

A PROJECT REPORT (CS323)
on
PREDICTION OF STUDENTS' CGPA USING ML
A report submitted in partial fulfilment of the requirement for the award of
The degree of
BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

by
Utkarsh Verma - 180178008 - CSE ML
Pratichi Painuly - 180178009 - CSE ML
Chetna Garg - 180178010 - CSE ML
Kartik Kumar - 180178014 - CSE ML

Under the Guidance of

Dr. Megha
Assistant Professor
School of Computing



SCHOOL OF COMPUTING
DIT UNIVERSITY, DEHRADUN

(State Private University through State Legislature Act No. 10 of 2013 of Uttarakhand and approved by UGC)

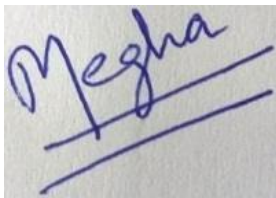
Mussoorie Diversion Road, Dehradun, Uttarakhand - 248009, India.

April, 2021

CANDIDATES DECLARATION

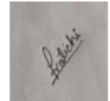
We hereby certify that the work, which is being presented in the Report, entitled **Prediction of Student's CGPA using ML**, in partial fulfilment of the requirement for the award of the Degree of **Bachelor of Technology** and submitted to the DIT University is an authentic record of our work carried out during the period January 2021 to April 2021 under the guidance of **Dr. Megha**.

Date: February 2021



Signature of the Mentor

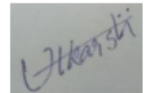
Pratichi Painuly



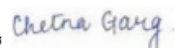
Kartik Kumar



Utkarsh Verma



Chetna Garg



Signature of the Candidates

ACKNOWLEDGEMENT

We would like to express our gratitude to our mentor, Dr Megha as well as the DIT University for giving us this opportunity to do this wonderful project on the topic **Prediction of Student's CGPA using ML**, which led us in doing a lot of research and we came across so many new things and topics which truly helped us to improve our skills as a coder. We are really thankful to them.

Secondly we would also like to thank our respective parents and each group member without whom this project could not have completed on the given time frame.

Utkarsh Verma – 180178008 - 100010770
Pratichi Painuly – 180178009 - 1000010808
Chetna Garg – 180178010 - 1000011235
Kartik Kumar – 180178014 - 1000010710

ABSTRACT

Students' academic performance is a very concerned issue as it is the reflection of both academic background and family support. It is very important for the educational institution to track the performance record because it can help them to improve the quality of their education.

Thus, the role of early CGPA prediction system comes into place. A CGPA Prediction system, this project seeks to create a model that can predict the CGPA of students based on some certain features. The main aim of this project is the prediction of the result in the form of CGPA of students for any subject considering schooling marks, continuous assessment and final marks during their academic semester using different Machine Learning models.

Because of some advantages and disadvantages present in every model we must find the most suitable model with maximum accuracy as our final model for the project. Some models that are widely been used and are found effective in this field are Linear Regression, Artificial Neural networks, and Deep Learning Techniques. As of now Deep Learning techniques are found most effective because of their high accuracy and performance, but it all depends on the attributes used in the prediction.

TABLE OF CONTENT

<u>CHAPTER</u>	<u>PAGE No.</u>
Candidate's Declaration	2
Acknowledgement	3
Abstract	4
Chapter 1 – Introduction	9-10
Chapter 2 –Project Description	11
2.1 Purpose	11
2.2 Problem statement	12
2.3 Special Features	12
Chapter 3 –Tools and Technologies	13
Example – Hardware and Software	
Chapter 4- Implementation Modules and Screen Shots	14-29
(List and explain all modules and every screenshot will have explanation and figure number)	
Chapter 5 – Conclusion &Road map of Phase-3	
Bibliography	

Plagiarism report 25% only

LIST OF FIGURES

<u>FIGURE NAME</u>	<u>PAGE</u>
1.1 Educational Data Mining	9
4.1 Survey Form	14-16
4.2 Response from survey	17
4.3 Snip of dataset	18
4.4 Importing libraries and dataset.	18
4.5 Encoding and handling missing values.	19
4.6 Feature Scaling	20
4.7 Splitting into training and testing sets.	20
4.8 Variable Explorer Window	20
4.9 Formula and calculation of MLR	21
4.10 Implementation of MLR	22
4.11 Working of Random Forest Regression	22
4.12 Implementation of Random Forest Regression	22
4.13 Working of Random Forest Classification	24
4.14 Implementation of Random Forest Classification	24
4.15 Formula and graph of Logistic Regression	24
4.16 Implementation of Logistic Regression	25
4.17 Working of Decision Tree Classification	25
4.18 Implementation of Decision Tree Classification	26
4.19 Formula of Naïve Bayes' Classifier	26
4.20 Implementation of Naïve Bayes' Classifier	26
4.21 Implementation of Error calculation and Visualising the results.	28
4.22 Visualising the result.	29

LIST OF TABLES

<u>TABLE NAME</u>	<u>PAGE</u>
4.1 Summarization of results	29

LIST OF ABBREVIATIONS

1. CGPA- Cumulative Grade Point Average
2. GPA- Grade Point Average
3. ML- Machine Learning
4. AI- Artificial Intelligence
5. ANN- Artificial Neural Network
6. SVM- Support Vector Machine
7. EDM- Educational Data Mining
8. MLR- Multiple Linear Regression
9. MAE- Mean Absolute Error
10. MSE- Mean Squared Error
11. RMSE- Root Mean Squared Error

Chapter 1 – Introduction

Generally, a process that analyses data and stats to predict every outcome taking the help of data models is called predictive model [1]. Predictive modelling is basically a method of building a model that can predict. Usually, any such model includes a machine learning algorithm that learns certain properties from a training dataset to make those predictions. Predictive modelling is divided into two areas: Regression and pattern classification. Regression models are based on the analysis of relationships between variables and trends to make predictions about continuous variables. Using machine learning algorithm, we can recognize patterns, this is termed as pattern recognition.

