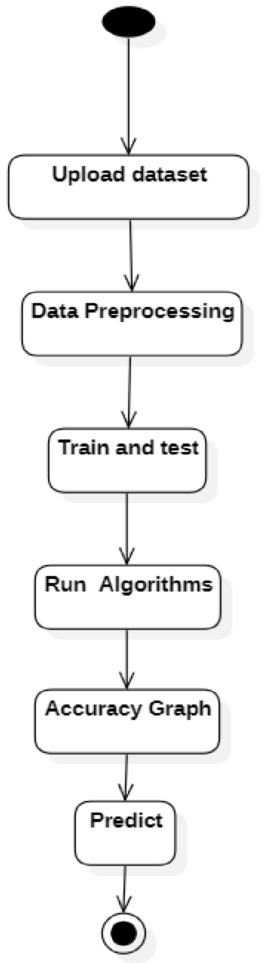


Figure 3.6: Activity Diagram for User and System for Airfare Prices Prediction Using Machine Learning techniques

4. IMPLEMENTATION

4. IMPLEMENTATION

4.1 SAMPLE CODE

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
import matplotlib.pyplot as plt
from tkinter.filedialog import askopenfilename
import pandas as pd
import numpy as np
import os
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.svm import SVR  #SVR regression
from sklearn.ensemble import RandomForestRegressor #random forest regression class
from sklearn.neural_network import MLPRegressor
from sklearn_extensions.extreme_learning_machines.elm import GenELMRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import BaggingRegressor
from sklearn.preprocessing import LabelEncoder
import timeit

main = Tk()
main.title("A Holistic Approach on Airfare Price Prediction Using Machine Learning
Techniques")
main.geometry("1300x1200")


global filename
global dataset
global X, Y, accuracy, le
sc = MinMaxScaler(feature_range = (0, 1))
global X_train, X_test, y_train, y_test

def uploadDataset():
    global filename



   global dataset
   text.delete('1.0', END)

 filename = askopenfilename(initialdir = "Dataset")
    tf1.insert(END,str(filename))
    text.insert(END,"Dataset Loaded\n\n")
```

```
dataset = pd.read_csv(filename,nrows=200)
    text.insert(END,str(dataset.head()))


    def preprocessDataset():
    global dataset
    global X, Y, sc, le
    global X_train, X_test, y_train, y_test
    text.delete('1.0', END)
    dataset.fillna(0, inplace = True)
    dataset.drop(['Date_of_Journey'], axis = 1,inplace=True)
    cols =
['Airline','Source','Destination','Route','Additional_Info','Dep_Time','Arrival_Time','Du
ration','Total_Stops']
    le = LabelEncoder()
    for i in range(len(cols)):
        dataset[cols[i]] = pd.Series(le.fit_transform(dataset[cols[i]].astype(str)))
    dataset = dataset.values
    Y = dataset[:,dataset.shape[1]-1]
    X = dataset[:,0:dataset.shape[1]-1]
    Y = Y.reshape(-1, 1)
    sc = MinMaxScaler(feature_range = (0, 1))

    X = sc.fit_transform(X)
    Y = sc.fit_transform(Y)
    text.insert(END,str(X)+"\n\n")
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)

    def predictPrice(algorithm, predict, test, algorithm_name, execution_time):
    global X, Y
    predict = predict.reshape(predict.shape[0],1)
    predict = sc.inverse_transform(predict)
    predict = predict.ravel()
    labels = sc.inverse_transform(test)


labels = labels.ravel()
    acc = algorithm.score(X,Y.ravel())
    accuracy.append(acc)
    text.insert(END,algorithm_name+" Accuracy : "+str(acc)+"\n\n")
    text.insert(END,algorithm_name+" Execution Time : "+str(execution_time)+"
Seconds\n\n")
    for i in range(0,20):

text.insert(END,"Original Price : "+str(labels[i])+algorithm_name+" Predicted Price :
"+str(predict[i])+"\n")
    plt.plot(labels, color = 'red', label = 'Original Test Prices')
    plt.plot(predict, color = 'green', label = algorithm_name+' Predicted Prices')
    plt.title(algorithm_name+' Predicted Prices')
    plt.xlabel('Original Test Prices')
    plt.ylabel(algorithm_name+' Airfare Predicted & Test Prices Comparison')
```

```
    plt.legend()

plt.show()

def runMLP():
    global X, Y, accuracy
    global X_train, X_test, y_train, y_test
    accuracy = []
    text.delete('1.0', END)
    start = timeit.default_timer()
    mlp = MLPRegressor()
    mlp.fit(X, Y.ravel())
    predict = mlp.predict(X_test)
    end = timeit.default_timer()
    predictPrice(mlp, predict, y_test, "MLP Algorithm", (end - start))

def runELM():
    global X, Y, accuracy
    global X_train, X_test, y_train, y_test
    accuracy = []
    text.delete('1.0', END)
    start = timeit.default_timer()
    elm_algorithm = GenELMRegressor()
    elm_algorithm.fit(X, Y.ravel())
    predict = elm_algorithm.predict(X_test)
    end = timeit.default_timer()
    predictPrice(elm_algorithm, predict, y_test, "ELM Algorithm", (end start))


