EXPERIMENT-0

Exploratory Data Analysis and Data Preprocessing using Python Libraries

1 Dataset Source

The dataset used in this experiment is a Student Performance Dataset, containing academic and behavioral attributes of students.

Dataset Source Link: https://github.com/MLDL-Lab/blob/main/datasets/student_performance.csv

2 Dataset Description

The dataset **student_performance.csv** contains student academic performance indicators.

Features (Independent Variables)

Feature	Description
Hours_Studied	Number of hours a student studied
Attendance	Attendance percentage
Assignment_Score	Marks obtained in assignments
Midterm_Score	Marks obtained in midterm exam

Target Variable

Target	Description
Final_Score	Final examination marks

Data Type

- All features are numerical
- Continuous data
- No categorical attributes initially

3 Mathematical Formulation

Since this experiment focuses on **statistical analysis and normalization**, the mathematical foundations used are:

Mean

$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$ μ=n1i=1∑nxi

Where:

- x_i xi = individual score
- n n = number of students

Median

Middle value after sorting data.

If n is even:

$Median = x_{n/2} + x_{(n/2+1)}$ 2Median=2xn/2+x(n/2+1)

Standard Deviation

$\sigma = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu)^2$ σ=n1i=1∑n(xi−μ)2

Measures spread of scores.

Min-Max Normalization

$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$ Xnorm=Xmax−XminX−Xmin

Transforms data into range [0,1].

Correlation (Pearson Correlation)

$r=\sum(x-\bar{x})(y-\bar{y})\sum(x-\bar{x})2\sum(y-\bar{y})2$r=∑(x−x¯)2∑(y−y¯)2∑(x−x¯)(y−y¯)

Used in heatmap analysis.

4 Algorithm Limitations

Limitations of Statistical Analysis

- Mean is sensitive to outliers
- Correlation only measures linear relationships
- Normalization does not remove skewness

Limitations of Visualization

- Scatter plots become unclear for very large datasets
- Heatmap only shows linear correlation
- EDA does not guarantee model accuracy

EDA is exploratory, not predictive.

5 Methodology / Workflow

Step 1: Data Loading

- Dataset loaded using Pandas
- Converted Final_Score to NumPy array

Step 2: Statistical Analysis

- Computed mean, median, standard deviation
- Applied Min-Max normalization

Step 3: Data Inspection

- Checked dataset shape
- Checked missing values
- Reviewed feature names

Step 4: Feature Engineering

- Created new categorical feature: Performance
    - Excellent
    - Good
    - Average
    - Poor

Step 5: Data Visualization

- Line Plot (Hours vs Final Score)
- Histogram (Distribution)
- Scatter Plot
- Correlation Heatmap
- Boxplot (Categorical comparison)

Workflow Diagram

Dataset
↓
Data Loading (Pandas)
↓
Data Cleaning & Inspection
↓
Statistical Analysis (NumPy)
↓
Feature Engineering
↓
Visualization (Matplotlib + Seaborn)
↓
Insights & Interpretation

6 Performance Analysis

Since no ML model was trained:

- This experiment focuses on data understanding.

- Correlation heatmap shows which features strongly influence Final_Score.
- Scatter plot shows positive relationship between Hours_Studied and Final_Score.
- Histogram shows distribution pattern (normal / skewed).

Observations

1)Numpy Basics [Mean,Median and Mode Calculated]

Code

```
import numpy as np
import pandas as pd
df = pd.read_csv('/mnt/data/student_performance.csv')
final_scores = df['Final_Score'].values


mean_val = np.mean(final_scores)
median_val = np.median(final_scores)
std_val = np.std(final_scores)

print("Mean:", mean_val)
print("Median:", median_val)
print("Standard Deviation:", std_val)


min_val = np.min(final_scores)
max_val = np.max(final_scores)
normalized = (final_scores - min_val) / (max_val - min_val)

print("\nFirst 5 Normalized Values:\n", normalized[:5])
```

```
Mean: 68.95
Median: 70.5
Standard Deviation: 8.71478628538876

First 5 Normalized Values:
 [0.         0.16129032 0.25806452 0.38709677 0.51612903]
```

