ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

Báo cáo đồ án

Đề tài: Linear Regression

Môn học: Toán ứng dụng và thống kê cho Công nghệ thông tin

Sinh viên thực hiện:
22127098 - Đinh Vũ Gia Hân

Giáo viên hướng dẫn:
Cô Phan Thị Phương Uyên
Thầy Nguyễn Ngọc Toàn
Thầy Nguyễn Văn Quang Huy
Thầy Vũ Quốc Hoàng

Ngày 16 tháng 8 năm 2024

Mục lục

Danh sách bảng

Danh sách hình vẽ

1 Lời cảm ơn

Để hoàn thành được đồ án này, em xin chân thành cảm ơn quý thầy, cô đã tạo cơ hội cho em được học tập, nghiên cứu và tích lũy kiến thức để thực hiện đồ án này. Trên hết, em xin được gửi lời cảm ơn chân thành nhất đến giảng viên Phan Thị Phương Uyên đã tận tình chỉ dẫn và đưa ra nhiều lời khuyên bổ ích giúp em hoàn thành đồ án này một cách tốt nhất.

Em cũng xin gửi lời tri ân tới những nhà phát triển của các thư viện Numpy, Pandas, Matplotlib và Seaborn. Nhờ có tài liệu hướng dẫn chi tiết trên trang web của họ, em đã dễ dàng làm quen và ứng dụng các hàm cần thiết cho đồ án này.

Đặc biệt, em xin gửi lời cảm ơn chân thành đến ChatGPT, mô hình ngôn ngữ lớn, vì đã hỗ trợ em trong việc thu thập và tóm tắt thông tin quan trọng liên quan đến việc thực hiện đồ án Linear Regression.

Do kiến thức của bản thân còn nhiều hạn chế và thiếu kinh nghiệm thực tiễn nên nội dung báo cáo khó tránh khỏi những thiếu sót. Em rất mong nhận được những lời góp ý thêm từ quý thầy cô.

Trân trọng.

2 Giới Thiệu

Linear Regression là một lĩnh vực quan trọng trong công nghệ thông tin và khoa học máy tính, với ứng dụng rộng rãi trong nhiều lĩnh vực như y tế, an ninh, giao thông, và truyền thông. Đồ án này nhằm nghiên cứu và ứng dụng các kỹ thuật xử lý thông tin để giải quyết một số vấn đề cụ thể trong thực tế.

Trong đồ án này, em sẽ tập trung vào việc phân tích và xử lý các dữ liệu bằng cách sử dụng các công cụ và thư viện mạnh mẽ như Numpy, Pandas, Matplotlib và Seaborn. Numpy sẽ hỗ trợ trong việc xử lý dữ liệu dưới dạng mảng đa chiều và Pandas sẽ cung cấp các phương thức để xử lý và thao tác trên các dữ liệu một cách hiệu quả. Matplotlib và Seaborn hỗ trợ việc hiển thị hình ảnh và vẽ biểu đồ.

Đồ án này không chỉ giúp em củng cố kiến thức về lý thuyết xử lý dữ liệu mà còn trang bị cho em những kỹ năng thực hành cần thiết để giải quyết các bài toán thực tế trong lĩnh vực này. Qua đó, em hy vọng sẽ đóng góp một phần nhỏ vào sự phát triển của công nghệ xử lý dữ liệu và mở ra những hướng nghiên cứu mới trong tương lai.

Bố cục các phần còn lại được tổ chức như sau, Phần 3 là các thông tin của sinh viên thực hiện báo cáo. Phần 4 với các ý tưởng và mô tả về thuật toán và các hàm tiện ích. Ở phần 5, ta sẽ phân tích khám phá dữ liệu đầu vào và cuối cùng phần 6 sẽ đưa ra các kết quả của mô hình xây dựng được và nhận xét.

