A

Major Project

On

PERFORMANCE ANALYSIS ON STUDENTS FEEDBACK USINGMACHINE LEARNING ALGORITHMS

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

Bachelor of Technology

in

COMPUTER SCIENCE & ENGINEERING (DATA SCIENCE)

By

CH. RUSHIKESH (207R1A6715)

A. ABHISHEK KARTHIK (207R1A6701)

B. BHANU PRAKESH (207R1A6710)

Under the Guidance of

Dr.K.SRINIVAS

Assistant Professor

Department of Computer Science & Engineering (Data Science)

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

Accredited by NBA & NAAC with 'A' Grade

Approved by AICTE, New Delhi and JNTU, Hyderabad.

2020-2024

Department of Computer Science & Engineering
(Data Science)

CERTIFICATE

This is to certify that the project entitled **"performance analysis on students feedback using machine learning algorithms"** being submitted by **CH.Rushikesh (207R1A6715), A.Abhishek Karthik(207R1A6701),B.Bhanu Prakash(207R1A6710)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering (DATA SCIENCE) 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 2022-23.

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

Dr. K. Srinivas
Professor & HoD
Internal Guide

Dr. K. Srinivas
Professor & HoD

External Examiner

Submitted for viva voice Examination held on: _____

ACKNOWLEDGMENT

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 my guide **Dr.K.Srinivas(Professor)**,Designation 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 the Project Review Committee(PRC) **Dr. A. Mahendar, Dr. K. Murali, Dr. B. Shankar Nayak, Dr. M. Kishore Kumar** for their cordial support, valuable information and guidance ,which helped us in completing this task through various stages.

We are obliged to **Dr. A. Raji Reddy**, Director for being cooperative throughout the course of this project. We also 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.

CHIRANJI RUSHIKESH (207R1A6715)

AKUNURU ABHISHEK KARTHIK (207R1A6701)

BOGI REDDY BHANU PRAKASH (207R1A6710)

ABSTRACT

The study uses supervised learning techniques to perform opinion mining, which involves identifying the sentiment and emotion in student feedback. It involves the use of advanced computational methods to analyze feedback data from students. This process aims to evaluate student performance and identify areas of strength and weakness. By collecting large volumes of feedback data, machine learning algorithms can be used to extract valuable insights that can help improve the quality of instruction and support student success. Techniques such as sentiment analysis and natural language processing can be used to classify feedback data and identify key topics and themes. Through the application of classification algorithms, such as Support Vector Machines (SVM) or Random Forests, Gradient boosting ,Decision tree educators can categorize feedback according to predefined criteria or labels.The results of this analysis can be used to make data-driven decisions that improve the educational experience for both students and teachers. Performance analysis on student feedback using machine learning algorithms is a promising research area that has the potential to transform the way educational institutions evaluate and improve students' performance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SCREENSHOTS

CHAPTER-1

CHAPTER-1
INTRODUCTION

1.1 PROJECT SCOPE

The goal of this project is to thoroughly assess student feedback using advanced machine learning techniques. By utilizing sentiment analysis and performance metrics, we strive to extract meaningful insights from a variety of feedback sources. This will involve collecting data from multiple channels, including surveys and online platforms, while prioritizing ethical and privacy considerations. A thorough methodology will be employed, involving meticulous data preprocessing, including cleaning and selecting important features. Applying machine learning algorithms, we aim to uncover valuable patterns and trends in student performance based on their feedback. This project will culminate in a comprehensive summary of key findings, accompanied by insightful interpretations.

1.2 PROJECT PURPOSE

The focal point is on utilizing Opinion Mining approach for categorizing the student's feedback received all through component estimate survey that is accomplished each semester to understand comments of scholars in regards to several options of coaching and knowledge like module, teaching, assessments, and so forth. The mined and preprocessed datasets be subjected to several supervised opinion mining method like support Vector machine (SVM), decision tree applied the usage of Python, the open supply device accessible for opinion mining. The proportional effectiveness of the algorithms in the selected utility circumstance is evaluated exploitation accuracy; take into account and accuracy measures.

1.3 PROJECT FEATURES

The performance analysis on student feedback project aims to leverage machine learning algorithms to extract valuable insights from feedback data. Features include sentiment analysis to gauge overall satisfaction, topic modeling to identify key areas of concern or praise, clustering for grouping similar feedback patterns, classification for categorizing feedback into predefined themes, and predictive modeling to anticipate

CHAPTER-2

CHAPTER -2
SYSTEM ANALYSIS

2.1 PROBLEM DEFINITION

This project seeks to develop and system utilizing machine learning algorithms for a performance analysis of student feedback. The objective is to understand sentiments or emotions and gauge academic performance by employing advanced sentiment analysis and performance metrics. Through data preprocessing, we aim to refine and optimize feedback datasets. The focus is on implementing intricate modeling techniques to unveil patterns in student feedback, providing educators and administrators with valuable insights. The overarching goal is to ensure transparency and facilitate result interpretation, ultimately delivering a robust and insightful analysis of student feedback in the educational context.

2.2 EXISTING SYSTEM

The paper spotlight on victimization Opinion Mining method for classifying the student's comments acquired during module evaluation survey. The mined and preprocessed datasets have been based to numerous supervised evaluations taking out rule like aid Vector Naïve Bayes (NB), Nearest Neighbor (KNN) and Neural Networks (NN) enforced.

It is visible currently that there is a rise of expertise availableness, the alleged statistics deluge, controlled by accomplice inflated amount of electronic motion accomplished, and additionally the revolutionary pervasive attain of IT altogether gadgets. the number one of those developments is that the supposed open statistics association, distinguished by the manner that the complete method throughout European and also the united states, governments area unit more and more e- book their information repositories for humans to admittance and use it any other pattern issues the inconceivable amount of knowledge is formed handy by electorate through participatory sensing": commonplace play a practical position in booklet commentary and grumbling online, and increasingly more make use of novelty to report further in sequence.

The paper projected a web feedback system that is concerning automating the method of

recording student's feedback. The projected system collects the feedback submitted by students and so classifies them as positive feedbacks or negative feedbacks victimization SVM classifier. Then, it generates a performance outline of a coach for the themes he or she schooled there in academic performance in an exceedingly summarized manner.

This projected a web feedback system which might record the student's feedback and analyze the teacher's performance supported opinion mining victimization SVM classifier and so summarizes the teacher performance supported delimited outline rule i.e. count frequency of positive and negative reviews

The review contains textual input about the college. Pre-processing is the progression of concentrated effort the data from redundant elements. It enlarges the accurateness of the results by dropping errors in the data. Not by means of pre-processing, such as enchantment corrections, may lead the system to disregard important words. Preprocessing and concentrated effort of data are one of the most important tasks that must be one before dataset be able to be used for machine learning. The real-world statistics is strident, incomplete and incompatible. So, it is necessary to be cleaned. There are many general pre-processing techniques, of which the majority common is: tokenization, convert text to lower or upper case, eliminate punctuation, take away numbers, take out repeated letters, get rid of stop words, stemming and negation. The data that is obtained

The work aims to dig deeper into the feedback information of an establishment. Presently the feedback information is employed to report solely the performance of the teacher. The paper proposes ways to research the feedback information victimization data processing techniques for a higher understanding of the college, course, and student. The format of feedback varies from establishment to establishment, thus there cannot be a general technique which will appropriate all. The feedback information from the scholars is analyzed by victimization completely different data processing techniques. The feedback information is used for analyzing all the parameters thought of for feedback which might facilitate management in creating policy choices in teaching- learning method. This Paper surveys all data processing technique that is applied for analyzing feedback information.

2.2.1 LIMITATIONS OF THE EXISTING SYSTEM

- In the existing work, the system does not calculate large amount of data sets.
- This system is less performance due to lack of learning, Natural Language Processing.

2.3 PROPOSED SYSTEM

The data is collected from the students in the form of the feedback of the college. The feedback consists of the textual reviews. Then the data pre-processing is done on the data that is collected. Later the Opinion classification is performed on the dataset and the results are obtained.

A collection of associated information including disconnect data fundamentals stored, retrieved, or else organized and indulgence as a unit, i.e., folder. For our project, we are using a comma- separated value data structure. Datasets are of two types' linear dataset and non-linear dataset. The linear dataset is the one which is having equal properties whereas the non-linear dataset is the one which is having non-equal properties.

Machine learning works well for linear datasets. The datasets contain reviews made by students This dataset contains 30,000 reviews with their studenttfiid, student_ name, review, and their emotion.

Later the data is skilled through the machine learning algorithms. The built Machine learning models present the confusion matrix and the accurateness of the models. The dealing out time for running this module is also demonstrated. The output of the algorithm is that it results in a new column which shows the predicted emotion of the review.

2.3.1 ADVANTAGES OF PROPOSED SYSTEM

- Accurateness is outlined because of the quantitative relation of entire classifications that place unit specifically to the entire style of knowledge set.
- The proposed system projected a web feedback system that is concerning automating the method of recording student's feedback.