Education is a very important aspect in terms of economy that is why many researchers are developing various techniques to improve the performance of the students, one way to do so is to track the performance of the students. Through R&D in this field, students can be benefitted in many ways, like teachers can give special attention to the students whose predicted CGPA is low or not up to the mark and this will be very helpful for the student and for the university as well as their all over result percentage will improve on the other hand students can also track their performance and hence can improve their study pattern and perform well.

Here the terminology Educational data mining [2] comes into role which is an emerging discipline, used to explore the distinctive and increasingly sizeable data obtained from educational institutes, and use those techniques to understand students and the methods in which they learn. Through to explore these large datasets, using different data mining techniques, one can identify unique patterns which will help to study, predict, and improve a student's academic performance.

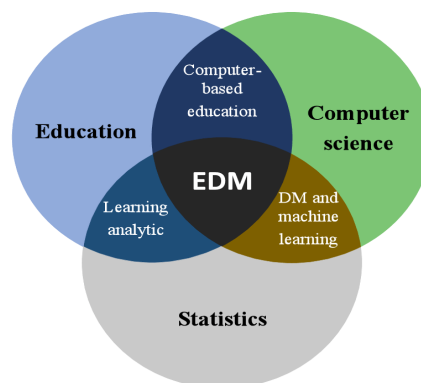


Figure 1.1 Educational Data Mining

As we all know COVID-19 outbreak has brought with it the unique challenges that was not expected by anyone earlier, especially in Education system. Due to the pandemic, schools, colleges, universities are closed since March 2020. Most of the exams has been cancelled or held online. In the higher education, the UGC has given the flexibility to the Universities to decide whether they will take the online exam or will transfer the students to next academic year based on the previous results attained by them which is not fair for the students. In-fact most importantly, the pandemic questions the value provided by a university which comprise social opportunities as well as education [3].

Some universities followed the former, while some went for promoting the students to next semester by providing them grades based on assignments marks and the marks scored by the students in the previous semester. As each year brings new subjects during higher education, assigning the grades based only on the result of previous semester is not a fair mean. Here we propose a model in which the results of students will not be dependent on one single feature, and along with that each subject result will be predicted by assigning different weights to each factor, to predict the result as fair as possible. For the same we will apply different Supervised Machine Learning techniques for predicting the results of the students, the models will be trained using the results of the senior students [4,5].

In phase 1, we performed literature study where we read research papers of different authors highlighting the same issue and learnt about which machine learning and deep learning models could be considered for our prediction.

In the second phase, we collected the dataset from the students at our university through Microsoft Survey forms and further pre-processed it and applied various machine learning models. Then we will compare different Machine Learning models by applying on the dataset to find the model that works best for predicting the student CGPA.[6]

And in final phase, the hybrid model will be proposed by considering and combining the techniques which has worked in the best way in all the research papers which we have considered so far in phase 1.

Chapter 2 – Project Description

The main features of this project are predicting the CGPA of student based on different features. This project uses some main features like quiz marks, internal assessment marks, attendance, IQ score & final marks will be used to predict the CGPA. In order to predict more accuracy and precision, different weightage will be provided to each attribute. This model will show if the student needs any special attention or if he is over-confident, also if he is taking proper rest or not. To minimize the complications of prediction, the information of senior students will be used to predict the marks of the present students.

2.1. Purpose

The main purpose of this project is to predict student's CGPA with the help of different features using machine learning techniques which will help the student to analyse what are the fields in which he is lagging behind, if he's being overconfident, lousy, hard-working, taking proper rest etc. Knowing all these things, students can self-improve and self-develop by working hard and perform better in college academics to achieve a better CGPA.

Also, this project can be really helpful in global pandemics like COVID-19 which has proved to be a major problem in every aspect like individuals' physical and balance, country's economy, job opportunities, educational sector, business etc. In such time, this project can contribute towards the educational sector by predicting students CGPA and helping them analyse their shorthand and their area of weaknesses.

This factor could have been a major problem towards the universities as it would be really hard for predicting student's CGPA when there was no college or exams were being conducted. The colleges had to promote the students to the next semester on the basis of their past performance and results. So at that time, this model could be really helpful to predict the student's CGPA precisely and promoting him to next semester.

2.2. Motivation

This project has been a helpful and very useful machine learning techniques' implementation. Predicting the student's CGPA seemed to be impossible if we look 15-20 years back from

now. If the CGPA could be predicted precisely and accurately, it could really benefit many students and help them achieve a lot more.

This project aims at predicting the CGPA of the student so that it can help students who are in need of special attention and care. There are so many other models and techniques that can be used to achieve the main objective of this project. There are many algorithms and models that show most accurate results, but their accuracy also varies with the attributes taken.

2.3. Problem Statement

We can segregate students into three categories:

- I) Performs well in the beginning but degrade their performance by the end.
- II) Performs worse in the beginning but improve their performance by the end.
- III) Who consistently performs either better or worse.

These variations can cause trouble for teachers to figure out the students who are at risk and might not perform well. Predicting their CGPA is a lot helpful for them to figure out which students need more attention and need to work hard.

As we all know the current scenario of COVID-19 outbreak, it was a major problem for the universities to promote the students to next semester without taking any exams or classes (for some cases), making it really hard to assign marks/CGPA to students. Predicting their CGPA precisely and assigning their marks accordingly makes it easy for the university to promote the students.

