VNUHCM-UNIVERSITY OF SCIENCE
FACULTY OF INFORMATION TECHNOLOGY

--🕮--

COURSE PROJECT REPORT

Course Title: Applied Mathematics and Statistics for Information Technology
Project Title: Linear Regression

Ho Chi Minh City, August, 2025

Table of contents

Part I. Project Idea

1. Overview

This project aims to predict a student's performance index based on their study habits, previous scores, extracurricular activities, sleep hours, and practice papers completed.

2. Input

A dataset containing the following features for each student:

- Hours Studied
- Previous Scores
- Extracurricular Activities (0 = No, 1 = Yes)
- Sleep Hours
- Sample Question Papers Practiced

3. Output

A predicted performance index (numerical value) for each student.

4. Objective

To build a linear regression model that can accurately estimate the performance index, and to compare different feature combinations to find the most effective model.

5. Main idea

Use exploratory data analysis (EDA) to understand the data, apply statistical methods to evaluate features, and train multiple regression models. Evaluate the models using Mean Absolute Error (MAE) and select the best-performing one.

Part II. Implementation Details

1. List of libraries

1.1. pandas (import pandas as pd)

- Purpose: For data manipulation and analysis, especially with tabular data.
- Usage in the project:
  - Reading CSV files (pd.read_csv).
  - Accessing columns.
  - Performing descriptive statistics (.describe(), .nunique(), .value_counts()) [1].

1.2. numpy (import numpy as np)

- Purpose: Provide support for numerical computations and handling of large, multi-dimensional arrays and matrices [2].
- Usage in the project:
  - Calculating average (e.g., np.mean).
  - Storing and manipulating numerical results like MAE score.

1.3. matplotlib.pyplot (import matplotlib.pyplot as plt)

- Purpose: For creating visualizations of data [3].
- Usage in the project:
  - Setting figure sizes.
  - Creating subplots.
  - Displaying charts.

1.4. seaborn (import seaborn as sns)

- Purpose: For statistical data visualization with more attractive and informative plots [4].
- Usage in the project:
  - Creating boxplot, scatterplot, barplot, lineplot, and heatmap for exploratory data analysis (EDA).

1.5. sklearn.linear_model.LinearRegression

- Purpose: To implement the Linear Regression model [5].
- Usage in the project:
    - Training regression models for predicting the performance index.
    - Obtaining model coefficients and intercepts.

1.6. sklearn.metrics.mean_absolute_error

- Purpose: To evaluate regression model performance.
- Usage in the project:
    - Calculating the Mean Absolute Error (MAE) between predicted and actual values.

1.7. sklearn.model_selection.KFold

- Purpose: To create K-Fold Cross Validation splits.
- Usage in the project:
    - Splitting the dataset into multiple folds for model evaluation and to avoid overfitting.

1.8. sklearn.model_selection.cross_val_score

- Purpose: To evaluate a model's performance using Cross Validation.
- Usage in the project:
    - Calculating MAE across multiple folds and obtaining the average score.

2. Detailed Function Descriptions

2.1. statistic(df, cols)

- Purpose: The statistic function is used to display an overview of key statistical information for selected columns in a DataFrame, helping to understand the nature and distribution of the data.
- How it works:
    - Loops through the specified column names (cols).
    - Prints the column's data type **(dtype()).**
    - Counts and displays the number of unique values **(nunique()).**
    - Shows the top 5 most frequent values **(value_counts()).**
    - If the column is numeric, prints **(describe())** descriptive statistics (count, mean, std, min, max, and quartiles).

2.2. show_chart(df)

- Purpose: Visualizes relationships and distributions of dataset features to support exploratory data analysis.
- How it works: Creates six subplots:
    - **Boxplot()** – Compares performance distribution by study hours.
    - **Scatterplot()** – Shows relationship between previous scores and performance.
    - **Barplot()** – Compares performance between extracurricular activity groups.
    - **Lineplot()** – Shows performance trend by sleep hours.
    - **Scatterplot()** – Examines link between practiced papers and performance.
    - **Heatmap()** – Displays correlation between features.

2.3. train_model(X_train, y_train)

- Purpose: The function is used to create and train a Linear Regression model on a given dataset, then display the model's regression formula.
- How it works:
    - Create a Linear Regression object.
    - Fit the model using the provided training features (**X_train**) and target values (**y_train**).
    - Retrieve the model's coefficients and intercept.
    - Display the regression equation showing the relationship between features and the target.

