**COLONEL’S ACADEMY**

****

**Subject code:** (065)

**“Student Academic Performance Analysis Report”**

**Submitted by: Submitted to:**

Gargi Naroliya Mrs Mala Patidar

**Class: Subject:**

12’A’ Information Practices

**Introduction**

This report provides a data-driven analysis of academic performance for five students across five subjects. The central theme revolves around **Education**, **Data Science**, and **Visualization**. The objective is to leverage Python’s data processing capabilities to compute total marks, percentages, and grades, and to present the findings through meaningful visualizations.

Python libraries such as Panda for data manipulation and Matplotlib for data visualisation, were used to structure and analyse the data. Visual tools like bar charts and pie charts make the report more intuitive and help uncover patterns such as subject-wise strengths and overall grade distribution.

This project demonstrates how programming can simplify and clarify academic performance evaluation.

**Importing libraries**

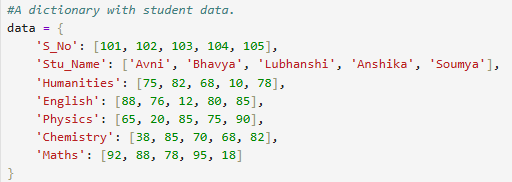
Libraries in Python are essential toolkits that support specific tasks. To begin, we must import the required libraries. The import statement is used to access these libraries within our program, and the as keyword is used to assign an alias for convenience.

. 

* Pandas enable us to create and manage structured data in tabular form.
* Matplotlib..pyplot allows us to for the creation of bar charts, pie charts, and other graphical representations..

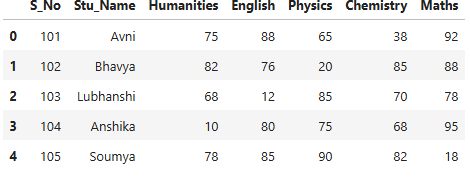
**Creating the Dataset**

To construct the dataset, we first organize the raw data using a nested dictionary. This structure is then converted into a tabular format using the DataFrame() function from Pandas.



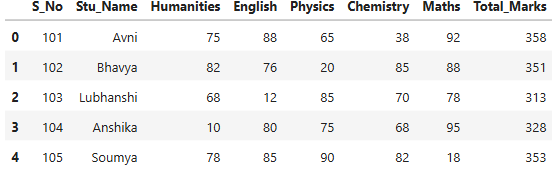


This code will show the output :



**Calculating Total Marks and Percentage**

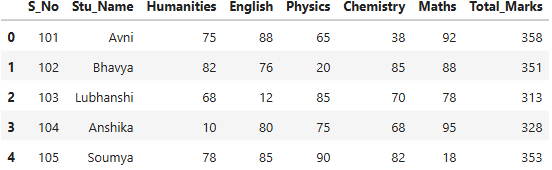
To compute the total marks for each student, we use the sum(axis=1) function, which aggregates data across each row (i.e., student-wise totals) 

This results in the creation of a new column, **Total\_Marks**. 

Next, we calculate each student’s percentage out of 500 marks. A new column, **Percentage**, is created to reflect this data.

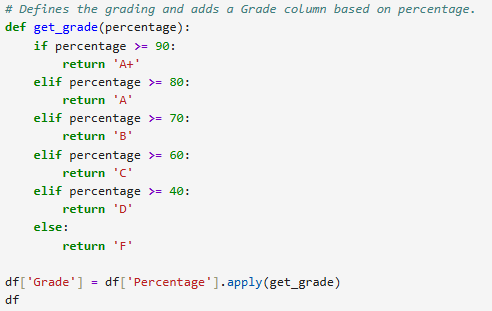


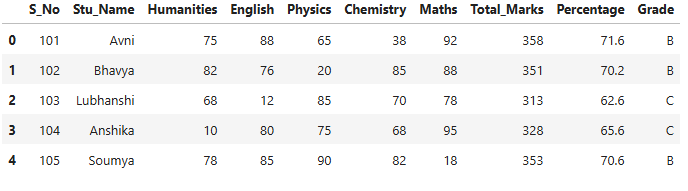
The generated output will be :