2.4. Special Features

The special features in our model will be as follows:

- Prediction of student's CGPA based on different features.
- Different weightage will be provided to each attribute.
- Features like daily social media interaction, class attendance, attention during lectures, self-study time (both daily and during end-semesters), frequency of physical activity and SGPA 1, SGPA 2 and SGPA 3, where SGPA 4 is the predicted value.

Chapter 3 – Tools and Technologies

Hardware Requirements

- Ram: 8 GB
- Hard disk: 512 GB
- Processor: 3rd Generation Intel Core i5

Software Requirements

- Tools:

WEKA: Waikato Environment for Knowledge Analysis is a collection of machine learning algorithm for data mining task. It contains tools for regression, pre-processing & clustering etc. It is written in JAVA programming language.

MATLAB: It is a programming language and numerical computing environment which was developed by MathWorks. It allows matrix manipulations, plotting of functions and data, implementation of algorithms & creation of user interfaces etc.

- IDE:

Anaconda Navigator: It is a GUI included in anaconda distribution used to launch applications.

Spyder: It is a free Integrated Development Environment included in Anaconda.

Jupyter Notebook: This is another IDE of Anaconda Navigator.

Programming Requirements

- Language: Python

Technology Requirements

- Machine Learning: It is a branch of AI that focuses on building applications that can use real data to learn themselves. It helps us to analyse data in large quantities. This process of learning initiates with the data (or the collection of information), then further looking up for a specific pattern or a particular system common throughout the data and then finally preparing a model that can analyse the same pattern in future.
- Deep Learning: It is a branch of ML and is designed in such a way that it mimics the working of human brain to understand and learn functions. It is a key technology used in preparing driverless cars, voice control systems like Google Assistant and Siri.

Chapter 4- Implementation Modules and Screen Shots

4.1 Data Collection

(Link for the survey form <https://forms.office.com/r/tLk27FhyVw>)

The data was collected from the students of Computer Science engineering at our university through Microsoft forms. It contained the attributes required to predict the result of the student. It contained both academic as well as personal attributes. The list of attributes taken is as follows:

- Daily social media interaction
- Physical activity frequency
- Programming language knowledge
- Class attendance
- Self-study duration
- External study references
- Attention during class.
- Frequency of doubt clearing during group study.
- SGPA of previous semesters

Screenshots from the survey form are as follows:

Student's Information

Fill this form to the best of your knowledge.

Please use the following scale for rating:

Excellent/Regular: 8-10 Good/Mostly: 6-7 Average/Quite often: 3-5 Poor/Not really: 1-2

Hi CHETNA, when you submit this form, the owner will be able to see your name and email address.

* Required

1. SAP ID *

Enter your answer

2. Course and Branch: *

Select your answer

Figure 4.1(a) Survey Form

3. Your daily social media interaction *

☐ Less than 1 hour

☐ 1-2 hours

☐ More than 2 hours

4. Rate your programming skills (Fluency with languages: C/C++/Python/Java etc.)
(Only for CSE, IT, BCA and MCA students)

1 2 3 4 5 6 7 8 9 10

☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

5. Rate your physical activity (Dedicated workout session/sports) *

1 2 3 4 5 6 7 8 9 10

☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

Figure 4.1(b) Survey Form

6. Your class attendance *

☐ 85% - 95%

☐ 75% - 85%

☐ 65% - 75%

☐ 55% - 65%

☐ Below 55%

7. Rate your attention during lectures *

1 2 3 4 5 6 7 8 9 10

☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

8. How often do you take references from Google, YouTube, library or any other platform to clear your concepts? *

1 2 3 4 5 6 7 8 9 10

☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

Figure 4.1(c) Survey Form


9. Your daily study duration *

☐ Less than 1 hour

☐ 1-2 hours

☐ 2-4 hours

☐ More than 4 hours

10. Study duration during end-sem exams * 

☐ 2-4 hours

☐ 4-6 hours

☐ 6-8 hours

☐ More than 8 hours

11. How often do you clear doubts/topics of your classmates during group study? *


1 2 3 4 5 6 7 8 9 10

☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

Figure 4.1(d) Survey Form

12. Your SGPA 1 *

13. Your SGPA 2 *

14. Your SGPA 3 * 

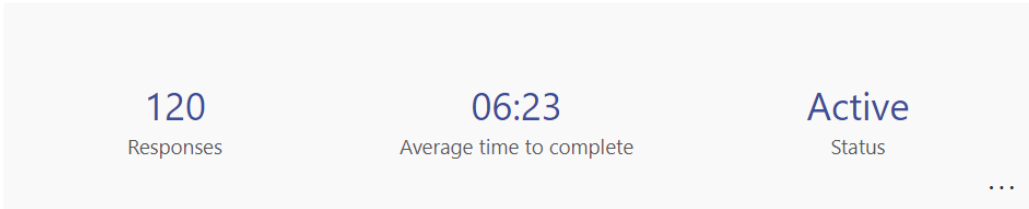
15. Your SGPA 4 *

16. Your SGPA 5 *

Figure 4.1(e) Survey Form

RESPONSE FROM THE SURVEY:

Student's Information



View results

Open in Excel

1. SAP ID

[More Details](#)

120 Responses

Latest Responses

"1000010627"
"1000011799"
"1000010965"

Figure 4.2(a) Response of survey

2. Course and Branch:

[More Details](#)

