

**“ STUDENT GRADE PREDICTION USING MACHINE LEARNING”**

**Project report submitted in partial fulfilment of the**

**Requirements for the award of the Degree of**

**Bachelor of Technology**

**in**

**CSE (Data Science)**

**By**

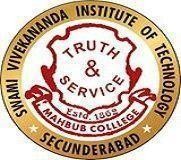
|  |  |
| --- | --- |
| **P. RAMYA** | **(20P71A6727)** |
| **V. RISHIKESH** | **(20P71A6728)** |
| **S. SOLOMON FELIZ** | **(20P71A6739)** |
| **P. VIKRAM** | **(20P71A6746)** |

**Under the Esteemed Guidance of**

**Mrs. M SUPRIYA SAMUEL, Associate Professor and HOD.**

**Department of Computer Science and Engineering**

**(Data Science)**



**SWAMI VIVEKANANDA INSTITUTE OF TECHNOLOGY**

**Mahbub College Campus, R.P Road, Secunderabad-500003**

**(Affiliated to JNTUH)**

**2020-2024**



**DEPARTMENT OF CSE (DATA SCIENCE)**

****

**CERTIFICATE**

This is to certify that the project report entitled **“STUDENT GRADE PREDICTION USING MACHINE LEARNING”** is being submitted by Name of P. Ramya(20P71A6726), V. Rishikesh(20P71A6728), S. Solomon Feliz(20P71A6739) and P. Vikram(20P71A6746) in partial fulfilment for the award of Degree of BACHELOR OF TECHNOLOGY in CSE (DATA SCIENCE), to the Jawaharlal Nehru Technological University-Hyderabad is a record of bonafied work carried out by him/her under my guidance and supervision.

Date:

Internal Guide

HOD – CSE (DS)

External Examiner Principal

**ACKNOWLEDGEMENT**

First, we thank our Project Guide Associate Professor and HOD Mrs. M Supriya Samuel of the Department of CSE (Data Science) for giving us an opportunity for developing this project.

This project has really helped us in enhancing our skills of programming, the perspective with which we should view projects and above all, our presentation, and interpersonal skills.

Last but not the least; we also thank the professors of our department for their help and wishes for successful completion of project.

P. RAMYA

20P71A6726

V. RISHIKESH

20P71A6728

S. SOLOMON FELIZ

20P71A6739

P. VIKRAM

20P71A6746

**DECLARATION**

We hereby declare that the work which is being presented in this Mini Project entitled, **“STUDENT GRADE PREDICTION USING MACHINE LEARNING”** submitted to **JNTU-H,** in the partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in **CSE (DATA SCIENCE),** is an authentic record of our own work carried out from August 2023 to December 2023 under the supervision of **Mrs. M Supriya Samuel, Associate Professor and HOD, Department of CSE (Data Science), SVIT, Mahbub Campus.**

*The matter embodied in this project report has not been submitted by me for the award of any other degree.*

Place: SECUNDERABAD

Date:

P. RAMYA

20P71A6727

V. RISHIKESH

20P71A6728

S. SOLOMON FELIZ

20P71A6739

P. VIKRAM

20P71A6746

**INDEX**

**Abstract vi**

**List of Figures vii**

**List of Tables viii**

1. Introduction 1

2. System Requirements and Analysis 2

2.1 Problem Definition 2

2.2 Problem Analysis 2

2.3 Feasibility Study 3

2.4 Software/Hardware Requirements 4

3. System Design 5

3.1 Module Description 5

3.2 UML Diagrams 6

4. Implementation Details 11

5. Code Templates 21

6. Testing 31

7. Output Screens 36

8. Conclusion 44

Bibliography/References 45

**ABSTRACT**

This system is designed to predict the final grade of the students based on the grades scored by him/her during his/her previous course and years. In order to predict the grade of the student it needs some data to be analysed and hence grade is predicted. Input is student information such as their name and roll number, and their previous academic information is retrieved from corresponding dataset using which student grade is predicted. Here system will generate a predicted output where he/she will get grade prediction using Linear Regression algorithm. The Linear Regression model is trained using the data which is available in the Dataset. This technique anticipates grade which can be utilized for confidence building, degree planning for students, tailored advice. This system can be used in schools, colleges and other educational institutes.

This project intention is to help students know what would be their CGPA after completing four years of their B-TECH without writing exams, this is only possible through Machine Learning. This project allows students who have completed either of first, second or third year to predict their Cumulative Grade Point Average (CGPA). This model can be implemented according to the requirements of educational institutions.

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Name of Figure** | **Page No.** |
| 3.2.1 | Class Diagram | 7 |
| 3.2.2 | Sequence Diagram | 8 |
| 3.2.3 | Use Case Diagram | 9 |
| 3.2.4 | Activity Diagram | 10 |
| 6.2 | Black box testing | 33 |
| 6.4 | White box testing | 34 |
| 7.1 | Insights of Dataset | 36 |
| 7.2 | Correlations in the Dataset | 37 |
| 7.3 | Predicted CGPA vs Actual CGPA | 38 |
| 7.4 | Predicted CGPA vs Actual CGPA in Model-1 | 39 |
| 7.5 | Predicted CGPA vs Actual CGPA in Model-2 | 40 |
| 7.6 | Predicted CGPA vs Actual CGPA in Model-3 | 41 |
| 7.7 | Predicted CGPA according to Model-1 | 42 |
| 7.8 | Predicted CGPA according to Model-2 | 42 |
| 7.9 | Predicted CGPA according to Model-3 | 43 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Name of Figure** | **Page No.** |
| 3.2.1 | Class Diagram | 7 |

**1. INTRODUCTION**

Prediction is a technique that has become prevalent in several fields and sectors including education, medicine, biology, politics, and finance. This prevalence is strongly attributed to recent advances in machine learning techniques.

Although various predictions systems adopt various approaches, they all follow the same notion – make an educated guess about the value of a parameter by observing what variables affect the parameter and how they have affected it in the past. The amount of data needed to make a good prediction depends on how complex the parameter being predicted is. In this project the parameter that is being predicted is CGPA (Cumulative Grade Point Average) and the variables that affect this parameter are grades obtained by the student in his/her previous year courses.

The possible grades that can be obtained by a student are O, A+, A, B+,B, C. The letter grades are assigned numerical value as 10,9,8,7,6,5. Using the Dataset which contains specific grades obtained by the student in a particular course as input to our system we are able to predict the student CGPA.

In this project we employ three models to predict the final CGPA. Model 1: predict final CGPA based on Grades of first year only. Model: 2 predict final CGPA based on grades of first two years. Model 3: predict final CGPA based on grades of first three years.

This prediction can be useful for educational institutions, advisors, and students themselves to assess academic progress and make informed decisions. By analysing historical data and applying statistical models, CGPA prediction aims to provide an accurate estimation of a student's future academic performance.

**2. SYSTEM REQUIREMENT AND ANALYSIS**

**2.1 Problem Definition**

Student Management System is an existing system, that doesn’t provide any prediction capabilities of student grade. It only provides various information about the student such as student identity, gender, age, attendance, grades obtained in various courses etc. This existing system doesn’t employ any machine learning algorithms, it is a simple software used in many educational institutions to keep record of student details.

**2.1.1 Disadvantages of Existing Systems:**

* It doesn’t have a option to predict students final CGPA.
* It cannot discriminate among levels of student performance.

**2.2 Problem Analysis**

The proposed system can overcome all the limitations of the existing system, the main idea is to make use of Machine Learning algorithms to make accurate predictions. Firstly we will be using a Dataset to train the model. The Dataset contains information regarding the performance of the students in previous semesters. The Machine Learning Algorithms that are employed in the proposed system is Linear Regression.

**2.2.1 Advantages if Proposed System:**

* Student Final CGPA i.e the Grade is predicted.
* Accurate and Efficient
* User friendly.

**2.3 Feasibility Study**

Feasibility study is a comprehensive evaluation of a proposed project that assesses its practicality, viability, and likelihood of success. Feasibility analysis has various purposes such as identifying potential issues and risks, make informed decisions, optimize resource allocation and enhance project success rate.

There are four considerations involved in feasibility study

1. Economic Feasibility

2. Technical Feasibility

3. Social Feasibility

4. Operational Feasibility

**1. Economic Feasibility:**

This evaluates the financial viability of the system. It involves analysing the costs associated with system development, implementation, maintenance, and potential benefits or cost savings it can generate. The hardware for this project is readily available the cost associated with it is relatively low, and software which is used in developing this project are all open source and free to use. Hence, this project was developed using the least cost. Hence the project is economically feasible.

