ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Khoa Công Nghệ Thông Tin

----------***----------

BÁO CÁO

Project 3 – Linear Regression

Tác giả

Nguyễn Thế Hiển - 22127107

Lớp

22CLC08

Môn học

Toán ứng dụng và thống kê

Công Nghệ Thông Tin

Giảng viên hướng dẫn

Vũ Quốc Hoàng

Nguyễn Văn Quang Huy

Nguyễn Ngọc Toàn

Phan Thị Phương Uyên

Thành phố Hồ Chí Minh – 2024

Mục Lục

1. Giới thiệu và yêu cầu đồ án:

Trong bối cảnh giáo dục hiện đại, việc đánh giá thành tích học tập của sinh viên là một yếu tố quan trọng giúp các nhà giáo dục hiểu rõ hơn về quá trình học tập và phát triển của sinh viên. Một trong những cách tiếp cận phổ biến để đánh giá thành tích này là sử dụng **Academic Student Performance Index** (Chỉ số thành tích học tập của sinh viên). Để hiểu rõ hơn về các yếu tố ảnh hưởng đến chỉ số này, đồ án đặt ra mục tiêu nghiên cứu về mối quan hệ giữa các yếu tố như số giờ học tập/nghiên cứu, hoạt động ngoại khóa, số giờ ngủ, và số bài kiểm tra mẫu đã luyện tập với thành tích học tập của sinh viên.

Bài toán này được đặt trong bối cảnh của một tập dữ liệu bao gồm thông tin chi tiết về các hoạt động học tập và sinh hoạt của 10,000 sinh viên. Mỗi sinh viên được đánh giá dựa trên các yếu tố như số giờ học tập/nghiên cứu, điểm số từ các bài kiểm tra trước đó, sự tham gia vào các hoạt động ngoại khóa, số giờ ngủ trung bình mỗi ngày, và số bài kiểm tra mẫu mà sinh viên đã luyện tập.

Cụ thể, các yếu tố chính được sử dụng trong đồ án bao gồm:

- **Hours Studied**: Tổng số giờ học tập/nghiên cứu của sinh viên.
- **Previous Scores**: Điểm số của sinh viên đạt được trong các bài kiểm tra trước đó.
- **Extracurricular Activities**: Sự tham gia của sinh viên vào các hoạt động ngoại khóa.
- **Sleep Hours**: Số giờ ngủ trung bình mỗi ngày của sinh viên.
- **Sample Question Papers Practiced**: Số bài kiểm tra mẫu mà sinh viên đã luyện tập.

Trong quá trình thực hiện, sinh viên được cung cấp hai tập dữ liệu chính: **train.csv** và **test.csv**. Tập dữ liệu train.csv bao gồm 9,000 mẫu, được sử dụng để huấn luyện mô hình dự đoán chỉ số thành tích học tập. Trong khi đó, tập dữ liệu test.csv bao gồm 1,000 mẫu, sẽ được sử dụng để kiểm tra và đánh giá hiệu quả của mô hình sau khi huấn luyện. Việc chia tách dữ liệu này cho phép mô hình được huấn luyện một cách hiệu quả, đồng thời đánh giá khả năng tổng quát hóa của mô hình trên dữ liệu mới.

Nhiệm vụ của đồ án là tìm hiểu và phân tích mối quan hệ giữa các yếu tố này với chỉ số thành tích học tập của sinh viên, từ đó đưa ra các dự đoán chính xác về thành tích học tập dựa trên mô hình hồi quy tuyến tính.

Đồ án sẽ được chia thành các yêu cầu cụ thể như sau:

1. **Phân tích khám phá dữ liệu (Exploratory Data Analysis - EDA) để hiểu rõ hơn về các đặc trưng của dữ liệu** (Ứng với **Câu 1**): Sinh viên sẽ thực hiện các phân tích thống kê và biểu đồ để quan sát và hiểu rõ hơn về các đặc trưng của dữ liệu, bao gồm việc xem xét sự phân bố của các đặc trưng, mối quan hệ giữa các đặc trưng với nhau, và mối quan hệ giữa các đặc trưng với chỉ số thành tích học tập.
2. **Xây dựng mô hình hồi quy tuyến tính sử dụng tất cả các đặc trưng để dự đoán chỉ số thành tích học tập** (Ứng với **Câu 2a**): Sinh viên sẽ sử dụng toàn bộ 5 đặc trưng đã được cung cấp để xây dựng mô hình hồi quy tuyến tính. Mô hình này sẽ được huấn luyện trên tập dữ liệu train.csv và kiểm tra trên tập dữ liệu test.csv. Kết quả dự đoán sẽ được so sánh với giá trị thực để đánh giá độ chính xác của mô hình.
3. **Tìm kiếm các mô hình hồi quy tối ưu bằng cách sử dụng các kết hợp khác nhau của các đặc trưng và kỹ thuật k-fold Cross Validation**:
    - **Câu 2b**: Xây dựng mô hình hồi quy tuyến tính chỉ sử dụng một đặc trưng duy nhất để dự đoán chỉ số thành tích học tập. Mô hình này sẽ được đánh giá bằng cách sử dụng kỹ thuật k-fold

Cross Validation để tìm ra đặc trưng đơn lẻ nào có ảnh hưởng mạnh nhất đến chỉ số thành tích học tập.

- **Câu 2c**: Sinh viên tự thiết kế các mô hình hồi quy tuyến tính khác nhau bằng cách kết hợp nhiều đặc trưng hoặc biến đổi các đặc trưng đã có. Các mô hình này cũng sẽ được đánh giá bằng kỹ thuật k-fold Cross Validation để tìm ra mô hình tối ưu nhất.

Kết quả của đồ án không chỉ giúp hiểu rõ hơn về các yếu tố ảnh hưởng đến thành tích học tập mà còn cung cấp các mô hình dự đoán có khả năng áp dụng thực tế trong việc đánh giá và hỗ trợ quá trình học tập của sinh viên.

2. Liệt kê và mô tả các hàm đã sử dụng

Trong đồ án này, chúng ta sử dụng các thư viện Python phổ biến để hỗ trợ trong việc xử lý, phân tích dữ liệu, và xây dựng mô hình dự đoán chỉ số thành tích học tập của sinh viên. Dưới đây là danh sách các thư viện đã được sử dụng và lý do sử dụng chúng:

Thư viện Pandas:

Mục đích:

Pandas được sử dụng để đọc, xử lý và thao tác với dữ liệu dạng bảng từ các file CSV. Nó cung cấp các cấu trúc dữ liệu mạnh mẽ và dễ sử dụng để phân tích dữ liệu.

Các hàm sử dụng:

- **pd.read_csv(filepath_or_buffer)**:
  - **Mô tả**: Hàm dùng để đọc dữ liệu từ file CSV và lưu vào một DataFrame.
  - **Input**:
    - **filepath_or_buffer** (str): Đường dẫn đến file CSV hoặc URL.
  - **Output**: DataFrame chứa dữ liệu từ file CSV.
  - **Vai trò trong đồ án**: Hàm này được sử dụng để tải dữ liệu từ các file train.csv và test.csv vào trong các biến train và test, làm cơ sở cho toàn bộ quá trình phân tích và xây dựng mô hình.
- **train.describe()**:
  - **Mô tả**: Hàm này cung cấp các thống kê cơ bản của dữ liệu trong DataFrame như giá trị trung bình, độ lệch chuẩn, min, max, và các phần trăm phân vị.
  - **Input**:
    - Không có tham số đầu vào.
  - **Output**: DataFrame chứa các giá trị thống kê cơ bản.
  - **Vai trò trong đồ án**: Hàm này được sử dụng trong phần phân tích khám phá dữ liệu để hiểu rõ hơn về các đặc trưng của dữ liệu huấn luyện train.
- **train.info()**:
  - **Mô tả**: Hàm này cung cấp thông tin về cấu trúc của DataFrame, bao gồm số lượng hàng, cột, kiểu dữ liệu của từng cột, và số lượng giá trị không phải là null.
  - **Input**:
    - Không có tham số đầu vào.
  - **Output**: Thông tin về cấu trúc của DataFrame.
  - **Vai trò trong đồ án**: Hàm này được sử dụng để kiểm tra thông tin về tập dữ liệu huấn luyện train, giúp xác định các cột nào có dữ liệu bị thiếu hoặc có kiểu dữ liệu không mong muốn.
- **train.isnull()**:
  - **Mô tả**: Hàm này kiểm tra các giá trị null trong DataFrame và trả về một DataFrame chứa các giá trị True hoặc False tương ứng với các vị trí có giá trị null hoặc không.
  - **Input**:

        - Không có tham số đầu vào.
    - **Output**: DataFrame chứa giá trị True hoặc False cho mỗi giá trị trong DataFrame gốc.
    - **Vai trò trong đồ án**: Hàm này được sử dụng để kiểm tra xem có bất kỳ giá trị thiếu nào trong dữ liệu train không.
- **train.sum()**:
    - **Mô tả**: Hàm này tính tổng các giá trị trong các cột của DataFrame.
    - **Input**:
        - Không có tham số đầu vào.
    - **Output**: Series chứa tổng các giá trị của mỗi cột.
    - **Vai trò trong đồ án**: Hàm này được sử dụng kết hợp với isnull() để đếm số lượng giá trị thiếu trong mỗi cột của DataFrame train.
- **train.columns**:
    - **Mô tả**: Thuộc tính này trả về danh sách tên cột của DataFrame.
    - **Input**:
        - Không có tham số đầu vào.
    - **Output**: Index object chứa tên các cột.
    - **Vai trò trong đồ án**: Được sử dụng để lấy tên các đặc trưng trong DataFrame train, đặc biệt trong việc xây dựng các mô hình hồi quy.
- **train.iloc[]**:
    - **Mô tả**: Hàm này cho phép truy xuất dữ liệu từ DataFrame dựa trên chỉ số hàng và cột.
    - **Input**:
        - **rows** (int, list, slice): Chỉ số của các hàng cần truy xuất.
        - **cols** (int, list, slice): Chỉ số của các cột cần truy xuất.
    - **Output**: Một phần của DataFrame (các dòng hoặc cột cần truy xuất).
    - **Vai trò trong đồ án**: Sử dụng để tách các đặc trưng và giá trị mục tiêu từ DataFrame train để xây dựng mô hình hồi quy.
- **train.reset_index(drop=True)**:
    - **Mô tả**: Hàm này đặt lại chỉ số của DataFrame về chỉ số mặc định (từ 0 đến n-1).
    - **Input**:
        - **drop** (bool, mặc định là False): Nếu True, cột chỉ số cũ sẽ bị loại bỏ.
    - **Output**: DataFrame với chỉ số được đặt lại.
    - **Vai trò trong đồ án**: Được sử dụng sau khi xáo trộn dữ liệu để đảm bảo chỉ số của DataFrame là tuần tự và không giữ lại các chỉ số cũ.