CSE	69
CSE-CCV	2
CSE-BDA	7
CSE-IOT	3
CSE-ML	25
CSE-CSF	0
IT	4
ME	4
ME-AE	0
PE	0
CE	3
ECE	2
EE	1
BCA	0
MCA	0

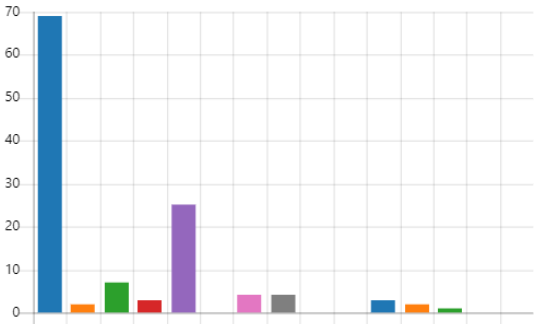


Figure 4.2(b) Response of survey

OUR DATASET (Link for the dataset: <https://bit.ly/2PXvv1c>)

1	6	8	85% - 95%	8	8	2	2	8	8.4	7.9	7.9	8.2
0	9	2	65% - 75%	1	8	0	2	8	7.8	7.5	7.9	7.8
0	8	7	75% - 85%	7	7	1	2	7	7.89	7.95	7.5	8
1	6	6	75% - 85%	7	8	1	0	5	9.3	8.83	8.67	8.68
0	4	9	75% - 85%	6	8	1	1	5	5.78	5.53	6	6
0	6	9	75% - 85%	7	9	1	3	6	6.38	7.51	6.67	7.91
1	9	10	85% - 95%	7	10	2	0	7	7.9	9	8.4	8.5

Figure 4.3 Snip of the dataset

4.2 Steps for Implementation

1. Importing the Libraries and Data Set

The libraries which we used were NumPy, Pandas and Matplotlib.

NumPy: This library is used to implement arrays. It's various functions are used for implementation in linear algebra, Fourier transform, and matrices.

Pandas: It is a software library used for data manipulation and analysis. It yields enhanced performance.

Matplotlib: It is a plotting library which is used for data visualization. It also allows us to generate fair quality line plots, scatter plots, histograms, bar charts etc.

```
#importing the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

#importing data set
dataset = pd.read_csv("Student's Information_cse - Copy.csv")
x=dataset.iloc[:,12].values
y=dataset.iloc[:,12].values
```

Figure 4.4 Importing libraries and dataset.

2. Data Pre-Processing

It is a step or a technique which is used to transform or encode data in a useful and efficient state so that the machine can easily understand or parse it.

a. Encoding

Label Encoder: When there is a dataset present which contains multiple labels in one or more columns which can be either words or numbers, to make it readable form we use label encoder.

One Hot Encoder: It allows categorical data representation to be more meaningful. Some machine learning algorithms cannot work with the categorical data directly. These categories must be converted into numbers. This is required for both input and output variables.

b. Handling missing values

It is a part of data pre-processing, real-world data may can have **missing values**, this can happen for several reasons such as observations that were over-looked or data corruption. Handling missing values is an important step in machine learning as the algorithms do not support data with missing values. There are multiple ways through which we can handle missing values in our dataset. Each comes with their own pros and cons. In our project we used Simple Imputer class to handle missing values. In simple words the missing values were replaced by the mean of the values of that specific column.

```
#label encoder
#one hot encoder
from sklearn.preprocessing import LabelEncoder
#from sklearn.compose import ColumnTransformer
labelencoder_x=LabelEncoder()
x[:,3]=labelencoder_x.fit_transform(x[:,3])
#ct=ColumnTransformer([('one_hot_encoder',OneHotEncoder(),[3])],remainder='passthrough')
#x=ct.fit_transform(x)

#missing values
from sklearn.impute import SimpleImputer
simpleimputer=SimpleImputer(missing_values=np.nan,strategy="mean")
simpleimputer=simpleimputer.fit(x[:,9:11])
x[:,9:11]=simpleimputer.transform(x[:,9:11])
```

Figure 4.5 Encoding and handling missing values.

c. Feature Scaling

It is a method that is used to uniform the independent fields. It is used to handle highly variable values. If this is not performed, then the ML model tends to weigh bigger values as higher and smaller values as lower.

```
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
```

Figure 4.6 Feature Scaling

d. Splitting the Data Set into Training & Test Set

This step is the core of data pre-processing, here we need to split the data into training and testing sets. Model is trained on the training set and then is applied on the testing set to check the model. Usually, it is divided by 20/80 rule, that is 20% data for testing and 80% for training. In our project we have performed the same.

```
#splitting the data set into training and testing sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 0)
```

Figure 4.7 Splitting into training and testing sets.

e. Variable Explorer

It allows us to correlatively browse and manage the objects generated running the code. It also gives us information on the name, size, type, and value of each object. Here is the variable explorer window of our dataset.

X_test	Array of float64	(22, 12)	<pre>[[-1.24740394 -0.3300165 -0.4203364 ... 0.185... 0.2491364 ... [0.21621668 -1.65008251 0.90072087 ... -1.450... -1.5778641 ...</pre>
X_train	Array of float64	(88, 12)	
classifier	ensemble._forest.RandomForestClassifier	1	RandomForestClassifier object of sklearn.ensemble._forest module
cm	Array of int64	(3, 3)	<pre>[[3 1 0] [2 15 0]</pre>
dataset	DataFrame	(110, 16)	Column names: Your daily social media interaction2, Rate your programm ...
labelencoder_x	preprocessing._label.LabelEncoder	1	LabelEncoder object of sklearn.preprocessing._label module
r2_test	float64	1	0.1287128712871285
sc_X	preprocessing._data.StandardScaler	1	StandardScaler object of sklearn.preprocessing._data module
x	Array of object	(110, 12)	ndarray object of numpy module
y	Array of int64	(110,)	<pre>[1 1 1 ... 1 1 1]</pre>
y_pred	Array of int64	(22,)	<pre>[1 1 1 ... 0 1 1]</pre>
y_test	Array of int64	(22,)	<pre>[1 1 1 ... 0 1 1]</pre>
y_train	Array of int64	(88,)	<pre>[1 1 0 ... 0 1 1]</pre>

Figure 4.8 Variable Explorer Window

4.3 The learning process: Applying the machine learning models.

After the data-preprocessing, our dataset is ready to be applied on machine learning models. Here we used, two types of algorithms to train and test our dataset: Regression and Classification.