### ****Assigning Grades****

Grades are assigned based on percentage values using a custom function defined with def. This function standardizes the grading logic and applies it across all student records:



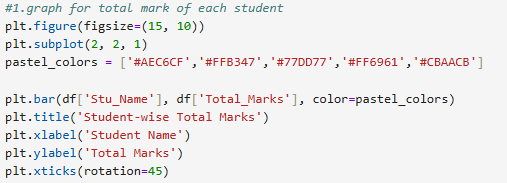


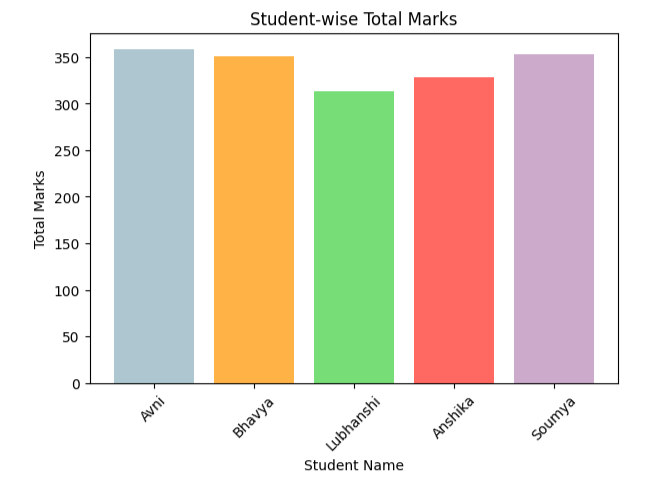
**Visualizations**

"Visualizations help us see patterns and comparisons more clearly than raw numbers alone.."

### With the data prepared and calculations complete, we now move on to the visualization phase to better understand and communicate the results through graphical representations."

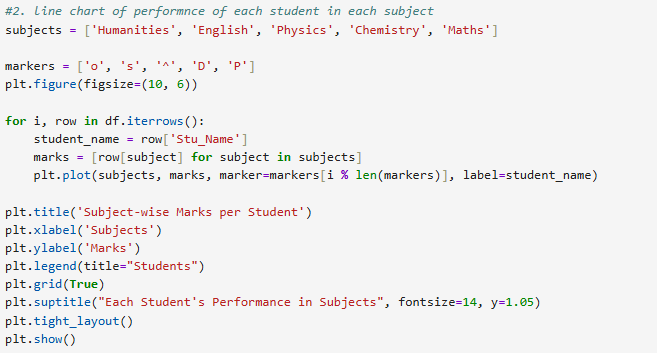
* Student-wise Total Marks – A bar chart showing each student's overall performance.