Thư viện NumPy:

Mục đích:

NumPy được sử dụng để thực hiện các phép toán số học trên mảng (array) và hỗ trợ việc chuẩn hóa dữ liệu.

Các hàm sử dụng:

- **np.array()**:
    - **Mô tả**: Hàm này tạo mảng từ danh sách hoặc các đối tượng tương tự mảng.
    - **Input**:
        - **object** (list, tuple, array-like): Đối tượng cần chuyển đổi thành mảng.
    - **Output**: NumPy array.
    - **Vai trò trong đồ án**: Được sử dụng để chuyển đổi danh sách các giá trị thành mảng NumPy, hỗ trợ các phép tính toán ma trận trong mô hình hồi quy.
- **np.mean(array)**:
    - **Mô tả**: Hàm này tính giá trị trung bình của các phần tử trong mảng.
    - **Input**:
        - **array** (array-like): Mảng hoặc danh sách các số cần tính trung bình.

- **Output**: Giá trị trung bình của các phần tử trong mảng.
- **Vai trò trong đồ án**: Sử dụng để tính trung bình của danh sách các giá trị MAE trong quá trình Cross Validation.

- **np.random.seed(seed)**:
  - **Mô tả**: Hàm này đặt seed cho trình tạo số ngẫu nhiên của NumPy, giúp đảm bảo các kết quả ngẫu nhiên có thể tái lập.
  - **Input**:
    - **seed** (int): Số nguyên dùng để thiết lập seed cho trình tạo số ngẫu nhiên.
  - **Output**: Không có đầu ra, chỉ thay đổi trạng thái của trình tạo số ngẫu nhiên.
  - **Vai trò trong đồ án**: Được sử dụng để đảm bảo việc xáo trộn dữ liệu trong quá trình Cross Validation có thể được lặp lại trong các lần chạy khác nhau.
- **np.random.permutation(x)**:
  - **Mô tả**: Hàm này xáo trộn ngẫu nhiên các phần tử của một mảng hoặc trả về một mảng mới với các chỉ số được xáo trộn.
  - **Input**:
    - **x** (array-like hoặc int): Mảng hoặc số nguyên đại diện cho số phần tử cần xáo trộn.
  - **Output**: Mảng với các phần tử đã được xáo trộn.
  - **Vai trò trong đồ án**: Sử dụng để xáo trộn dữ liệu huấn luyện trước khi thực hiện các bước Cross Validation.

Thư viện Scikit-learn:

Mục đích:

Scikit-learn là thư viện chính được sử dụng để xây dựng, huấn luyện, và đánh giá mô hình hồi quy tuyến tính, cũng như áp dụng các phương pháp xử lý và tiền xử lý dữ liệu.

Các hàm sử dụng:

- **LinearRegression(fit_intercept=True, normalize=False):**
  - **Mô tả**: Hàm này khởi tạo mô hình hồi quy tuyến tính.
  - **Input:**
    - fit_intercept (bool, mặc định là True): Quyết định có tính toán hệ số chặn (intercept) cho mô hình hay không.
    - normalize (bool, mặc định là False): Nếu True, dữ liệu sẽ được chuẩn hóa trước khi huấn luyện.
  - **Output:** Một đối tượng mô hình hồi quy tuyến tính.
  - **Vai trò trong đồ án**: Được sử dụng để xây dựng các mô hình hồi quy tuyến tính cho từng đặc trưng và cho các đặc trưng kết hợp. Mô hình này sẽ được huấn luyện và đánh giá dựa trên dữ liệu.
- **mean_absolute_error(y_true, y_pred):**
  - **Mô tả:** Hàm này tính toán sai số tuyệt đối trung bình (MAE) giữa giá trị thực (y_true) và giá trị dự đoán (y_pred).
  - **Input:**
    - y_true (array-like): Mảng các giá trị thực tế.
    - y_pred (array-like): Mảng các giá trị dự đoán.
  - **Output:** Giá trị MAE.
  - **Vai trò trong đồ án:** Được sử dụng để đánh giá hiệu suất của các mô hình hồi quy bằng cách tính toán độ lệch trung bình giữa giá trị dự đoán và giá trị thực tế trên tập kiểm tra.
- **KFold(n_splits):**
  - **Mô tả**: Hàm này khởi tạo một đối tượng K-Fold, dùng để chia dữ liệu thành các phần (fold) để thực hiện Cross Validation.
  - **Input:**
    - n_splits (int): Số lượng fold để chia dữ liệu.

- **Output:** Một đối tượng K-Fold.
- **Vai trò trong đồ án:** Sử dụng để chia dữ liệu huấn luyện thành 5 phần để thực hiện k-fold Cross Validation, giúp đánh giá mô hình một cách toàn diện hơn.

- **PolynomialFeatures(degree, include_bias=False):**
  - **Mô tả:** Hàm này tạo ra các đặc trưng đa thức từ các đặc trưng ban đầu, giúp xây dựng các mô hình phi tuyến tính từ dữ liệu tuyến tính.
  - **Input:**
    - degree (int, mặc định là 2): Bậc của đa thức.
    - include_bias (bool, mặc định là False): Nếu True, sẽ bao gồm thêm cột bias (1's).
  - **Output:** Một đối tượng PolynomialFeatures.
  - **Vai trò trong đồ án:** Được sử dụng để tạo các đặc trưng mới từ các đặc trưng gốc bằng cách tính bình phương chúng, nhằm xây dựng các mô hình hồi quy phi tuyến tính.

Thư viện Seaborn:

Mục đích:

Seaborn là thư viện mở rộng của Matplotlib, được sử dụng để tạo ra các biểu đồ có tính thẩm mỹ cao và dễ dàng hơn trong việc trực quan hóa dữ liệu phức tạp.

Các hàm sử dụng:

- **sns.countplot(x, data):**
  - **Mô tả:** Hàm này vẽ biểu đồ cột (countplot) cho dữ liệu phân loại.
  - **Input:**
    - x (str): Tên cột dữ liệu phân loại trong DataFrame.
    - data (DataFrame): DataFrame chứa dữ liệu cần vẽ.
  - **Output:** Biểu đồ cột.
  - **Vai trò trong đồ án:** Được sử dụng để hiển thị số lượng sinh viên tham gia hoạt động ngoại khóa, giúp kiểm tra xem hoạt động ngoại khóa có ảnh hưởng đến hiệu suất học tập hay không.
- **sns.boxplot(data):**
  - **Mô tả:** Hàm này vẽ biểu đồ hộp (boxplot) cho các cột số trong DataFrame.
  - **Input:**
    - data (DataFrame): DataFrame chứa dữ liệu cần vẽ.
  - **Output:** Biểu đồ hộp.
  - **Vai trò trong đồ án:** Sử dụng để hiển thị phân phối của các cột số trong tập dữ liệu huấn luyện, giúp kiểm tra xem có bất kỳ ngoại lệ nào trong dữ liệu hay không.
- **sns.scatterplot(x, y, data):**
  - **Mô tả**: Hàm này vẽ biểu đồ tán xạ (scatterplot) cho hai biến số.
  - **Input**:
    - x (str): Tên cột dữ liệu trục x.
    - y (str): Tên cột dữ liệu trục y.
    - data (DataFrame): DataFrame chứa dữ liệu cần vẽ.
  - **Output**: Biểu đồ tán xạ.
  - **Vai trò trong đồ án:** Được sử dụng để hiển thị mối quan hệ giữa các đặc trưng với chỉ số thành tích học tập (Performance Index), từ đó xác định đặc trưng nào có mối tương quan mạnh nhất.
- **sns.lineplot(data):**
  - **Mô tả:** Hàm này vẽ biểu đồ đường (lineplot) cho dữ liệu chuỗi.
  - **Input:**
    - data (Series hoặc DataFrame): Cột dữ liệu chuỗi hoặc DataFrame cần vẽ.
  - **Output**: Biểu đồ đường.
  - **Vai trò trong đồ án:** Được sử dụng để hiển thị xu hướng luyện tập bài kiểm tra mẫu, kiểm tra xem có sự thay đổi nào về tần suất luyện tập theo thời gian hay không.

- **sns.heatmap(data, annot=True, cmap='coolwarm'):**
  - **Mô tả**: Hàm này vẽ bản đồ nhiệt (heatmap) cho ma trận tương quan giữa các cột số.
  - **Input**:
    - data (DataFrame hoặc numpy array): Ma trận dữ liệu cần vẽ heatmap.
    - annot (bool, mặc định là True): Hiển thị các giá trị số trong các ô của heatmap.
    - cmap (str, mặc định là 'coolwarm'): Bảng màu sử dụng cho heatmap.
  - **Output**: Bản đồ nhiệt.
  - **Vai trò trong đồ án**: Sử dụng để hiển thị mối tương quan giữa các đặc trưng trong tập dữ liệu huấn luyện, giúp tìm ra các đặc trưng có mối quan hệ mạnh nhất.