Regression:

It is a supervised learning algorithm, works with labeled datasets. It is used to find the connections between output (dependent) and input (independent) variables. The main task of this algorithm is to search for the function to map the input variable(s) to the output variable which is a continuous value.

This type of algorithm mainly solves the prediction problems like CGPA prediction, rain prediction and more.

In our project, we applied regression, to predict the SGPA 4 of the student using various factors which we collected through survey and taking the continuous progress of student through his/her previous SGPA's.

The models we applied here are:

- a. Multiple Linear Regression
- b. Random Forest Regression

Multiple Linear Regression: It is a statistical technique that uses more than one dependent variable to predict the output variable. It is also known as Multiple Regression. The main aim of this algorithm is the predict and model the linear relationship between the independent variables (input) and dependent (response/output) variable. It is an extension of Simple Linear regression that uses only one dependent variable.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} + \epsilon$$

where, for $i = n$ observations:

y_i = dependent variable
 x_i = explanatory variables
 β_0 = y-intercept (constant term)
 β_p = slope coefficients for each explanatory variable
 ϵ = the model's error term (also known as the residuals)

Figure 4.9 Formula and Calculation of Multiple Linear Regression

```
#fitting Multiple linear regression to the training set
from sklearn.linear_model import LinearRegression
regressor=LinearRegression()
regressor.fit(X_train,y_train)

#predicting the test set results
y_pred=regressor.predict(X_test)
```

Figure 4.10 Implementation of Multiple Linear Regression

Random Forest Regression: It is a supervised Machine learning algorithm; it uses **ensemble learning** method for regression. Ensemble learning is a technique that combines the predictions from multiple machine learning algorithms to make a more accurate prediction than a single model. In this case we have combined multiple decision tree models to result into a single random Forest Regression.

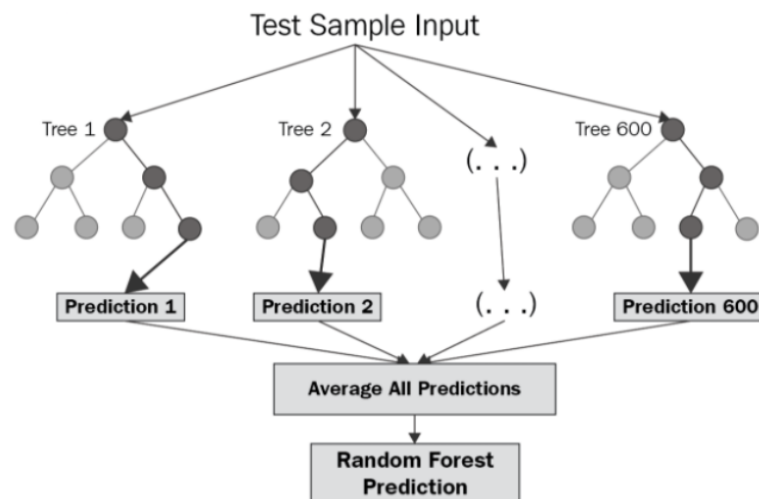


Figure 4.11 Working of Random Forest Regression.

```
#Fitting the Random forest Regression model to the data set
from sklearn.ensemble import RandomForestRegressor
regressor=RandomForestRegressor(n_estimators=200,random_state=0)
regressor.fit(x,y)

#Predict Result
y_pred=regressor.predict(X_test)
```

Figure 4.12 Implementation of Random Forest Regression

Classification:

It is a supervised learning algorithm, where the dataset is divided into classes based on different parameters. In this process, the dataset is trained on training set and according to it, it divides the data into different classes.

In our project, we divided the previous SGPA of the students in 3 classes, according to which we trained the model and predicted the SGPA 4.

The 3 classes of SGPA are of the range as follows:

SGPA range	Class
0-6.5	0
6.5-8.5	1
8.5 above	2

The models here we applied are:

- Random Forest Classification
- Logistic Regression Classification
- Decision Tree Classification
- Naïve Bayes' Classification

Random Forest Classification: It is a supervised learning algorithm, here we used it to classify. This algorithm makes decision trees on data and then predicts, and finally selects the best solution by means of voting/majority.