3 Thông tin sinh viên

- Họ tên: Đinh Vũ Gia Hân
- MSSV: 22127098
- Lớp: 22CLC08
- Email: dvghan22@clc.fitus.edu.vn

4 Ý tưởng thực hiện và mô tả các hàm chức năng:

4.1 Giới thiệu chung về đồ án

Trong thời đại dữ liệu lớn, việc xây dựng các mô hình dự đoán chính xác trở nên ngày càng quan trọng. Hồi quy tuyến tính, với tính đơn giản và hiệu quả, là một trong những công cụ thống kê cơ bản nhưng vô cùng hữu ích. Đồ án này tập trung vào việc xây dựng và đánh giá một mô hình hồi quy tuyến tính để đánh giá mức độ ảnh hưởng của các yếu tố bên ngoài đến thành tích học tập của sinh viên. Qua đó, ta mong muốn tìm ra mối quan hệ giữa các biến độc lập và biến phụ thuộc, từ đó đưa ra những dự đoán chính xác và xây dựng mô hình phù hợp với thực tế.

4.2 Các thư viện cần dùng

- Numpy: dùng để thực hiện các phép tính liên quan đến ma trận.
- Pandas: dùng để đọc và thực hiện các phép biến đổi trên DataFrame.
- Matplotlib: dùng để hiển thị hình ảnh.
- Seaborn: dùng để hiển thị hình ảnh
- Warnings: dùng để phớt lờ các cảnh cáo khi vẽ biểu đồ.

4.3 Các hàm từ thư viện đã dùng (ngoại trừ Numpy)

- warnings.filterwarnings: dùng để loại bỏ các thông báo cảnh cáo không cần thiết.
- pandas.read_csv: dùng để đọc tập giá trị được phân cách bằng dấu phẩy vào DataFrame.
- pandas.DataFrame: dùng để chuyển dữ liệu dạng array thành DataFrame.
- pandas.Series: dùng để chuyển dữ liệu dạng array thành Series.

- pandas.DataFrame.copy: dùng để tạo một bản sao của dữ liệu này.
- pandas.DataFrame.to_numpy: dùng để chuyển dữ liệu dạng DataFrame thành numpy array.
- pandas.DataFrame.iloc: dùng để lập chỉ mục dựa trên index được truyền vào.
- pandas.DataFrame.loc: dùng để truy cập một nhóm hàng và cột theo nhãn hoặc mảng boolean.
- pandas.DataFrame.drop: dùng để loại bỏ các nhãn được chỉ định khỏi hàng hoặc cột.
- pandas.DataFrame.info: dùng để in ra tóm tắt về dữ liệu.
- pandas.DataFrame.describe: dùng để in ra số liệu thống kê của dữ liệu.
- pandas.DataFrame.duplicated: trả về chuỗi boolean biểu thị hàng trùng lặp.
- pandas.DataFrame.isnull: dùng để phát hiện các giá trị bị thiếu.
- pandas.DataFrame.hist: dùng để tạo histogram của các cột đột của DataFrame.
- pandas.DataFrame.corr: dùng để tính giá trị tương quan giữa các cặp theo cột.
- seaborn.boxplot: dùng để vẽ biểu đồ hình hộp để thể hiện sự phân bố theo các danh mục.
- seaborn.heatmap: dùng để vẽ dữ liệu hình chữ nhật dưới dạng ma trận được mã hóa màu.
- seaborn.barplot(): dùng để hiển thị ước tính điểm và lỗi dưới dạng thanh hình chữ nhật.
- seaborn.lineplot(): dùng để vẽ biểu đồ đường.
- matplotlib.pyplot.figure: dùng để tạo một hình mới.
- matplotlib.pyplot.subplot: thêm trục vào hình hiện tại hoặc truy xuất trục.
- matplotlib.pyplot.ylabel: dùng để đặt nhãn cho trục y.
- matplotlib.pyplot.show: dùng để hiển thị hình ảnh.
- matplotlib.pyplot.title: dùng để đặt tiêu đề cho hình.

4.4 Mô tả các hàm chức năng

4.4.1 def preprocess(x, features)

1. **Mô tả:** preprocess được dùng để thêm một cột 1 vào trước dữ liệu bảng được truyền vào. Việc thêm cột 1 này vào sẽ tạo ra một hệ số chặn giúp cho đường hồi quy không đi qua tâm.
2. **Tham số đầu vào:** Một biến DataFrame chứa bảng dữ liệu cần xử lý và danh sách tên các đặc trưng của bảng đó.
3. **Các bước thực hiện:**
   - Sử dụng kết hợp hàm numpy.ones và numpy.hstack để tạo ra một cột với số dòng bằng số dòng của bảng dữ liệu và thêm nó vào đầu bảng.
   - Tạo tên cho cột mới thêm vào.
   - Do sau khi thêm cột mới thì dữ liệu trở thành numpy array nên ta sẽ dùng hàm pandas.DataFrame để chuyển nó về lại DataFrame.
   - Và cuối cùng trả về bảng dữ liệu đã được xử lý.
4. **Kết quả đầu ra:** Bảng dữ liệu sau khi được xử lý.