**2. Technical Feasibility:**

This assesses whether the necessary technology infrastructure, hardware, software, and technical expertise are available or can be acquired to implement and support the system effectively. Every system that is created must not put a heavy burden on the technological resources that are readily accessible. As a result, the technological resources will be put under a lot of stress. Because of this, the client will face high expectations. As there are no or very few changes need to deploy the established system, it must have a low requirement.

**3. Social Feasibility:**

Checking the user’s level oof system acceptability is one of the study’s aspects. This covers the procedure for instructing the use on how to operate the system effectively. The user must accept the system as a requirement rather than perceive it as a danger. The techniques used to inform and acquaint users with the system are the sole determinants of how much acceptance it receives from them. As he is the system’s final user, his degree of confidence must be increased so that he can offer some helpful feedback as well.

**4. Operational Feasibility:**

Operational feasibility is a measure of how well a proposed system solves problems and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development project fits in with the existing business environment and objectives about the development schedule, delivery date, corporate culture, and existing business processes. This system operationally eliminates all the tensions of the Admin and helps in effectively tracking the project progress. This kind of automation will surely reduce the time and energy, which previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

**2.4 System Requirements**

**Hardware Requirements:**

Processor : Core i3 processor

RAM : 4GB

Hard Disk : 240GB

**Software Requirements:**

Operating System : Windows, Linux

Programming Language : Python

Libraries : NumPy, Pandas, Seaborn

**3. SYSTEM DESIGN**

**3.1 Module Description:**

**3.1.1 Admin:**

Admin has full access to all the modules of this system. They can add, modify and delete the data in this system and can update and upgrade the system.

Features of Admin module

* Load the Dataset.
* Build and Train the model.
* Pass necessary commands and view results.

**3.1.2 System:**

System is the main working module, it is the soul of the project. It handles various computations and operations to give us the expected output. It is responsible for performing all operations which are being requested by the admin.

Features of System module

* Preprocessing Data
  + - Data Exploration.
    - Cleaning the data.
    - Discover and visualize the data.
    - Look for correlations.
* Perform Linear Regression
* Predict CGPA.

**3.1.3 User:**

User have restricted access i.e., they have access to some modules only.

Feature of User

* Enter Name and Roll\_No.
* View Predicted CGPA.

**3.2 UML Diagrams**

UML stands for Unified Modelling Language. It is a general-purpose modelling language with standards, UML is used in the area of object-oriented software engineering. The objective of UML is to establish itself as a standard language for modelling object-oriented computer programs. The Unified Modelling Language is a standard language used for business modelling, non-software systems, and defining, visualizing, building, and documenting the artefacts of software systems. The UML is a crucial component of the software development process and the creation of objects-oriented software. The UML primarily employs pictorial notations to convey software project design. The following are the UML’s primary objectives:

1. Visualizing and communicating system structure.
2. Designing system architecture.
3. Facilitating software development and maintenance.
4. Modelling system behaviour.
5. Capturing system requirements.

**3.2.1 Class Diagram:** A class diagram is a type of static structure diagram used in software engineering that displays the classes, attributes, operations (or methods), and relationships between the classes to explain the structure of a system. Class diagrams provide a visual representation of how classes interact and collaborate to achieve system functionalities. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of software application.

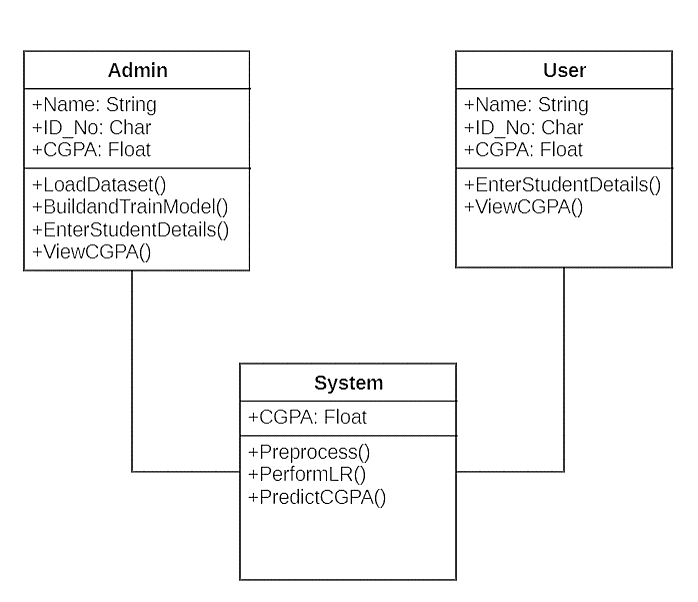


Fig 3.1 Class Diagram for Student Grade Prediction

**3.3.3 Use Case Diagram:** A use case diagram is a type of behavioural diagram that is specified by and produced from a use-case analysis in the Unified Modelling Language (UML). Its aim is to provide a graphical summary of the functionality of a system offered in terms of actors, their objectives (represented as use cases), and any independencies between those use cases.

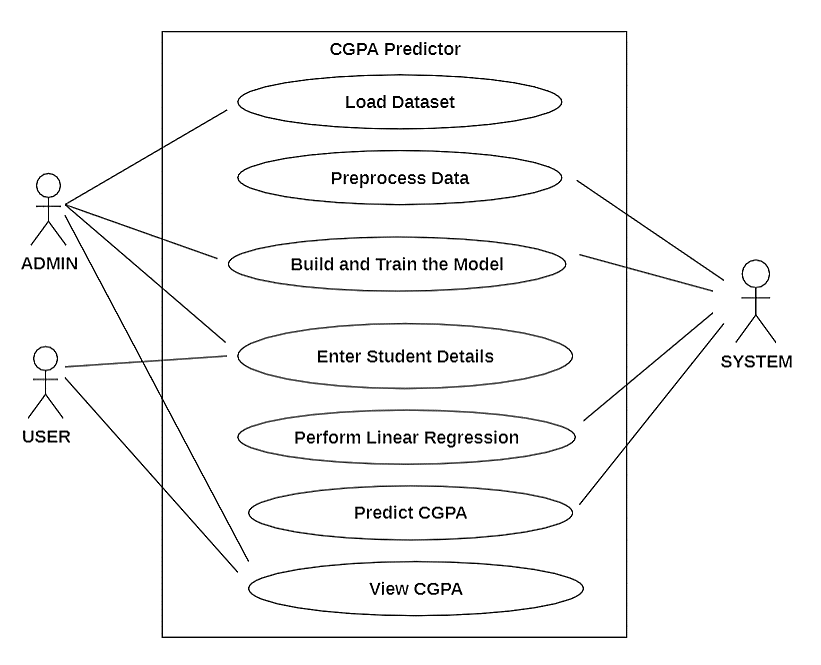


Fig 3.3 Use Case Diagram for Student Grade Prediction

**3.2.2 Sequence Diagram:** An interaction diagram that depicts how processes engage with one another and in what order is known as a sequence diagram in the UML. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. We can also use the terms event diagrams how and in what order the objects in a system function.

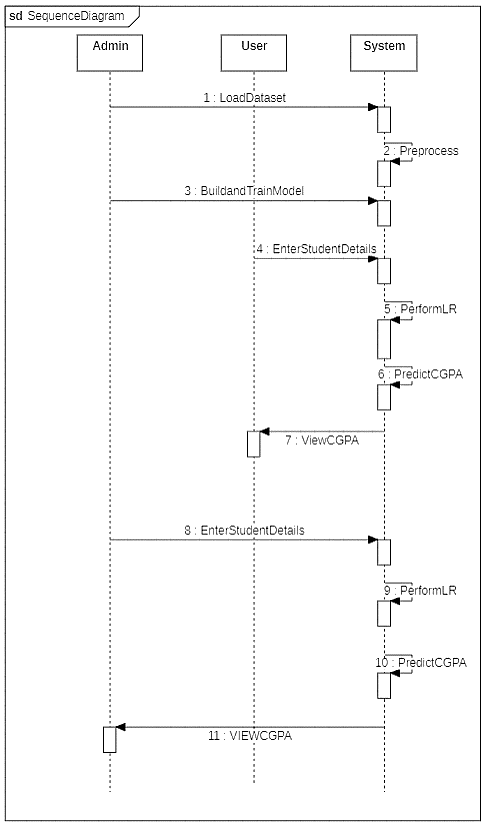


Fig 3.2 Sequence Diagram for Student Grade Prediction

**3.3.4 Activity Diagram:** Activity diagrams are graphical representations of workflows of stepwise activities and action 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.