def runRandomForest():
    text.delete('1.0', END)
    start =  timeit.default_timer()
    rf_algorithm = RandomForestRegressor()
    rf_algorithm.fit(X, Y.ravel())
    predict = rf_algorithm.predict(X_test)
    end = timeit.default_timer()
    predictPrice(rf_algorithm, predict, y_test, "Random Forest Algorithm", (end -
start))

def runRegressionTree():
    text.delete('1.0', END)
    start = timeit.default_timer()
    rt_algorithm = DecisionTreeRegressor()
    rt_algorithm.fit(X, Y.ravel())
    predict = rt_algorithm.predict(X_test)
    end = timeit.default_timer()
    predictPrice(rt_algorithm, predict, y_test, "Regression Tree Algorithm", (end -
start))

def runBagging():
    text.delete('1.0', END)
    start = timeit.default_timer()
```

```
    br_algorithm = BaggingRegressor()

br_algorithm.fit(X, Y.ravel())
    predict = br_algorithm.predict(X_test)
    end = timeit.default_timer()
    predictPrice(br_algorithm, predict, y_test, "Bagging Regressor Algorithm",
(end - start))

def runPolySVM():
    text.delete('1.0', END)
    start = timeit.default_timer()
    poly_svm_algorithm = SVR(kernel="poly")
    poly_svm_algorithm.fit(X, Y.ravel())
    predict = poly_svm_algorithm.predict(X_test)
    end = timeit.default_timer()
    predictPrice(poly_svm_algorithm, predict, y_test, "Polynomial SVM
Algorithm", (end - start))

def runLinearSVM():
    text.delete('1.0', END)


start = timeit.default_timer()

linear_svm_algorithm = SVR(kernel="linear")

linear_svm_algorithm.fit(X, Y.ravel())

predict = linear_svm_algorithm.predict(X_test)
    end = timeit.default_timer()
    predictPrice(linear_svm_algorithm, predict, y_test, "Linear SVM Algorithm",
(end - start))

def runLinearRegression():
    text.delete('1.0', END)
    start = timeit.default_timer()
    linear_algorithm = LinearRegression()

 linear_algorithm.fit(X, Y.ravel())
    predict = linear_algorithm.predict(X_test)
    end = timeit.default_timer()
    predictPrice(linear_algorithm, predict, y_test, "LinearRegression Algorithm",
(end - start))


def graph():
    height = accuracy
    bars = ('MLP','ELM','Random Forest','Regression Tree', 'Bagging Regressor',
'Poly SVM', 'Linear SVM', 'Linear Regression')
    y_pos = np.arange(len(bars))
    plt.bar(y_pos, height)
    plt.xticks(y_pos, bars)
```

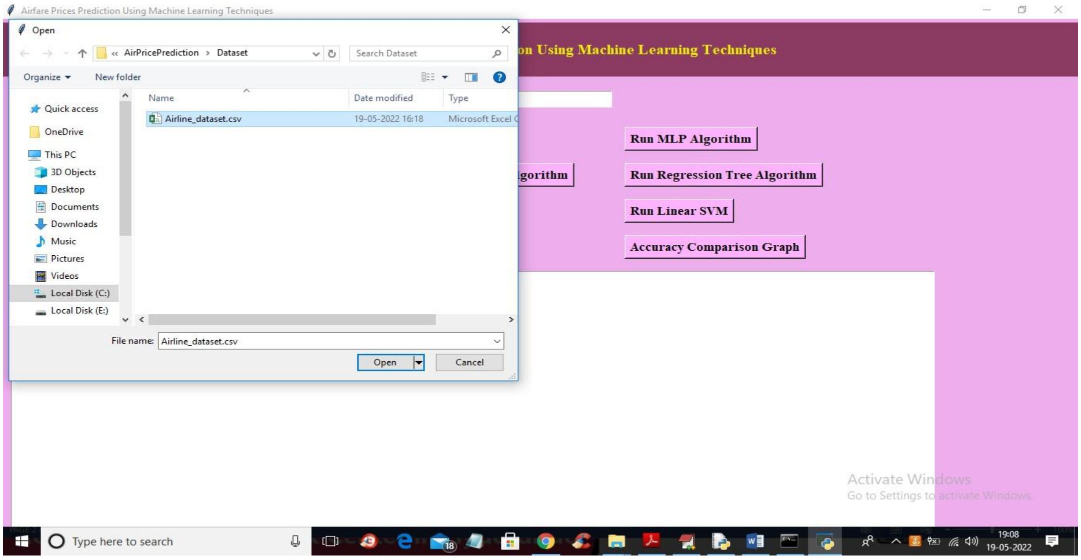
```
plt.title("All Algorithms Accuracy Comparison")

plt.show()

font = ('times', 15, 'bold')
title = Label(main, text='A Holistic Approach on Airfare Price Prediction Using
Machine Learning Techniques')
title.config(bg='HotPink4', fg='yellow2')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)

font1 = ('times', 13, 'bold')
ff = ('times', 12, 'bold')


l1 = Label(main, text='Dataset Location:')