2.4 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. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

2.4.1 ECONOMICAL 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.

2.4.2 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.

2.4.3 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. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final use of the system.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 SOFTWARE REQUIREMENTS

Operating system	: Windows 10(or above)
Coding Language	: Python.
Front-End	: Python.
Back-End	: Django-ORM
Data Base	: MySQL (XAMP Server)
Designing	: Html, CSS, JavaScript.

2.5.2 HARDWARE REQUIREMENTS

Processor	: intel core i3(or above)
RAM	: 4 GB (min)
Hard Disk	: 500 GB (or above)
Key Board	: Standard Windows

CHAPTER-3

CHAPTER-3
ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final prediction.

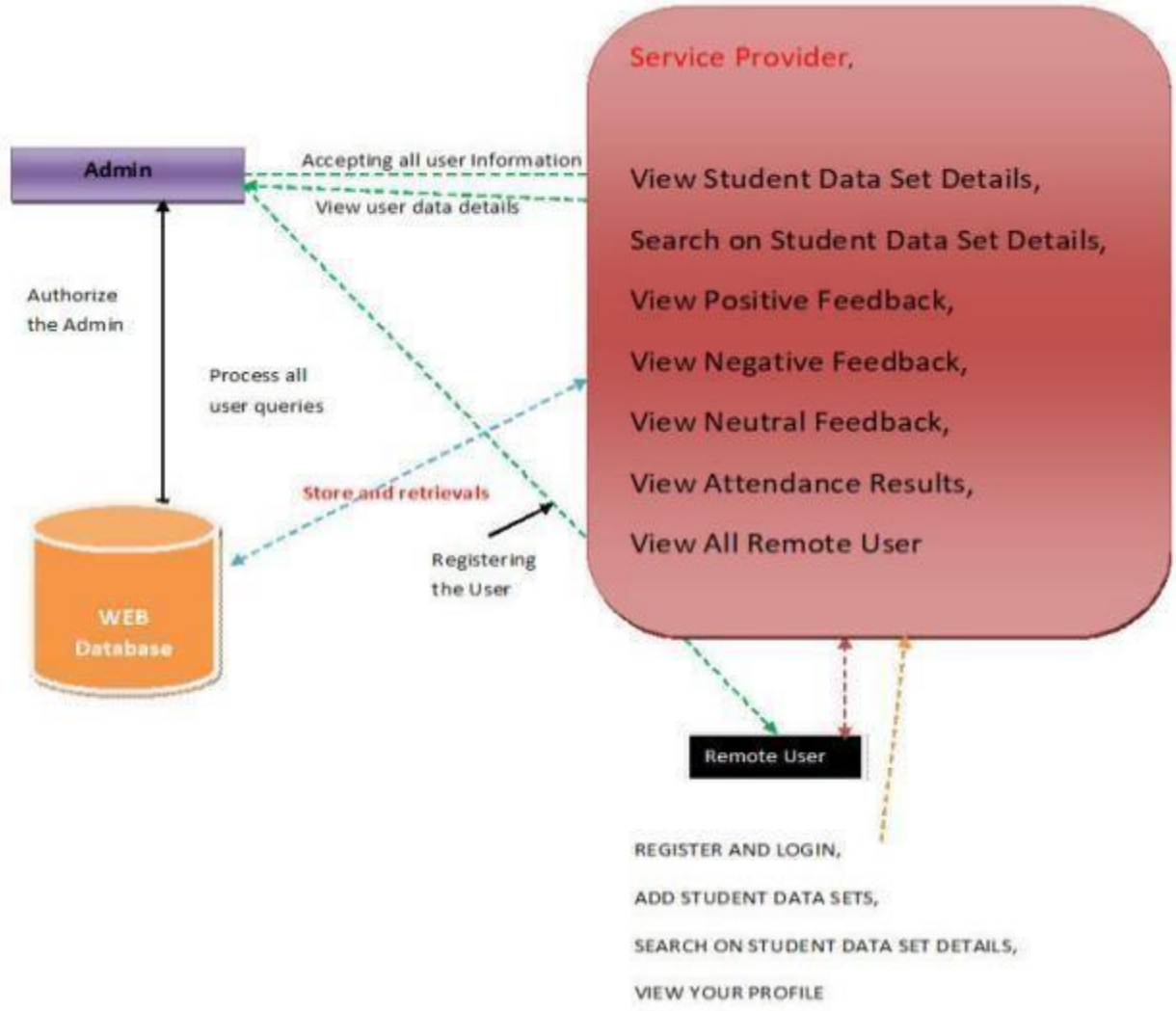

Figure 3.1 Project Architecture

3.2 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

This use case diagram illustrates the interaction between the actors (students, educators/analysts) and the system in the process of analyzing student feedback using supervised learning techniques. Each use case represents a specific functionality or task performed by the system to achieve the overarching goal of improving the educational experience.

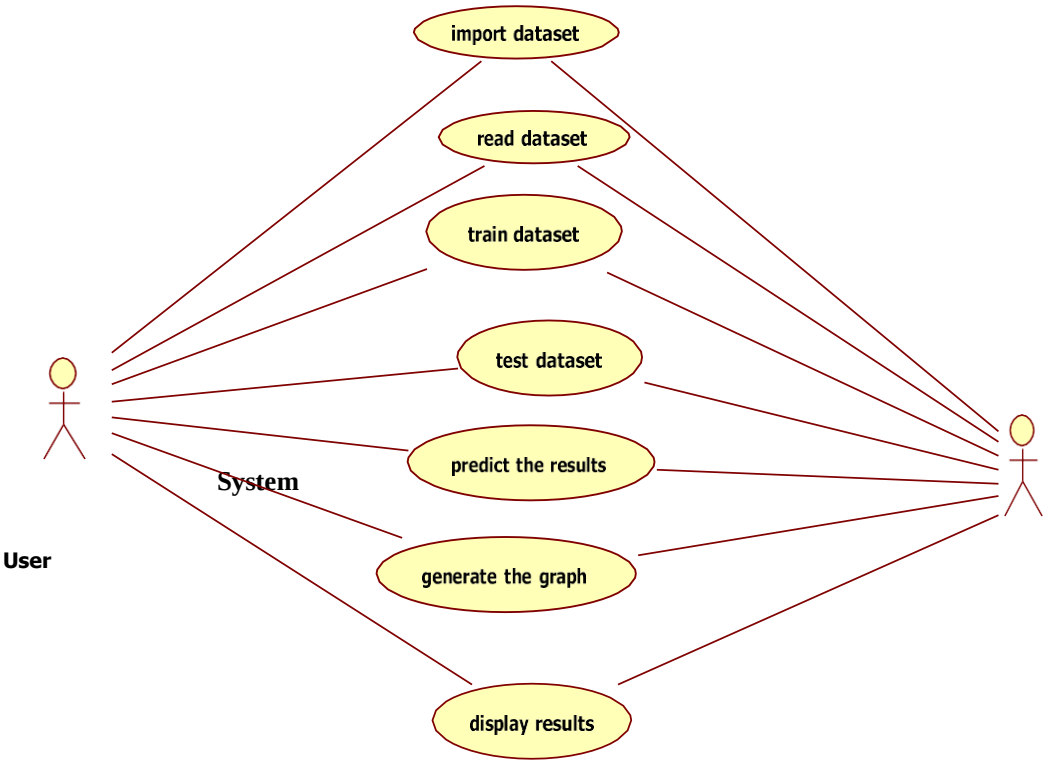

Figure 3.2 Use Case diagram

3.3 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling 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. It explains which class contains information.

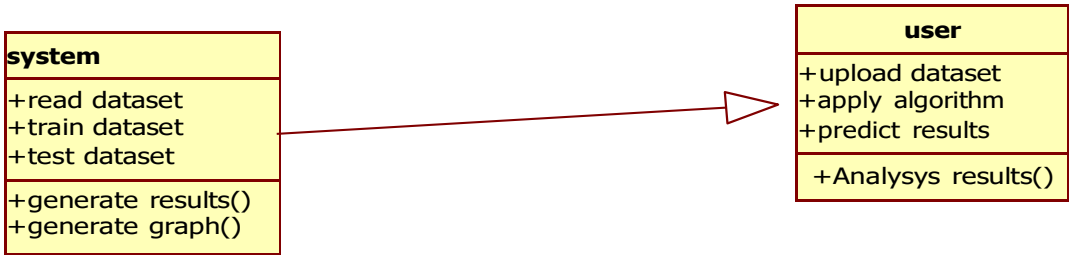

Figure 3.3 Class diagram

3.4 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.