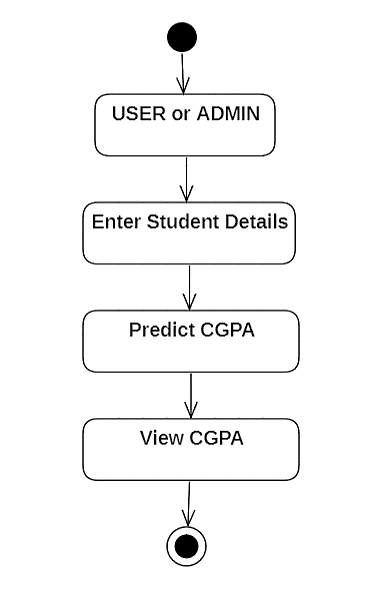


Fig 3.4 Activity Diagram for Student Grade Prediction

**4. IMPLEMENTATION DETAILS**

**4.1 Functional Requirements**

In Software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behaviour, and outputs. Functional requirements may be calculations, technical details, data manipulation, processing, and other specific functionality that define what a system is supposed to accomplish. Behavioural requirements describing all the cases where the system uses the functional requirements. Generally, functional requirements are expressed in the form “**system shall do <requirement>”.** In requirements engineering, functional requirements specify particular results of a system. Functional requirements drive the application architecture of a system. A requirement analyst generates use cases after gathering and validating a set of functional requirements.

**4.2 Non-Functional Requirements**

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours

The non-functional requirements are

* **Availability**

A system’s availability is the amount of time that is operational and available for use.

* **Efficiency**

Specifies how well the software utilizes scarce resources: CPU cycles, disk space, memory, bandwidth etc.

* **Flexibility**

If the organization intends to increase or extend the functionality of the software after it is deployed, that should be planned from the beginning; it influences choices made during the design, development, testing and deployment of the system. New modules can be easily integrated to our system without disturbing the existing modules or modifying the logical database schema of the existing applications.

* **Portability**

Portability specifies the case with which the software can be installed on all necessary platforms, and the platforms on which it is expected to tun. This allows the application to be easily operated on any operating system.

* **Scalability**

Software that is scalable has the ability to handle a wide variety of system configuration sizes. The non-functional requirements should specify the ways in which the system may be expected to scale up (by increasing hardware capacity, adding machines etc.). Our system can be easily expandable. Any additional requirements such as hardware or software which increase the performance of the system can be easily added.

* **Integrity**

Integrity requirements define the security attributes of the system, restricting access to features or data to certain users and protecting the privacy of data entered into the software.

* **Usability**

Ease-of-use requirements address the factors that constitute the capacity of the doftware to be understood, learned, and widely used by its intended users.

* **Performance**

The performance constraints specify the timing characteristics of the software.

**4.3 Software Used:**

**4.3.1 PYTHON**

Python is an open source, high-level programming language developed by Guido van Rossum in the late 1980s and presently administered by Python Software Foundation. It came from the ABC language that he helped create early on in his career. Python is a powerful language that we can use to create games, write GUIs, and develop web applications. It is a high-level language. Reading and writing codes in Python is much like reading and writing regular English statements, because they are not written in machine-readable language. Python programs need to be processed before machines can run them. Python is an interpreted language this means that every time a program runs its interpreter runs through the code and translates into machine readable code.

Python is an object-oriented language that allows users to manage and control data structures or objects to create and run programs. Everything n Python is, in fact, first class. All objects, data types, functions, methods, and classes take equal position in Python. Some important features of Python are as follows:

* **Easy to learn and read**

Python has a clean and readable syntax, making it beginner-friendly and easy to understand. Its simplicity allows developers to write code more quickly and efficiently.

* **Large standard library**

Python comes with a comprehensive standard library that provides a wide range of modules and functions for various tasks. This extensive library reduces the need for developers to write code from scratch, saving time and effort.

* **Cross-platform compatibility**

Python is a cross-platform language, meaning it can run on various operating systems such as Windows, macOS, and Linux. This makes it highly versatile and accessible for developers.

* **Third-part libraries and frameworks**

Python has a vast ecosystem of third-party libraries and frameworks that extend its capabilities. These libraries, such as NumPy, Pandas, and TensorFlow, enable developers to easily handle complex tasks like data analysis, machine learning, web development, and more.

* **Object-oriented programming (OOP) support**

Python supports object-oriented programming, allowing developers to create reusable and modular code. OOP concepts like classes, objects, and inheritance enhance code organization and maintainability.

* **Interpreted language**

Python is an interpreted language, which means it does not require compilation before execution. Developers can write and run code directly, making it convenient for prototyping and testing.

**Steps to install python**

1. Go to the official Python download page (<https://www.python.org/downloads/>) and download the latest stable release of Python for Windows.
2. Double-click the downloaded Python installer file.
3. Click the "Install Now" button.
4. Make sure the "Add Python 3.11 to PATH" checkbox is selected.
5. Click the "Install Now" button again.
6. The Python installation process will begin. Once the installation is complete, click the "Close" button.

**4.3.2 Python Libraries Used**

1. **NumPy**

NumPy, which stands for Numerical Python, is a fundamental library for scientific computing with Python. It provides a powerful and efficient way to work with multidimensional arrays, matrices, and linear algebra operations. NumPy is widely used in various fields, including data science, machine learning, and scientific research. Key features of NumPy are:

* N-dimensional arrays - NumPy's core data structure is the ndarray, which represents a homogeneous n-dimensional array. This means that all elements in the array have the same data type, and the array can have any number of dimensions.

12

* Efficient Array Operations - NumPy provides a comprehensive set of functions for operating on arrays. These functions are highly optimized and take advantage of hardware acceleration to achieve significant performance gains compared to built-in Python lists.
* Linear Algebra Operations - NumPy includes a rich set of functions for linear algebra operations, such as matrix multiplication, matrix inversion, and eigenvalue decomposition. These functions are essential for various scientific and engineering applications.
* Broadcasting - NumPy's broadcasting mechanism allows for efficient element-wise operations between arrays of different shapes. This feature simplifies complex calculations and reduces the need for explicit loops.
* Matplotlib Integration - NumPy works seamlessly with Matplotlib, a popular Python library for data visualization. NumPy arrays can be directly passed to Matplotlib plotting functions to create charts and graphs.

**Steps to install Numpy**

1. Once Python is installed, you can install NumPy using pip, the package manager for Python. Open a terminal window and type the following command:

Bash

pip3 install numpy

1. This will download and install the NumPy library.

**2. Pandas**

Pandas is a powerful Python library for data manipulation and analysis. It provides high-performance data structures and tools for working with tabular data, making it an essential tool for data scientists, analysts, and anyone who works with data in Python. Key features of Pandas are:

* DataFrames - Pandas' primary data structure is the DataFrame, a two-dimensional, size-mutable, labelled tabular data structure with rows as observations and columns as variables. DataFrames allow for efficient data organization, storage, and manipulation.
* Data Manipulation - Pandas offers a comprehensive set of functions for data cleaning, transformation, and manipulation. These functions include filtering, selecting, sorting, merging, and aggregating data.
* Data Analysis - Pandas provides built-in statistical functions and tools for data analysis. These tools allow for descriptive statistics, hypothesis testing, and time series analysis.

13

* Data Visualization - Pandas integrates seamlessly with Matplotlib and Plotly, popular Python libraries for data visualization. We can easily create charts, graphs, and plots to visualize your data.
* Flexibility and Extensibility - Pandas is highly flexible and extensible. It allows you to create custom functions and apply them to your data.

**Steps to install Pandas**

1. Open a terminal or command prompt window and enter the following command:

Bash

pip3 install pandas

1. This will download and install the Pandas library along with its dependencies.

**3. Seaborn**

Seaborn is a Python library for making statistical data visualizations. It is built on top of Matplotlib and provides a higher-level interface that focuses on statistical plotting. Seaborn makes it easy to create beautiful and informative plots that can help you explore and understand your data. Key features of Seaborn are:

* High – level interface: Seaborn provides a high-level interface that makes it easy to create complex plots with just a few lines of code.
* Statistical plots – Seaborn specializes in statistical plots, such as histograms, scatter plots, and box plots.
* Easy to customize – Seaborn is also easy to customiza, so we can create plots that match our specific needs.