l1.config(font=font1)
l1.place(x=50,y=100)


tf1 = Entry(main,width=60)
tf1.config(font=font1)

tf1.place(x=230,y=100)
uploadButton = Button(main, text="Upload Airfare Prices Dataset",
command=uploadDataset, bg='#ffb3fe')
uploadButton.place(x=50,y=150)
uploadButton.config(font=font1)

preprocessButton = Button(main, text="Preprocess Dataset",
command=preprocessDataset, bg='#ffb3fe')
preprocessButton.place(x=470,y=150)

preprocessButton.config(font=font1)

mlpButton = Button(main,text="Run MLP Algorithm", command=runMLP,
bg='#ffb3fe')
mlpButton.place(x=790,y=150)
mlpButton.config(font=font1)

elmButton = Button(main,text="Run ELM Algorithm", command=runELM,
bg='#ffb3fe')
elmButton.place(x=50,y=200)
elmButton.config(font=font1)

rfButton = Button(main,text="Train Random Forest Algorithm",
command=runRandomForest, bg='#ffb3fe')
rfButton.place(x=470,y=200)
rfButton.config(font=font1)
```

```
rtButton = Button(main,text="Run Regression Tree Algorithm",
command=runRegressionTree, bg='#ffb3fe')
rtButton.place(x=790,y=200)
rtButton.config(font=font1)

brButton = Button(main,text="Run BaggingRegressor
Algorithm",command=runBagging, bg='#ffb3fe')


brButton.place(x=50,y=250)
brButton.config(font=font1)



polysvmButton = Button(main,text="Run Polynomial SVM",
command=runPolySVM, bg='#ffb3fe')

polysvmButton.place(x=470,y=250)
polysvmButton.config(font=font1)

linearsvmButton = Button(main,text="Run Linear SVM",
command=runLinearSVM, bg='#ffb3fe')
linearsvmButton.place(x=790,y=250)
linearsvmButton.config(font=font1)

linearButton = Button(main,text="Run LinearRegression Algorithm",
command=runLinearRegression, bg='#ffb3fe')
linearButton.place(x=50,y=300)
linearButton.config(font=font1)

graphButton = Button(main,text="Accuracy Comparison Graph",
command=graph, bg='#ffb3fe')

graphButton.place(x=790,y=300)
graphButton.config(font=font1)

font1 = ('times', 13, 'bold')
text=Text(main,height=20,width=130)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=10,y=350)
text.config(font=font1)