Thư viện Matplotlib:

Mục đích:

Matplotlib được sử dụng để vẽ các biểu đồ và trực quan hóa dữ liệu, giúp hiểu rõ hơn về các đặc trưng và mối quan hệ giữa chúng.

Các hàm sử dụng:

- **plt.figure(figsize=(width, height)):**
  - **Mô tả:** Hàm này tạo ra một figure mới với kích thước xác định.
  - **Input:**
    - figsize (tuple, mặc định là None): Kích thước của figure, truyền dưới dạng tuple (width, height).
  - **Output**: Một figure mới.
  - **Vai trò trong đồ án:** Được sử dụng trước khi vẽ các biểu đồ để tùy chỉnh kích thước của các figure, đảm bảo chúng được hiển thị rõ ràng và phù hợp với mục đích phân tích.
- **plt.title(label):**
  - **Mô tả:** Hàm này đặt tiêu đề cho biểu đồ.
  - **Input:**
    - label (str): Tiêu đề của biểu đồ.
  - **Output:** Tiêu đề được hiển thị trên biểu đồ.
  - **Vai trò trong đồ án:** Sử dụng để đặt tiêu đề cho các biểu đồ, giúp người đọc hiểu rõ nội dung mà biểu đồ đang thể hiện.
- **plt.show():**
  - **Mô tả:** Hàm này hiển thị tất cả các figure hiện đang có.
  - **Input:**
    - Không có tham số đầu vào.
  - **Output:** Các figure được hiển thị trên màn hình.
  - **Vai trò trong đồ án:** Được sử dụng sau khi vẽ các biểu đồ để hiển thị chúng, giúp người phân tích trực tiếp xem kết quả phân tích dữ liệu.

Thư viện Tabulate:

Mục đích:
Tabulate được sử dụng để tạo ra các bảng kết quả đẹp mắt và dễ đọc trong quá trình hiển thị dữ liệu trên console hoặc trong báo cáo.
Các hàm sử dụng:

- **tabulate(tabular_data, headers, tablefmt='pipe'):**
  - **Mô tả:** Hàm này tạo bảng từ dữ liệu tabular và hiển thị nó dưới dạng bảng dễ đọc.
  - **Input:**
    - tabular_data (list of lists hoặc DataFrame): Dữ liệu dạng bảng cần hiển thị.
    - headers (list hoặc str): Tiêu đề cho các cột trong bảng.
    - tablefmt (str, mặc định là 'pipe'): Định dạng bảng đầu ra (các tùy chọn khác như 'plain', 'grid', 'html', v.v.).
  - **Output:** Chuỗi văn bản đại diện cho bảng.
  - **Vai trò trong đồ án:** Được sử dụng để hiển thị kết quả MAE cho các mô hình khác nhau một cách rõ ràng và dễ nhìn, giúp người đọc dễ dàng so sánh hiệu suất của các mô hình.

3. Ý tưởng thực hiện, giải thích, đánh giá và nhận xét:

3.1) Câu 1:

Ý tưởng thực hiện:

Bước 1: Thống kê mô tả dữ liệu

- **Cách thức hoạt động**:
  - train.describe() tính toán các thống kê mô tả cơ bản như giá trị trung bình, độ lệch chuẩn, giá trị nhỏ nhất, giá trị lớn nhất, và các phần trăm vị (25%, 50%, 75%) cho các cột số trong DataFrame train. Thống kê này cung cấp một cái nhìn tổng quan về phân phối của các đặc trưng.

Bước 2: Kiểm tra thông tin cấu trúc dữ liệu

- **Cách thức hoạt động**:
  - train.info() hiển thị thông tin về cấu trúc của DataFrame train, bao gồm số lượng cột, tên cột, kiểu dữ liệu của mỗi cột, và số lượng giá trị không có trong mỗi cột. Điều này giúp nhận biết liệu có cần xử lý dữ liệu thiếu hay không.

Bước 3: Kiểm tra giá trị thiếu

- **Cách thức hoạt động**:
  - train.isnull().sum() kiểm tra và tính tổng số lượng giá trị thiếu trong mỗi cột của DataFrame train. Đây là bước quan trọng để đảm bảo rằng không có giá trị nào bị thiếu trước khi tiến hành các phân tích tiếp theo.

Bước 4: Trực quan hóa dữ liệu với biểu đồ cột

- **Cách thức hoạt động**:
  - sns.countplot(x='Extracurricular Activities', data=train) tạo biểu đồ cột để hiển thị số lượng sinh viên tham gia và không tham gia vào các hoạt động ngoại khóa. Biểu đồ cột này giúp dễ dàng nhận biết sự phân phối của biến phân loại này.
  - plt.title() và plt.show() được sử dụng để đặt tiêu đề và hiển thị biểu đồ.

Bước 5: Trực quan hóa dữ liệu với biểu đồ hộp

- **Cách thức hoạt động**:
    - plt.figure(figsize=(12, 8)) tạo một cửa sổ hình ảnh (figure) mới với kích thước 12x8 inch. sns.boxplot() tạo biểu đồ hộp cho các cột số trong train. Biểu đồ hộp này giúp phát hiện các giá trị ngoại lệ (outliers) và cung cấp thông tin về phân vị của các đặc trưng. plt.title() và plt.show() được sử dụng để đặt tiêu đề và hiển thị biểu đồ.

Bước 6: Trực quan hóa mối quan hệ giữa các đặc trưng và chỉ số thành tích

- **Cách thức hoạt động**:
    - features = train.columns[:-1] lấy danh sách các tên cột trừ cột cuối cùng (là cột mục tiêu Performance Index). Vòng lặp for lặp qua từng cột trong features, tạo một biểu đồ tán xạ cho mỗi cột với Performance Index. sns.scatterplot() tạo biểu đồ tán xạ cho mỗi quan hệ giữa từng cột và Performance Index. plt.title() và plt.show() được sử dụng để đặt tiêu đề và hiển thị từng biểu đồ.

Bước 7: Trực quan hóa xu hướng luyện tập bài kiểm tra mẫu

- **Cách thức hoạt động**:
    - sns.lineplot() tạo biểu đồ đường cho cột Sample Question Papers Practiced. Biểu đồ đường này giúp theo dõi xu hướng của việc luyện tập qua thời gian hoặc số lượng các giá trị khác nhau. plt.title() và plt.show() được sử dụng để đặt tiêu đề và hiển thị biểu đồ.

Bước 8: Phân tích mối tương quan giữa các đặc trưng

- **Cách thức hoạt động**:
    - train.corr() tính toán ma trận tương quan giữa các cột số trong DataFrame. sns.heatmap() tạo bản đồ nhiệt dựa trên ma trận tương quan này. Các ô vuông trong bản đồ nhiệt sẽ có màu sắc khác nhau tùy thuộc vào mức độ tương quan giữa các cặp đặc trưng. annot=True hiển thị giá trị tương quan trong mỗi ô. plt.title() và plt.show() được sử dụng để đặt tiêu đề và hiển thị bản đồ nhiệt.

Báo cáo và nhận xét

Báo cáo Kết quả Phân tích Khám phá Dữ liệu

1. **Thống kê Cơ bản của Tập Huấn Luyện:**

Thống kê cơ bản của tập huấn luyện:

	Hours Studied	Previous Scores	Extracurricular Activities	\
count	9000.000000	9000.000000	9000.000000	
mean	4.976444	69.396111	0.493667	
std	2.594647	17.369957	0.499988	
min	1.000000	40.000000	0.000000	
25%	3.000000	54.000000	0.000000	
50%	5.000000	69.000000	0.000000	
75%	7.000000	85.000000	1.000000	
max	9.000000	99.000000	1.000000	

	Sleep Hours	Sample Question Papers Practiced	Performance Index
count	9000.000000	9000.000000	9000.000000
mean	6.535556	4.590889	55.136333
std	1.695533	2.864570	19.187669
min	4.000000	0.000000	10.000000
25%	5.000000	2.000000	40.000000
50%	7.000000	5.000000	55.000000
75%	8.000000	7.000000	70.000000
max	9.000000	9.000000	100.000000

- **Giá trị Trung bình (mean):**

- Hours Studied: 4.976 giờ.
- Previous Scores: 69.396 điểm.
- Extracurricular Activities: 0.494 (gần 50% sinh viên tham gia hoạt động ngoại khóa).
- Sleep Hours: 6.536 giờ.
- Sample Question Papers Practiced: 4.591 (trung bình mỗi sinh viên thực hành khoảng 4-5 bài kiểm tra mẫu).
- Performance Index: 55.136 điểm.