**How to install Seaborn**

1. To install Seaborn using pip, open a terminal window and type the following command

Bash

pip install seaborn

1. This will download and install Seaborn along with its dependencies.

**4.3.3 Jupyter Notebook**

The Jupyter Notebook is an open source web application that we can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter

Jupyter Notebooks are a spin-off from the IPython project, which used to have an IPython Notebook project itself. The name Jupyter comes from the core supported programming languages that it supports: Juila, Python, and R. Pupyter ships with the IPython kernel, which allows us to write programs in Python, but there are currently over 100 other kernals that we can use. Some important features of Jupyter Notebook are as follows:

* **Interactive Computing**

Jupyter Notebook provides an interactive computing environment where users can write and execute code in real-time. It supports multiple programming languages, including Python, R, Julia, and more, making it versatile for various data analysis and scientific computing tasks.

* **Code execution and visualization**

Users can write and execute code cells within the notebook, making it easy to experiment, test, and iterate on code. The output of code cells, such as text, tables, plots, and interactive visualizations, can be displayed directly within the notebook, enhancing the ability to analyze and communicate results.

* **Markdown support**

Jupyter Notebook supports Markdown, a lightweight markup language, allowing users to create richly formatted text cells. Markdown cells can include headings, lists, links, images, mathematical equations, and more, enabling the creation of well-documented and visually appealing notebooks.

* **Collaboration and sharing**

Jupyter Notebook facilitates collaboration by allowing users to share their notebooks with others. Notebooks can be shared as static files or through platforms like JupyterHub or JupyterLab. This feature promotes reproducibility and enables teams to work together on data analysis projects.

**Notebook extensions and widgets**

Jupyter Notebook offers a range of extensions and widgets that enhance its functionality. Extensions provide additional features like code linting, code folding, and table of contents, while widgets enable interactive elements like sliders, dropdowns, and buttons, enhancing the interactivity and user experience of the notebook.

**Steps to install Jupyter Notebook**

1. To install Jupyter Notebook using pip enter the following commands in a terminal

Bash

pip install notebook

1. This will download and install Jupyter Notebook.

**4.3.4 Microsoft Visual Studio Code**

Microsoft Visual Studio Code, also commonly referred to as VS Code, is a free and open-source code editor developed by Microsoft for Windows, Linux, and macOS. It is a popular choice for developers due to its extensibility, support for a wide range of programming languages, and lightweight performance. Key features of Microsoft Visual Studio code are:

15

* Extensibility - VS code is highly extensible, with a vast ecosystem of extension that add new features and functionalities.
* Support for Multiple Programming Languages - VS Code supports a wide range of programming languages such as JavaScript, Python, C++, Java etc.
* Lightweight Performance – VS code is a lightweight and fast code editor making it suitable for resource – constrained machines. It starts up quickly and runs smoothly, even with large projects.
* Integrated Debugging – VS code has a built-in debugger that allows you to step through code, inspect variables, and set breakpoints. It supports various debugging protocols, including python.
* Git Integration – VS code has built-in Git integration, allowing you to manage our Git repositories directly from the editor.

**Steps to install Microsoft Visual Studio Code**

1. Visit the official Visual Studio Code download page:
2. Choose the appropriate download option for your operating system. For Windows, you can download the 32-bit or 64-bit version.
3. Once the download is complete, locate the downloaded file. For Windows, it will be a .exe file.
4. Double-click the downloaded file to start the installation process.
5. The installation wizard will appear. Follow the instructions on the screen.
6. Accept the license agreement.
7. Choose the installation location. You can leave it as the default or select a different location.
8. Select the components you want to install. You can install all of them or choose specific ones.
9. Click the "Install" button.
10. The installation process will begin. Once it's complete, you can click the "Launch" button to start Visual Studio Code.

**4.3.5 Jupyter Lab**

JupyterLab is a web-based interactive computing environment that provides a rich user interface for notebooks, code, and data. It is a next-generation notebook environment that extends the classic Jupyter Notebook with a variety of new features, including:

* A modular and extensible architecture that allows users to customize their JupyterLab experience with a wide range of extensions.
* A more powerful and flexible notebook editor with support for Markdown, LaTeX, and other popular markup languages.
* A variety of new widgets and extensions for data visualization, scientific computing, and machine learning.
* A more integrated development environment (IDE) with support for debugging, code completion, and code refactoring.

JupyterLab is a powerful tool for data scientists, machine learning engineers, and other scientists and engineers who need to interact with data and code in a rich and interactive environment. It is also a popular tool for teaching and education, as it provides a user-friendly interface for students to learn about programming and data science.Here are some of the key features of JupyterLab:

* **Modular and extensible architecture:**

JupyterLab is built on a modular and extensible architecture that allows users to customize their JupyterLab experience with a wide range of extensions. There are over 2,000 extensions available for JupyterLab, which provide a variety of new features and functionality.

* **Powerful and flexible notebook editor:**

The JupyterLab notebook editor is more powerful and flexible than the classic Jupyter Notebook editor. It supports Markdown, LaTeX, and other popular markup languages, and it has a variety of features for code completion, code refactoring, and debugging.

* **Variety of widgets and extensions:**

JupyterLab has a variety of widgets and extensions for data visualization, scientific computing, and machine learning. These widgets and extensions make it easy to visualize data, perform complex calculations, and build machine learning models.

* **Integrated development environment (IDE):**

JupyterLab is a more integrated development environment (IDE) than the classic Jupyter Notebook. It supports debugging, code completion, and code refactoring, which makes it easier to develop and debug code.

**4.3.6 HTML and CSS**

HTML (HyperText Markup Language) and CSS (Cascading Style Sheets) are the two essential building blocks of web pages. HTML provides the structure and content of a web page, while CSS styles the appearance and layout of the page. HTML is a markup language that tells browsers how to display a web page. It uses tags to define different elements of a web page, such as headings, paragraphs, images, and links. HTML tags are enclosed in angle brackets (< >). CSS is a style sheet language that tells browsers how to style the elements of a web page. It uses selectors to identify elements and properties to specify their style. CSS properties are defined in curly braces ({ }). The details of HTML are as follows

* Basic Structure:
* HTML documents have a basic structure with a head and a body.
* The <head> section contains metadata like title, charset, etc.
* The <body> section contains the content of the web page.
* HTML Elements:
* Elements are defined by tags (e.g., <p> for paragraphs).
* Tags can have attributes (e.g., <a href="https://example.com">Link</a>).
* Text Formatting:
* Headings: <h1> to <h6> for different levels of headings.
* Paragraphs: <p> tag.
* Bold: <strong> or <b>.
* Italics: <em> or <i>.
* Line Break: <br>.
* Lists:
* Ordered List: <ol> with <li> items.
* Unordered List: <ul> with <li> items.
* Links and Images:
* Hyperlinks: <a> with the href attribute.
* Images: <img> with the src attribute.
* Forms:
* Input fields: <input> for text, password, etc.
* Buttons: <button> or <input type="button">.
* Form: <form> to create a form, with various input elements.
* Tables:
* Table: <table> creates a table.
* Rows: <tr> defines a row.
* Data: <td> defines a data cell.
* Headers: <th> defines a header cell.
* Semantics:
* Semantic tags like <header>, <nav>, <article>, <section>, <footer> improve document structure.
* Comments:
* <!-- This is a comment --> for adding comments in HTML.

The details of CSS are as follows

**5. CODE TEMPLATES**

import numpy as np

import pandas as pd

from flask import Flask, render\_template, request

import csv

app = Flask(\_\_name\_\_)

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

csv\_file\_path = r"C:\Users\rishi\OneDrive\Desktop\MINI PROJECT\CGPA Predictor\Student details.csv"

def encrypt\_grade(grade):

encryption\_mapping = {'O': 10, 'A+': 9, 'A': 8, 'B+': 7, 'B': 6, 'C': 5}

return encryption\_mapping.get(grade, 0)

file\_path = r"C:\Users\rishi\OneDrive\Desktop\MYPROJECT\MYDATASET.csv"

df = pd.read\_csv(file\_path)

input = df.drop(["4Predictive Analytics","4Web and Social Media Analytics","4Professional Elective - 4","4Professional Elective - 5","4Open Elective - 2","4Seminar","4Web and Social Media Analytics Lab","4Mini Project ","4Project Stage - 1","4Organizational Behaviour","4Professional Elective - 6","4Open Elective - 3","4Project Stage - 2","No","CGPA"], axis=1)

target = df[["CGPA"]]

grades\_enc = {'O':10, 'A+':9, 'A':8, 'B+':7, 'B':6, 'C':5}

type(input)

for item in input:

input[item] = input[item].replace(grades\_enc)

df = pd.concat([input, target], axis=1)

model1\_columns = []

model2\_columns = []

model3\_columns = []

for item in [input]:

for i in item:

if i[0]=='1':

model1\_columns.append(i)

elif i[0]=='2':

model2\_columns.append(i)

else:

model3\_columns.append(i)

model2\_columns = model1\_columns + model2\_columns

model3\_columns = model2\_columns + model3\_columns

labels = df['CGPA'].values

features = df[list(model1\_columns)].values

from sklearn.model\_selection import train\_test\_split

random\_seed=42

X\_train, X\_test, y\_train, y\_test = train\_test\_split(features, labels, test\_size=0.30,random\_state=random\_seed)

from sklearn import linear\_model

lr\_1 = linear\_model.LinearRegression()

lr\_1.fit(X\_train, y\_train)

y\_train\_predict = lr\_1.predict(X\_train)

y\_test\_predict = lr\_1.predict(X\_test)

labels = df['CGPA'].values

features = df[list(model2\_columns)].values

from sklearn.model\_selection import train\_test\_split

random\_seed=42

X\_train, X\_test, y\_train, y\_test = train\_test\_split(features, labels, test\_size=0.30,random\_state=random\_seed)

from sklearn import linear\_model

Bulid a new Model

lr\_2 = linear\_model.LinearRegression()