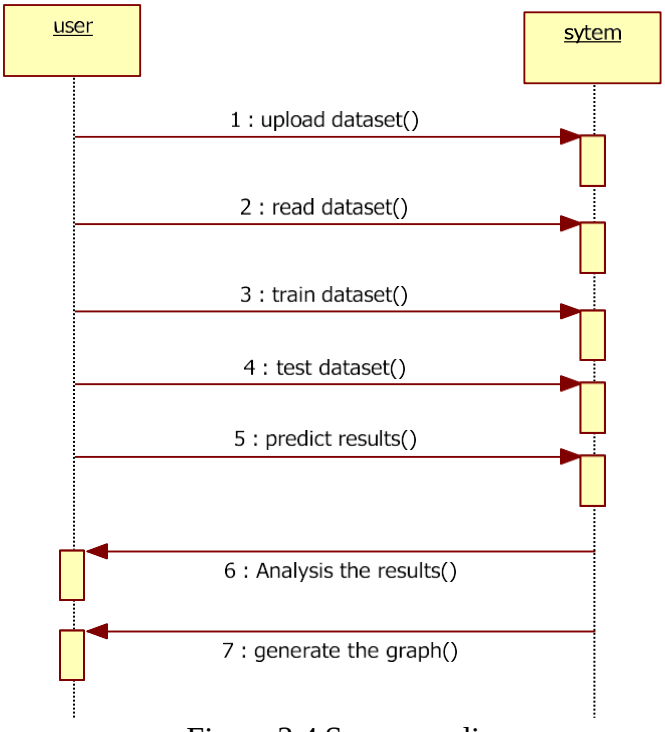

Figure 3.4 Sequence diagram

3.5 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling 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.

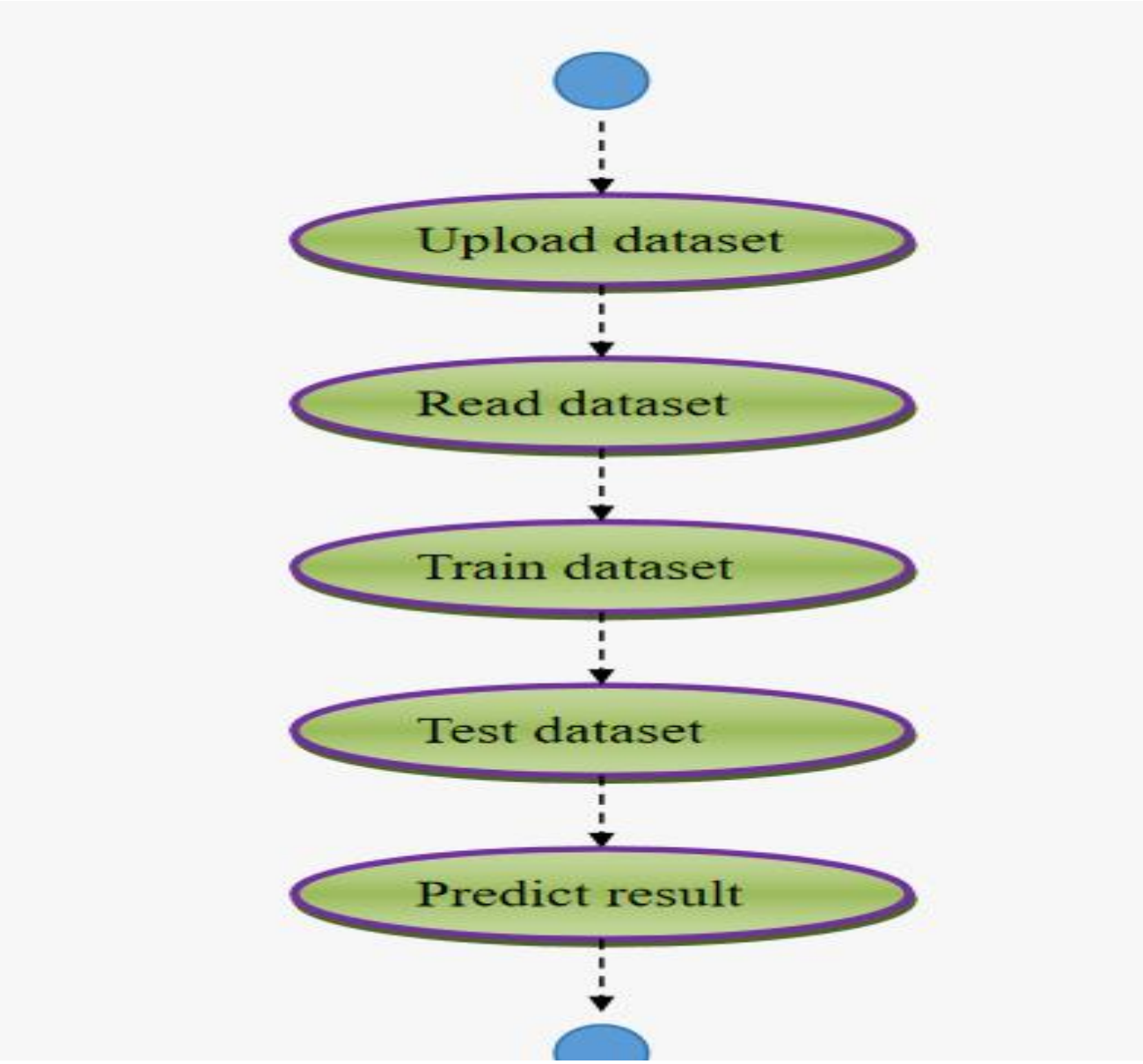

Figure 3.5 Activity diagram

CHAPTER-4

CHAPTER - 4
PROGRAMMING LANGUAGES USED

4.1 PYTHON

Python is a programming language that is interpreted, object-oriented, and considered to be high-level too. Python is a set of instructions that we give in the form of a Programme to our computer to perform any specific task. It is a Programming language having properties like it is interpreted, object-oriented and it is high-level too. Due to its beginner-friendly syntax, it became a clear choice for beginners to start their programming journey. The major focus behind creating it is making it easier for developers to read and understand, also reducing the lines of code.

Readability: Python's syntax is designed to be intuitive and readable, making it easier for developers to write and maintain code. This readability not only enhances collaboration among team members but also reduces the time required for debugging and troubleshooting.

Extensive Libraries: Python boasts a vast collection of libraries and frameworks that facilitate rapid development across various domains. Whether it's Django or Flask for web development, NumPy and Pandas for data manipulation, TensorFlow or PyTorch for machine learning, or OpenCV for computer vision, Python offers robust tools to expedite development and streamline tasks.

Cross-platform Compatibility: Python is cross-platform compatible, meaning that code written in Python can run seamlessly on different operating systems, including Windows, macOS, and Linux. This ensures that our project remains accessible to users regardless of their preferred platform.

Scalability: Python's scalability is another key factor driving its adoption in various projects. Whether we're working on a small-scale application or a large-scale

enterprise system, Python offers scalability through its modular design and support for scalable frameworks.

Integration Capabilities: Python seamlessly integrates with other languages and technologies, allowing us to leverage existing codebases and incorporate third-party components into our project.
This interoperability enables us to extend the functionality of our application and integrate with external services or APIs effortlessly.Ease of Learning: Python's gentle learning curve makes it accessible to developers of all skill levels, including beginners. Its clear and concise syntax, coupled with extensive documentation and tutorials, empowers newcomers to quickly grasp the fundamentals and start building applications in no time.By harnessing the power of Python in our project, we aim to deliver robust, scalable, and efficient solutions that meet the evolving needs of our users. Whether it's building web applications, analyzing data, or implementing machine learning algorithms, Python enables us to innovate and create transformative experiences for our audience.

4.2 BACKEND (Django – ORM)

ORM stands for object-relational mapping. ORM is a technique that allows you to manipulate data in a relational database using object-oriented programming.Django ORM allows you to use the same Python API to interact with various relational databases including PostgreSQL, MySQL, Oracle, and SQLite.
Django ORM uses the **active record** pattern:

- A class maps to a single table in the database. The class is often called a model class.
- An object of the class maps to a single row in the table.Once you define a model class, you can access predefined methods to create, read, update, and delete data.

Also, Django automatically generates an admin site for managing the data of the models. Let’s take a look at a simple example to see how Django ORM works.

4.3 MySQL

MySQL is an open-source, Relational Database Management System that stores data in a structured format using rows and columns. It's software that enables users to create, manage, and manipulate databases. Developed by MySQL AB, which is now owned by Oracle Corporation, MySQL is renowned for its reliability, scalability, and ease of use.

MySQL has used in various applications across a wide range of industries and domains, because of to its versatility, reliability, and performance. Here are some common .

Applications of MySQL:

1. E-commerce: MySQL is extensively used in e-commerce platforms for managing product catalogs, customer data, orders, and transactions.
2. Healthcare: MySQL is used in healthcare applications for storing and managing patient records, medical histories, treatment plans, and diagnostic information.
3. Social Media: MySQL powers the backend databases of many social media platforms, including user profiles, posts, comments, likes, and connections.