2)Pandas Data Handling

Code

```
import pandas as pd

df = pd.read_csv('/mnt/data/student_performance.csv')
```

```
print("Shape:", df.shape)
print("\nColumns:\n", df.columns)
print("\nMissing Values:\n", df.isnull().sum())

def label(score):
    if score >= 85:
        return "Excellent"
    elif score >= 70:
        return "Good"
    elif score >= 50:
        return "Average"
    else:
        return "Poor"

df['Performance'] = df['Final_Score'].apply(label)

df.head()
```

```
Shape: (20, 5)

Columns:
 Index(['Hours_Studied', 'Attendance', 'Assignment_Score', 'Midterm_Score',
       'Final_Score'],
      dtype='object')

Missing Values:
 Hours_Studied       0
Attendance           0
Assignment_Score     0
Midterm_Score        0
Final_Score          0
dtype: int64
```

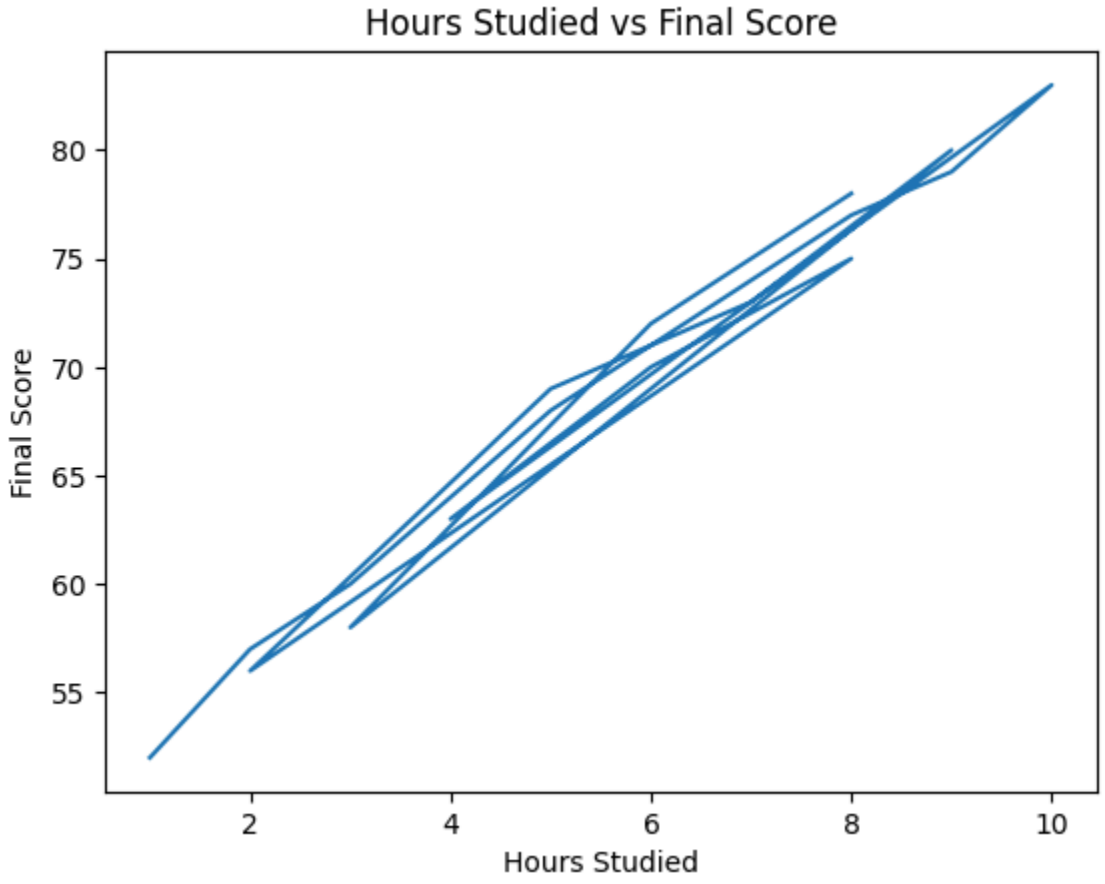
	Hours_Studied	Attendance	Assignment_Score	Midterm_Score	Final_Score	Performance
0	1	60	55	50	52	Average
1	2	65	58	55	57	Average
2	3	70	60	58	60	Average
3	4	75	65	62	64	Average
4	5	80	68	65	68	Average