Train the Model

lr\_2.fit(X\_train, y\_train)

y\_train\_predict = lr\_2.predict(X\_train)

y\_test\_predict = lr\_2.predict(X\_test)

labels = df['CGPA'].values

features = df[list(model3\_columns)].values

from sklearn.model\_selection import train\_test\_split

random\_seed=42

X\_train, X\_test, y\_train, y\_test = train\_test\_split(features, labels, test\_size=0.30,random\_state=random\_seed)

from sklearn import linear\_model

lr\_3 = linear\_model.LinearRegression()

lr\_3.fit(X\_train, y\_train)

y\_train\_predict = lr\_3.predict(X\_train)

y\_test\_predict = lr\_3.predict(X\_test)

@app.route('/', methods=['GET', 'POST'])

def home():

if request.method == 'POST':

name = request.form['name']

roll\_number = request.form['roll\_number']

model = request.form['model'] # This is a string, not an int

if model in ['1', '2', '3']: # Use strings in the list for comparison

# Find the row corresponding to the entered name and roll number

selected\_row = None

with open(csv\_file\_path, 'r') as csvfile:

reader = csv.DictReader(csvfile)

for row in reader:

if row["Name"] == name and row["Roll Number"] == roll\_number:

selected\_row = row

break

if selected\_row is not None:

if model == '1':

columns = model1\_columns

lr\_model = lr\_1

elif model == '2':

columns = model2\_columns

lr\_model = lr\_2

else:

columns = model3\_columns

lr\_model = lr\_3

print(f"\n{name}'s Grades for Model {model}:")

for column in columns:

grade = selected\_row.get(column, "N/A")

print(f"{column}: {grade}")

grades = [selected\_row.get(column, 0) for column in columns]

encrypted\_grades = [encrypt\_grade(grade) for grade in grades]

input\_lr = [float(grade) for grade in encrypted\_grades]

prediction = lr\_model.predict([input\_lr])

return render\_template('result.html', name=name, prediction=prediction[0])

else:

return render\_template('result.html', name="Details not found", prediction="N/A")

else:

return render\_template('result.html', name="Invalid model", prediction="N/A")

return render\_template('index.html')

@app.route('/predict', methods=['POST'])

def predict():

csv\_file\_path = r"C:\Users\rishi\OneDrive\Desktop\MINI PROJECT\CGPA Predictor\Student details.csv"

if request.method == 'POST':

name = request.form['name']

roll\_number = request.form['roll\_number']

model = request.form['model']

if model in ['1', '2', '3']

selected\_row = None

with open(csv\_file\_path, 'r') as csvfile:

reader = csv.DictReader(csvfile)

for row in reader:

if row["Name"] == name and row["Roll Number"] == roll\_number:

selected\_row = row

break

if selected\_row is not None:

if model == '1':

columns = model1\_columns

lr\_model = lr\_1

elif model == '2':

columns = model2\_columns

lr\_model = lr\_2

else:

columns = model3\_columns

lr\_model = lr\_3

grades = [selected\_row.get(column, 0) for column in columns]

encrypted\_grades = [encrypt\_grade(grade) for grade in grades]

input\_lr = [float(grade) for grade in encrypted\_grades]

prediction = lr\_model.predict([input\_lr])

return render\_template('result.html', name=name, prediction=prediction[0])

else:

return render\_template('result.html', name="Details not found", prediction="N/A")

else:

return render\_template('result.html', name="Invalid model", prediction="N/A")

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**6. TESTING**

**SYSTEM 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.

**6.1 Types of testing**

**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.

**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.

The following are the types of Integration Testing:

1. Top-down Integrating Testing: This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through

the control hierarchy, beginning with the main program module. The module subordinates to the main program module which is incorporated into the structure in wither depth first or breadth first manner.

2. Bottum-up Integration Testing: This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to give level is always available and the needs for stubs is eliminated.

**Functional Testing:** Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, systemdocumentation, and user manuals.Functional testing is dependent 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.

**6.2 Black Box Testing**

**What is Black Box Testing?**

Black box testing is a software testing technique in which functionality of the software under test (SUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on the software requirements and specifications.

In Black Box Testing we just focuses on inputs and outputs of the software system without bothering about internal knowledge of the software program.



Fig 6.2 Black box testing

The above Black Box can be any software system you want to test. For example; an operating system like Windows, a website like Google, a database like Oracle or even our own custom application. Under Black Box testing, we can test these applications by just focusing on the inputs and outputs without knowing their internal code implementations.

**Black box testing – Steps**

Here are the generic steps followed to carry out any type of Black Box Testing.

* Initially requirements and specifications of the system are examined.
* Tester chooses valid inputs (positive test scenario) to check whether SUT processes

them correctly. Also, some invalid inputs (negative test scenario) are chosen to verify that SUT is able to detect them.

* Tester determines expected outputs for all those inputs.
* Software tester constructs test cases with the selected inputs.
* The test cases are executed.
* Software tester compares the actual outputs with the expected outputs.
* Defects if any are fixed and re-tested.

**Types of Black Box Testing**

There are many types of Black Box Testing but following are the prominent ones

* **Functional testing –** This black box testing type is related to functional requirements of a system, it is done by software testers.
* **Non-functional testing** – This type of black box testing is not related to testing of a specific functionality, but non-functional requirements such as performance, scalability, usability.
* **Regression testing** – Regression testing is done after codes fixes, upgrades pr maintenance to check whether the new code has not affected the existing code.

**6.3 White Box Testing**

White Box testing of software solution’s internal ending and infrastructure. It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability. White box testing is also known as clear, open, structural, and glass box testing.

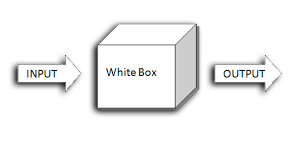


Figure 6.3 Black Box Testing

It is one of parts of “box testing” approach of software testing. Its counterpart black box testing from an external or end-user type perspective. On the other hand, White box testing is based on the inner workings of an application and revolves around internal testing. The term “white box” was used because of the see-through box concept. The clear box or white box names symbolizes the ability to see through the software’s outer shell (or “box”) into its inner workings. Likewise, the “black box” in “black box testing” symbolizes not being able to see the inner workings of the software so that the only end- user experience can be tested.

* **What do we verify in White Box Testing?**

White box testing involves the testing of the software code for the following: Internal security holes.

* Broken or poorly structured paths in the coding processes
* The flow of specific inputs through the code
* Expected output
* The functionality of conditional loops
* Testing of each statement object and function on an individual basis.

The testing can be done at system, integration and unit levels of software development. One of the basic goals of white box testing is to verify a working flow for an application. It involves testing a series of predefined inputs against expected or desired outputs so that when a specific input does not result in the expected output, you have encountered a bug.