4.4.2 def fit(self, X, y) (class OLSLinearRegression)

1. **Ý tưởng:**
   - Ta cần tìm nghiệm của phương trình $Ax \approx b$.
   - Xét ma trận A có kích thước m x n (m > n) và vector (cột) b có kích thước m. Ta có chuẩn Euclidean của bình phương phần dư r của $Ax - b$ như sau:

$$r = ||Ax - b||^2 \quad (1)$$

   - Để giải được nghiệm x cho hệ phương trình, ta thực hiện tối thiểu hóa công thức (1) được nghiệm x của hệ phương trình được tính như sau:

$$x = (A^T A)^{-1} A^T b \quad (2)$$

   - Note: $(A^T A)^{-1} A^T$ là ma trận giả nghịch đảo của A.
   - Đồ này sử dụng tên gọi khác cho đầu vào, đầu ra và tham số trong hồi quy tuyến tính như sau:
     - $A \to X$

  - $b \to y$
  - $x \to w$ (w: weight)
- $Ax \approx b \to Xw \approx y$ hay $Xw = y$ (y được gọi là đường hồi quy (regression line)).

2. **Mô tả:** fit được sử dụng để huấn luyện mô hình trên dữ liệu. Hàm sử dụng phương pháp Ordinary Least Squares để tìm ra các tham số tối ưu.

3. **Tham số đầu vào:** Một biến DataFrame chứa bảng dữ liệu cần xử lý và một biến Series chứa tên các giá trị mục tiêu.

4. **Các bước thực hiện:**

   - Sử dụng hàm pandas.to_numpy để chuyển dữ liệu đầu vào thành dạng numpy array để thuận tiện cho việc tính toán và lưu vào X_np và y_np.
   - Sử dụng hàm numpy.linalg().pinv kết hợp với phép '@' (nhân ma trận) đê tìm trọng số w theo công thức (2).
   - Và cuối cùng trả chính object hiện tại.

5. **Kết quả đầu ra:** Trả về chính object hiện tại.

4.4.3 def get_params(self) (class OLSLinearRegression)

1. **Mô tả:** get_params dùng để lấy các trọng số của mô hình.

2. **Tham số đầu vào:** Không có tham số đầu vào.

3. **Các bước thực hiện:**

   - Trả về trọng số của mô hình dưới dạng array.

4. **Kết quả đầu ra:** Trả về trọng số của mô hình.

4.4.4 def predict(self, X) (class OLSLinearRegression)

1. **Mô tả:** predict dùng để dự đoán giá trị mục tiêu dựa trên dữ liệu đầu vào.

2. **Tham số đầu vào:** Một biến DataFrame chứa bảng dữ liệu để dự đoán.

3. **Các bước thực hiện:**

   - Sử dụng hàm pandas.to_numpy để chuyển bảng dữ liệu đầu vào thành numpy array và lưu vào X_np.

- Nhân ma trận X_np và self.w theo công thức (2) (đã nên ở phần 4.3.2) để tìm giá trị dự đoán.
- Trả về các giá trị mục tiêu dự đoán dưới dạng Series

4. **Kết quả đầu ra:** Trả về giá trị mục tiêu dự đoán.

4.4.5 def MAE(y, y_hat)

1. **Ý tưởng:** MAE được dùng để ước lượng trung bình của sai số (độ lỗi) tuyệt đối, được tính bằng công thức:

$$MAE = \frac{1}{n}\sum_{n}^{i=1}|y_i - \hat{y}_i| \quad (3)$$

Trong đó:

- n: số lượng mẫu quan sát
- y_i: giá trị mục tiêu của mẫu thứ i
- $\hat{y}_i$: giá trị mục tiêu của mẫu thứ i được dự đoán từ mô hình hồi quy tuyến tính

2. **Mô tả:** MAE dùng để tính độ lỗi tuyệt đối trung bình.
3. **Tham số đầu vào:** Một Series chứa các giá trị mục tiêu thực tế và một series chứa các giá trị mục tiêu dự đoán
4. **Các bước thực hiện:**

- Ta chuyển các tham số đầu vào thành dạng array bằng hàm numpy.array. Sau đó ta lấy trị tuyệt đối của hiệu hai ma trận vừa tạo bằng hàm numpy.abs. Cuối cùng ta dùng hàm numpy.mean() để tính giá trị trung bình theo như công thức (3) và trả về giá trị vừa tính được.