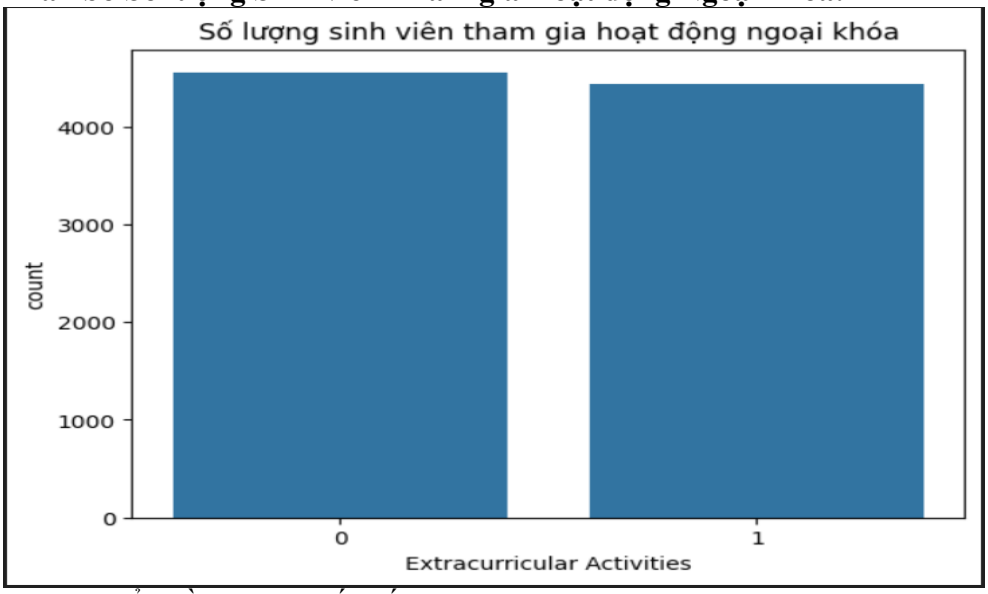
- **Giá trị Tối thiểu (min) và Tối đa (max):**
  - Hours Studied: Từ 1 đến 9 giờ.
  - Previous Scores: Từ 40 đến 99 điểm.
  - Extracurricular Activities: Hoạt động ngoại khóa chia đều giữa 0 và 1.
  - Sleep Hours: Từ 4 đến 9 giờ.
  - Sample Question Papers Practiced: Từ 0 đến 9 bài.
  - Performance Index: Từ 10 đến 100 điểm.
- **Nhận xét:**
  - Previous Scores có độ lệch chuẩn cao nhất (std = 17.37), cho thấy sự khác biệt lớn về điểm số của sinh viên.
  - Performance Index có độ lệch chuẩn khá cao (std = 19.19), điều này chỉ ra sự biến động đáng kể trong chỉ số thành tích học tập của sinh viên.

2. **Thông tin về Dữ liệu Huấn Luyện:**

```
Thông tin về dữ liệu huấn luyện:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9000 entries, 0 to 8999
Data columns (total 6 columns):
...
Sleep Hours                         0
Sample Question Papers Practiced    0
Performance Index                   0
dtype: int64
```

- Dữ liệu gồm 9000 mục nhập (entries) với 6 cột (columns). Không có giá trị null trong bất kỳ cột nào, đảm bảo tính toàn vẹn của dữ liệu.

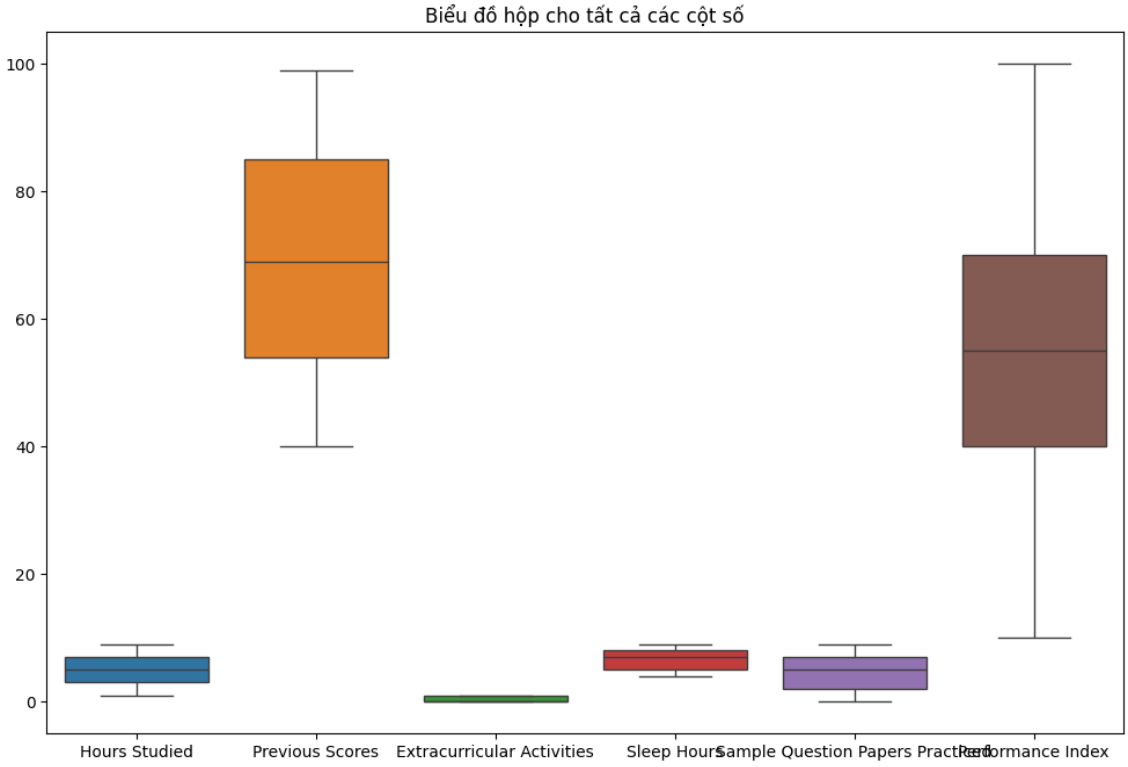
3. **Phân bố Số lượng Sinh viên Tham gia Hoạt động Ngoại khóa:**



- Biểu đồ cột cho thấy số lượng sinh viên tham gia và không tham gia hoạt động ngoại khóa là

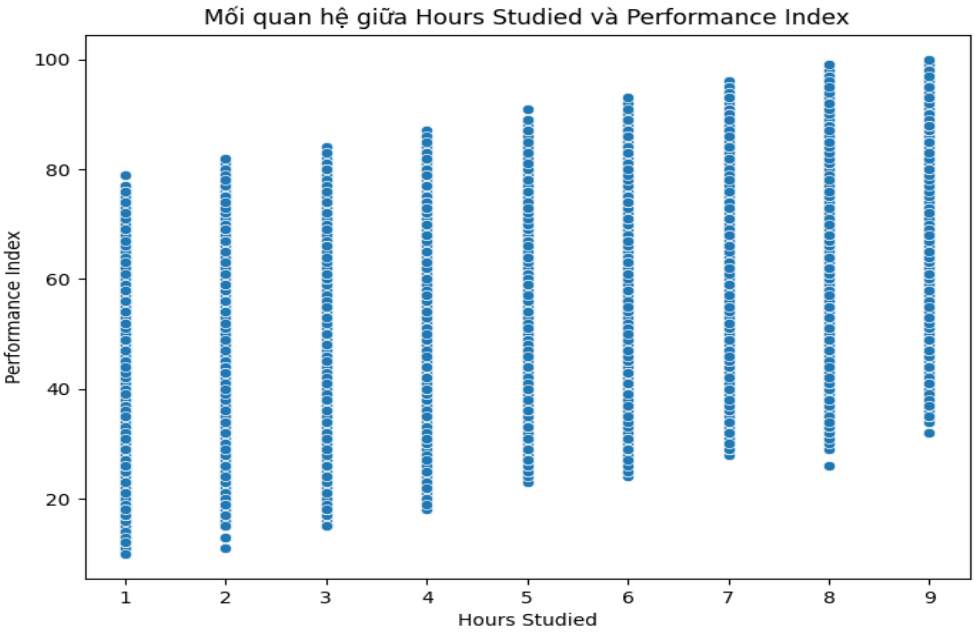
tương đương, với gần 50% sinh viên tham gia hoạt động ngoại khóa.

4. **Phân bố Dữ liệu qua Biểu đồ Hộp (Boxplot):**



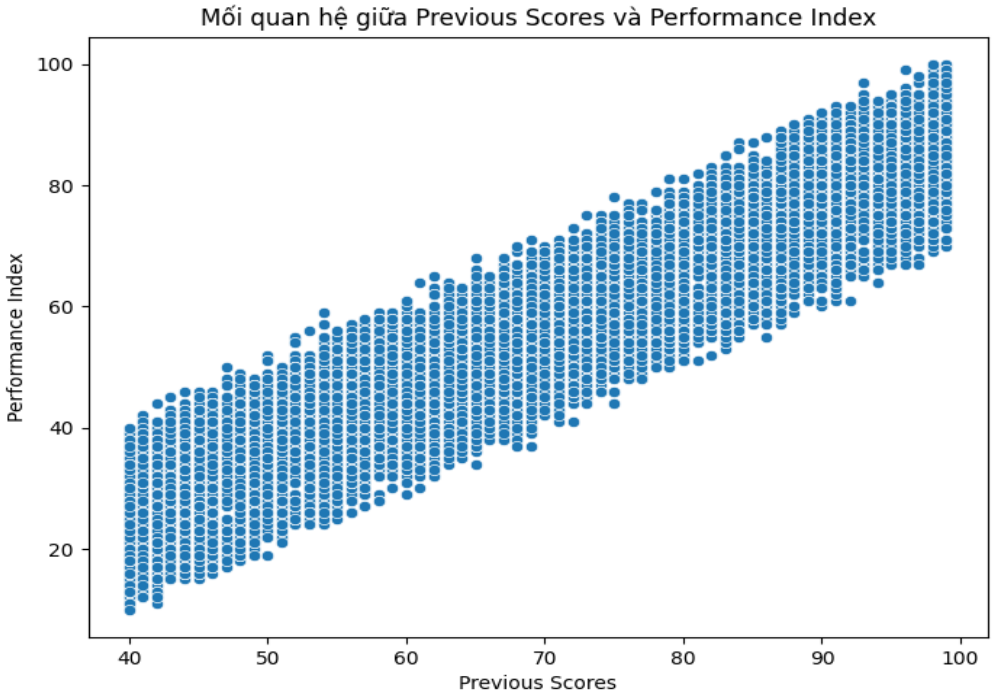
   - Previous Scores có phân bố tương đối rộng, với điểm số dao động từ 40 đến 99, và phần lớn các sinh viên đạt điểm số trung bình quanh mức 69.
   - Hours Studied, Sleep Hours, và Sample Question Papers Practiced có sự phân bố khá đồng đều với một số ngoại lệ.