main.config(bg='plum2')
main.mainloop()
```

5. SCREENSHOTS

5.SCREENSHOTS

5.1 UPLOAD DATASET

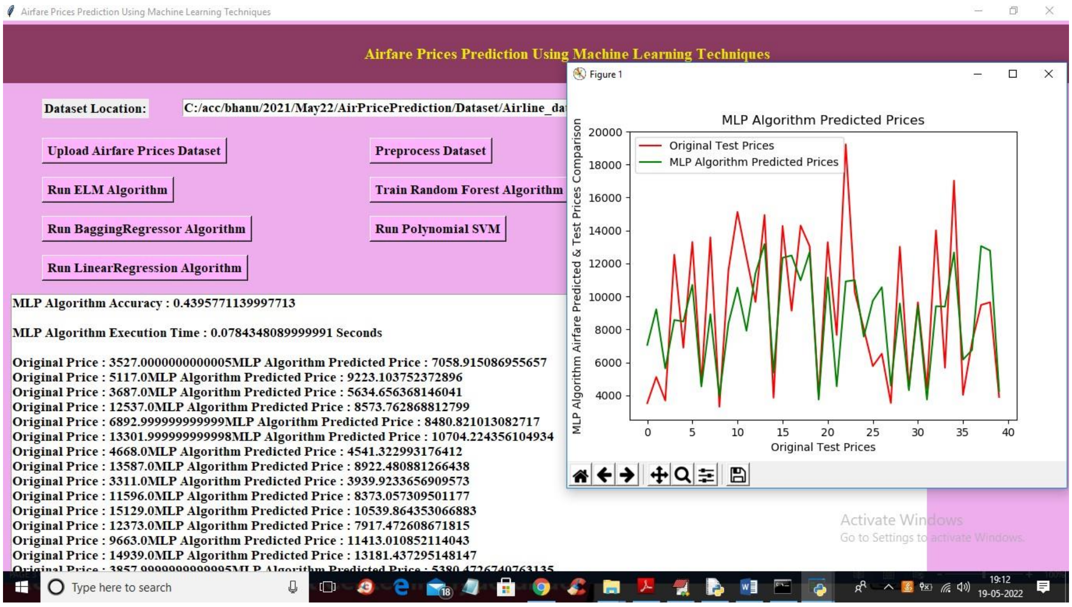


Screenshot 5.1 : In above screen selecting and uploading dataset and then click on ‘Open’ button to load dataset and get below output

5.2 DATA PREPROCESSING

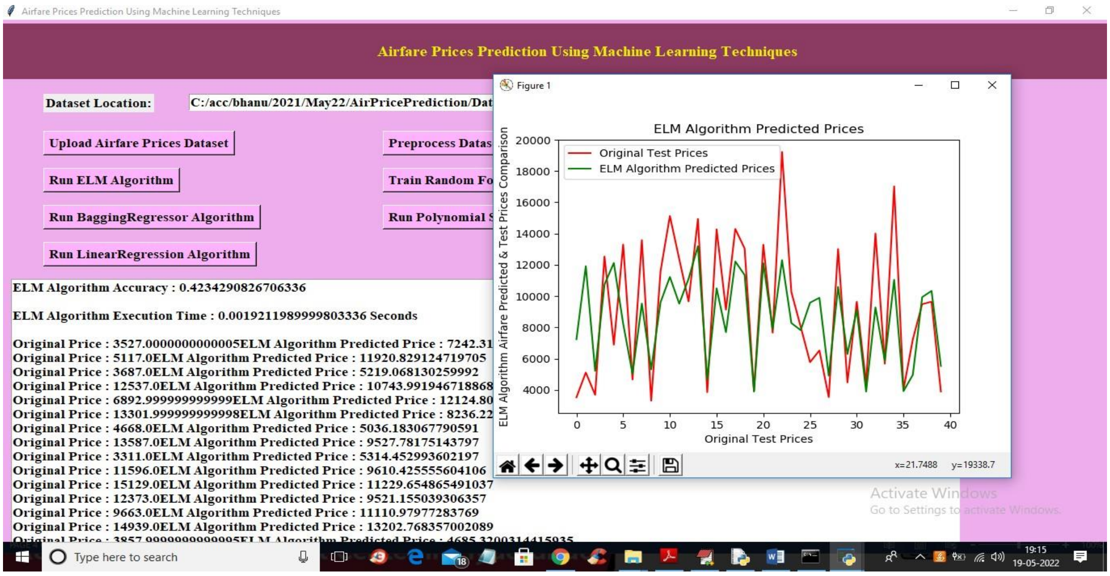
Screenshot 5.2 In above screen all dataset values converted to numeric and now click on ‘Run MLP Algorithm’ button to train MLP and apply trained model on test data to predict prices and calculate its prediction accuracy

5.3 RUN MLP ALGORITHM



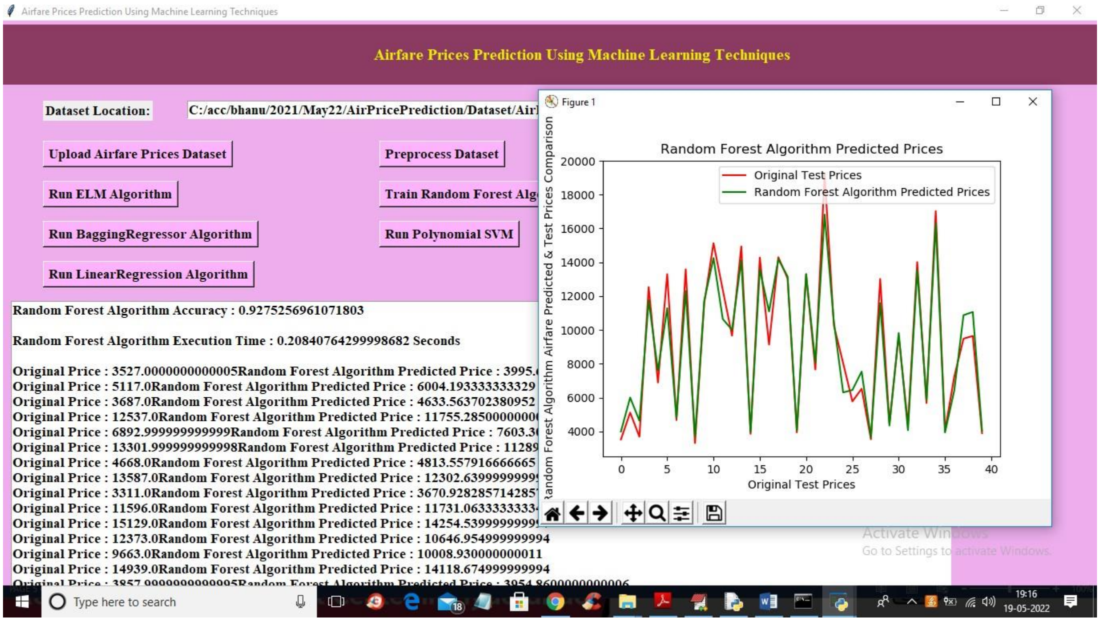
Screenshot 5.3 In above screen text area we can see MLP accuracy as 0.42% and its execution time is 0.07 seconds and then we can see original TEST prices and then prices predicted by MLP and same prices we are plotting in above graph.

5.4 RUN ELM ALGORITHM



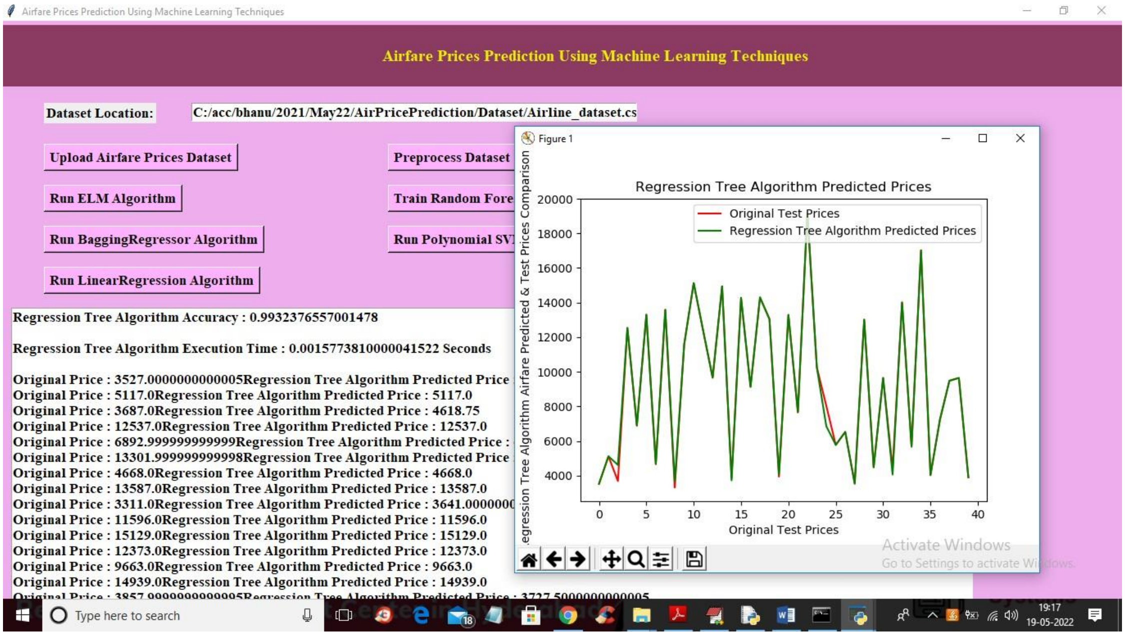
Screenshot 5.4: In above screen we can see ELM accuracy as 0.42 and its prediction is also not good and now click on Random Forest button to get below output

5.5 RUN RANDOM FOREST ALGORITHM



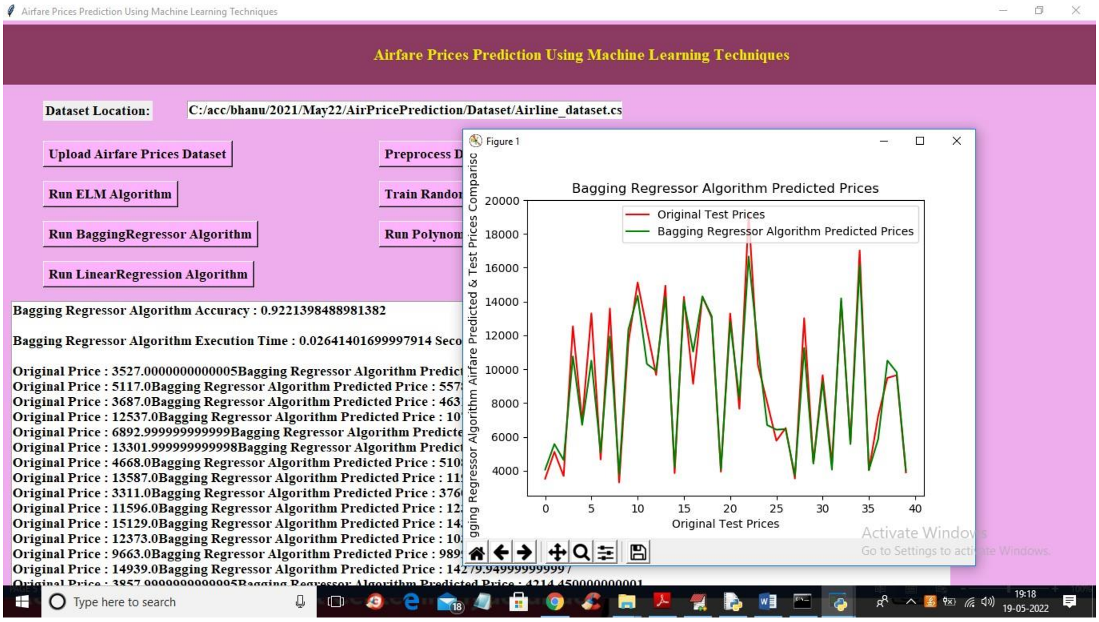
Screenshot 5.5: In above screen with Random Forest we got 0.92% accuracy and in graph we can see both lines are closely overlapping

5.6 RUN REGRESSION TREE ALGORITHM



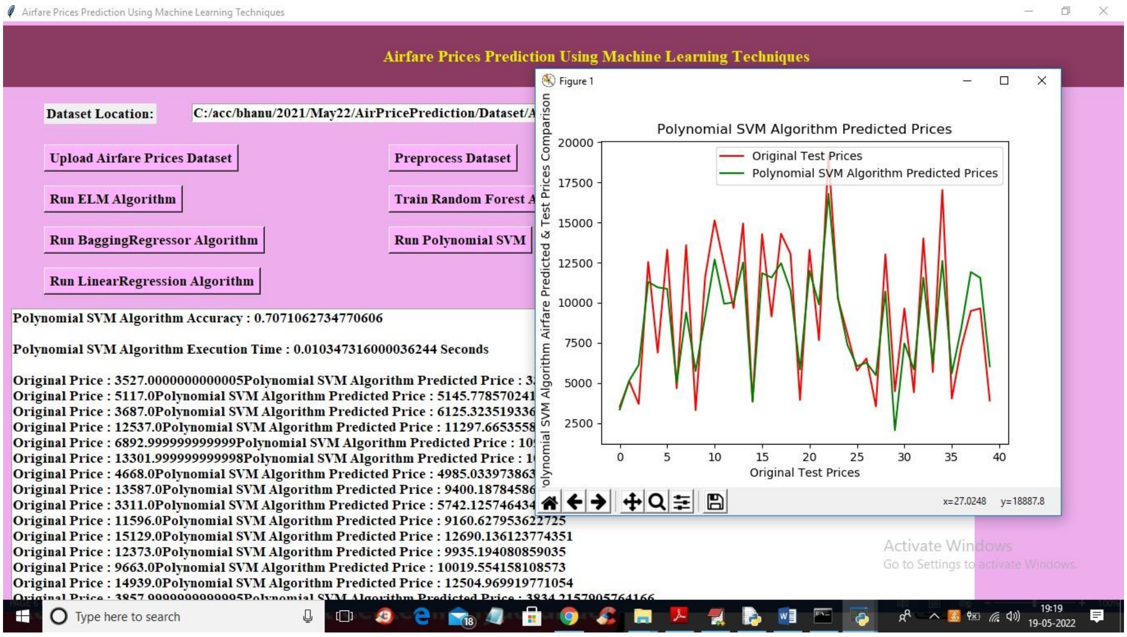
Screenshot 5.6: In above screen with Regression Tree we got 0.99% accuracy and in graph also we can see both line closely matching so its performance is good and now click on ‘Run Bagging Regression Algorithm’ button to get below output

5.7 RUN BAGGING REGRESSION TREE ALGORITHM



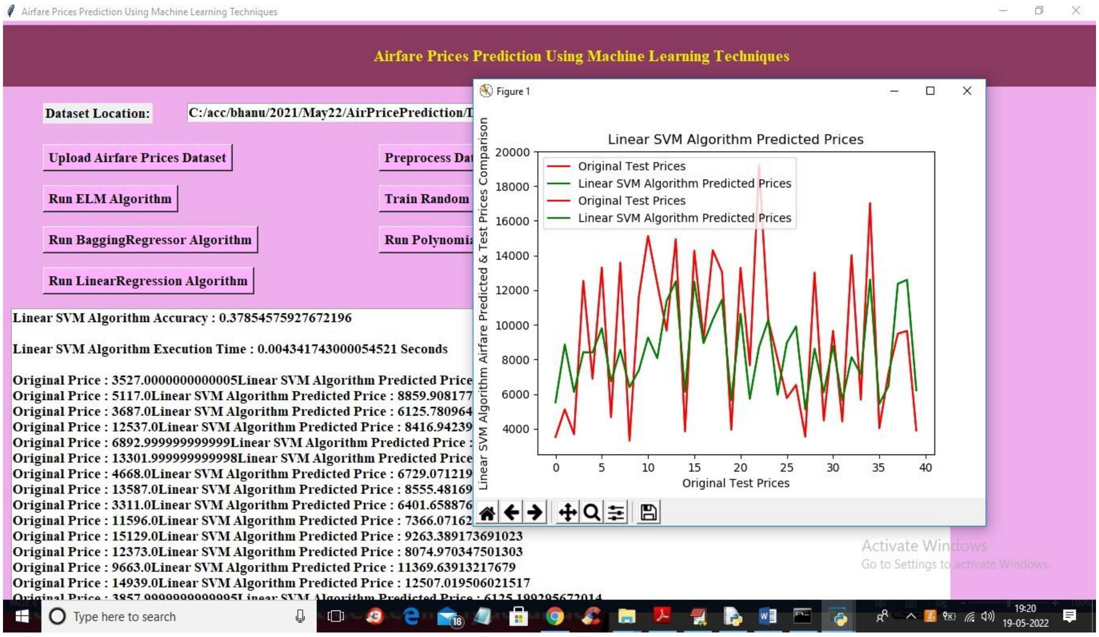
