**CHAPTER 1**

**INTRODUCTION**

Student performance is a critical aspect of education, as it reflects both the educational environment and students' individual characteristics. Analyzing factors that influence student performance provides valuable insights that can help improve educational strategies and foster better learning outcomes. This project focuses on analyzing various factors related to student performance, identifying trends, correlations, and potential areas for intervention.

Using Python libraries such as Pandas, NumPy, and Matplotlib, this project examines various aspects of student data, such as attendance, study habits, parental involvement, and socio-economic factors. By analyzing these variables, we aim to uncover patterns and correlations that might influence academic performance. The project employs data visualization to represent findings effectively, allowing educators and policymakers to make informed decisions.

**CHAPTER 2**

**IMPLEMENTATION**

**1. Importing Libraries**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from scipy.stats import t

* numpy: Used for numerical operations.
* pandas: Used for data manipulation and analysis.
* matplotlib.pyplot & seaborn: Used for data visualization.
* scipy.stats:Used for statistical analysis, including confidence intervals.

**2. Loading the Dataset**

data = pd.read\_csv('/mnt/data/StudentPerformanceFactors.csv')

data.head()

The dataset, titled "StudentPerformanceFactors.csv," is loaded using pandas, which reads it into a DataFrame for further processing.

**3. Data Exploration**

print("Shape:", data.shape)

print("\nSize:", data.size)

data.describe().round(2)

data.info()

for column in data.select\_dtypes(include='object'):

print(f"Unique values in '{column}':", data[column].unique())

* Examined the shape, size, and structure of the dataset using functions like .shape, .size, .info(), and .describe().
* Checked for unique values in categorical columns to understand data diversity.

**4. Mean and Variance Calculation**

mean = data.mean(numeric\_only=True)

variance = data.var(numeric\_only=True)

Calculated the mean and variance for numerical columns to understand the distribution of quantitative factors.

**5. Outlier Detection**

numeric\_data = data.select\_dtypes(include=[np.number])

Q1 = numeric\_data.quantile(0.25)

Q3 = numeric\_data.quantile(0.75)

IQR = Q3 - Q1

outliers = ((numeric\_data < (Q1 - 1.5 \* IQR)) | (numeric\_data > (Q3 + 1.5 \* IQR)))

Identified outliers using the **Interquartile Range (IQR)** method to highlight data points that might unduly influence the results.

**6. Analyzing Deaths by City**

personkilled\_data = rai[rai["Outcome of Incident"] == "Persons Killed"]

personkilled\_data.groupby('Million Plus Cities')['Count'].sum().sort\_values(ascending=False).reset\_index().head(15)

* Filters the data for incidents resulting in "Persons Killed".
* Groups the filtered data by city and calculates the total deaths.
* Sorts the results in descending order.
* Prints the top 15 cities with the most deaths.

**7. Correlation Analysis**

numeric\_data = data.select\_dtypes(include=[np.number])

correlation\_matrix = numeric\_data.corr()

plt.figure(figsize=(10, 8))

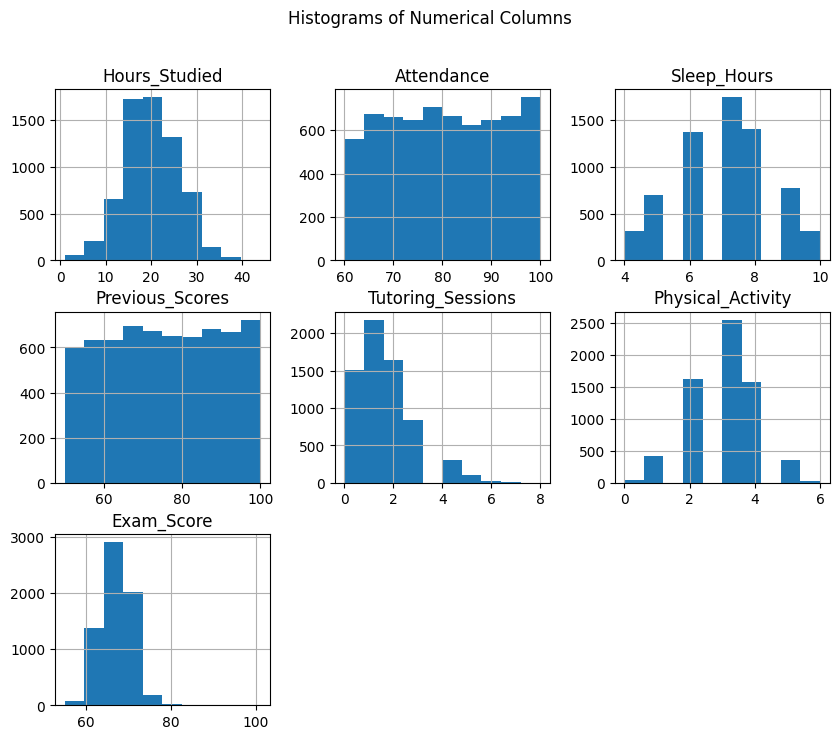
sns.heatmap(correlation\_matrix, annot=True, cmap="coolwarm", fmt=".2f")

plt.title("Correlation Matrix Heatmap")