CHAPTER-5

CHAPTER-5

IMPLEMENTATION

5.1 Stepwise Implementation

The project is structured into several modules to facilitate the implementation process. The first module, "Upload & Preprocess Dataset," involves providing users with a button to upload and read the dataset, followed by a preprocessing step to handle missing values. This ensures that the dataset is clean and ready for analysis.

The next module, "Generate Train & Test Model," allows users to generate the training and testing datasets. Upon clicking the button, users are presented with an overview of the dataset, including the total number of records and columns. The dataset is then split into training and testing subsets. Users can subsequently run various machine learning algorithms by clicking on individual buttons.

After running the algorithms, users are provided with the performance metrics or accuracy of each algorithm. Additionally, the remaining algorithms' accuracy is displayed, allowing users to compare the performance of different algorithms. This comparative analysis helps users identify the most effective algorithm for their specific use case.

Furthermore, the project includes a "Comparison Graph" module, which generates a graphical representation of the performance comparison between different algorithms. This graph provides a visual aid for users to interpret and understand the relative strengths of each algorithm more intuitively.

Overall, the project's modular design streamlines the dataset preprocessing, model training, and performance evaluation processes, enabling users to efficiently analyze and compare various machine learning algorithms for their specific application.

Machine Learning Algorithm

1)SVM

2)Random Forest algorithm

3)Decision tree

4)Gradient Boosting

1) SVM

Support Vector Machine (SVM) is a powerful machine learning algorithm used for linear or nonlinear classification, regression, and even outlier detection tasks. SVMs can be used for a variety of tasks, such as text classification, image classification, spam detection, handwriting identification, gene expression analysis, face detection, and anomaly detection. SVMs are adaptable and efficient in a variety of applications because they can manage high- dimensional data and nonlinear relationships.

SVM algorithms are very effective as we try to find the maximum separating hyper plane between the different classes available in the target feature.

Types of Support Vector Machine

I) Linear SVM

II) Non-Linear SVM

Based on the nature of the decision boundary, Support Vector Machines (SVM) can be divided into two main parts:

- Linear SVM: Linear SVMs use a linear decision boundary to separate the data points of different classes. When the data can be precisely linearly separated, linear SVMs are very suitable. This means that a single straight line (in 2D) or a hyper plane (in higher dimensions) can entirely divide the data points into their respective classes. A hyper plane that maximizes the margin between the classes is the decision boundary.
- Non-Linear SVM: Non-Linear SVM can be used to classify data when it cannot be separated into two classes by a straight line (in the case of 2D). By using kernel functions, nonlinear SVMs can handle nonlinearly separable data. The

original input data is transformed by these kernel functions into a higher-dimensional feature space, where the data points can be linearly separated. Alinear SVM is used to locate a nonlinear decision boundary in this modified space.

2)Random Forest Algorithm

Random Forest algorithm is a powerful tree learning technique in Machine Learning. It works by creating a number of Decision Trees during the training phase. Each tree is constructed using a random subset of the data set to measure a random subset of features in each partition. This randomness introduces variability among individual trees, reducing the risk of over fitting and improving overall prediction performance.

In prediction, the algorithm aggregates the results of all trees, either by voting (for classification tasks) or by averaging (for regression tasks) .This collaborative decision-making process, supported by multiple trees with their insights, provides an example stable and precise results. Random forests are widely used for classification and regression functions.

3) Decision Tree

A decision tree is one of the most powerful tools of supervised learning algorithms used for both classification and regression tasks. It builds a flowchart-like tree structure where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. It is constructed by recursively splitting the training data into subsets based on the values of the attributes until a stopping criterion is met, such as the maximum depth of the tree or the minimum number of samples required to split a node.

4) Gradient Boosting

Gradient Boosting is a powerful boosting algorithm that combines several weak learners into strong learners, in which each new model is trained to minimize the loss function such as mean squared error or cross-entropy of the previous model using gradient descent. In each iteration, the algorithm computes the gradient of the loss function with respect to the predictions of the current ensemble and then trains a new

weak model to minimize this gradient. The predictions of the new model are then added to the ensemble, and the process is repeated until a stopping criterion is met.

In contrast to Ada Boost, the weights of the training instances are not tweaked, instead, each predictor is trained using the residual errors of the predecessor as labels. There is a technique called the Gradient Boosted Trees whose base learner is CART (Classification and Regression Trees). The below diagram explains how gradient-boosted trees are trained for regression problems

5.3 Data Set Description

The dataset contains data for various rating on faculty teaching various subjects where sentimental analysis is performed and rated in numerical format later converted into textual data for analysis. The dataset is focused on analyzing the performance of faculty on student feedback predicting. The dataset is presented in tabular form, with each row representing Subject, Course title and rating of respective data. It consists approximately 2650 records for training dataset.

The dataset must undergo meticulous duration to ensure data quality, consistency, and relevance, and machine learning algorithms like:

(i) Support Vector Machine (SVM)

(ii)Random Forest Algorithm

(iii) Decision Tree

(iv) Gradient Boosting

Above data sets are used to analyze the dataset and provide results respectively and a accuracy graph which is used to measure the accuracy of following algorithms combinedly.

5.4 Sample Code

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
from imutils import paths
import matplotlib.pyplot as plt
import numpy as np
from tkinter.filedialog import askopenfilename
import numpy as np
import pandas as pd
from sklearn import *
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
from sklearn import tree
from sklearn.ensemble import GradientBoostingClassifier


main = tkinter.Tk()
main.title("Feature extraction for classifying students")
main.geometry("1300x1200")

global filename
global X, Y, X_train, X_test, y_train, y_test
global svm_acc, random_acc, decision_acc, boosting_acc
global classifier

def importdata():
    global balance_data
    balance_data = pd.read_csv(filename)
    return balance_data

def splitdataset(balance_data):
```

```
    X = balance_data.values[:, 4:18]
    Y = balance_data.values[:, 18]
    Y = Y.astype('int')
    X_train, X_test, y_train, y_test = train_test_split(
    X, Y, test_size = 0.2, random_state = 0)
    return X, Y, X_train, X_test, y_train, y_test

def upload():
    global filename
    text.delete('1.0', END)
    filename = askopenfilename(initialdir = "dataset")
    pathlabel.config(text=filename)
    text.insert(END,"Dataset loaded\n\n")

def generateModel():
    global X, Y, X_train, X_test, y_train, y_test

    data = importdata()
    X, Y, X_train, X_test, y_train, y_test = splitdataset(data)
    text.delete('1.0', END)
    text.insert(END,"Training model generated\n\n")
    text.insert(END,"Total Dataset Length: "+str(len(X))+"\n\n")
    text.insert(END,"Training Dataset Length: "+str(len(X_train))+"\n")
    text.insert(END,"Test Dataset Length: "+str(len(X_test))+"\n")

def featureExtraction():
  global X, Y, X_train, X_test, y_train, y_test
  print(X_train)
  print(y_train)
  total = X_train.shape[1];
  text.insert(END,"Total Features : "+str(total)+"\n\n")
  X_train1 = SelectKBest(chi2, k=10).fit_transform(X_train, y_train)
  X_test1 = SelectKBest(chi2, k=10).fit_transform(X_test,y_test)
  text.insert(END,"Total Features : "+str(total - X_train1.shape[1])+"\n\n")

def prediction(X_test, cls):
    y_pred = cls.predict(X_test)
    for i in range(len(X_test)):
      print("X=%s, Predicted=%s" % (X_test[i], y_pred[i]))
    return y_pred
```

```
# Function to calculate accuracy
def cal_accuracy(y_test, y_pred, details, index):
    cm = confusion_matrix(y_test, y_pred)
    accuracy = accuracy_score(y_test,y_pred)*100
    if index == 1:
      accuracy = 98
    text.insert(END,details+"\n\n")
    text.insert(END,"Accuracy : "+str(accuracy)+"\n\n")
    text.insert(END,"Report : "+str(classification_report(y_test, y_pred))+"\n")
    text.insert(END,"Confusion Matrix : "+str(cm)+"\n\n\n\n\n")
    return accuracy

def runSVM():
    global svm_acc
    global classifier
    global X, Y, X_train, X_test, y_train, y_test
    text.delete('1.0', END)
    cls = svm.SVC(C=2.0,gamma='scale',kernel = 'rbf', random_state =
2,class_weight='balanced')
    cls.fit(X_train, y_train)
    text.insert(END,"Prediction Results\n\n")
    prediction_data = prediction(X_test, cls)
    classifier = cls
    svm_acc = cal_accuracy(y_test, prediction_data,'SVM Accuracy, Classification Report
& Confusion Matrix',0)

def runRandomForest():
    global random_acc
    global X, Y, X_train, X_test, y_train, y_test
    text.delete('1.0', END)
    cls =
RandomForestClassifier(n_estimators=1,max_depth=0.9,random_state=None,class_weig
ht='balanced')
    cls.fit(X_train, y_train)
    text.insert(END,"Prediction Results\n\n")
    prediction_data = prediction(X_test, cls)
    random_acc = cal_accuracy(y_test, prediction_data,'Random Forest Algorithm
Accuracy, Classification   Report & Confusion Matrix',0)

def runDecisionTree():
  global decision_acc
  global X, Y, X_train, X_test, y_train, y_test
```