Screenshot 5.7: In above screen with Bagging Regression we got 0.92% accuracy and now click on 'Run Polynomial SVM' button to get below output

5.8 RUN POLYNOMIAL SVM ALGORITHM



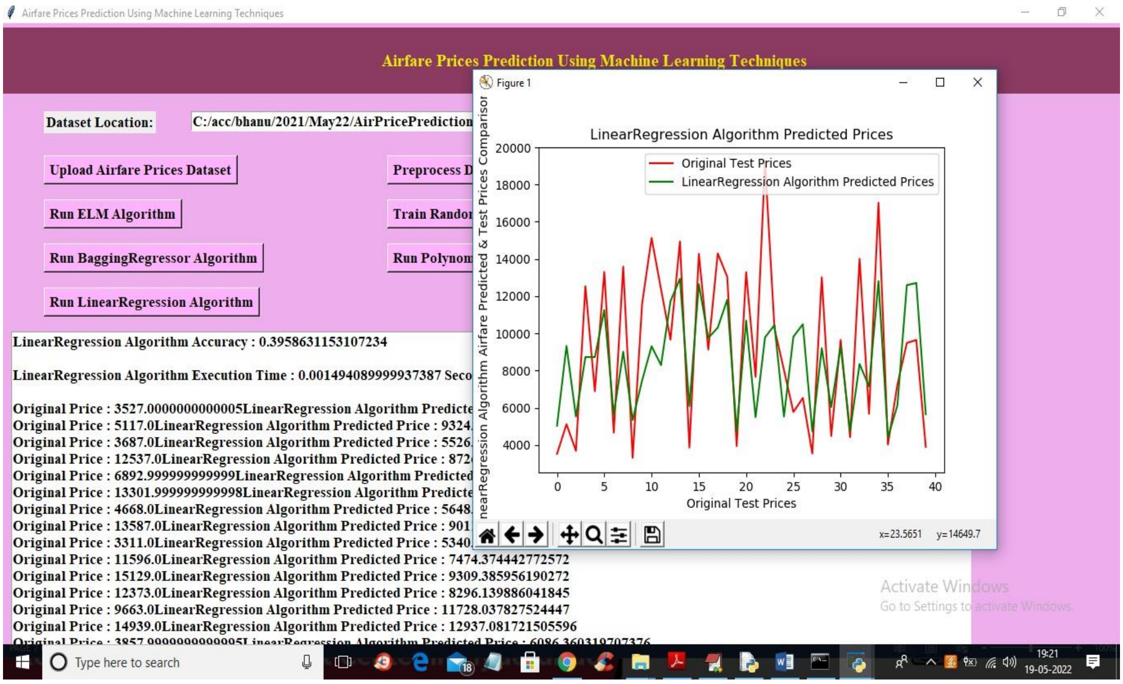
Screenshot 5.8 : In above screen with Polynomial SVM' we got 0.70% accuracy and now click on 'Run Linear SVM Algorithm' button to get below output

5.9 RUN LINEAR SVM ALGORITHM



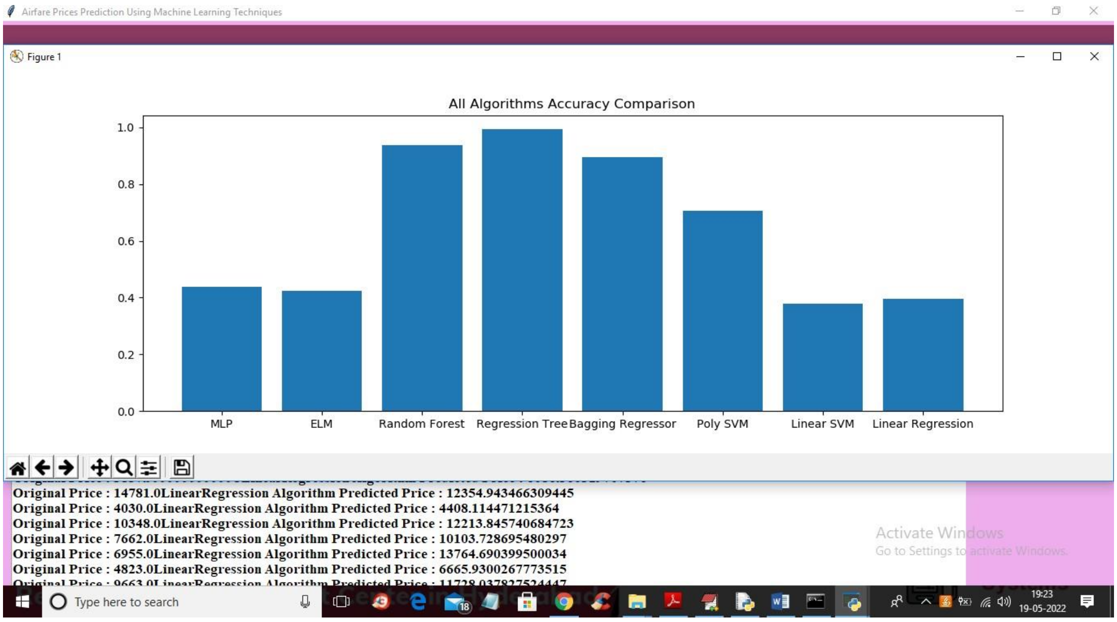
Screenshot 5.9 : In above screen with Linear SVM we got 0.37% accuracy and now click on ‘Run Linear Regression Algorithm’ button to get below output

5.10 RUN LINEAR REGRESSION ALGORITHM



Screenshot 5.10: In above screen with Linear Regression we got 0.39% accuracy and now close above graph and then click on 'Accuracy Comparison Graph' button to get below graph

5.11 ACCURACY GRAPH



Screenshot 5.11: In above graph x-axis represents algorithm names and y-axis represents accuracy of those algorithms and in all algorithms Regression Tree got high accuracy

6. TESTING

6.TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available the specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be xercised.
Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes.

6.2.4 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.2.5 WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6.2.6 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document.

6.3 TEST CASES

Test case ID	Test case Description	input	Expected output	Actual output	Status(pass/fail)
1	Verify system's ability to predict airfare price for a given route	Route: New York to London, Date: 30 days from today	Predicted airfare price is displayed	Predicted price of $550 displayed	Pass
2	Check prediction accuracy using real-time data	Route: Paris to Tokyo, Date: 15 days from today	Prediction changes based on real-time data integration	Prediction updated to $850 after live data integration	Pass