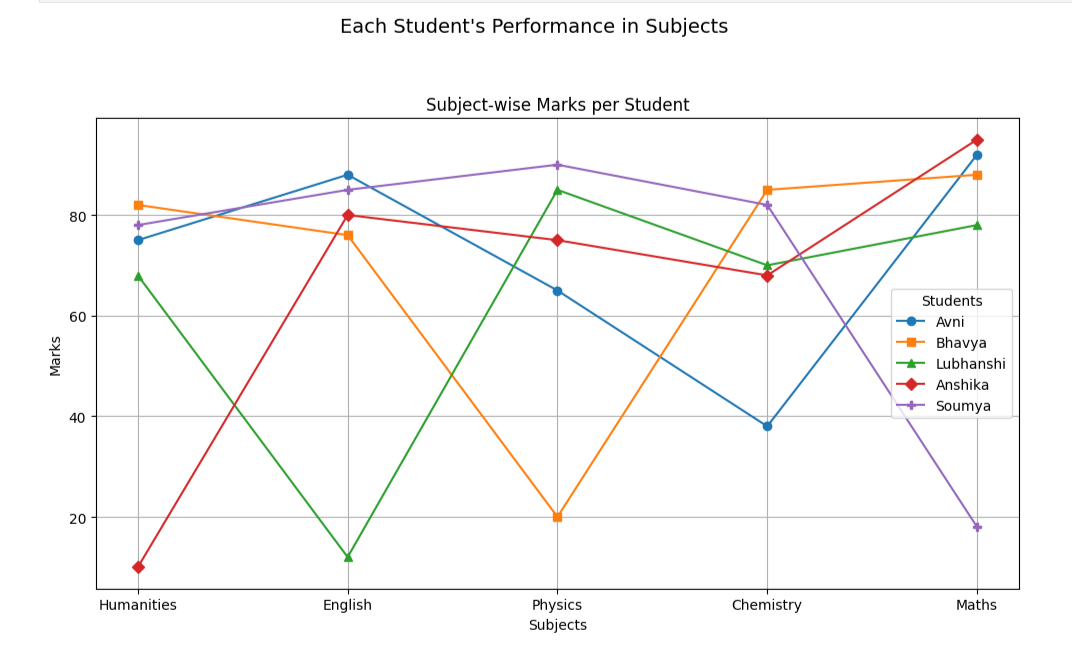
  
(Below is the graphical representation of each student’s total marks.



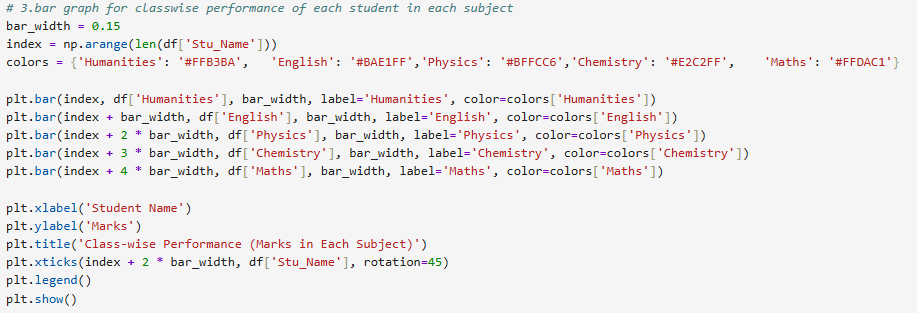
* **Subject-wise Average Marks** – A bar chart representing the average marks in each subject.

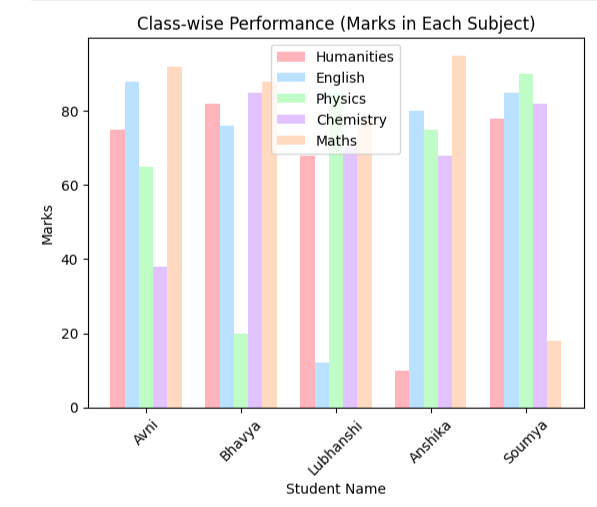
(line chart)