**6.4 Test Cases:**

Test cases for CGPA predictor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case** | **Expected Output** | **Actual Output** | **Result** |
| T1 | Predict CGPA using Model 1 | Final CGPA | Final CGPA is calculated | Pass |
| T2 | Predict CGPA using Model 2 | Final CGPA | Final CGPA is calculated | Pass |
| T3 | Predict CGPA using Model 3 | Final CGPA | Final CGPA is calculated | Pass |

**7. OUTPUT SCREENS**

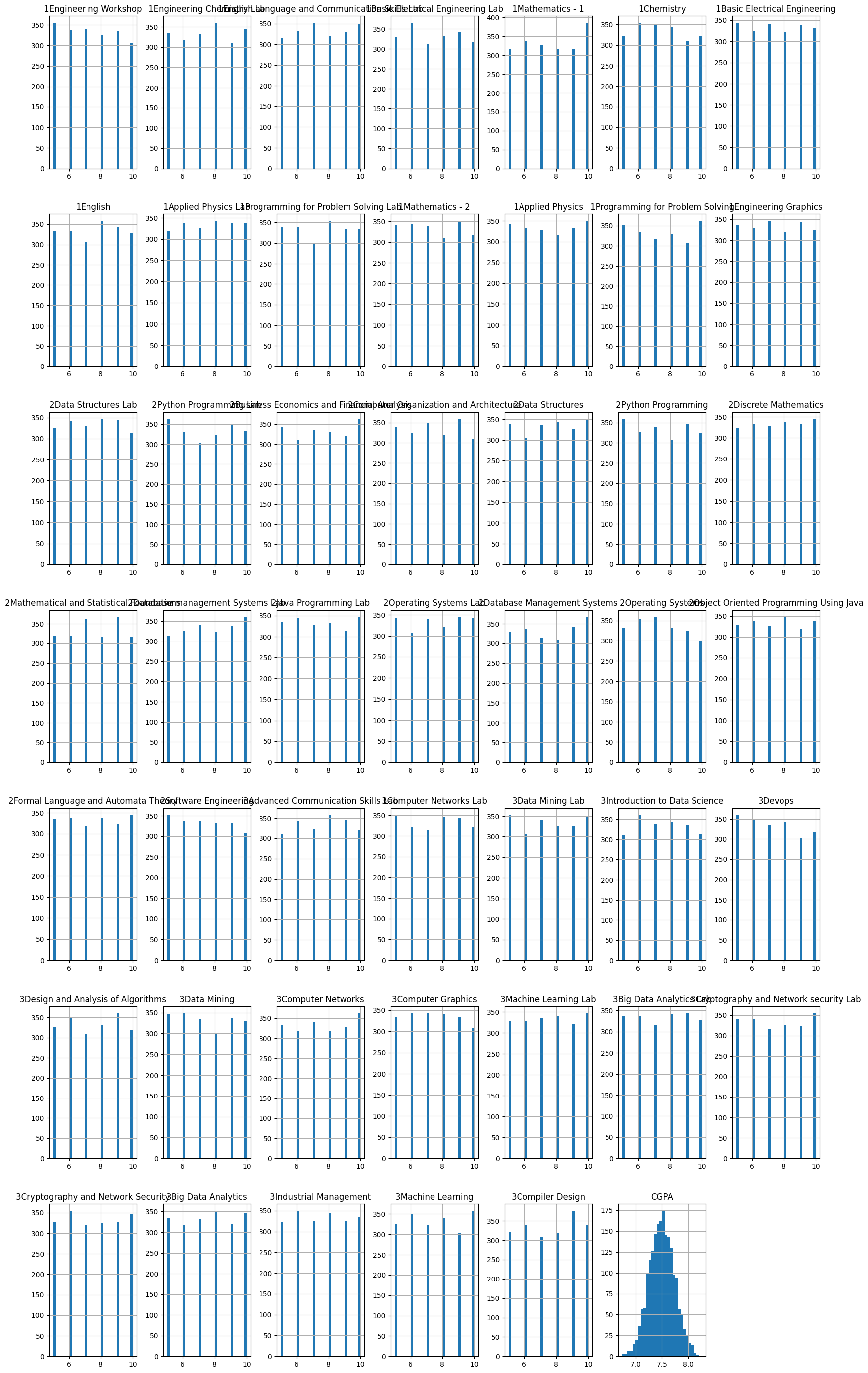


Fig 7.1 Insights of Dataset

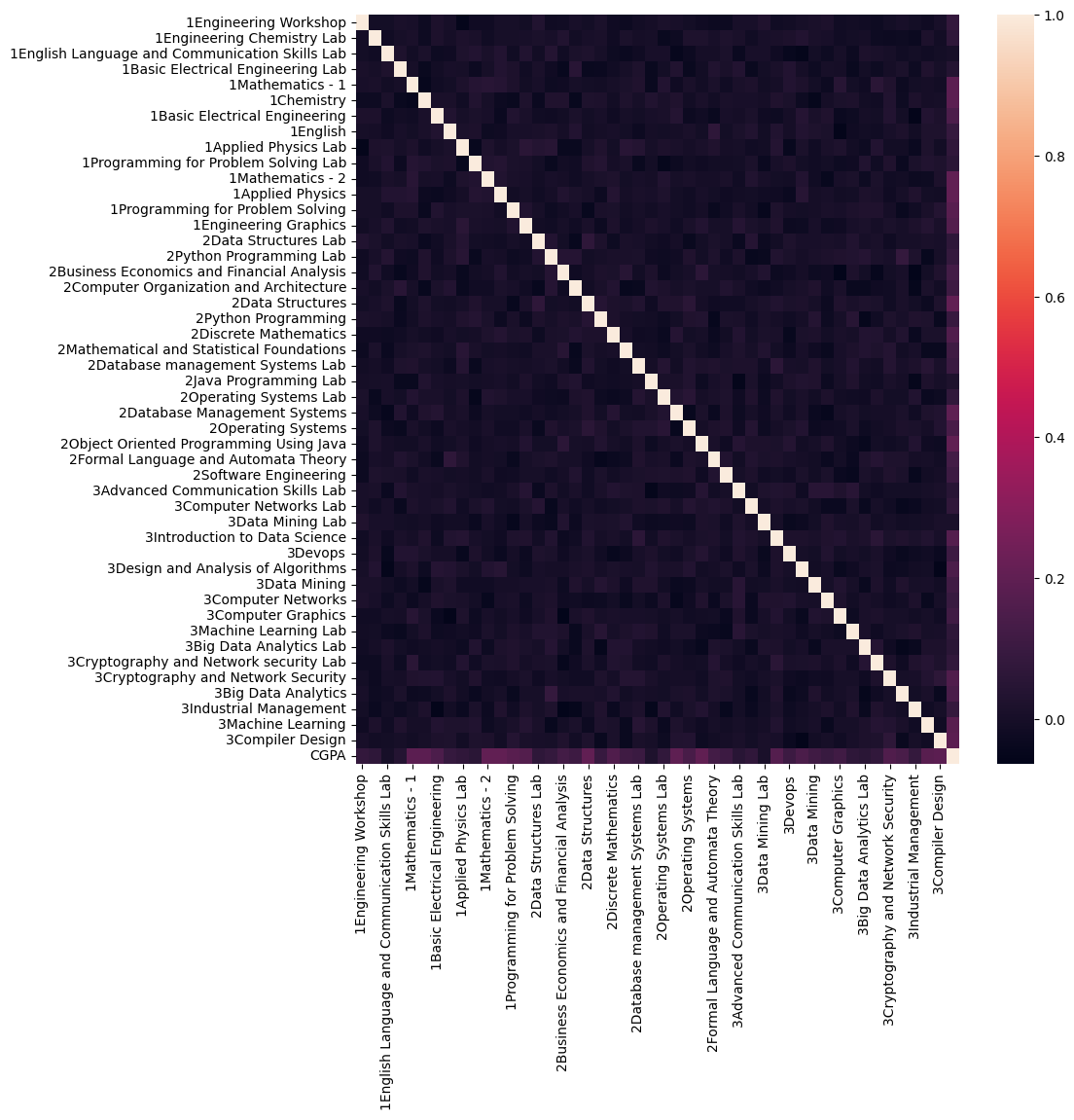
****

Fig 7.2 Correlations in the Dataset

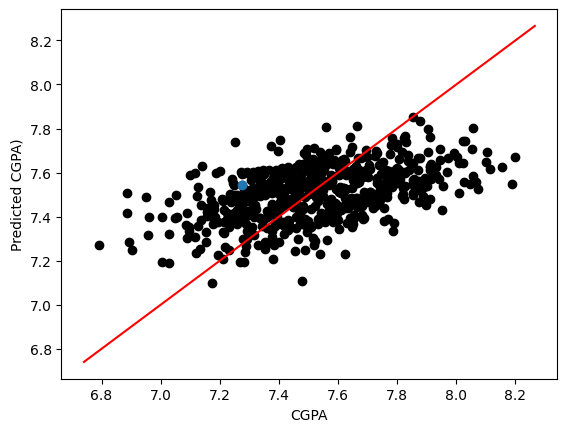
****

Fig 7.3 Predicted CGPA vs Actual CGPA

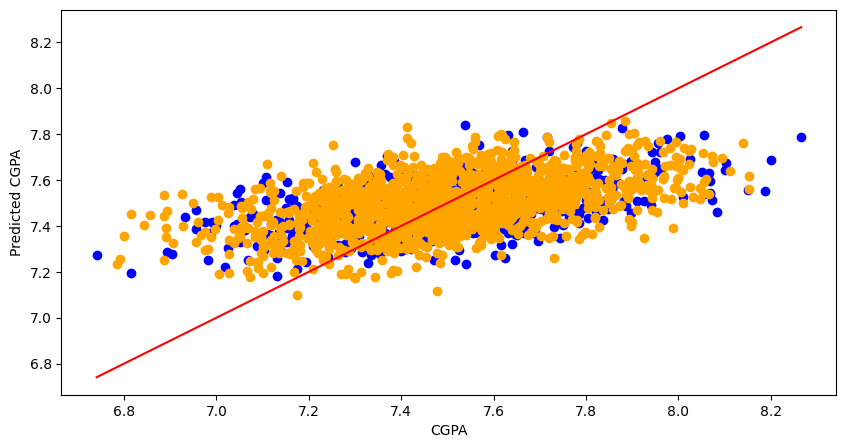
****

Fig 7.4 Predicted CGPA vs Actual CGPA in Model 1

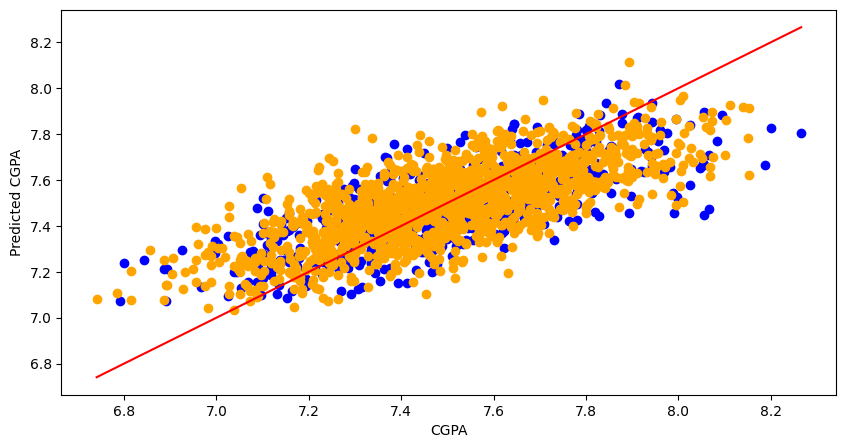
****

Fig 7.5 Predicted CGPA vs Actual CGPA in Model 2

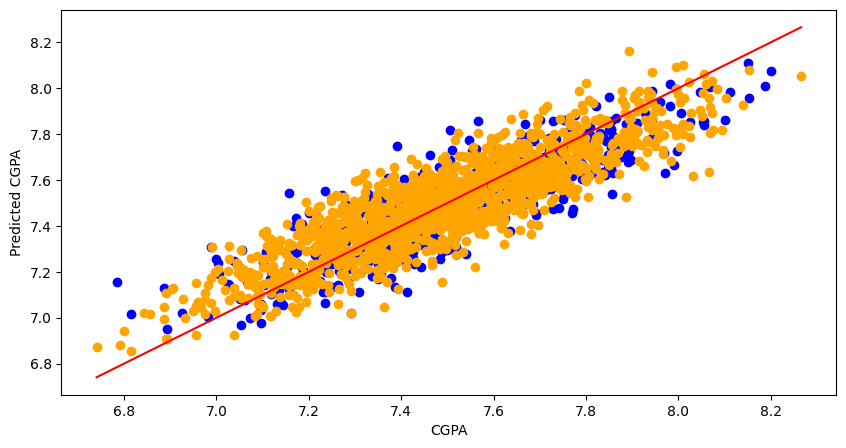