3	Check system’s ability to adapt to volatile conditions	Sudden spike in demand for a flight route	Updated price predictions to reflect demand surge	Prices increased by $100 due to demand spike	Pass
4	Test system's ability to handle multiple user preferences	User Preferences: Airline: United, Non-stop flights	Personalized prediction displayed based on preferences	Non-stop United flight price of $650 displayed	Pass

6.3.1 DESCRIPTION

The test case table validates the Airfare Price Prediction system’s accuracy, personalization, and adaptability. It includes scenarios for verifying route-specific predictions, user preference-based recommendations, real-time data integration, and price category classifications. These tests ensure the system provides reliable, tailored, and up-to-date airfare predictions.

7. CONCLUSION AND FUTURE SCOPE

7.CONCLUSION AND FUTURE SCOPE

7.1 PROJECT CONCLUSION

The Airfare Price Prediction Using Machine Learning project is designed to tackle the challenges of fluctuating airfare prices by employing various machine learning algorithms to predict optimal booking times. Through the analysis of historical flight data and real-time information, this system identifies patterns and trends influenced by factors such as demand, booking times, and airline pricing strategies. By classifying flights into distinct pricing categories (low, medium, high) and offering personalized predictions tailored to user preferences, such as preferred airlines and travel dates, it enables users to make well-informed booking decisions and manage their travel expenses effectively.