5. **Mối Quan hệ giữa Các Đặc Trưng với Performance Index:**
   - **Hours Studied và Performance Index:** Dữ liệu cho thấy một xu hướng tăng nhẹ, chứng tỏ việc học tập nhiều giờ hơn có thể dẫn đến chỉ số thành tích cao hơn, mặc dù mức tăng không quá rõ rệt.

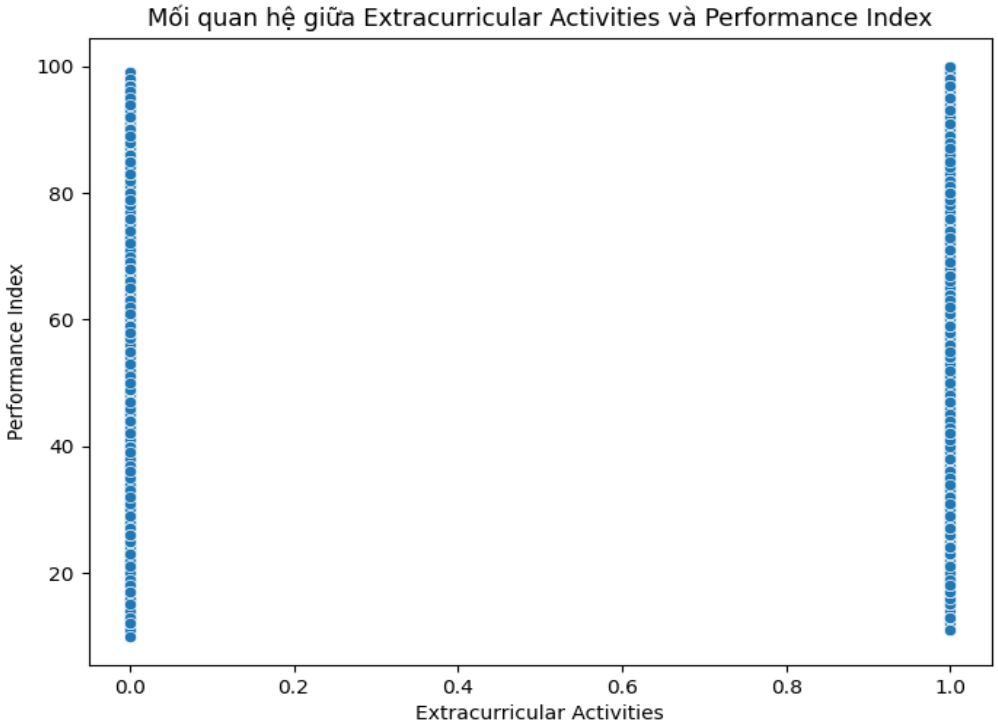


   - **Previous Scores và Performance Index:** Mối quan hệ này là mạnh nhất, với một sự tương

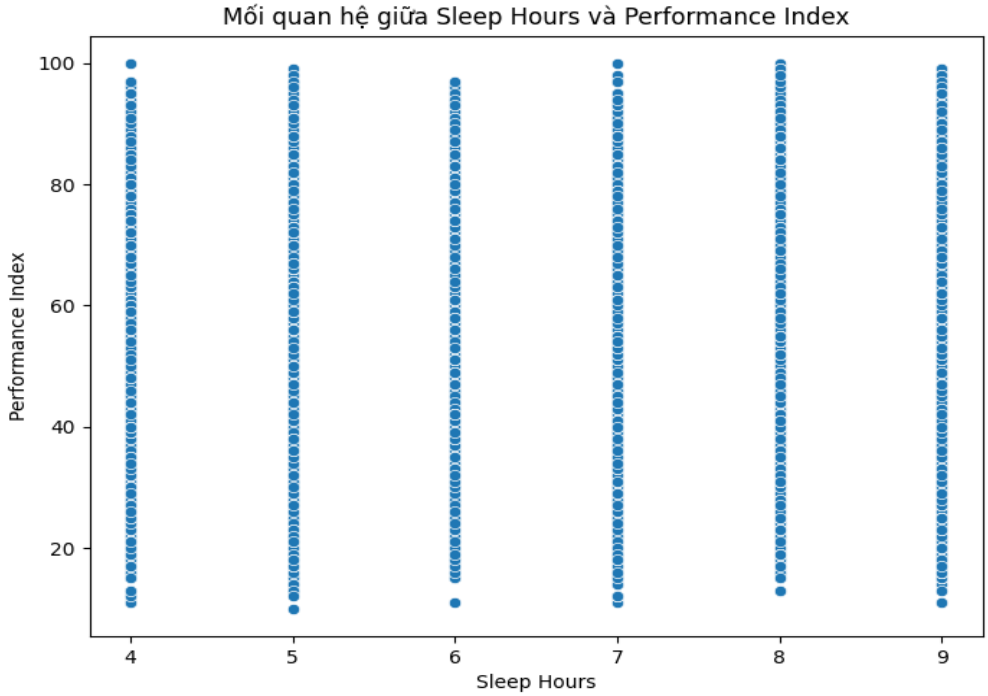
quan tuyến tính rõ ràng. Sinh viên có điểm số trước đó cao hơn thường có chỉ số thành tích học tập cao hơn.



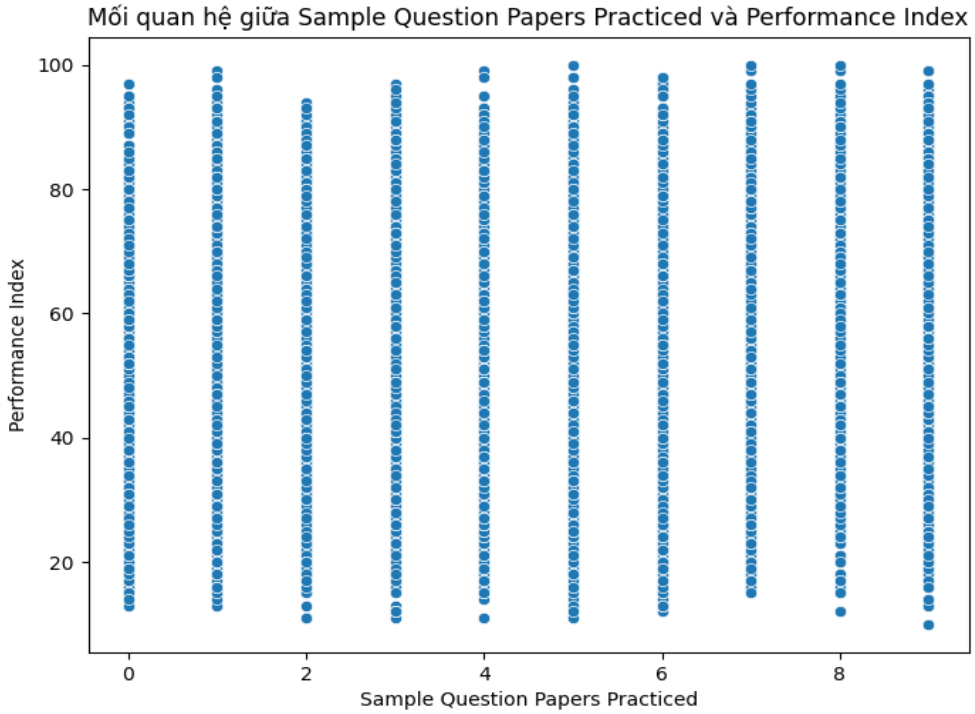
- **Extracurricular Activities và Performance Index:** Không có sự khác biệt lớn giữa sinh viên tham gia hoặc không tham gia hoạt động ngoại khóa, chỉ số thành tích không bị ảnh hưởng nhiều bởi yếu tố này.



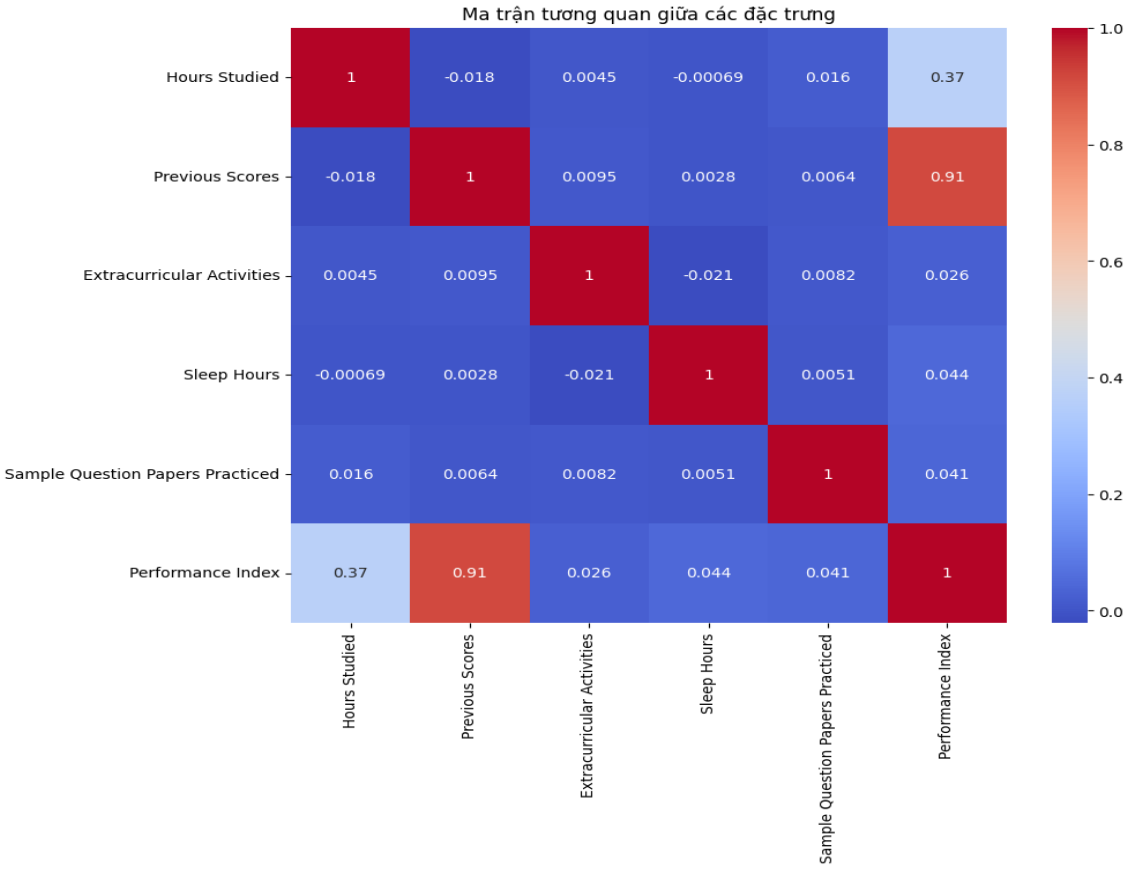
- **Sleep Hours và Performance Index:** Không có sự khác biệt rõ ràng, mặc dù sinh viên ngủ ít giờ hơn có xu hướng có chỉ số thành tích thấp hơn.