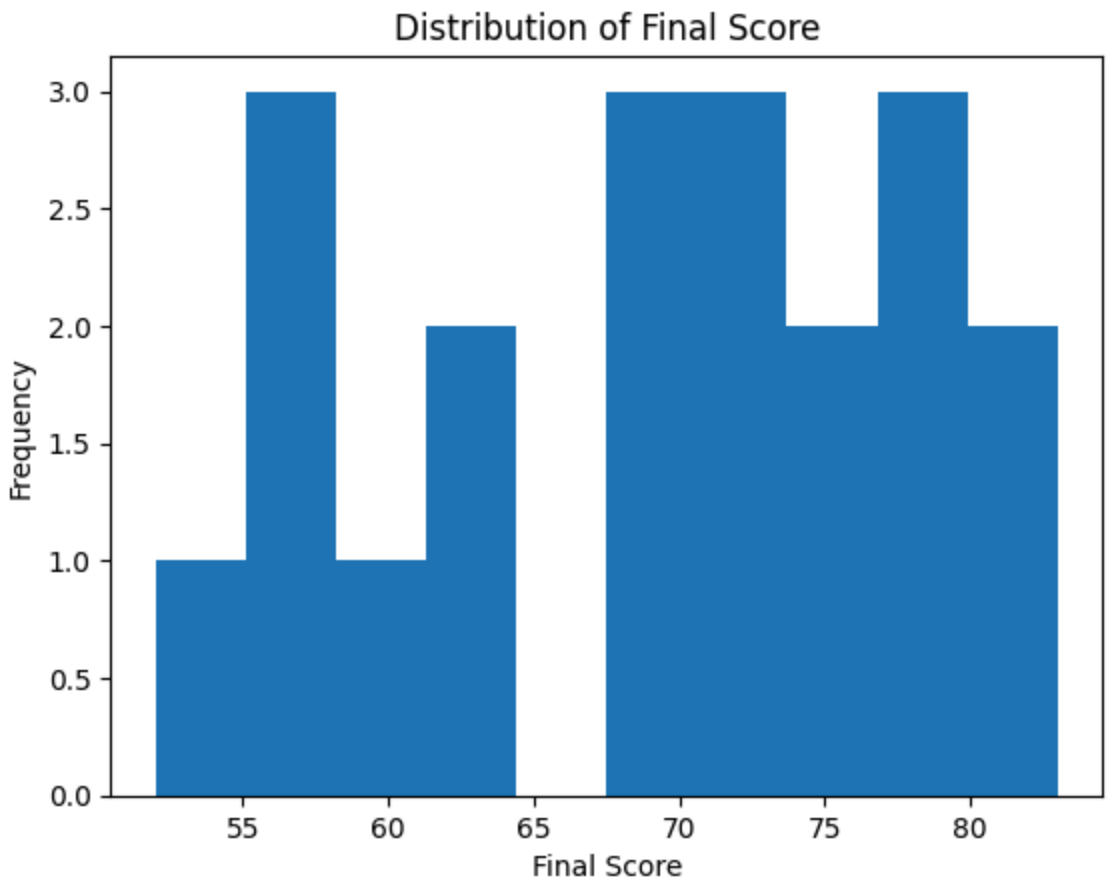
3)Matplotlib Visualization

Code

```
import matplotlib.pyplot as plt
import pandas as pd

df = pd.read_csv('/mnt/data/student_performance.csv')

plt.figure()
plt.plot(df['Hours_Studied'], df['Final_Score'])
plt.xlabel("Hours Studied")
plt.ylabel("Final Score")
```

```
plt.title("Hours Studied vs Final Score")
plt.show()


plt.figure()
plt.hist(df['Final_Score'], bins=10)
plt.xlabel("Final Score")
plt.ylabel("Frequency")
plt.title("Distribution of Final Score")
plt.show()
```