This project addresses significant limitations in existing airfare prediction systems by incorporating a wider range of data sources and providing transparent insights into the reasons behind each price prediction, building user trust. It also adapts dynamically to sudden market changes, such as seasonal peaks or airline promotions, ensuring users receive timely and relevant advice. The project’s future potential includes the integration of additional data sources, such as weather patterns and global events, to improve prediction accuracy, as well as mobile app development for greater accessibility. Ultimately, this system offers a comprehensive, accurate, and user-centered solution for airfare prediction, benefiting both consumers seeking cost savings and airlines aiming to optimize revenue strategies.

7.2 FUTURE SCOPE

The future scope of the Airfare Price Prediction Using Machine Learning project includes several potential enhancements:

1. **Integration of More Diverse Data**: Expanding the system to incorporate additional real-time data, such as weather patterns, social media trends, and geopolitical events, to improve prediction accuracy.
2. **Advanced Machine Learning Models**: Future developments could include more sophisticated models, such as hybrid or ensemble methods, to handle larger datasets and increase predictive performance.
3. **Dynamic Pricing for Airlines**: The system could be extended to help airlines implement dynamic pricing strategies, adjusting fares in real-time based on market demand and other conditions.
4. **Multi-Leg and International Flight Support**: Adding support for complex routes, including multi-leg flights and international routes, would make the system more comprehensive.