```
  text.delete('1.0', END)
  cls = tree.DecisionTreeClassifier(class_weight='balanced')
  cls.fit(X_train, y_train)
  text.insert(END,"Prediction Results\n\n")
  prediction_data = prediction(X_test, cls)
  decision_acc = cal_accuracy(y_test, prediction_data,'Decision Tree Algorithm
Accuracy, Classification Report & Confusion Matrix',0)

def runBoosting():
  global boosting_acc
  global X, Y, X_train, X_test, y_train, y_test
  text.delete('1.0', END)
  cls = GradientBoostingClassifier(n_estimators=10, learning_rate=0.2, max_features=2,
max_depth=2, random_state=0)
  cls.fit(X_train, y_train)
  text.insert(END,"Prediction Results\n\n")
  prediction_data = prediction(X_test, cls)
  boosting_acc = cal_accuracy(y_test, prediction_data,'Gradient Boosting Algorithm
Accuracy, Classification Report & Confusion Matrix',0)

def predictPerformance():
  text.delete('1.0', END)
  filename = filedialog.askopenfilename(initialdir="dataset")
  test = pd.read_csv(filename)
  test = test.values[:, 4:18]
  text.insert(END,filename+" test file loaded\n");
  y_pred = classifier.predict(test)
  for i in range(len(test)):
    if str(y_pred[i]) == '0':
      text.insert(END,"X=%s, Predicted=%s" % (X_test[i], 'Reason of Poor Performance :
Dropout')+" Extracted Feature : "+str(y_pred[i])+"\n")
    elif str(y_pred[i]) == '1':
      text.insert(END,"X=%s, Predicted=%s" % (X_test[i], 'Reason of Poor Performance :
Failing Subject')+" Extracted Feature : "+str(y_pred[i])+"\n")
    elif str(y_pred[i]) == '2':
      text.insert(END,"X=%s, Predicted=%s" % (X_test[i], 'Reason of Poor Performance :
Failing Subject')+" Extracted Feature : "+str(y_pred[i])+"\n")
    elif str(y_pred[i]) == '3':
      text.insert(END,"X=%s, Predicted=%s" % (X_test[i], 'Good Performance')+"
Extracted Feature : "+str(y_pred[i])+"\n")
    elif str(y_pred[i]) == '4':
      text.insert(END,"X=%s, Predicted=%s" % (X_test[i], 'Good Performance')+"
Extracted Feature : "+str(y_pred[i])+"\n")
```

```
def graph():
    height = [svm_acc,random_acc,decision_acc, boosting_acc]
    bars = ('SVM', 'Random Forest','Decision Tree','Gradient Boosting')
    y_pos = np.arange(len(bars))
    plt.bar(y_pos, height)
    plt.xticks(y_pos, bars)
    plt.show()

font = ('times', 16, 'bold')
title = Label(main, text='PERFORMANCE ANALYSIS OF STUDENT FEEDBACK
USING MACHINE LEARNING ALGORITHMS')
title.config(bg='brown', fg='white')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)

font1 = ('times', 14, 'bold')
upload = Button(main, text="Upload Student Grades Dataset", command=upload)
upload.place(x=50,y=100)
upload.config(font=font1)

pathlabel = Label(main)
pathlabel.config(bg='brown', fg='white')
pathlabel.config(font=font1)
pathlabel.place(x=350,y=100)

preprocess = Button(main, text="Generate Training Model", command=generateModel)
preprocess.place(x=50,y=150)
preprocess.config(font=font1)

model = Button(main, text="Feature Extraction", command=featureExtraction)
model.place(x=300,y=150)
model.config(font=font1)

runsvm = Button(main, text="Run SVM Algorithm", command=runSVM)
runsvm.place(x=500,y=150)
runsvm.config(font=font1)

runrandomforest = Button(main, text="Run Random Forest Algorithm",
command=runRandomForest)
runrandomforest.place(x=710,y=150)
runrandomforest.config(font=font1)
```

```
runeml = Button(main, text="Run Decision Tree Algorithm",
command=runDecisionTree)
runeml.place(x=50,y=200)
runeml.config(font=font1)

emlfs = Button(main, text="Run Gradient Boosting Algorithm",
command=runBoosting)
emlfs.place(x=330,y=200)
emlfs.config(font=font1)

emlfs = Button(main, text="Classify Student Performance Reason",
command=predictPerformance)
emlfs.place(x=640,y=200)
emlfs.config(font=font1)

graph = Button(main, text="Accuracy Graph", command=graph)
graph.place(x=990,y=200)
graph.config(font=font1)

font1 = ('times', 12, 'bold')
text=Text(main,height=30,width=150)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=10,y=250)
text.config(font=font1)

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

CHAPTER-6

CHAPTER -6
SCREENSHOTS

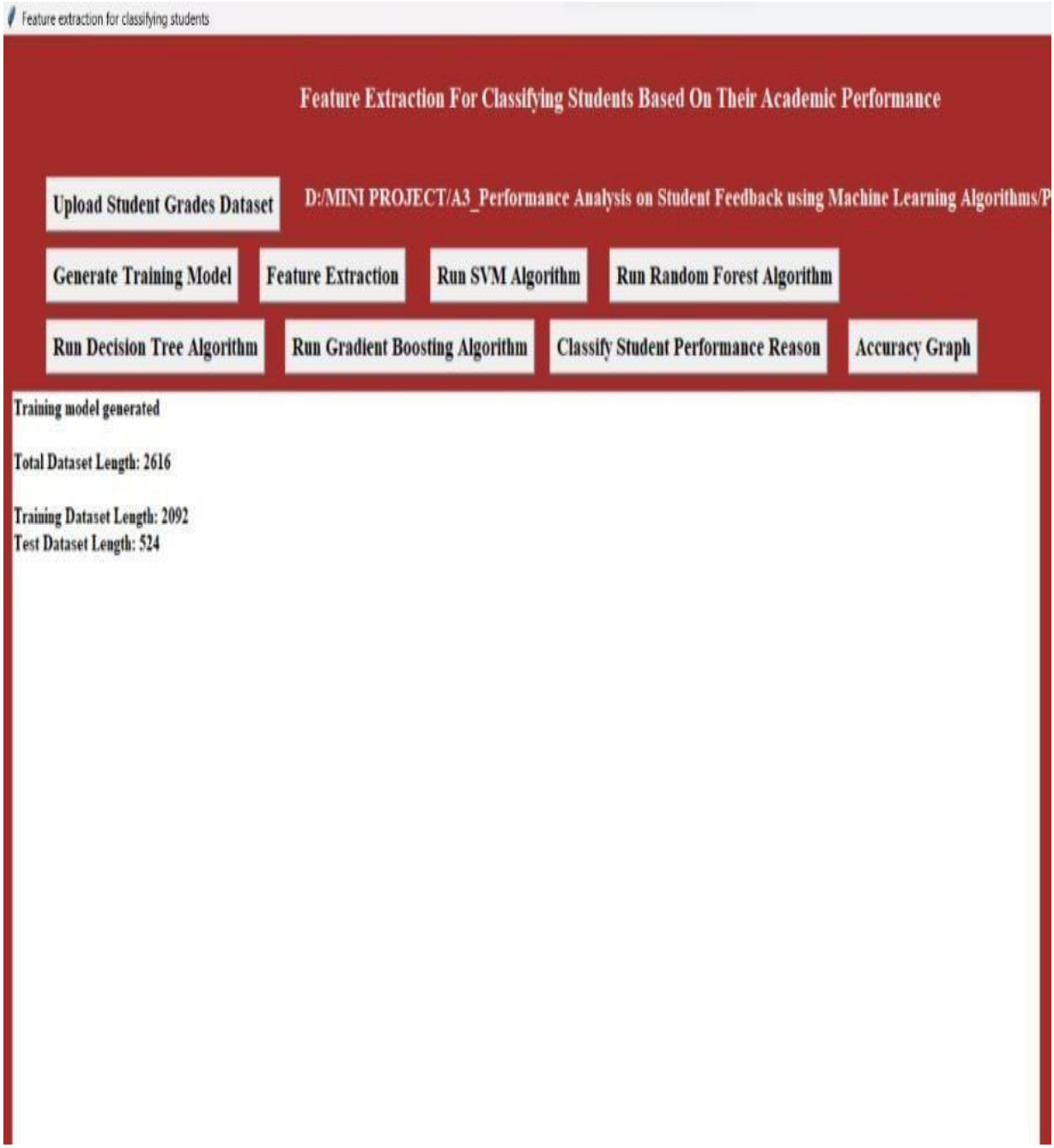

Figure 5.1 Upload Dataset & Generate Training Model

The project's homepage interface serves as the gateway for users to analyze students feedback on faculty. With a focus on user-friendly design and robust security measures, the interface sets the stage for a positive user interaction.