4)Seaborn Visualization

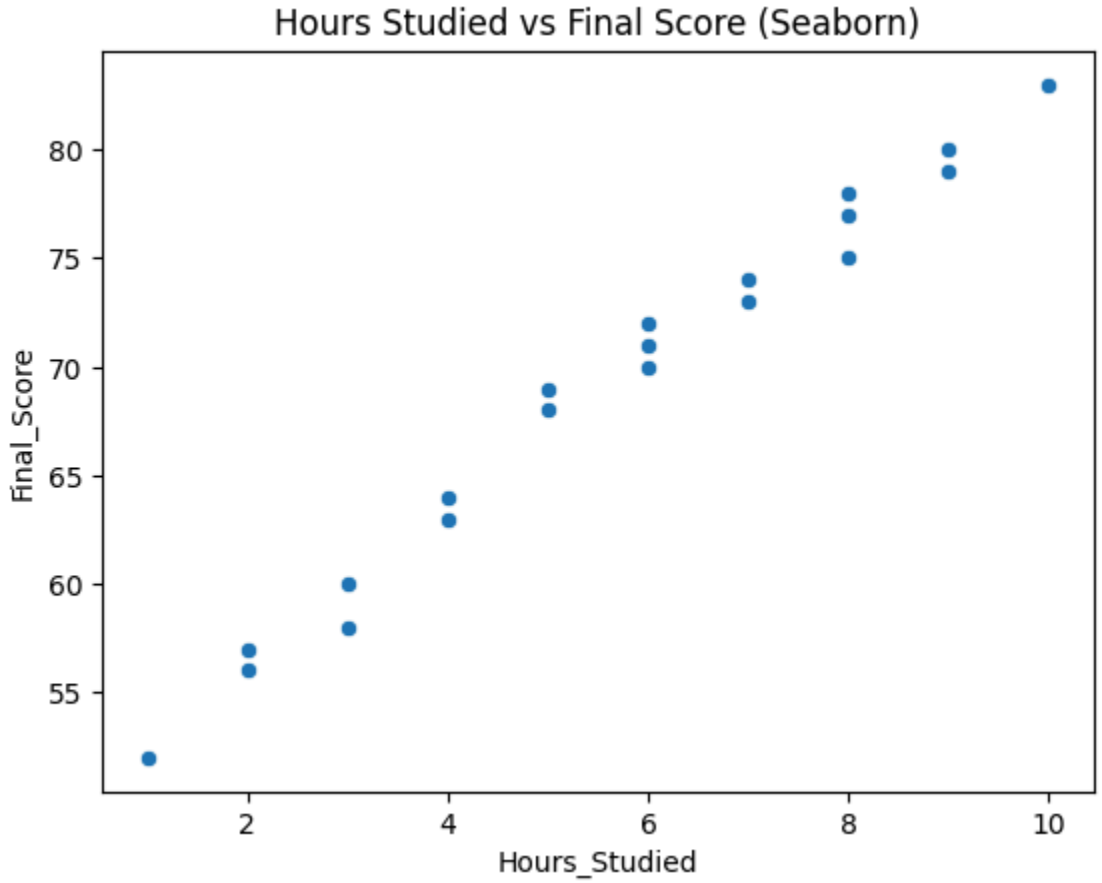
Code

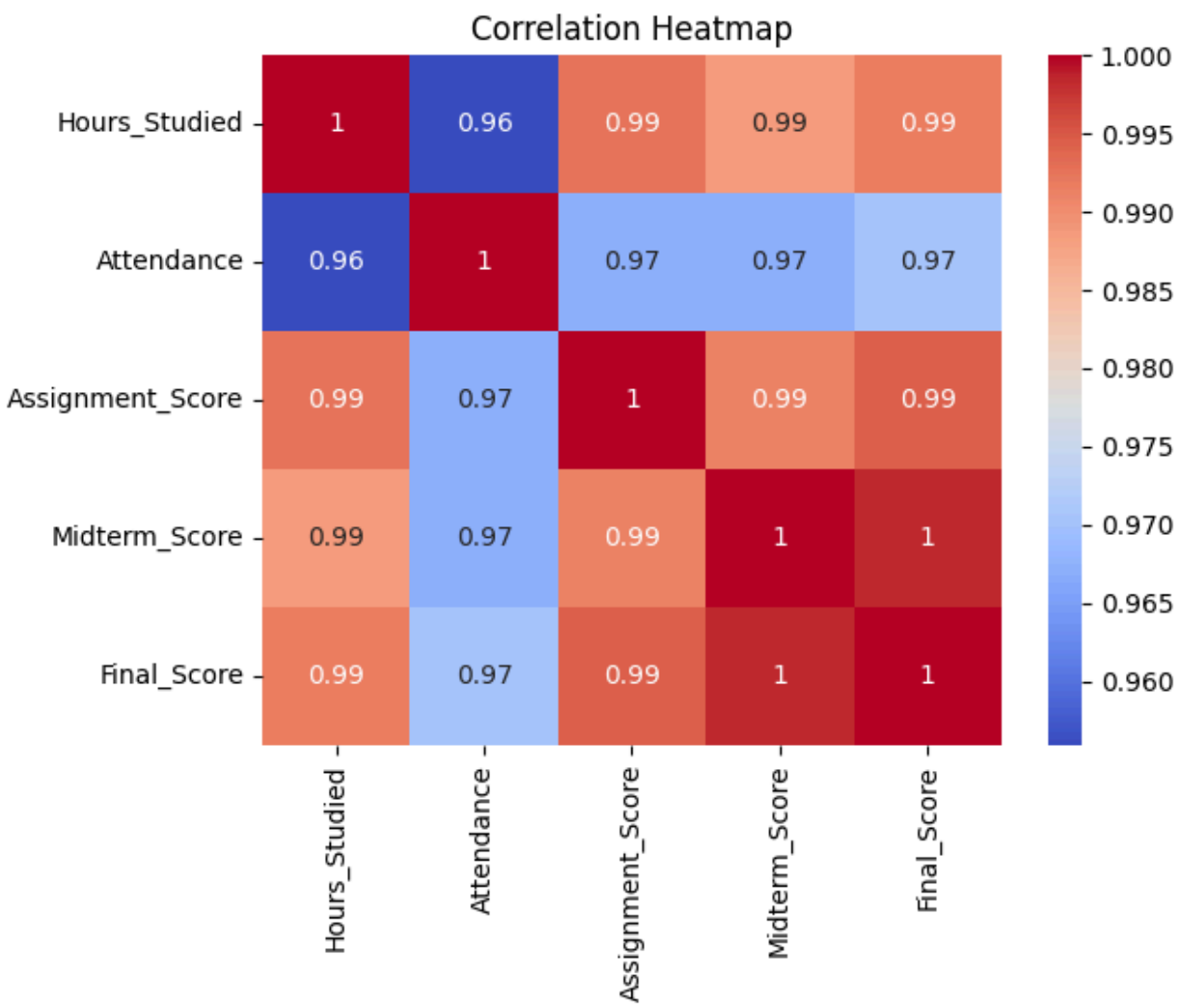
```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

df = pd.read_csv('/mnt/data/student_performance.csv')

def label(score):
    if score >= 85:
        return "Excellent"
    elif score >= 70:
        return "Good"
    elif score >= 50:
        return "Average"
    else:
        return "Poor"

df['Performance'] = df['Final_Score'].apply(label)
```