5. **Mobile and User-Centric Enhancements**: Developing a mobile app with personalized notifications for price changes and user preferences, making it a more proactive and user-friendly tool.
6. **Collaboration with Travel Platforms**: Partnering with airlines and booking platforms for direct integration, allowing users to seamlessly book flights based on the system’s predictions.

These advancements would significantly expand the system’s capabilities, making it a more powerful tool for travelers and the airline industry.

8. BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

[1] P. Malighetti, S. Paleari and R. Redondi, "Pricing strategies of low-cost airlines: The Ryanair case study," Journal of Air Transport Management, vol. 15, no. 4, pp. 195-203, 2009.

[2] P. Malighetti, S. Paleari and R. Redondi, "Has Ryanair's pricing strategy changed over time? An empirical analysis of its 2006–2007 flights," Tourism Management, vol. 31, no. 1, pp. 36-44, 2010.

[3] W. Groves and M. Gini, "A regression model for predicting optimal purchase timing for airline tickets," Technical Report 11-025, University of Minnesota, Minneapolis, 2011.

[4] W. Groves and M. Gini, "An agent for optimizing airline ticket purchasing," 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013), St. Paul, MN, May 06 - 10, 2013, pp. 1341-1342.

[5] M. Papadakis, "Predicting Airfare Prices," 2014.

[6] T. Janssen, "A linear quantile mixed regression model for prediction of airline ticket prices," Bachelor Thesis, Radboud University, 2014.

[7] R. Ren, Y. Yang and S. Yuan, "Prediction of airline ticket price," Technical Report, Stanford Univerisy, 2015.

[8] S. Haykin, Neural Networks – A Comprehensive Foundation. Prentice Hall, 2nd Edition, 1999.

8.2 WEBSITES

https://github.com/Vinodkumar0113/mini-project-airfare