5. **Kết quả đầu ra:** Trả về độ lỗi tuyệt đối trung bình.

4.4.6 def __init__(self, k, features, models, data) (class KFoldsCrossValidation)

1. **Ý tưởng:**

- Cross validation là một kỹ thuật lấy mẫu để đánh giá mô hình học máy trogn trường hợp dữ liệu không được dồi dào.
- Tham số quan trọng trong kỹ thuật này là k, đại diện cho số nhóm mà dữ liệu sẽ được chia ra. Vì lý do đó, nó được mang tên k-fold cross-validation. Khi giá trị của k được lựa

chọn, người ta sử dụng trực tiếp giá trị đó trong tên của phương pháp đánh giá. Ví dụ với k = 5, phương pháp sẽ mang tên 5-fold cross-validation.

- Theo bài viết từ trang trituenhantao.io [1], kỹ thuật này bao gồm các bước như sau:
  - Xáo trộn dataset một cách ngẫu nhiên
  - Chia dataset thành k nhóm
  - Với mỗi nhóm: Sử dụng nhóm hiện tại để đánh giá hiệu quả mô hình. Các nhóm còn lại được sử dụng để huấn luyện mô hình. Huấn luyện mô hình. Đánh giá và sau đó hủy mô hình
  - Tổng hợp hiệu quả của mô hình dựa từ các số liệu đánh giá
- Dựa vào kỹ thuật trên kết hợp cùng kiến thức tham được từ nguồn [2], ta sẽ xây dựng lớp KFoldsCrossValidation để thực hiện cho cross validation.

2. **Mô tả:** __init__ dùng khởi tạo của lớp KFoldsCrossValidation.
3. **Tham số đầu vào:** Một biến int thể hiện số fold sẽ được chia thành, danh sách đặc trưng, danh sách mô hình và một DataFrame chứa bảng dữ liệu để thực hiện cross validation.
4. **Các bước thực hiện:**
   - Gán các tham số đầu vào cho object hiện tại.
5. **Kết quả đầu ra:** Không có giá trị trả về

4.4.7 def shuffle_data(self) (class KFoldsCrossValidation)

1. **Mô tả:** shuffle_data dùng xáo trộn dữ liệu trước khi split
2. **Tham số đầu vào:** Không có tham số đầu vào.
3. **Các bước thực hiện:**
   - Đầu tiên, ta dùng hàm numpy.random.seed với seed ở trạng thái 42 để dữ liệu của ta chỉ bị xáo trộn một lần duy nhất, nghĩa là kết quả mỗi lần xáo trộn sẽ ra giống nhau.
   - Chuyển dữ liệu data của lớp thành array và lưu vào data_arr. Sử dụng hàm numpy.random.shuffle để xáo trộn dữ liệu.
   - Chuyển dữ liệu đã xáo trộn về dạng bảng và lưu vào biến data_shuffled của lớp.
4. **Kết quả đầu ra:** Không có giá trị trả về

4.4.8 def split_to_k_folds(self) (class KFoldsCrossValidation)

1. **Mô tả:** split_to_k_folds dùng để chia dữ liệu đã trộn thành k fold bằng nhau.
2. **Tham số đầu vào:** Không có tham số đầu vào.
3. **Các bước thực hiện:**
   - Tính số dòng của dữ liệu và lưu vào biến rows.
   - Với từng fold trong k fold, sử dụng pandas.DataFrame.iloc để lấy các dòng từ chỉ số bắt đầu đến chỉ số kết thúc của fold thứ i từ DataFrame và dùng hàm pandas.DataFrame.reset_index để đặt lại chỉ số của DataFrame mới tạo, bỏ qua chỉ số cũ.
4. **Kết quả đầu ra:** Không có giá trị trả về