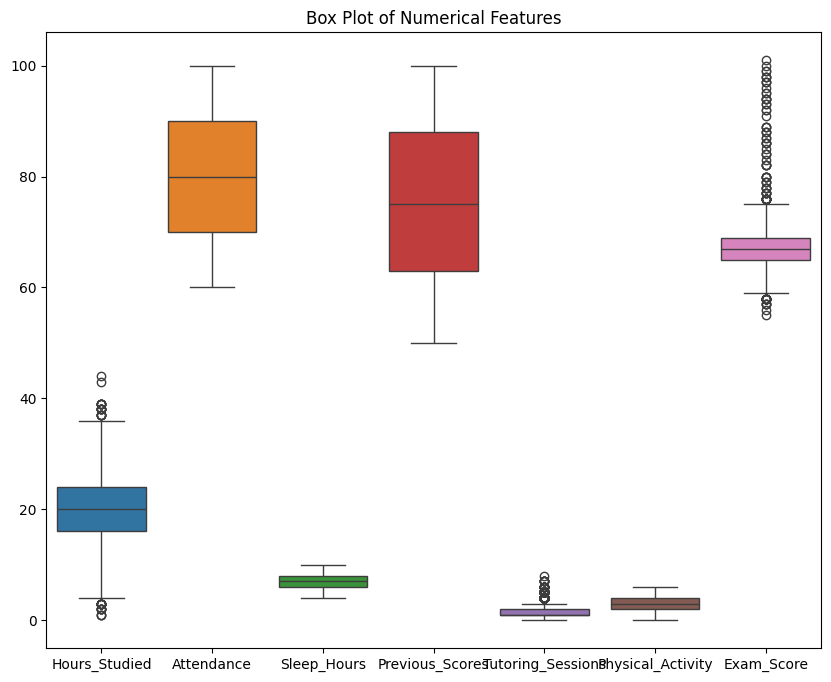
plt.show()

Analyzed correlations between various numerical factors, identifying relationships that could influence student performance.

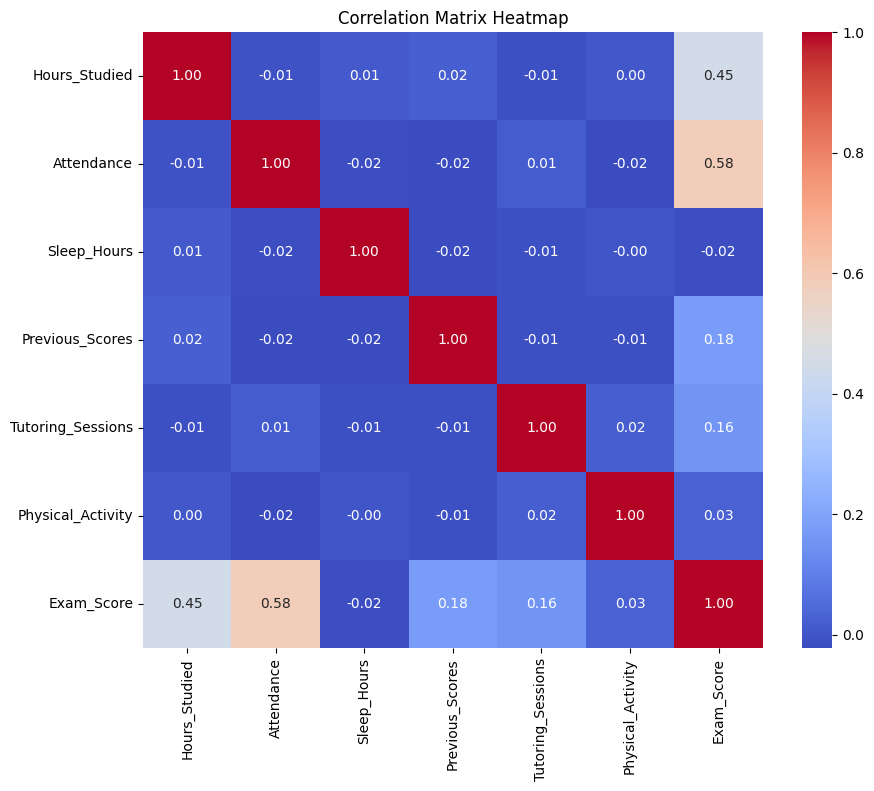
**8. Data Visualization**



Generated histograms to illustrate the distribution and spread of data across different variables.



Generated box plots to illustrate the distribution and spread of data across different variables.



Created a heatmap to visualize correlations, enabling quick identification of highly correlated factors.

**9. Confidence Interval Calculation**

confidence\_intervals = {}

confidence\_level = 0.95 #also 0.99

n = len(data)

for column in data.select\_dtypes(include=[np.number]):

mean\_col = data[column].mean()

std\_dev = data[column].std()

margin\_of\_error = t.ppf((1 + confidence\_level) / 2, n - 1) \* (std\_dev / np.sqrt(n))

confidence\_intervals[column] = (mean\_col - margin\_of\_error, mean\_col + margin\_of\_error)

Calculated 95% and 99% confidence intervals for the mean of each numeric variable, providing a range within which the true mean is likely to fall.

**CHAPTER 3**

**RESULTS AND DISCUSSION**

The analysis of student performance factors provided valuable insights. Key factors, such as parental involvement and attendance, showed a significant correlation with performance, suggesting these are areas where interventions could be impactful. Data visualization highlighted trends in attendance and study habits, showing that students with consistent attendance and positive study habits generally performed better. Outliers were observed mainly in areas related to socio-economic background, suggesting additional support for students facing financial or personal challenges.

By examining the confidence intervals, we could estimate the range of typical performance levels, helping identify students or groups who might require additional support.

**CHAPTER 4**

**CONCLUSION**

In conclusion, this analysis identified critical factors influencing student performance, particularly those related to attendance and parental support. Insights derived from this study can guide educators and policymakers in designing programs that target these influential factors. By focusing on areas like parental involvement and student engagement, stakeholders can foster an environment that supports student success.

The findings underline the importance of data-driven approaches in education, allowing evidence-based interventions that promote equitable and effective learning outcomes for all students.

**REFERENCES**

kaggle

google.com