2.4. evaluate_model_mae(model, X_test, y_test)

- Purpose: The function evaluates a trained regression model's performance on a test dataset by calculating the Mean Absolute Error (MAE).
- How it works:
    - Use the trained model to generate predictions for **X_test**.
    - Compare predictions with **y_test** to compute the Mean Absolute Error.

2.5. find_best_single_feature(X_train, y_train, k=5, random_state=42)

- Purpose: The function identifies the single feature that is the best regression performance, measured by Mean Absolute Error (MAE), using k-fold Cross Validation.
- How it works:

- Create a KFold object with shuffle and a fixed random state for reproducibility.
- For each feature, train a Linear Regression model and evaluate it with Cross Validation, calculating the average MAE.
- Print the MAE for all single-feature models.
- Find the feature with the lowest MAE and return it along with all MAE scores.

2.6. best_feature_model(X_train, y_train, X_test, best_feature)

- Purpose: The function trains a Linear Regression model using only the best-performing feature and returns the trained model along with the corresponding test data for evaluation.
- How it works:
  - Extract the training and testing subsets containing only the chosen best feature.
  - Fit a Linear Regression model on the selected feature.
  - Print the regression equation with the coefficient and intercept.

2.7. evaluate_models(X_model_1, X_model_2, X_model_3, y_train, k=5, random_state=42)

- Purpose: The function compares the performance of three different regression models using k-fold Cross Validation, returning the average Mean Absolute Error (MAE) for each model.
- How it works:
  - Set up a KFold object with shuffle and a fixed random state for reproducibility.
  - For each feature set, train a Linear Regression model and compute the mean MAE using Cross Validation.
  - Print the average MAE for all three models to compare their performance.
  - Return the MAE values for further analysis or model selection.

2.8. best_combined_model(mae_list, feature_names, X_model_list, y_train, X_test)

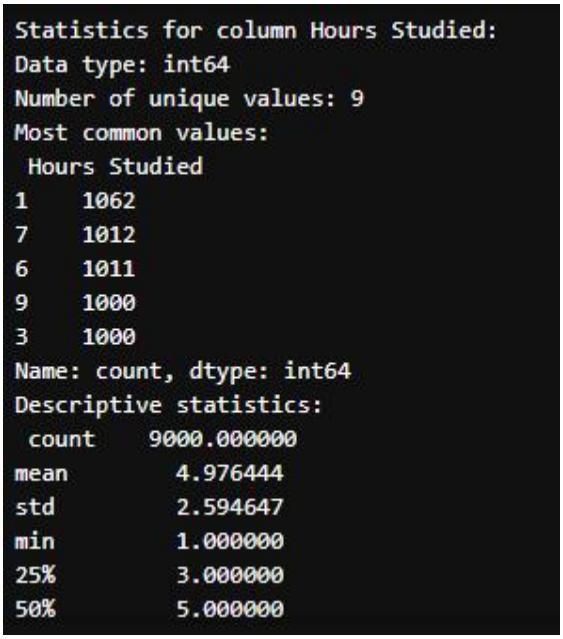
- Purpose: The function identifies the best-performing model from a list of candidates based on their MAE scores, trains it on the provided dataset, and displays its regression formula.
- How it works:
  - Finds the index of the model with the lowest MAE from the provided list.

- Retrieves the training and test feature sets corresponding to the best model.
- Fits a Linear Regression model using the selected feature set and training data.
- Prints the regression equation showing the relationship between the combined feature(s) and the target variable.