4.4.9 def cross_validation(self) (class KFoldsCrossValidation)

1. **Mô tả:** cross_validation dùng để thực hiện cross validation trên các mô hình và đặc trưng đã cho.
2. **Tham số đầu vào:** Không có tham số đầu vào.
3. **Các bước thực hiện:**
   - Tạo một biến avg_maes_list trong lớp để lưu giá trị MAE trung bình của mỗi model.
   - Với mỗi model, tạo một biến mae_list để lưu giá trị MAE tương ứng với mỗi fold của model đó.
     - Với từng fold trong folds, ta sẽ lấy fold hiện tại đó để đánh giá mô hình và các fold còn lại để huấn luyện mô hình.
       * Dùng hàm pandas.DataFrame.loc để lọc dữ liệu theo nhãn của model và lưu vào biến X_test_i. Dùng hàm pandas.DataFrame.iloc để lọc dữ liệu theo vị trí -1 (tương ứng với Performance Index) và lưu vào y_test_i.
       * Tạo tập train_fold từ dữ liệu đã được xáo trộn và bỏ đi dữ liệu ở fold đã gán cho X_test_i và y_test_i. Dùng hàm pandas.DataFrame.loc để lọc dữ liệu theo nhãn của model và lưu vào biến X_train_i. Dùng hàm pandas.DataFrame.iloc để lọc dữ liệu theo vị trí -1 (tương ứng với Performance Index) và lưu vào y_train_i.
       * Huấn luyện cho mô hình hiện tại trên tập train bằng hàm OLSLinearRegression.fit.
       * Tính giá trị dự đoán y_hat_i bằng hàm OLSLinearRegression.predict.

        * Tính mae_i bằng hàm MAE và lưu nó vào mae_list.
    - Dùng hàm numpy.mean để tính giá trị MAE trung bình của model và lưu vào biến avg_maes_list của lớp.

4. **Kết quả đầu ra:** Không có giá trị trả về

4.4.10 def best_model(self) (class KFoldsCrossValidation)

1. **Mô tả:** best_model dùng để trả về danh sách chứa giá trị MAE trung bình của các model và index của model có MAE trung bình nhỏ nhất.
2. **Tham số đầu vào:** Không có tham số đầu vào.
3. **Các bước thực hiện:**
    - Dùng hàm numpy.argmin để tìm index của model có MAE trung bình nhỏ nhất
4. **Kết quả đầu ra:** Trả về danh sách chứa giá trị MAE trung bình của các model và index của model có MAE trung bình nhỏ nhất.

5 Phân tích khám phá dữ liệu

5.1 Mô tả dữ liệu

- Hours Studied: Tổng số giờ học của mỗi sinh viên.
- Previous Scores: Điểm số học sinh đạt được trong các bài kiểm tra trước đó.
- Extracurricular Activities: Sinh viên có tham gia hoạt động ngoại khóa không (Có hoặc Không).
- Sleep Hours: Số giờ ngủ trung bình mỗi ngày của sinh viên.
- Sample Question Papers Practiced: Số bài kiểm tra mẫu mà học sinh đã luyện tập.
- Performance Index: Thước đó thành tích tổng thể cho mỗi sinh viên.

5.2 Mục tiêu

Performance Index: Thước đó thành tích tổng thể cho mỗi sinh viên. Nó thể hiện trình độ học tập của sinh viên và đã được làm tròn thành đến số thập phân thứ nhất. Giá trị của nó nằm trong khoảng [10; 100] với những giá trị cao hơn chỉ ra thành tích tốt hơn.