****

Fig 7.6 Predicted CGPA vs Actual CGPA in Model 3

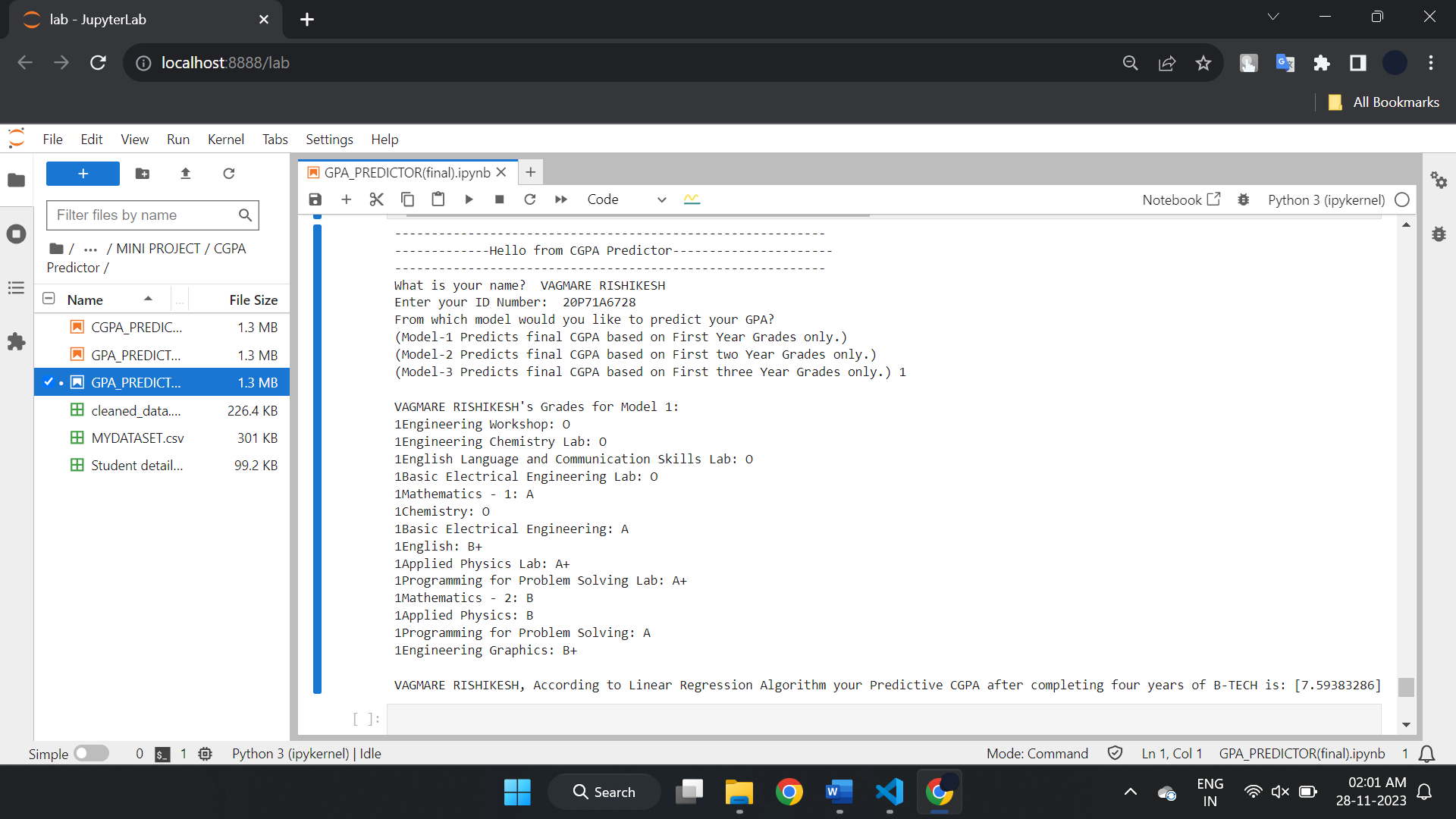


Fig 7.7 Predicted CGPA according to Model 1

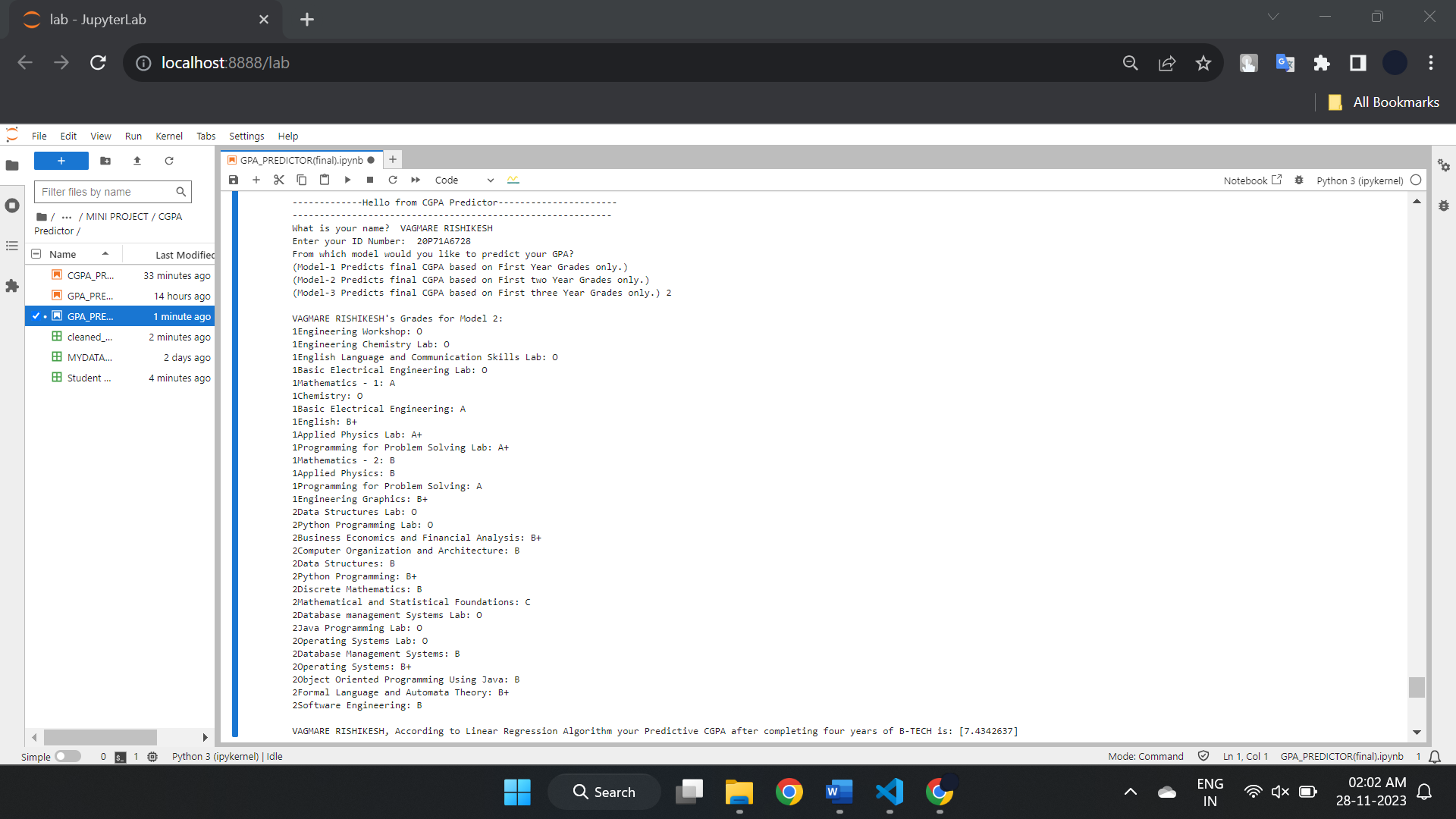


Fig 7.8 Predicted CGPA according to Model 2

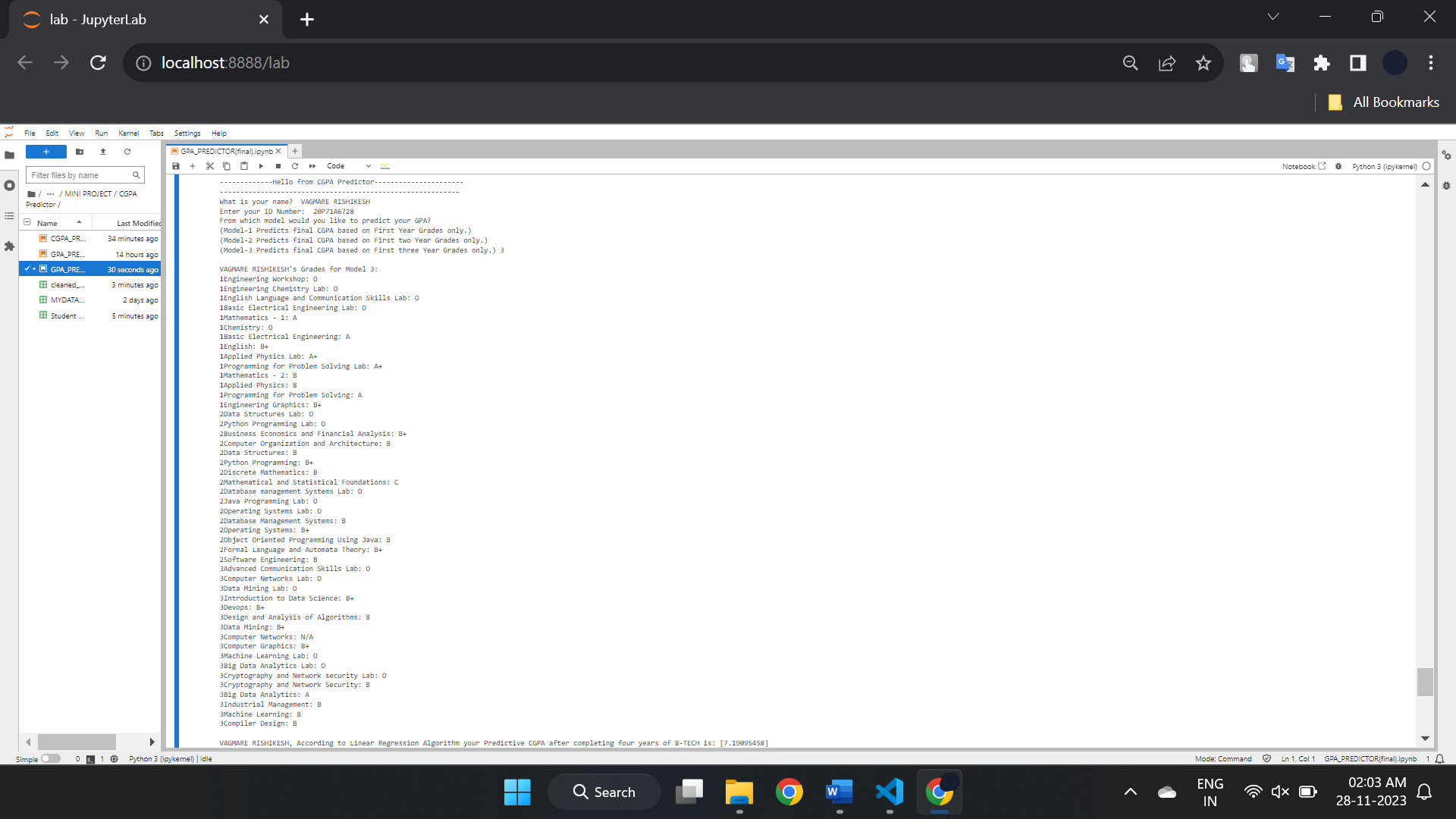


Fig 7.8 Predicted CGPA according to Model 3

**8. CONCLUSION**

One of the important performance indicators that might assist instructors in keeping tabs on students academic achievement is predicting their grades. As a result, it's crucial to have a predictive model that can lower the degree of uncertainty in the result for a dataset that is unbalanced. Based on the prior student final examination results from the previous year courses, this project suggests a Linear Regression prediction model with three predictive models to forecast the final student grades. To assess the performance accuracy of student grade prediction.

**REFERENCES**

1. Mustafa Agaoglu, ―Predicting Instructor Performance Using Data Mining

Techniques in Higher Education," IEEE Access , Volume: 4 ,2016

2. Tripti Mishra,Dr. Dharminder Kumar,Dr. Sangeeta Gupta,"Mining Students’

Data for Performance Prediction," in fourth International Conference on Advanced

Computing & Communication Technologies,2014.

3. Crist´obal Romero," Educational Data Mining: A Review of the State of the

Art," IEEE Transactions On Systems, Man, And Cybernetics—Part C:

Applications And Reviews, Vol. 40, No. 6, November 2010.

4. Carlos Márquez Vera, Cristóbal Romero Morales and Sebastián Ventura

Soto,―Predicting of school failure and dropout by using data mining techniques‖,

The IEEE Journal of Latin-American Learning Technologies (IEEE-RITA) , Vol.

8, No. 1, pp 7-14, Feb 2013.

5. R.S.J.D Baker and K.Yacef, ―The State of Educational Data Mining in 2009: A

Review and Future Visions‖ , Journal of Educational Data Mining, 1, Vol 1, No 1,

2009