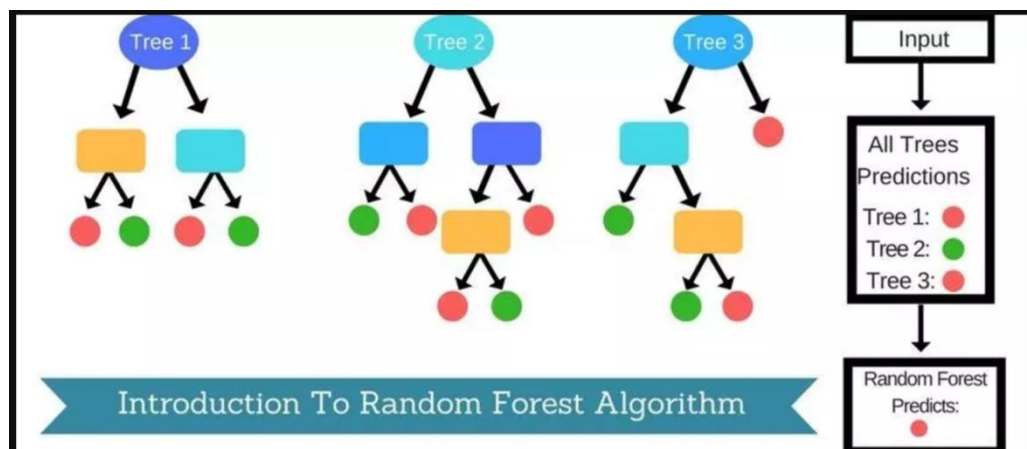


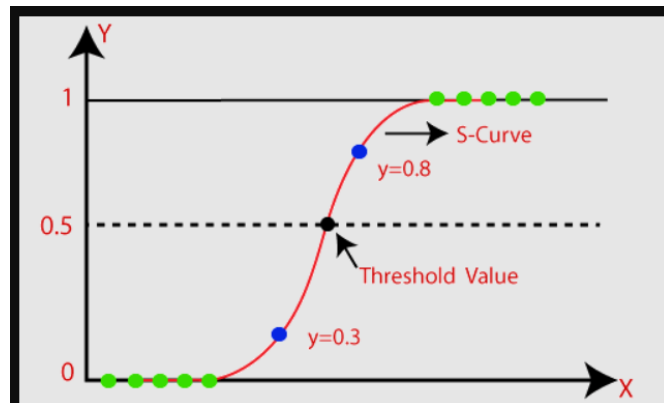
Figure 4.13 Working of Random Forest Classification

```
#Fitting Random Forest Classification to the training set
from sklearn.ensemble import RandomForestClassifier
classifier=RandomForestClassifier(n_estimators=20,criterion="entropy",random_state=0)
classifier.fit(X_train,y_train)

#Predicting the test set results
y_pred=classifier.predict(X_test)
```

Figure 4.14 Implementation of Random Forest Classification

Logistic Regression Classification: It is a supervised classification algorithm which works on predictive modelling; therefore, it is called logistic regression, but is used for classifying the samples, hence, it is called the classification algorithm. It is used to predict the output of a categorical dependent variable. The outcome must be a categorical or discrete value. It gives the probabilistic values which lie between 0 and 1.



$$g(z) = \frac{1}{1+e^{-z}}$$

Figure 4.15 Formula and graph for Logistic Regression

```
#Fitting Logistic Regression to the training set
from sklearn.linear_model import LogisticRegression
classifier=LogisticRegression(random_state=0, max_iter=200)
classifier.fit(X_train,y_train)

#Predicting the test set results
y_pred=classifier.predict(X_test)

#Making the confusion Matrix
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,y_pred)
```

Figure 4.16 Implementation of Logistic Regression

Decision Tree Classification: It is a supervised ML method, It is a tree-structured classifier. It consists of two nodes, Decision Node and Leaf Node. Decision nodes makes the decision and Leaf nodes are the output of those decisions.

It is so called because it is like a tree, it starts with the root node, and expands on further branches constructing a tree-like structure.

In simple words it asks a question and based on the answer (Yes/No), it further split the tree into subtrees.

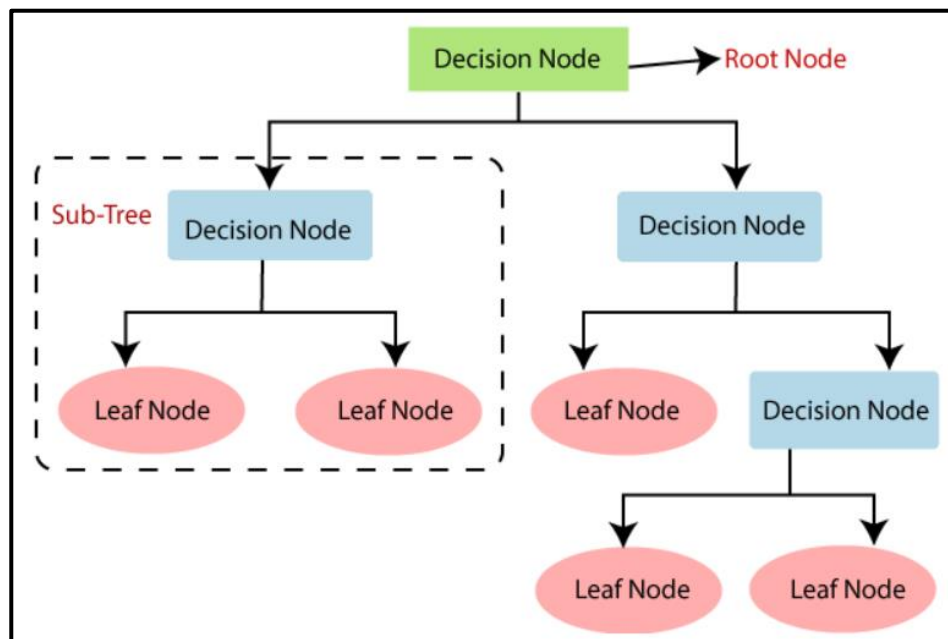


Figure 4.17 Working of Decision Tree Classification

```
#Fitting Decision Tree to the training set
from sklearn.tree import DecisionTreeClassifier
classifier=DecisionTreeClassifier(criterion="entropy",random_state=0)
classifier.fit(X_train,y_train)

#Predicting the test set results
y_pred=classifier.predict(X_test)
```

Figure 4.18 Implementation of Decision Tree Classification

Naïve Bayes Classification: It is a supervised ML method, which is based on Bayes theorem and used to classify. It is a probabilistic classifier, that is, it predicts based on the probability of an object. Formula for Naïve Bayes is given here:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Figure 4.19 Formula of Naïve Bayes' Classifier

Where,

$P(A|B)$ is termed as **Posterior probability** which means probability of hypothesis event A on the observed event B.

$P(B|A)$ is termed as **Likelihood probability** which means probability of the evidence given that the probability of a hypothesis is true.