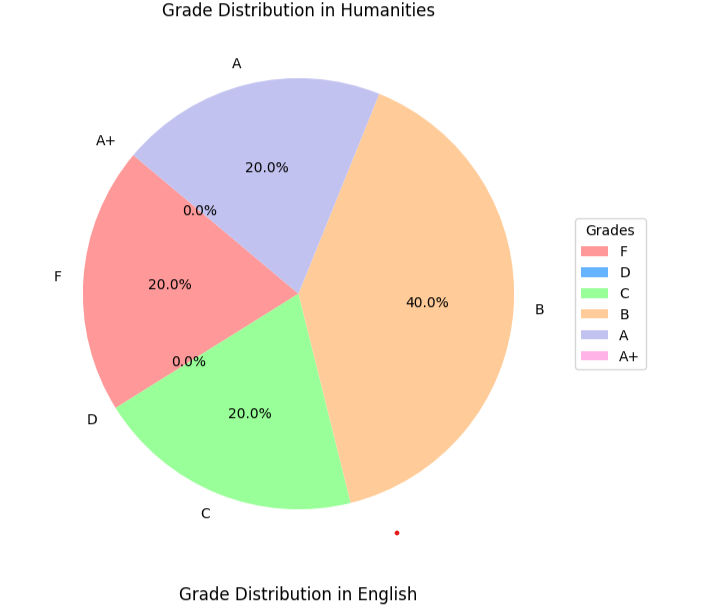
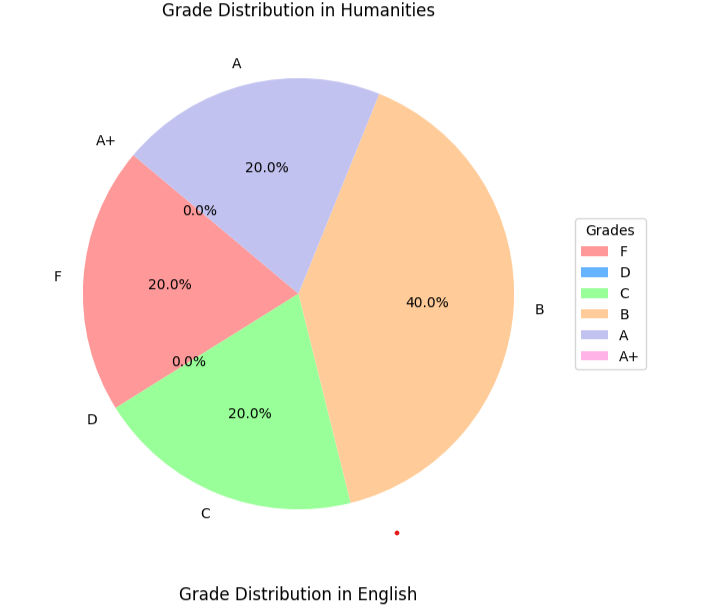
(bar graph)