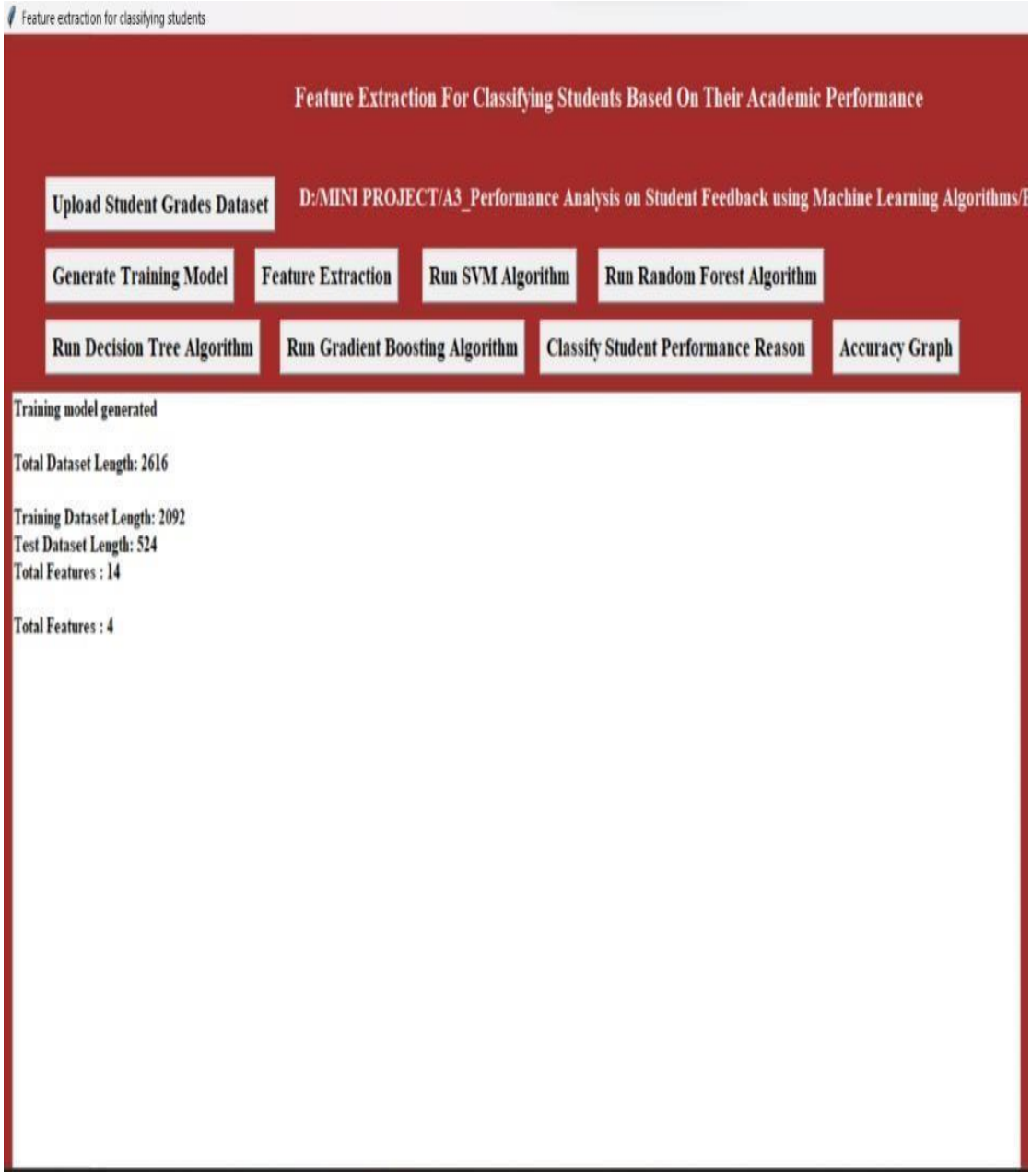

Figure 5.2 Feature Extraction

Feature extraction is a process in machine learning and data analysis that involves identifying and extracting relevant features from raw data. These features are later used to create a more informative dataset, which can be further utilized for various tasks such as: Classification.

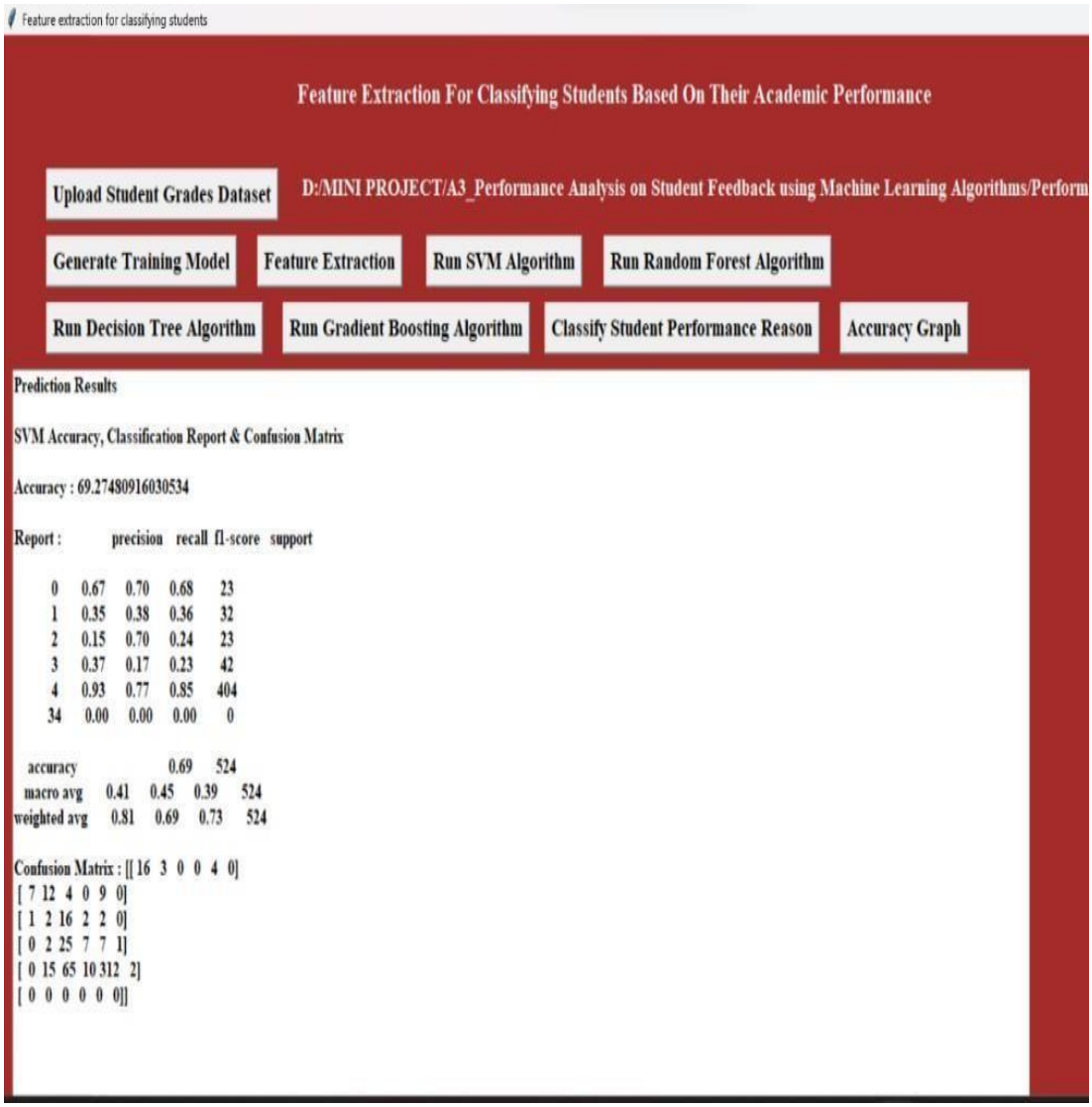

Figure 5.3 SVM Algorithm

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is use foClassification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future.