$P(A)$ is termed as **Prior Probability** which means probability of hypothesis before observing the evidence.

$P(B)$ is termed as **Marginal Probability** which means the probability of evidence.

```
#Fitting Naive Bayes to the training set
from sklearn.naive_bayes import GaussianNB
classifier=GaussianNB()
classifier.fit(X_train,y_train)

#Predicting the test set results
y_pred=classifier.predict(X_test)
```

Figure 4.20 Implementation of Naïve Bayes' Classifier

4.4 Error Calculation and Visualising the results

Error Calculation

Here we used different error calculating techniques such as mean absolute error, mean squared error, and root mean squared error.

Mean Absolute Error

The mean absolute error is the average of all absolute error. Its formula is:

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - x_i|}{n}$$

MAE = mean absolute error

y_i = prediction

x_i = true value

n = total number of data points

Mean Squared Error

It is the mean or average of the square of the difference between actual and estimated values.

Its formula is:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

MSE = mean squared error

n = number of data points

Y_i = observed values

\hat{Y}_i = predicted values

Root Mean Squared Error

It is the square root of the mean of the square of all the error.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N (x_i - \hat{x}_i)^2}{N}}$$

RMSE = root-mean-square deviation

i = variable i

N = number of non-missing data points

x_i = actual observations time series

\hat{x}_i = estimated time series

```
#Error calculation
from sklearn import metrics
def print_error(X_test, y_test, model_name):
    prediction = model_name.predict(X_test)
    print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, prediction))
    print('Mean Squared Error:', metrics.mean_squared_error(y_test, prediction))
    print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, prediction)))

print_error(X_test,y_test, classifier)

print('Train Score: ', classifier.score(X_train, y_train))
print('Test Score: ', classifier.score(X_test, y_test))

from sklearn.metrics import r2_score
r2_test = r2_score(y_test, y_pred)
print('R2 score: ',r2_test)

#Graph
plt.title("Random Forest Classifier")
plt.plot(y_test)
plt.plot(y_pred)
plt.xlabel("Number of students")
plt.ylabel("SGPA 4")
plt.legend(["Test Values","Predicted Values"])
plt.show()
```

Figure 4.21 Implementation of Error calculation and Visualising the results.

Visualising the results

After the error calculation, we visualized the results through graphs with the help of Matplotlib library. Following is the sample of one graph visualisation of Random Forest Regression:

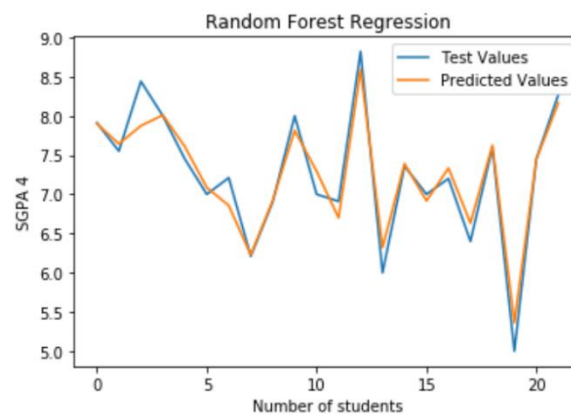


Figure 4.22 Visualising the result

4.4 Summarization of results

Following is the table comparing all the machine learning models applied till now based on the different errors and the accuracy.

Table 4.1 Summarization of results

S No.	Name of the model	MAE	MSE	RMSE	Train Score	Test Score
1.	Multiple Linear Regression	0.39	0.30	0.55	0.79	0.51
2.	Random Forest Regression	0.16	0.046	0.21	0.97	0.93
3.	Random Forest Classification	0.18	0.18	0.42	1.0	0.81
4.	Logistic Regression	0.18	0.18	0.42	0.97	0.81
5.	Decision Tree Classification	0.27	0.27	0.52	1.0	0.73
6.	Naïve Bayes' Classification	0.28	0.27	0.50	0.92	0.72

REFERENCES

- [1] Ganorkar, S. S., Tiwari, N., & Namdeo, V. (2020). Analysis and Prediction of Student Data Using Data Science: A Review. *Smart Trends in Computing and Communications: Proceedings of SmartCom 2020*, 443-448.
- [2] Nghe, N. T., Janecek, P., & Haddawy, P. (2020). A comparative analysis of techniques for predicting academic performance. In *37th annual frontiers in education conference-global engineering: knowledge without borders, opportunities without passports* (pp. T2G-7). IEEE.
- [3] Shetu, S. F., Saifuzzaman, M., Moon, N. N., Sultana, S., & Yousuf, R. Student's Performance Prediction Using Data Mining Technique Depending on Overall Academic Status and Environmental Attributes. In *International Conference on Innovative Computing and Communications* (pp. 757-769). Springer, Singapore.
- [4] Lau, E. T., Sun, L., & Yang, Q. (2019). Modelling, prediction and classification of student academic performance using artificial neural networks. *SN Applied Sciences*, 1(9), 982.
- [5] Rifat, M. R. I., Al Imran, A., & Badrudduza, A. S. M. (2019, May). EduNet: a deep neural network approach for predicting CGPA of undergraduate students. In *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)* (pp. 1-6). IEEE.
- [6] Kumar, V., & Garg, M. L. (2019). Comparison of machine learning models in student result prediction. In *International Conference on Advanced Computing Networking and Informatics* (pp. 439-452). Springer, Singapore.
- [7] Dhamija, P., Nandal, R., & Sehwat, H. A REVIEW PAPER ON PREDICTION ANALYSIS: PREDICTING STUDENT RESULT ON THE BASIS OF PAST RESULT.
- [8] Pushpa, S. K., Manjunath, T. N., Mrunal, T. V., Singh, A., & Suhas, C. (2017, August). Class result prediction using machine learning. In *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)* (pp. 1208-1212). IEEE.
- [9] Kumar, M., Singh, A. J., & Handa, D. (2017). Literature survey on student's performance prediction in education using data mining techniques. *International Journal of Education and Management Engineering*, 7(6), 40-49.
- [10] Sikder, M. F., Uddin, M. J., & Halder, S. (2016, May). Predicting students yearly performance using neural network: A case study of BSMRSTU. In *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)* (pp. 524-529). IEEE.
- [11] Halde, R. R., Deshpande, A., & Mahajan, A. (2016, May). Psychology assisted prediction of academic performance using machine learning. In *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)* (pp. 431-435). IEEE.
- [12] Sebastian, S., & Puthiyidam, J. J. (2015). Evaluating students performance by artificial neural network using weka. *International Journal of Computer Applications*, 119(23).