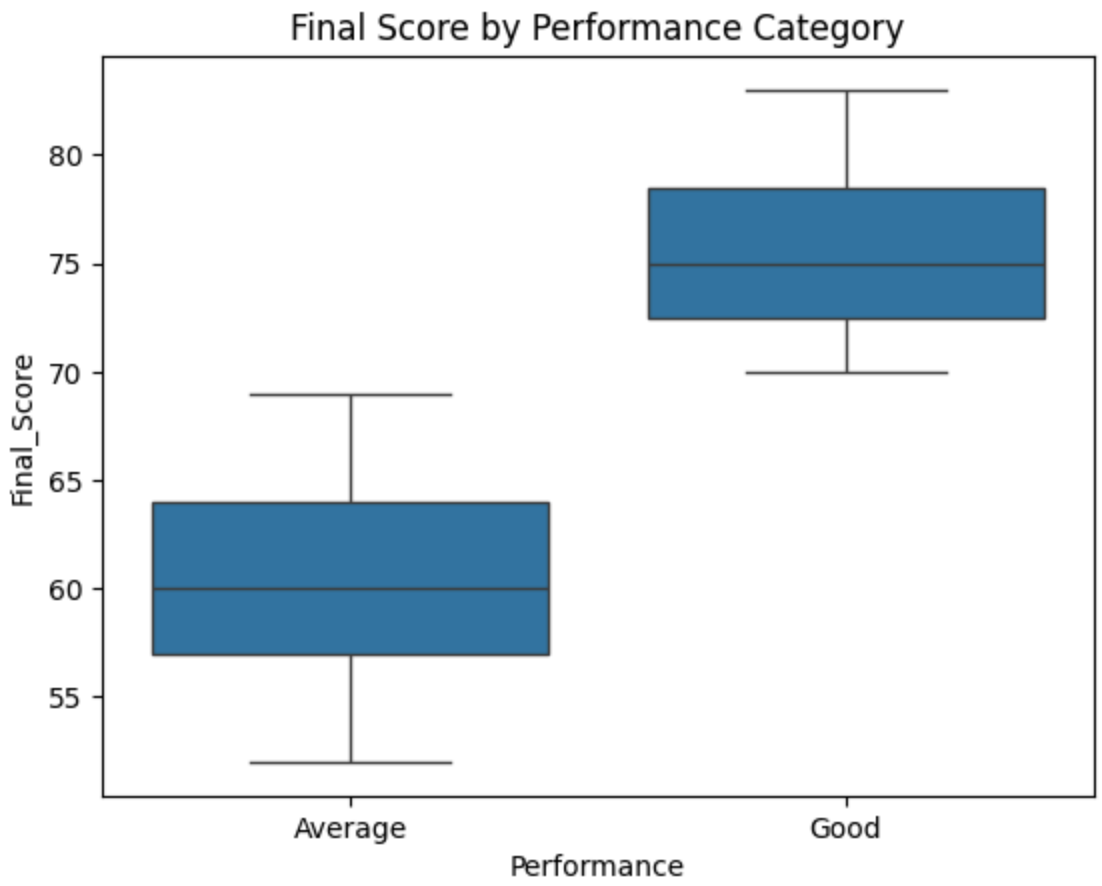
```
plt.figure()
sns.scatterplot(x='Hours_Studied', y='Final_Score', data=df)
plt.title("Hours Studied vs Final Score (Seaborn)")
plt.show()
```

```
plt.figure()
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```

```
plt.figure()
sns.boxplot(x='Performance', y='Final_Score', data=df)
plt.title("Final Score by Performance Category")
plt.show()
```




Correlation Heatmap
Hours_Studied
Attendance
Assignment_Score
Midterm_Score
Final_Score
1
0.96
0.99
0.99
0.99
0.96
1
0.97
0.97
0.97
0.99
0.97
1
0.99
0.99
0.99
0.97
0.99
1
1
0.99
0.97
0.99
1
1
1.000
0.995
0.990
0.985
0.980
0.975
0.970
0.965
0.960



Conclusion

This experiment successfully demonstrated:

- Data handling using Pandas
- Numerical computation using NumPy
- Data visualization using Matplotlib and Seaborn
- Feature engineering
- Correlation analysis

It provided foundational understanding required before building any Machine Learning model.