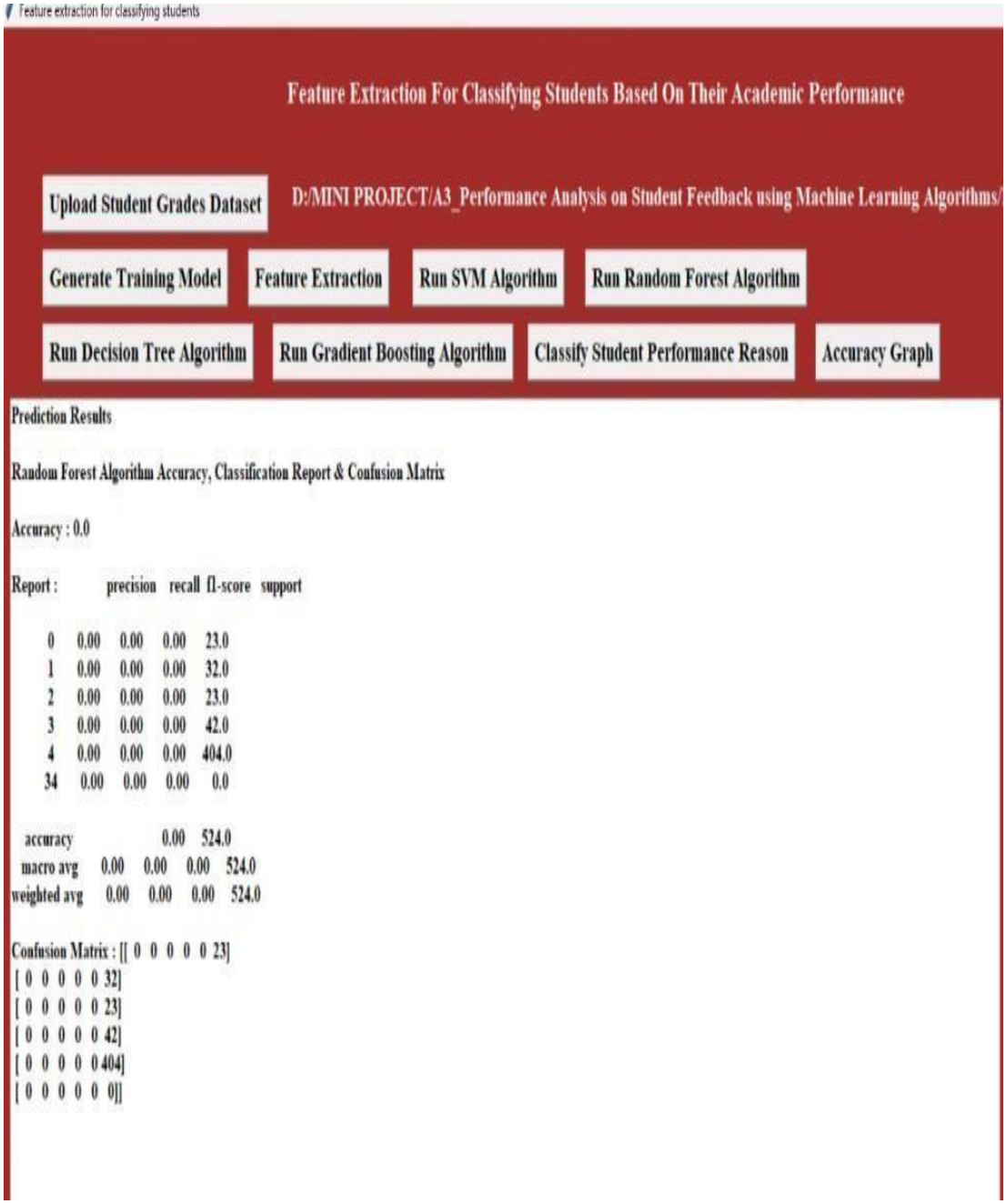

Figure 5.4 Random Forest Algorithm

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

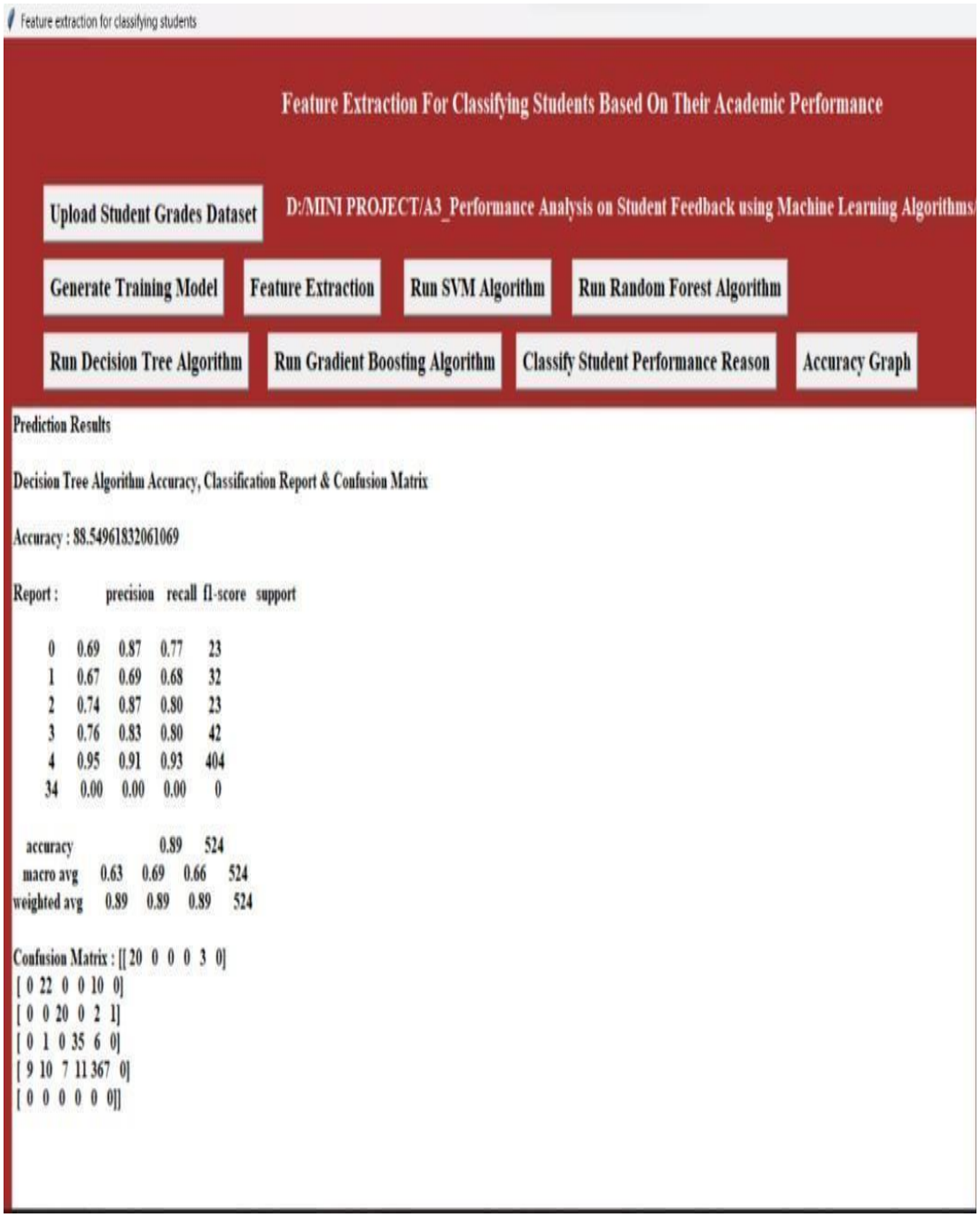

Figure 5.5 Decision Tree Algorithm

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

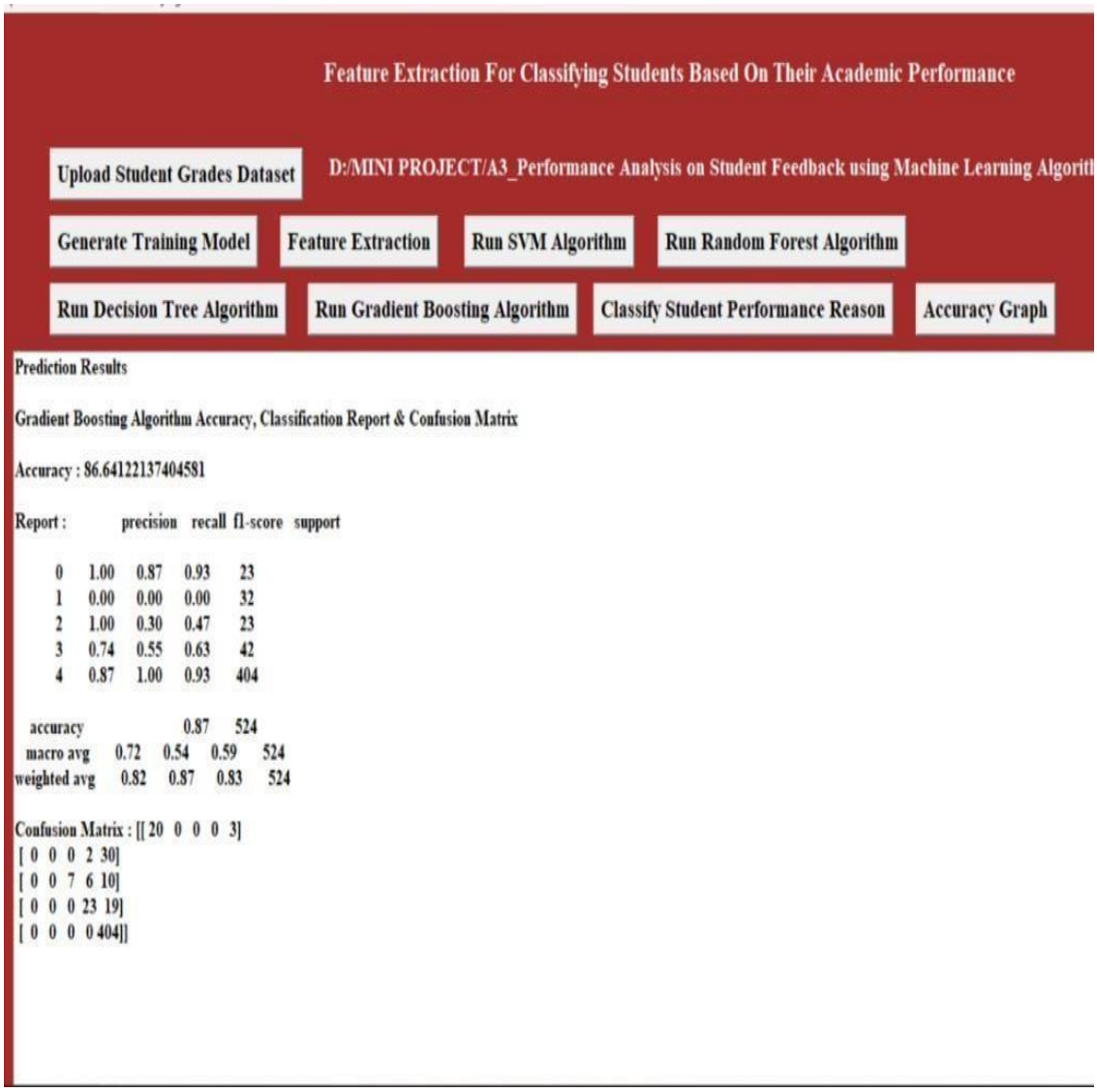

Figure 5.6 Gradient Boosting Algorithm

Gradient Boosting is a powerful boosting algorithm that combines several weak learners into strong learners, in which each new model is trained to minimize the loss function such as mean squared error or cross-entropy of the previous model using gradient descent. In each iteration, the algorithm computes the gradient of the loss function with respect to the predictions of the current ensemble and then trains a new weak model to minimize this gradient.

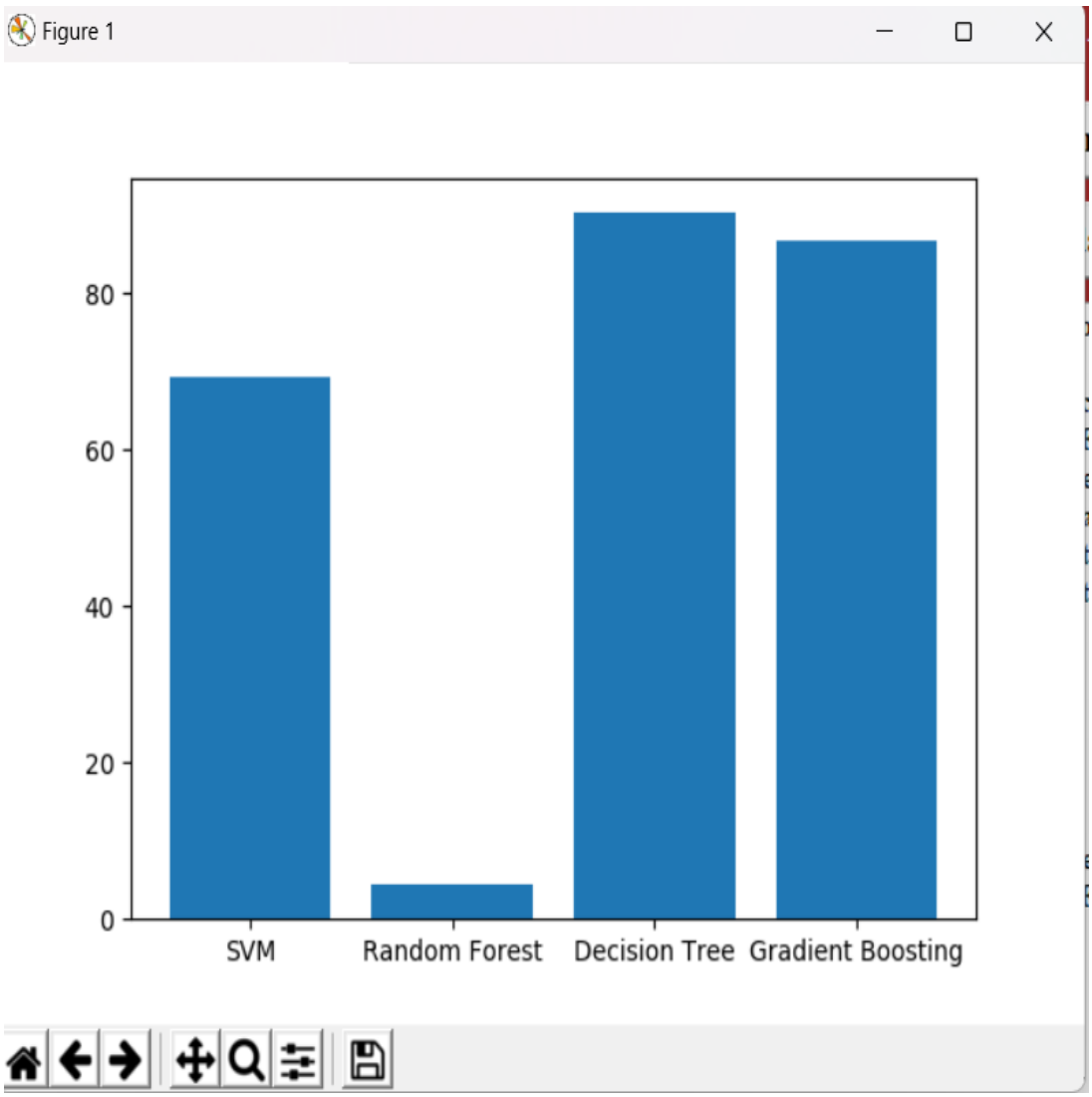

Figure 5.8 Accuracy Graph of algorithm

The combination of 4 different types of algorithms are displayed in the form of accuracy graph in order to analyze the graphs.

CHAPTER-7

CHAPTER-7
TESTING

TYPES OF TESTS

7.1 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.

7.2 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

7.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as 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 exercised. Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

CHAPTER-8

Chapter -8
CONCLUSION & FUTURE SCOPE

8.1 PROJECT CONCLUSION

Sentence level sentiment mining become accustomed removes the remarked normal alternatives and evaluation words from the contribution dataset. A Student Feedback Mining System is work to inquire about points and their slants as of understudy produced criticism. This strategy will be useful to improve student knowledge and educator's process for conveyance. Automating the student's feedback may give several advantages together with saving price, time and creating economical report generation, etc. the utilization of opining mining will facilitate in summarizing the feedback report effectively and