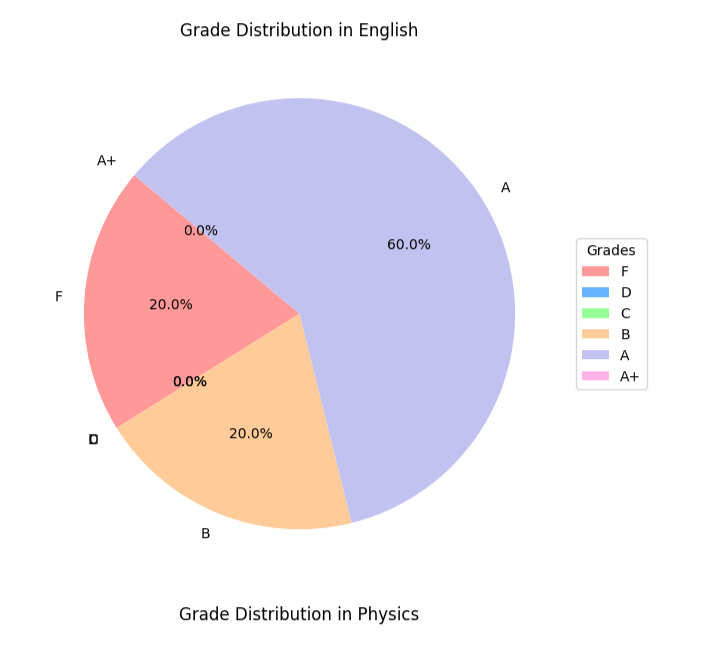
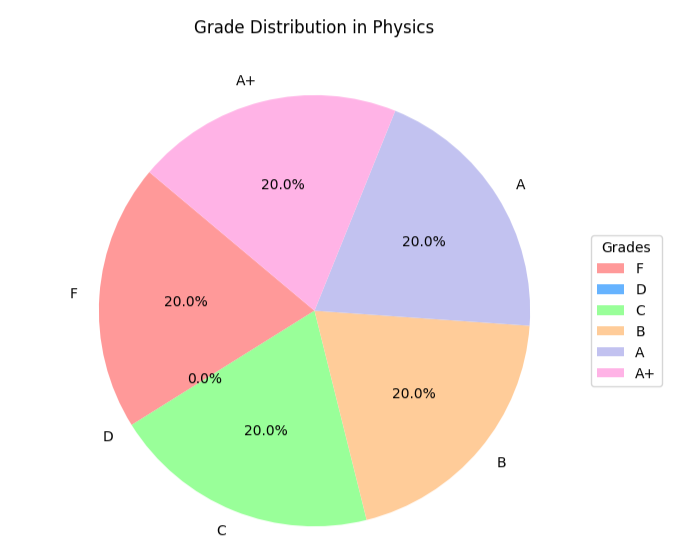
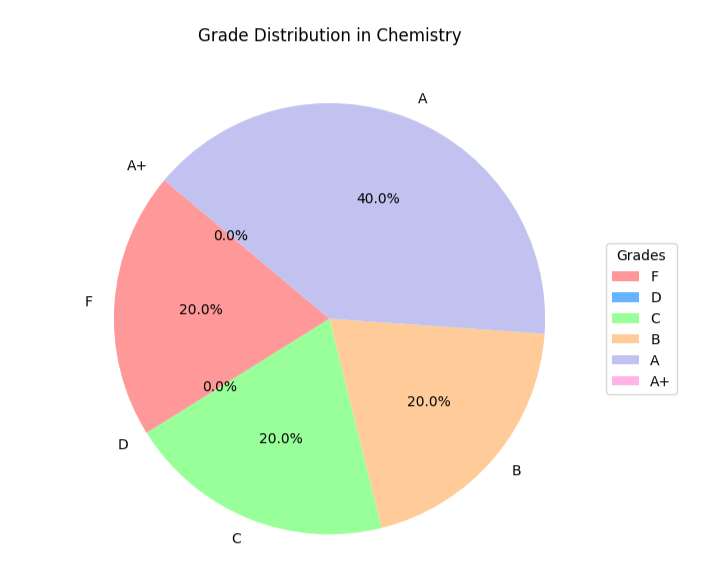
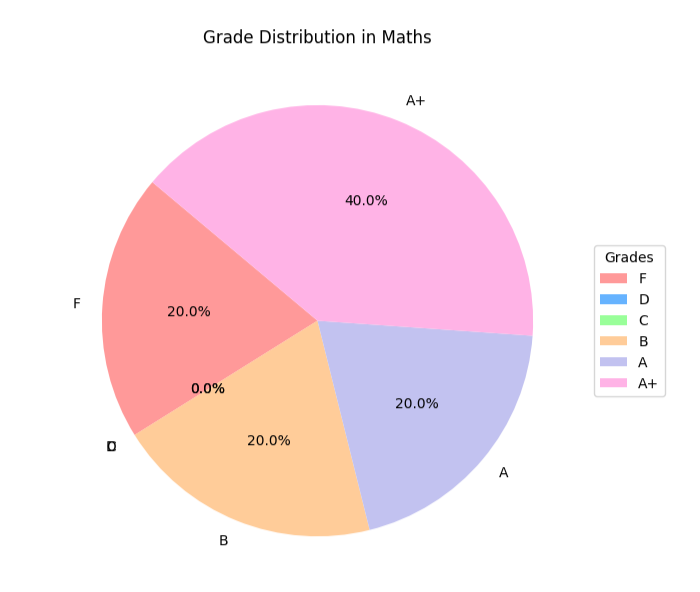




* **Grade Distribution** – A pie chart illustrating how grades are distributed among the students.





### ****Conclusion****

This project demonstrates the practical application of Python in educational data analysis. It simplifies the task of computing performance metrics and presents insights in a visually engaging manner. Through the use of libraries like Pandas and Matplotlib, we effectively transformed raw data into actionable information.