- **Sample Question Papers Practiced và Performance Index:** Mối quan hệ không rõ ràng; việc luyện tập nhiều bài kiểm tra mẫu không đảm bảo chỉ số thành tích học tập cao hơn.



6. **Ma trận Tương Quan (Heatmap):**



- Previous Scores có mối tương quan rất mạnh với Performance Index (hệ số tương quan = 0.91), khẳng định rằng điểm số trước đó là yếu tố quan trọng nhất quyết định thành tích học tập.
- Hours Studied cũng có một mối tương quan đáng kể với Performance Index (hệ số tương quan = 0.37), nhưng yếu hơn so với Previous Scores.
- Các đặc trưng khác như Extracurricular Activities, Sleep Hours, và Sample Question Papers Practiced có mối tương quan yếu với Performance Index.

Nhận xét:

- Dựa trên phân tích trên, có thể thấy rằng Previous Scores và Hours Studied là hai đặc trưng có mối quan hệ chặt chẽ nhất với Performance Index. Các đặc trưng khác như Extracurricular Activities, Sleep Hours, và Sample Question Papers Practiced có tác động nhỏ hơn hoặc không rõ rệt đến thành tích học tập.
- Do đó, khi xây dựng các mô hình dự đoán, việc sử dụng Previous Scores và Hours Studied có thể đem lại kết quả tốt hơn so với việc chỉ sử dụng các đặc trưng khác.

3.2) Câu 2a:

Ý tưởng thực hiện:

1. **Khởi tạo và huấn luyện mô hình hồi quy tuyến tính:**

- **Ý tưởng:** Xây dựng một mô hình hồi quy tuyến tính để dự đoán chỉ số thành tích học tập dựa trên các đặc trưng của dữ liệu huấn luyện.
- **Cách thực hiện:**
  - Đoạn mã model = LinearRegression() khởi tạo một mô hình hồi quy tuyến tính từ thư viện scikit-learn.

- Sau đó, model.fit(X_train, y_train) huấn luyện mô hình bằng cách sử dụng dữ liệu huấn luyện (X_train và y_train). Mô hình này học cách ánh xạ các đặc trưng (features) đến chỉ số thành tích (target) thông qua việc tối ưu hóa các hệ số trọng số (coefficients) và hệ số chặn (intercept).

2. **Dự đoán giá trị mục tiêu trên tập kiểm tra:**

- **Ý tưởng:** Áp dụng mô hình đã được huấn luyện để dự đoán chỉ số thành tích học tập trên dữ liệu mới (tập kiểm tra).
- **Cách thực hiện:**
  - Đoạn mã y_pred = model.predict(X_test) sử dụng mô hình đã huấn luyện để dự đoán chỉ số thành tích cho các mẫu trong tập kiểm tra (X_test). Kết quả dự đoán (y_pred) là các giá trị được tính toán từ phương trình hồi quy tuyến tính dựa trên các đặc trưng của từng mẫu trong tập kiểm tra.

3. **Tính toán lỗi trung bình tuyệt đối (MAE):**

- **Ý tưởng:** Đánh giá hiệu suất của mô hình bằng cách so sánh giá trị dự đoán với giá trị thực tế.
- **Cách thực hiện:**
  - Đoạn mã mae = mean_absolute_error(y_test, y_pred) tính toán sai số trung bình tuyệt đối (MAE) giữa giá trị thực tế (y_test) và giá trị dự đoán (y_pred). MAE là một thước đo hiệu suất phổ biến, phản ánh mức độ sai lệch trung bình giữa dự đoán của mô hình và kết quả thực tế.

4. **Trích xuất và hiển thị các hệ số hồi quy:**

- **Ý tưởng:** Hiểu rõ tầm quan trọng của từng đặc trưng trong việc dự đoán chỉ số thành tích học tập.
- **Cách thực hiện:**
  - Đoạn mã model_coefficients = model.coef_ và intercept = model.intercept_ trích xuất các hệ số trọng số và hệ số chặn của mô hình. Các hệ số này thể hiện mức độ ảnh hưởng của từng đặc trưng đến kết quả dự đoán.
  - Sau đó, print("Feature Weights:") và vòng lặp for feature, coef in zip(X_train.columns, model_coefficients): giúp hiển thị tên đặc trưng cùng với hệ số trọng số tương ứng, đồng thời các hệ số này được làm tròn đến 3 chữ số thập phân để dễ dàng trình bày.

5. **Xây dựng và hiển thị công thức hồi quy:**

- **Ý tưởng:** Thể hiện rõ mối quan hệ giữa các đặc trưng và chỉ số thành tích học tập dưới dạng công thức toán học.
- **Cách thực hiện:**
  - Đoạn mã formula = "Student Performance = " + f"{round(intercept, 3)} + " + " + ".join([...]) tạo ra một công thức hồi quy tuyến tính bằng cách kết hợp hệ số chặn và các hệ số trọng số của các đặc trưng.
  - Cuối cùng, print("\nRegression Equation:") và print(formula) hiển thị công thức hồi quy, trong đó mỗi đặc trưng được nhân với hệ số trọng số tương ứng và cộng với hệ số chặn, tạo thành một phương trình tuyến tính mô tả mối quan hệ này.

Báo cáo và nhận xét:

1. **Kết quả trọng số của các đặc trưng:**

```
Feature Weights:
Hours Studied: 2.852
Previous Scores: 1.018
Extracurricular Activities: 0.604
Sleep Hours: 0.474
Sample Question Papers Practiced: 0.192

Regression Equation:
Student Performance = -33.969 + 2.852 * Hours Studied + 1.018 * Previous Scores + 0.604 * Extracurricular Activities + 0.474 * Sleep Hours + 0.192 * Sample Question Papers Practiced
```

2. **Giá trị lỗi trung bình tuyệt đối (MAE) trên tập kiểm tra:**

```
Mean Absolute Error (MAE): 1.596
```

3. Nhận xét:

- **Trọng số của các đặc trưng:** Từ kết quả trên, chúng ta có thể thấy rằng **Hours Studied** có trọng số lớn nhất (2.852), điều này cho thấy rằng số giờ học tập ảnh hưởng mạnh mẽ nhất đến chỉ số thành tích học tập (Student Performance). Tiếp theo là **Previous Scores** với trọng số 1.018, cho thấy điểm số trước đây cũng là một yếu tố quan trọng. Các đặc trưng còn lại như hoạt động ngoại khóa (**Extracurricular Activities**), số giờ ngủ (**Sleep Hours**), và số bài kiểm tra mẫu đã luyện tập (**Sample Question Papers Practiced**) có trọng số thấp hơn, nhưng vẫn có tác động đến chỉ số thành tích.
- **Phương trình hồi quy:** Phương trình hồi quy thể hiện mối quan hệ tuyến tính giữa các đặc trưng và chỉ số thành tích học tập. Giá trị **intercept** (-33.969) cho thấy giá trị dự đoán của chỉ số thành tích học tập khi tất cả các đặc trưng bằng 0.
- **Giá trị MAE:** Với giá trị **MAE** là 1.596, mô hình này có khả năng dự đoán khá chính xác chỉ số thành tích học tập trên tập kiểm tra. Giá trị MAE càng thấp thì mô hình càng chính xác, và giá trị 1.596 là khá tốt trong bối cảnh này.
- **Kết luận:** Mô hình hồi quy tuyến tính đã tìm ra mối quan hệ giữa các đặc trưng và chỉ số thành tích học tập. **Hours Studied** và **Previous Scores** là hai yếu tố quan trọng nhất ảnh hưởng đến chỉ số này, và mô hình đã chứng minh khả năng dự đoán chính xác với lỗi dự đoán thấp. Tuy nhiên, cần lưu ý rằng mô hình này chỉ có thể dự đoán chính xác trong phạm vi dữ liệu huấn luyện và kiểm tra đã sử dụng, và có thể không chính xác khi áp dụng trên các dữ liệu khác với phạm vi đặc trưng này.

3.3) Câu 2b:

Ý tưởng thực hiện:

Mục tiêu của câu 2b là tìm ra đặc trưng đơn lẻ tốt nhất trong số 5 đặc trưng đã cho để dự đoán chính xác chỉ số thành tích học tập (**Performance Index**). Để đạt được điều này, thì sẽ sử dụng phương pháp k-fold Cross Validation để đánh giá từng đặc trưng riêng lẻ, sau đó chọn đặc trưng có chỉ số lỗi trung bình tuyệt đối (**Mean Absolute Error** - MAE) thấp nhất. Mô hình sẽ được huấn luyện lại với đặc trưng này trên toàn bộ tập huấn luyện, và sau đó được kiểm tra trên tập kiểm tra độc lập để đánh giá hiệu suất thực tế.