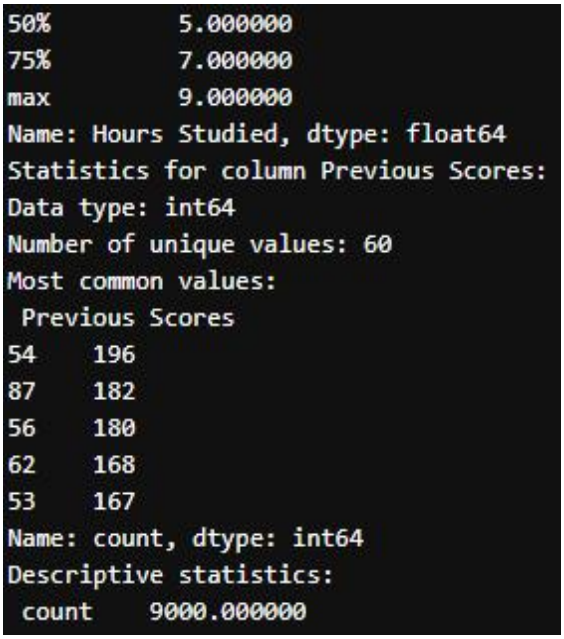
Part III. Result and Conclusion

1. Exploratory Data Analysis

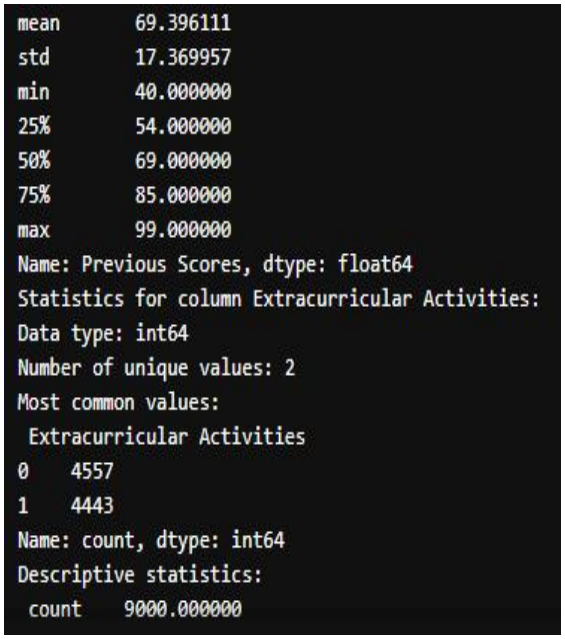
1.1. Statistical analysis of data features



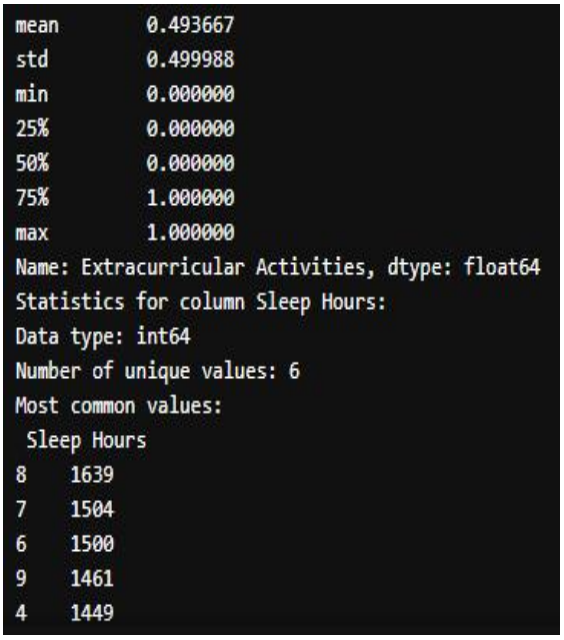
```
Statistics for column Hours Studied:
Data type: int64
Number of unique values: 9
Most common values:
 Hours Studied
1    1062
7    1012
6    1011
9    1000
3    1000
Name: count, dtype: int64
Descriptive statistics:
 count    9000.000000
mean        4.976444
std         2.594647
min         1.000000
25%         3.000000
50%         5.000000
```



```
50%         5.000000
75%         7.000000
max         9.000000
Name: Hours Studied, dtype: float64
Statistics for column Previous Scores:
Data type: int64
Number of unique values: 60
Most common values:
 Previous Scores
54    196
87    182
56    180
62    168
53    167
Name: count, dtype: int64
Descriptive statistics:
 count    9000.000000
```



```
mean       69.396111
std        17.369957
min        40.000000
25%        54.000000
50%        69.000000
75%        85.000000
max        99.000000
Name: Previous Scores, dtype: float64
Statistics for column Extracurricular Activities:
Data type: int64
Number of unique values: 2
Most common values:
 Extracurricular Activities
0    4557
1    4443
Name: count, dtype: int64
Descriptive statistics:
 count    9000.000000
```