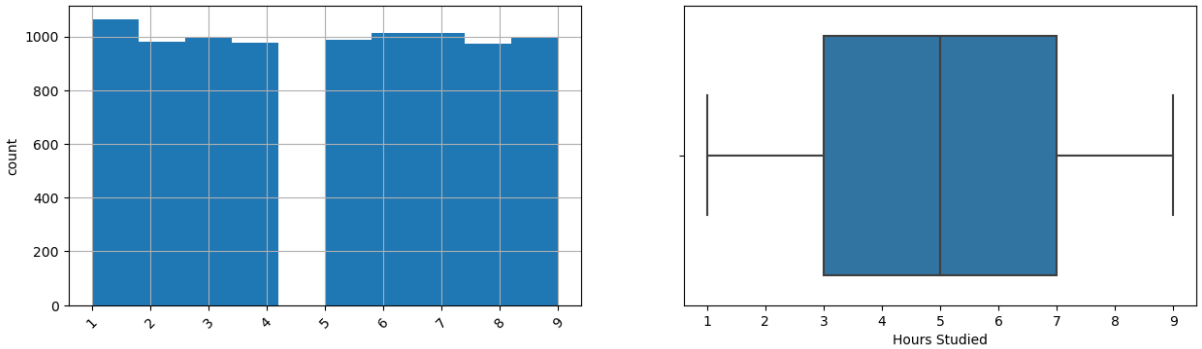
5.3 Phân tích dữ liệu

Sau khi sử dụng các hàm chức năng trong thư viện Pandas để phân tích, có thể thấy dữ liệu trên đã được thực hiện tiền xử lý và có một số đặc tính như sau:

- Dữ liệu gồm 9000 dòng và 6 cột.
- Dữ liệu gồm toàn các giá trị dưới dạng số. Trong đó 5 loại dữ liệu dạng số nguyên và 1 loại dữ liệu dạng số thực.
- Dữ liệu không có giá null nào.
- Có 103 giá trị giống nhau.
- Số giờ học trung bình của một sinh viên xấp xỉ 5 tiếng.
- Điểm số trước đó trung bình rơi vào khoảng 69 điểm, điểm thấp nhất là 40 điểm và không có sinh viên đạt điểm tuyệt đối.
- Số sinh viên tham gia và không tham gia hoạt động ngoại khóa gần như cân bằng.
- Trung bình một sinh viên ngủ 6 tiếng rưỡi một ngày, tương đối so với một giấc ngủ tiêu chuẩn.
- Số bài tập một sinh viên trung từ 4 đến 5 bài (4.59).
- Thành tích học tập trung bình của sinh viên là khoảng 55 điểm.

5.4 Trực quan hóa dữ liệu

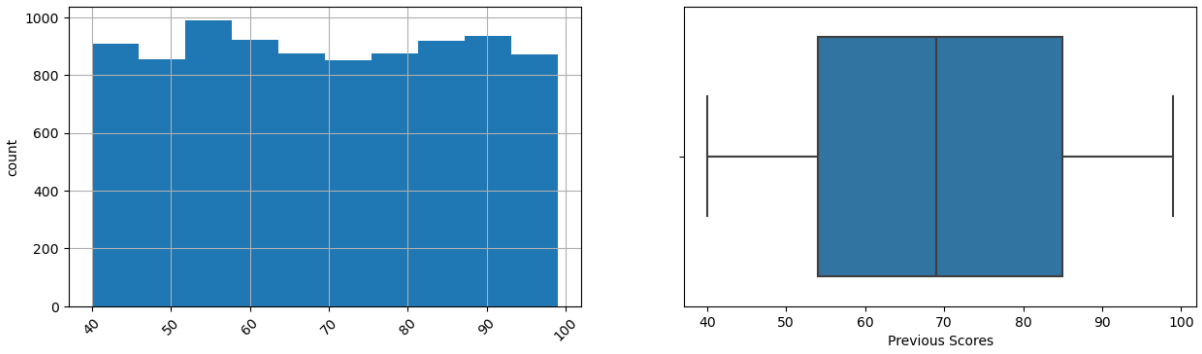
Đầu tiên, ta sẽ dùng Histogram và Boxplot để có cái nhìn tổng quan về các đặt trưng.



Hình 1: Biểu đồ về Hours Studied

Nhận xét:

