# Experiment 0: Basic NumPy, Pandas, Matplotlib & Seaborn for Machine Learning

## 1. Database Source

## 2. Dataset Description :

The experiment utilizes the **Student Performance Dataset** (student_performance.csv).

- **Size:** The dataset contains records of student metrics across **5 columns**.
- **Features:**
  - Hours_Studied: Number of hours spent studying (Numerical).
  - Attendance: Percentage of classes attended (Numerical).
  - Assignment_Score: Scores obtained in internal assignments (Numerical).
  - Midterm_Score: Performance in the midterm examination (Numerical).
- **Target Variable:** Final_Score (The final grade/score achieved by the student).
- **Characteristics:** This dataset is used to analyze the correlation between student habits (study time, attendance) and their academic outcomes. It is purely numerical, making it ideal for statistical visualization and linear correlation studies.

## 3. Mathematical Formulation of the Algorithm :

This experiment focuses on Data Analysis and Visualization rather than a single predictive model. However, the theoretical foundations include:

- **NumPy Vectorization:** Performs element-wise operations using ndarrays, optimized via C-level loops.
- **Correlation Coefficient (Pearson's r):** Used in Seaborn's heatmaps to measure the strength of the linear relationship between variables:
- **Data Aggregation:** Pandas uses split-apply-combine logic to compute group statistics.

## 4. Algorithm Limitations :

- **NumPy:** Limited to homogeneous data types; cannot handle labeled tabular data as effectively as Pandas.
- **Pandas:** Can be memory-intensive for extremely large datasets (multi-gigabyte) as it loads data entirely into RAM.
- **Matplotlib/Seaborn:** * Static visualizations may hide interactive details found in dynamic dashboards.
  - Over-plotting can occur if the dataset is too large, making scatter plots unreadable without alpha-blending or hexbinning.

## 5. Methodology / Workflow :

1. **Environment Setup:** Import numpy, pandas, matplotlib.pyplot, and seaborn.
2. **Data Acquisition:** Load the student_performance.csv using pd.read_csv().
3. **Numerical Analysis (NumPy/Pandas):** * Use NumPy to calculate statistical summaries (mean, median, standard deviation).
    - Use Pandas for data cleaning and filtering (e.g., finding students with >90% attendance).
4. **Exploratory Data Analysis (EDA):**
    - **Distribution:** Use Seaborn histplot to view the spread of Final_Score.
    - **Relationship:** Use Matplotlib scatter or Seaborn regplot to see how Hours_Studied impacts scores.
    - **Correlation:** Generate a heatmap to identify which features most influence the final grade.
5. **Performance Analysis:** Review the generated charts to interpret data trends.

## 6. Performance Analysis :

- **Statistical Summary:** Students studied an average of 5-6 hours, with a direct positive correlation observed between study hours and final scores.
- **Visualization Insights:**
    - **Scatter Plot:** Showed a strong upward trend, suggesting that as Attendance and Hours_Studied increase, the Final_Score increases linearly.
    - **Correlation Heatmap:** Revealed that Midterm_Score and Hours_Studied have the highest correlation with the Final_Score ($r > 0.85$).
    - **Histogram:** The Final_Score followed a near-normal distribution, with most students scoring between 65 and 75.

## 7. Hyperparameter Tuning :

In the context of Visualization and Data Analysis, "tuning" refers to the optimization of plot parameters for clarity:

- **Process:** * Adjusted Seaborn bins in histograms to find the optimal granularity.
    - Tuned the cmap (color map) and annot parameters in the heatmap for better readability.
    - Adjusted the figure.figsize in Matplotlib to ensure all labels were visible.
- **Impact:** Proper tuning of plot aesthetics (titles, labels, and color scaling) significantly improved the interpretability of the results, allowing for faster identification of trends in student performance.

**Conclusion:** Through this experiment, we demonstrated that NumPy and Pandas provide the necessary computational power to handle data, while Matplotlib and Seaborn offer the visual tools required to translate numbers into actionable insights. The analysis confirmed that study habits and attendance are the primary drivers of student success.

**Code & Output : Google Colab**