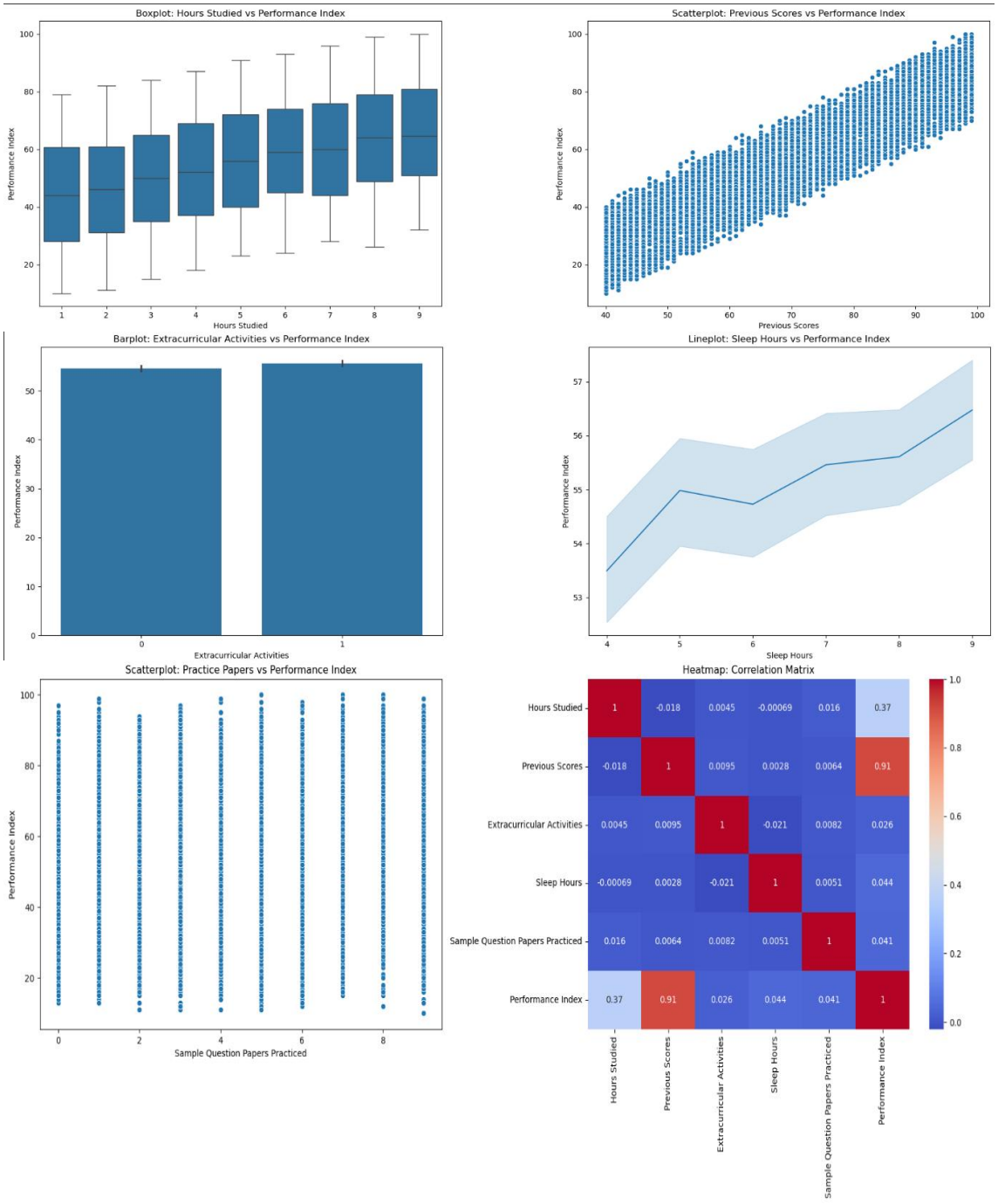
```
mean       0.493667
std        0.499988
min        0.000000
25%        0.000000
50%        0.000000
75%        1.000000
max        1.000000
Name: Extracurricular Activities, dtype: float64
Statistics for column Sleep Hours:
Data type: int64
Number of unique values: 6
Most common values:
 Sleep Hours
8    1639
7    1504
6    1500
9    1461
4    1449
```

```
Name: count, dtype: int64
Descriptive statistics:
 count    9000.000000
mean        6.535556
std         1.695533
min         4.000000
25%         5.000000
50%         7.000000
75%         8.000000
max         9.000000
Name: Sleep Hours, dtype: float64
Statistics for column Sample Question Papers Practiced:
Data type: int64
Number of unique values: 10
Most common values:
 Sample Question Papers Practiced
6    954
9    948
```

```
8    931
3    927
5    920
Name: count, dtype: int64
Descriptive statistics:
 count    9000.000000
mean        4.590889
std         2.864570
min         0.000000
25%         2.000000
50%         5.000000
75%         7.000000
max         9.000000
Name: Sample Question Papers Practiced, dtype: float64
Statistics for column Performance Index:
Data type: float64
Number of unique values: 91
```

```
Most common values:
 Performance Index
67.0    167
56.0    165
40.0    164
61.0    164
74.0    161
Name: count, dtype: int64
Descriptive statistics:
 count    9000.000000
mean       55.136333
std        19.187669
min        10.000000
25%        40.000000
50%        55.000000
75%        70.000000
max       100.000000
Name: Performance Index, dtype: float64
```