1. **Xáo trộn dữ liệu (Shuffle data):**
    - **Mục đích:** Đảm bảo rằng dữ liệu huấn luyện được phân phối ngẫu nhiên trước khi thực hiện k-fold Cross Validation để tránh hiện tượng mô hình bị overfitting.
    - **Cách thực hiện:** Sử dụng np.random.seed(42) để đảm bảo kết quả xáo trộn có thể tái lập.

np.random.permutation(len(train)) tạo ra một thứ tự ngẫu nhiên cho các chỉ số của tập dữ liệu. Sau đó, train.iloc[shuffled_indices].reset_index(drop=True) sắp xếp lại dữ liệu huấn luyện theo thứ tự mới này.

2. **Thiết lập Cross Validation:**
   - **Mục đích:** Đánh giá hiệu suất của mô hình hồi quy đơn biến cho từng đặc trưng bằng cách chia tập dữ liệu thành 5 phần (folds), với mỗi phần lần lượt được sử dụng làm tập kiểm tra trong khi các phần còn lại làm tập huấn luyện.
   - **Cách thực hiện:** Sử dụng KFold(n_splits=5) từ scikit-learn để chia dữ liệu thành 5 folds. Trong mỗi vòng lặp, một fold sẽ được sử dụng làm tập kiểm tra, trong khi 4 fold còn lại làm tập huấn luyện.
3. **Đánh giá từng đặc trưng (Evaluate each feature):**
   - **Mục đích:** Xác định đặc trưng đơn lẻ nào có khả năng dự đoán chính xác nhất (dựa trên giá trị MAE thấp nhất).
   - **Cách thực hiện:**
     - Với mỗi đặc trưng trong danh sách features, model.fit(X_train_fold, y_train_fold) được sử dụng để huấn luyện mô hình hồi quy đơn biến. Sau đó, model.predict(X_val_fold) dự đoán kết quả trên tập kiểm tra của từng fold.
     - mean_absolute_error(y_val_fold, y_pred_fold) được sử dụng để tính toán MAE cho mỗi fold. Các giá trị MAE sau đó được tính trung bình để đưa ra một giá trị MAE duy nhất cho từng đặc trưng.
4. **Chọn đặc trưng tốt nhất (Select the best feature):**
   - **Mục đích:** Chọn ra đặc trưng có giá trị MAE trung bình thấp nhất, từ đó xây dựng mô hình dựa trên đặc trưng này.
   - **Cách thực hiện:** Đặc trưng tốt nhất được xác định bằng cách so sánh giá trị MAE trung bình của các đặc trưng và chọn đặc trưng có giá trị thấp nhất. Mô hình hồi quy sau đó được huấn luyện lại trên toàn bộ tập huấn luyện với đặc trưng này. Kết quả huấn luyện bao gồm intercept và hệ số hồi quy của đặc trưng tốt nhất, được in ra dưới dạng công thức hồi quy.
5. **Huấn luyện lại mô hình và đánh giá (Retrain the model and evaluate):**
   - **Mục đích:** Huấn luyện lại mô hình hồi quy đơn biến với đặc trưng tốt nhất trên toàn bộ tập huấn luyện và đánh giá nó trên tập kiểm tra.
   - **Cách thực hiện:**
     - Mô hình được huấn luyện lại với toàn bộ dữ liệu huấn luyện sử dụng best_model.fit(X_train_best, y_train_best).
     - Sau đó, mô hình được kiểm tra trên tập kiểm tra độc lập. mean_absolute_error(y_test_best, y_pred_test_best) tính toán giá trị MAE để đánh giá hiệu suất thực tế của mô hình trên dữ liệu mới.
     - Kết quả MAE cuối cùng trên tập kiểm tra được in ra để đánh giá độ chính xác của mô hình.

Báo cáo và nhận xét chi tiết:

1. **Kết quả Cross-Validation với các đặc trưng**

STT	Mô hình với 1 đặc trưng	MAE
1	Hours Studied	15.4493
2	Previous Scores	6.6187
3	Extracurricular Activities	16.1942
4	Sleep Hours	16.1863
5	Sample Question Papers Practiced	16.1837

Dựa trên bảng kết quả, mô hình hồi quy tuyến tính đơn giản đã được huấn luyện với từng đặc trưng

riêng biệt. Kết quả cho thấy đặc trưng "Previous Scores" (Điểm số trước đó) có MAE thấp nhất (6.6187), cho thấy đây là đặc trưng có ảnh hưởng lớn nhất đến chỉ số thành tích học tập (Performance Index) trong số các đặc trưng đã kiểm tra. Những đặc trưng khác như "Hours Studied", "Extracurricular Activities", "Sleep Hours", và "Sample Question Papers Practiced" có MAE cao hơn, nghĩa là chúng không dự đoán chính xác chỉ số thành tích học tập bằng đặc trưng "Previous Scores".

Giải thích: Đặc trưng "Previous Scores" phản ánh trực tiếp mức độ học tập của sinh viên trong quá khứ và do đó có khả năng dự đoán mạnh mẽ về thành tích học tập hiện tại. Các đặc trưng khác, mặc dù cũng có ảnh hưởng đến thành tích học tập, nhưng có thể không trực tiếp hoặc có thể bị ảnh hưởng bởi nhiều yếu tố khác, dẫn đến MAE cao hơn.

2. Mô hình hồi quy tốt nhất

```
Mô hình hồi quy tốt nhất dựa trên đặc trưng: Previous Scores
Performance Index = -14.99 + 1.01 * Previous Scores
```

Sau khi xác định "Previous Scores" là đặc trưng tốt nhất, mô hình hồi quy tuyến tính đã được huấn luyện lại trên toàn bộ tập huấn luyện sử dụng đặc trưng này. Công thức hồi quy được xác định là:

$$\text{Performance Index} = -14.99 + 1.01 \times \text{Previous Scores}$$

Điều này có nghĩa là với mỗi điểm số tăng thêm trong "Previous Scores", chỉ số thành tích học tập sẽ tăng khoảng 1.01 đơn vị. Giá trị intercept là -14.99 cho thấy rằng nếu "Previous Scores" bằng 0, chỉ số thành tích học tập sẽ bắt đầu từ -14.99, tuy nhiên, trong thực tế, "Previous Scores" thường không bằng 0.

Giả thuyết: Mô hình này hợp lý vì "Previous Scores" là chỉ số đo lường trực tiếp khả năng học tập trước đó của sinh viên, nên không ngạc nhiên khi nó là đặc trưng dự đoán tốt nhất. Việc sử dụng "Previous Scores" làm biến đầu vào chính có thể giúp mô hình dự đoán chính xác hơn so với việc sử dụng các đặc trưng khác.

3. Kết quả trên tập kiểm tra

```
Kết quả trên tập kiểm tra với mô hình tốt nhất: MAE = 6.5443
```

Khi áp dụng mô hình tốt nhất trên tập kiểm tra, MAE đạt được là 6.5443. Đây là một kết quả tốt, cho thấy mô hình có khả năng dự đoán khá chính xác chỉ số thành tích học tập trên tập dữ liệu kiểm tra.

Nhận xét: MAE trên tập kiểm tra gần bằng với MAE trong quá trình Cross-Validation (6.6187), điều này cho thấy mô hình không bị overfitting và có thể tổng quát hóa tốt cho dữ liệu mới. Việc sử dụng "Previous Scores" làm đặc trưng dự đoán đã giúp mô hình đạt được hiệu suất tốt nhất, hợp lý với giả thuyết ban đầu rằng đặc trưng này có ảnh hưởng lớn nhất đến chỉ số thành tích học tập.

3.4) Câu 2c:

Ý tưởng thực hiện:

Mục tiêu của bài toán là tìm ra mô hình hồi quy tuyến tính tốt nhất dựa trên các đặc trưng đã chọn hoặc tạo mới, sau đó kiểm tra hiệu suất của mô hình này trên tập kiểm tra. Để đạt được điều này, các bước chính bao gồm xáo trộn dữ liệu, thiết lập K-Fold Cross Validation, thiết kế và đánh giá các mô hình khác nhau, chọn mô hình tốt nhất và kiểm tra nó trên tập kiểm tra.

1. **Xáo trộn dữ liệu**:
    - **Ý tưởng**: Để tránh việc mô hình bị thiên vị do thứ tự của dữ liệu, chúng ta xáo trộn dữ liệu

trước khi áp dụng K-Fold Cross Validation.

- **Cách thực hiện**:
  - Dùng np.random.seed(42) để đảm bảo kết quả xáo trộn có thể tái lập.
  - Sử dụng np.random.permutation(len(train)) để tạo ra một chuỗi các chỉ số ngẫu nhiên, sau đó áp dụng chuỗi này để xáo trộn dữ liệu bằng train.iloc[shuffled_indices].reset_index(drop=True).