- [13] Arsad, P. M., &Buniyamin, N. (2014, April). Neural Network and Linear Regression methods for prediction of students' academic achievement. In *2014 IEEE Global Engineering Education Conference (EDUCON)* (pp. 916-921). IEEE.
- [14] Kabakchieva, D. (2013). Predicting student performance by using data mining methods for classification. *Cybernetics and information technologies*, 13(1), 61-72.
- [15] Kabra, R. R., &Bichkar, R. S. (2011). Performance prediction of engineering students using decision trees. *International Journal of computer applications*, 36(11), 8-12.
- [16] Ibrahim, Z., &Rusli, D. (2007, September). Predicting students' academic performance: comparing artificial neural network, decision tree and linear regression. In *21st Annual SAS Malaysia Forum*, 5th September.

BIBLIOGRAPHY

1. www.google.com
2. <https://towardsdatascience.com/>
3. <https://www.coursera.org/>
4. <https://www.sciencedirect.com/>
5. <https://datasetsearch.research.google.com/>
6. <https://www.springer.com/gp>
7. <https://scholar.google.com/>
8. <https://ieeexplore.ieee.org/Xplore/home.jsp>
9. <https://dl.acm.org/>
10. <https://medium.com/>

Plagiarism Report

G49

ORIGINALITY REPORT

20%	14%	8%	15%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Govind Ballabh Pant Engineering College, Pauri-Garhwal Student Paper	2%
2	Submitted to Staffordshire University Student Paper	2%
3	www.javatpoint.com Internet Source	2%
4	chayabakshi.medium.com Internet Source	1%
5	www.grrroups.com Internet Source	1%
6	Submitted to DIT university Student Paper	1%
7	machinelearningmastery.com Internet Source	1%
8	Submitted to Mar Baselios Engineering College Student Paper	1%
9	coek.info	

	Internet Source	1%
10	Submitted to northcap Student Paper	1%
11	en.wikipedia.org Internet Source	1%
12	John Jacob, Kavya Jha, Paarth Kotak, Shubha Puthran. "Educational Data Mining techniques and their applications", 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), 2015 Publication	1%
13	ieeexplore.ieee.org Internet Source	1%
14	medevel.com Internet Source	<1%
15	barmaja-ar.blogspot.com Internet Source	<1%
16	Submitted to Lovely Professional University Student Paper	<1%
17	dokumen.pub Internet Source	<1%
18	Submitted to Rochester Institute of Technology Student Paper	<1%

19	Submitted to Jaypee University of Information Technology Student Paper	<1%
20	Submitted to Multimedia University Student Paper	<1%
21	www.3tier.com Internet Source	<1%
22	Submitted to De Montfort University Student Paper	<1%
23	jultika.oulu.fi Internet Source	<1%
24	open.uct.ac.za Internet Source	<1%
25	"2020 7th International Conference on Smart Structures and Systems (ICSSS) - Full Conference Proceedings", 2020 7th International Conference on Smart Structures and Systems (ICSSS), 2020 Publication	<1%
26	www.digitalocean.com Internet Source	<1%
27	"Advances in Distributed Computing and Machine Learning", Springer Science and Business Media LLC, 2021 Publication	<1%

28	Submitted to University of Essex Student Paper	<1%
29	Vladimir L. Uskov, Jeffrey P. Bakken, Kaustubh Gayke, Juveriya Fatima, Brandon Galloway, Keerthi Sree Ganapathi, Divya Jose. "Chapter 1 Smart Learning Analytics: Student Academic Performance Data Representation, Processing and Prediction", Springer Science and Business Media LLC, 2020 Publication	<1%
30	www.coursehero.com Internet Source	<1%
31	www.kaggle.com Internet Source	<1%
32	"ICCCE 2020", Springer Science and Business Media LLC, 2021 Publication	<1%
33	"International Conference on Innovative Computing and Communications", Springer Science and Business Media LLC, 2021 Publication	<1%
34	LAURA I. BURKE, ROBERT H. STORER, LAURA L. LANSING, SETH W. FLANDERS. "A neural-network approach to prediction of vehicle driving comfort", IIE Transactions, 1996 Publication	<1%

35	doku.pub Internet Source	<1 %
36	ijritcc.org Internet Source	<1 %
37	vijay-choubey.medium.com Internet Source	<1 %
38	www.ijeat.org Internet Source	<1 %
39	www.ijrte.org Internet Source	<1 %
40	www.irphouse.com Internet Source	<1 %
41	"Emerging Research in Electronics, Computer Science and Technology", Springer Science and Business Media LLC, 2019 Publication	<1 %
42	Sinkon Nayak, Mahendra Kumar Gourisaria, Manjusha Pandey, Siddharth Swarup Rautaray. "Prediction of Heart Disease by Mining Frequent Items and Classification Techniques", 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 2019 Publication	<1 %

Exclude quotes Off
Exclude bibliography Off

Exclude matches Off