1.2. Using charts to analyze/visualize data features



1.3. Conclusion

1.3.1 Hours Studied

- Statistics: Ranges from 1 to 9 hours, mean ~4.98 hours, with most common values being 1, 7, 9, and 3 hours.
- Plot Insight: Performance Index tends to increase as study hours increase, but variation remains wide for all groups. Median scores show a clear upward trend with hours studied.

→ More study hours are generally associated with better performance, though other factors also play a role.

1.3.2. Previous Scores

- Statistics: Range 40 - 99, mean ~69.40, high correlation (0.91) with Performance Index.
- Plot Insight: Strong positive linear relationship - students with higher previous scores almost always achieve higher current performance.

→ Previous academic achievement is the strongest predictor of current performance.

1.3.3. Extracurricular Activities

- Statistics: Binary variable (0 = no, 1 = yes), mean participation rate ~49.37%.
- Plot Insight: Minimal difference in average Performance Index between participants and non-participants.

→ Participation in extracurricular activities does not show a strong direct impact on academic performance.

1.3.4. Sleep Hours

- Statistics: Range 4 - 9 hours, mean ~6.54, relatively small variation.
- Plot Insight: Slight positive trend - students with more sleep hours tend to score marginally higher.

→ Adequate sleep may have a small but positive effect on performance.

1.3.5. Sample Question Papers Practiced

- Statistics: Range 0 - 9, mean ~4.59.
- Plot Insight: No strong linear trend; students with all levels of practice can achieve high or low scores.

→ Practice papers alone are not a strong determinant of performance; their benefit may depend on other study habits.

1.3.6. Performance Index

- Statistics: Range 10 - 100, mean ~55.14, standard deviation ~19.19.
- Insight from Heatmap:
    - Strongest correlation: Previous Scores (0.91)
    - Moderate correlation: Hours Studied (0.37)

- Other factors show very weak correlation.

→ The dataset indicates that Previous Scores and Hours Studied are the most important factors affecting Performance Index. While other features contribute to individual cases, they show weaker direct correlations in this dataset. This suggests that maintaining consistent study habits and leveraging existing knowledge are the most effective strategies for performance improvement.

2. Building a predictive model using Linear Regression

2.1. Using all 5 features

Model Formula:

```
Regression Model Formula:
Performance Index = -33.969 + (2.852 x Hours Studied) + (1.018 x Previous Scores) + (0.604 x Extracurricular Activities) + (0.474 x Sleep Hours) + (0.192 x Sample Question Papers Practiced)
```

Model Performance:

```
MAE on the test set: 1.596
```

Explanation:

- Strongest features:
  - **Previous Scores** (coef ≈ 1.018) → strongest overall impact due to wide range (≈40 - 99).
  - **Hours Studied** (coef ≈ 2.852) → second most influential but narrower range (1–9).
- **Sleep Hours** (≈ 0.474) and **Sample Papers** (≈ 0.192) add small improvements; **Extracurricular** (≈ 0.604, binary) has minimal direct effect.
- Strong linear patterns for **Previous Scores** and **Hours Studied** match their coefficients; weaker EDA signals for other features align with small effects.

Conclusion:

- The model confirms that building on prior knowledge achievement and increasing study hours are the most impactful strategies for boosting performance.
- Secondary factors (extracurricular activities, sleep, practice papers) contribute marginally but can still provide benefits.

- The low MAE demonstrates that the Linear Regression model is well-suited for this dataset.