2. **Thiết lập K-Fold Cross Validation**:
   - **Ý tưởng**: Sử dụng K-Fold Cross Validation để chia dữ liệu thành nhiều phần và đánh giá mô hình trên các phần này, nhằm đảm bảo mô hình không phụ thuộc vào một phần cụ thể của dữ liệu.
   - **Cách thực hiện**:
     - Thiết lập K-Fold với 5 lần gấp bằng KFold(n_splits=5). Điều này chia dữ liệu thành 5 phần, và mỗi phần lần lượt được sử dụng làm tập kiểm tra trong khi các phần còn lại làm tập huấn luyện.
3. **Thiết kế các mô hình**:
   - **Ý tưởng**: Tạo ra ít nhất 3 mô hình khác nhau với các đặc trưng khác nhau, bao gồm các đặc trưng đã chọn và các đặc trưng tạo mới, để đánh giá xem mô hình nào hoạt động tốt nhất.
   - **Cách thực hiện**:
     - Các mô hình được định nghĩa dưới dạng từ điển models, với mỗi mô hình là một tập hợp các đặc trưng cụ thể:
       - Mô hình sử dụng 2 đặc trưng: Hours Studied và Previous Scores.
       - Mô hình sử dụng 3 đặc trưng: Hours Studied, Previous Scores, và Sleep Hours.
       - Mô hình sử dụng đặc trưng tạo mới bằng cách bình phương Previous Scores và Sleep Hours.
       - Mô hình sử dụng nhân của 2 đặc trưng Hours Studied và Previous Scores.
4. **Tìm kiếm mô hình tốt nhất**:
   - **Ý tưởng**: Áp dụng K-Fold Cross Validation cho từng mô hình để tính toán MAE, từ đó chọn mô hình có MAE thấp nhất.
   - **Cách thực hiện**:
     - Với mỗi mô hình, chạy qua các lần gấp trong K-Fold, huấn luyện mô hình trên tập huấn luyện và kiểm tra nó trên tập kiểm tra.
     - Nếu mô hình cần tạo mới đặc trưng, sử dụng PolynomialFeatures để tạo đặc trưng đa thức, hoặc nhân các đặc trưng với nhau để tạo đặc trưng tương tác.
     - Tính toán MAE cho từng lần gấp và lấy MAE trung bình để đánh giá mô hình.
5. **Huấn luyện và kiểm tra mô hình tốt nhất**:
   - **Ý tưởng**: Sau khi chọn được mô hình tốt nhất, huấn luyện lại mô hình này trên toàn bộ tập dữ liệu huấn luyện và kiểm tra nó trên tập kiểm tra.
   - **Cách thực hiện**:
     - Huấn luyện mô hình tốt nhất với toàn bộ tập huấn luyện, sau đó kiểm tra nó trên tập kiểm tra (test.csv) để tính toán MAE cuối cùng.
     - Công thức hồi quy của mô hình tốt nhất cũng được in ra, cho thấy các trọng số và intercept của mô hình.

Báo cáo và nhận xét chi tiết

1. **Kết quả Cross-Validation với các mô hình:**

```
Kết quả MAE cho từng mô hình:
|   STT | Mô hình                                                          |     MAE |
|------:|:-----------------------------------------------------------------|--------:|
|     1 | Sử dụng 2 đặc trưng (Hours Studied, Previous Scores)             |  1.8159 |
|     2 | Sử dụng 3 đặc trưng (Hours Studied, Previous Scores, Sleep Hours) |  1.7015 |
|     3 | Sử dụng đặc trưng tạo mới (Previous Scores^2 + Sleep Hours^2)    |  6.6005 |
|     4 | Sử dụng nhân của 2 đặc trưng (Hours Studied * Previous Scores)   | 11.0818 |
```

Dựa trên bảng kết quả, mô hình hồi quy tuyến tính đã được huấn luyện với nhiều tổ hợp đặc trưng khác nhau. Kết quả cho thấy mô hình sử dụng tổ hợp 3 đặc trưng **"Hours Studied"**, **"Previous Scores"**, và **"Sleep Hours"** có MAE thấp nhất (1.7015), chứng tỏ rằng sự kết hợp này có ảnh hưởng lớn nhất đến chỉ số thành tích học tập (Performance Index) so với các mô hình khác. Trong khi đó, các mô hình khác sử dụng đặc trưng tạo mới hoặc nhân các đặc trưng lại với nhau có MAE cao hơn, cho thấy rằng các biến này không dự đoán chính xác chỉ số thành tích học tập bằng cách kết hợp 3 đặc trưng cơ bản.

Giải thích:

- Mô hình tốt nhất có MAE thấp nhất vì nó kết hợp các yếu tố quan trọng như số giờ học tập (Hours Studied), điểm số trước đó (Previous Scores), và số giờ ngủ (Sleep Hours). Những yếu tố này phản ánh trực tiếp nỗ lực học tập, nền tảng kiến thức và sức khỏe của sinh viên. Việc thêm một đặc trưng sức khỏe (Sleep Hours) đã giúp mô hình cải thiện đáng kể khả năng dự đoán. Những mô hình khác, mặc dù tạo ra các đặc trưng mới hoặc sử dụng phép nhân của các đặc trưng, có thể làm tăng sự phức tạp mà không mang lại nhiều lợi ích, dẫn đến MAE cao hơn.

2. **Mô hình hồi quy tốt nhất:**

```
Công thức hồi quy của mô hình tốt nhất:
Student Performance = -32.82 + 2.856 * Hours Studied + 1.018 * Previous Scores + 0.472 * Sleep Hours
```

Sau khi xác định rằng mô hình kết hợp 3 đặc trưng **"Hours Studied"**, **"Previous Scores"**, và **"Sleep Hours"** là tốt nhất, mô hình hồi quy tuyến tính đã được huấn luyện lại trên toàn bộ tập huấn luyện sử dụng 3 đặc trưng này. Công thức hồi quy được xác định là:

$$\text{Student Performance} = -32.82 + 2.856 \times \text{Hours Studied} + 1.018 \times \text{Previous Scores} + 0.472 \times \text{Sleep Hours}$$

Điều này có nghĩa là:

- Với mỗi giờ học tập tăng thêm, chỉ số thành tích học tập sẽ tăng khoảng 2.856 đơn vị.
- Với mỗi điểm số tăng thêm trong "Previous Scores", chỉ số thành tích học tập sẽ tăng khoảng 1.018 đơn vị.
- Với mỗi giờ ngủ tăng thêm, chỉ số thành tích học tập sẽ tăng khoảng 0.472 đơn vị.

Giả thuyết:

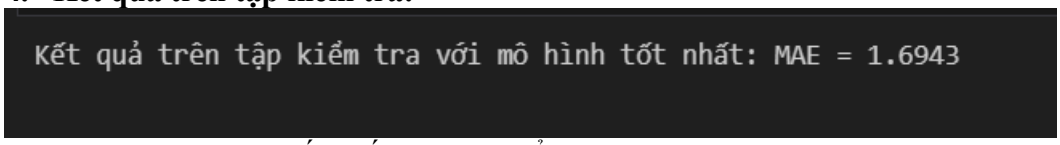
- Mô hình này hợp lý vì nó bao gồm cả yếu tố học tập (Hours Studied), nền tảng kiến thức (Previous Scores), và sức khỏe (Sleep Hours) - những yếu tố có ảnh hưởng trực tiếp đến thành tích học tập của sinh viên. Việc kết hợp các đặc trưng này giúp mô hình dự đoán chính xác hơn và phản ánh được thực tế rằng học tập, sức khỏe và kiến thức đều đóng vai trò quan trọng trong việc đạt được thành tích cao.

3. **Nhận xét tổng quan về các mô hình khác:**

- **Mô hình sử dụng 2 đặc trưng (Hours Studied, Previous Scores)**: Mô hình này cũng có kết quả tốt, với MAE là 1.8159. Tuy nhiên, việc thêm đặc trưng Sleep Hours đã cải thiện đáng kể độ chính xác của mô hình, thể hiện qua việc giảm MAE từ 1.8159 xuống 1.7015. Điều này cho thấy rằng số giờ ngủ cũng đóng vai trò quan trọng trong việc xác định thành tích học tập, mặc dù có thể không mạnh mẽ như hai yếu tố còn lại.
- **Mô hình sử dụng đặc trưng tạo mới (Previous Scores^2 + Sleep Hours^2)** và **mô hình sử dụng nhân của 2 đặc trưng (Hours Studied * Previous Scores)** có MAE cao hơn nhiều. Điều này có thể

được lý giải do các đặc trưng mới tạo ra có thể không phù hợp hoặc quá phức tạp đối với mô hình hồi quy tuyến tính đơn giản, dẫn đến kết quả dự đoán kém chính xác hơn.

4. Kết quả trên tập kiểm tra:



Khi áp dụng mô hình tốt nhất trên tập kiểm tra, MAE đạt được là **1.6943**. Đây là một kết quả rất tốt, cho thấy mô hình có khả năng dự đoán rất chính xác chỉ số thành tích học tập trên tập dữ liệu kiểm tra.

Nhận xét:

- MAE trên tập kiểm tra rất gần với MAE trong quá trình Cross-Validation (1.7015), điều này chứng tỏ mô hình không bị overfitting và có khả năng tổng quát hóa tốt cho dữ liệu mới. Việc sử dụng kết hợp cả 3 đặc trưng "Hours Studied", "Previous Scores", và "Sleep Hours" đã giúp mô hình đạt được hiệu suất tốt nhất, hợp lý với giả thuyết rằng cả ba yếu tố này đều có ảnh hưởng lớn đến chỉ số thành tích học tập của sinh viên.

4. Tài liệu tham khảo

[1] Student Performance [Trực tuyến]. Available: https://www.kaggle.com/datasets/nikhil7280/student-performance-multiple-linear-regression [Đã truy cập 12 8 2024].

[2] J. A. C. a. c. F. Lundh, "Pillow Documentation," Pillow, 2010-2014. [Trực tuyến]. Available: https://pillow.readthedocs.io/en/stable/. [Đã truy cập 15 6 2024].

[3] D. D. E. F. M. D. a. t. M. d. t. John Hunter, "Using Matplotlib," The Matplotlib development team, 2012–2024. [Trực tuyến]. Available: https://matplotlib.org/stable/users/index. [Đã truy cập 12 8 2024].

[4] N. Developers, "NumPy reference," 16 9 2023. [Online]. Available: https://numpy.org/doc/stable/reference/index.html. [Đã truy cập 12 8 2024].

[5] KFold -scikit learn 1.5.1 Documentation [Trực tuyến]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html. [Đã truy cập 12 8 2024].

[5] Pandas Documentation 2.2.2 [Trực tuyến]. Available: https://pandas.pydata.org/docs/user_guide/dsintro.html#dataframe [Đã truy cập 12 8 2024].