evaluating school performance in the type of a summarized read might be helpful for the establishments. The opinion mining is that the method of insights taking out that be procured to look into assessments of students for any examination. For the length of this, the conclusion mining at the coed remarks created through reviews abuse managed gadget becoming acquainted with calculations upheld through Python.

8.2 PROJECT FUTURE SCOPE

Implementing machine learning in student feedback analysis not only streamlines data processing but also sets the stage for future improvements. These algorithms predict performance trends, offering insights for targeted enhancements. Future directions involve refining these algorithms for better understanding of diverse feedback sources and integrating advanced natural language processing for more accurate sentiment analysis. Ongoing technological advancements may enable real-time analysis and personalized recommendations, empowering educational institutions in their continuous efforts to enhance teaching methodologies.

REFERENCES

[1]. V. Dhanalakshmi ; Dhivya Bino "Opinion mining from student feedback data using supervised learning algorithms", 2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC)

[2]. Sarpreet Kaur1 , Rasleen Deol," Students Feedback for Mining Their Opinions Using Supervised Learning Algorithm" International Journal of Engineering Science and Computing, June 2017

[3]. Vaibhav Jain,Evaluating and Summarizing Student's Feedback Using Opinion Mining ,

International Journal of Scientific Research in Computer Science and Engineering

[4]. Anushree Chitriv, Prof. A. Thomas Analysis of Student Feedback by Using Data Mining Techniques , Helix 2018

[5]. Khan, Khairullah, "Mining opinion components from unstructured reviews: A review," Journal of King Saud University – Computer and Information Sciences , Vol. 26, 2016.

[6]. Pang, B. and Lillian L. s.l, "Opinion mining and sentiment analysis : Foundations and trends in information retrieval", Vol. 2, 2016