2.2. Building model using 1 feature to identify the best model

```
No.     Single-feature model                    MAE
1       Hours Studied                           15.449
2       Previous Scores                         6.618
3       Extracurricular Activities              16.196
4       Sleep Hours                             16.187
5       Sample Question Papers Practiced        16.188
The best feature is: Previous Scores with MAE = 6.618
```

2.2.1. Model: Hours Studied

- MAE (Cross-Validation): 15.449
- High prediction error suggests that hours studied alone is a weak predictor of academic performance.
- Possible reasons:
    - Additional study hours may not improve performance.
    - Effective learning matters more than raw study time.
    - Should be combined with other features (e.g., prior knowledge).

2.2.2. Model: Previous Scores

- MAE (Cross-Validation): 6.618
- Best-performing single-feature model, indicating that past academic performance is the strongest predictor of current results.
- The linear regression formula:

```
Model formula: y = 1.011 * (Previous Scores) + (-14.989)
```

- Test MAE (6.544): Confirms model reliability on unseen data.

```
MAE on the test set: 6.544
```

- Reflects cumulative knowledge and learning consistency.

2.2.3. Model: Extracurricular Activities

- MAE (Cross-Validation): 16.196
- Worst-performing feature, indicating no clear linear relationship with academic performance.
- Possible explanation:
    - Some activities (e.g., debate) may enhance critical thinking, while others (e.g., sports) could reduce study time.

2.2.4. Model: Sleep Hours

- MAE (Cross-Validation): 16.187
- Similarly poor performance as extracurriculars, but marginally better (~0.01 MAE difference).
- Possible reason:
    - Too little sleep harms cognition, but too much sleep might cause lethargy.

2.2.5. Model: Sample Question Papers Practiced

- MAE (Cross-Validation): 16.188
- MAE is almost the same as Sleep Hours, so by itself it doesn't predict much.
- Possible reasons:
    - Passive and active problem-solving impact effectiveness.
    - Unseen exam topics could render practice irrelevant.

2.3 Design models and identify the best-performing one

```
Model 1: Using 2 original features
Average MAE (cross-validation): 1.816

Model 2: Multiplication of 2 features
Average MAE (cross-validation): 15.598

Model 3: Squared feature
Average MAE (cross-validation): 6.767
```

2.3.1. Model 1 - Two original features (Previous Scores + Hours Studied)

- Reason: EDA showed the strongest linear signal for Previous Scores and a clear positive effect for Hours Studied (higher medians with more hours). They were weakly correlated with each other, so we expected complementary, additive information.
- Outcome:
    - MAE = 1.816 (best); Test MAE = 1.839 (≈ CV) ⇒ stable generalization.

```
MAE on the test set: 1.839
```

→ Selected as the best custom model because it balances accuracy, stability. This simple 2-feature linear model already captures most of the explainable variance.

```
Best model: Previous Scores + Hours Studied with MAE = 1.816
Model formula: y = 1.018 * (Previous Scores + Hours Studied) + (-29.747)
```

2.3.2. Model 2 - Interaction: Hours Studied × Sleep Hours

- Reason: I tested a synergy idea from domain intuition: sleep might amplify the benefit of study hours (rest × effort).
- Outcome:
  - MAE = 15.598 (poor).
  - Likely causes: both bases are weak individually; dropping main effects removes the dominant linear parts; relation may still be mostly additive.

2.3.3. Model 3 - (Previous Scores)²

- Reason: To test whether performance grows nonlinearly with prior ability (e.g., accelerating gains at higher baselines or saturation at extremes).
- Outcome:
  - MAE = 6.767 (worse than the linear single-feature model with Previous Scores).
  - Likely causes: near-linearity in data; variance inflation from squaring; omission of the linear term.

Part V. References

[1] "DataFrame," [Online]. Available: https://pandas.pydata.org/docs/reference/frame.html#. [Accessed 8 August 2025].

[2] "NumPy reference," 14 December 2024. [Online]. Available: https://numpy.org/doc/2.2/reference/index.html. [Accessed 8 August 2025].

[3] T. M. d. team, "matplotlib.pyplot," [Online]. Available: https://matplotlib.org/3.5.3/api/_as_gen/matplotlib.pyplot.html. [Accessed 8 August 2025].

[4] M. Waskom, "API reference," [Online]. Available: https://seaborn.pydata.org/api.html. [Accessed 9 August 2025].

[5] s.-l. developers, "API Reference," [Online]. Available: https://scikit-learn.org/stable/api/index.html. [Accessed 9 August 2025].

Part V. Acknowledgement

The project was supported by tools such as ChatGPT, DeepSeek, which assisted in program structure overview and in implementing **k-fold Cross Validation** and performing exploratory data analysis (EDA).