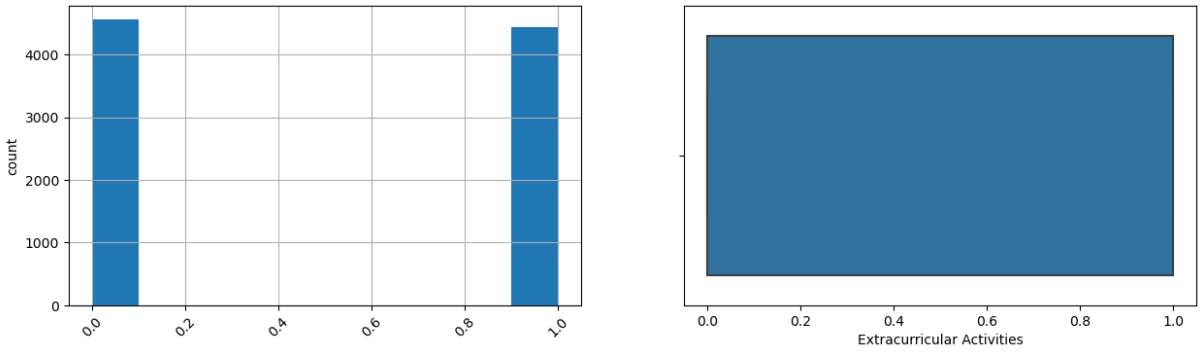
- Thời gian học tập được phân bố ổn đinh, phần lớn học sinh có thói quen học tập tương đối ổn định, dành một lượng thời gian tương đối bằng nhau cho việc học tập.
- Không có một nhóm học sinh nào dành quá nhiều hoặc quá ít thời gian cho việc học so với những người khác.



Hình 2: Biểu đồ về Previous Scores

Nhận xét:

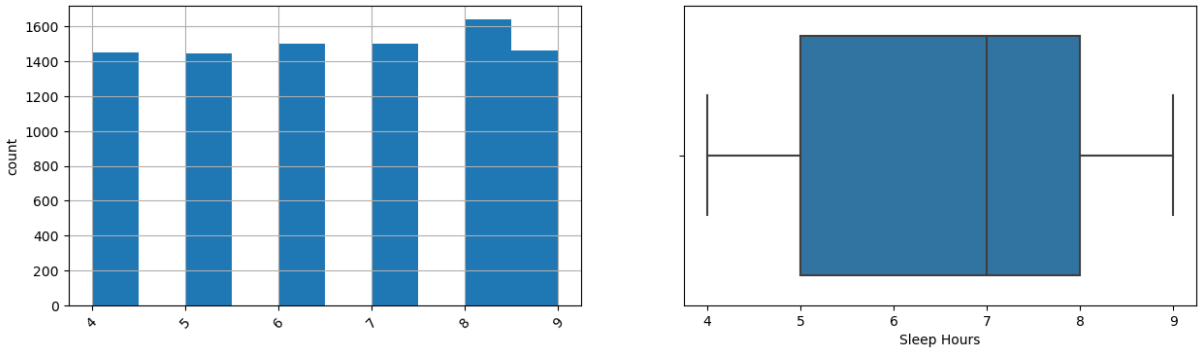
- Phần lớn học sinh có điểm số trước đó nằm trong khoảng từ 60 đến 80 điểm.
- Có một số học sinh đạt điểm rất cao và một số học sinh đạt điểm rất thấp. Tuy nhiên điểm số vẫn có sự phân bố khá đều.



Hình 3: Biểu đồ về Extracurricular Activities

Nhận xét:

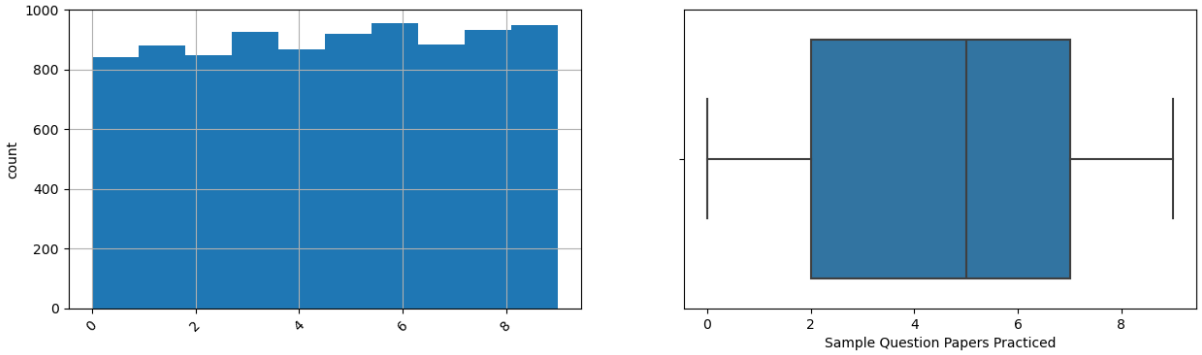
- Phân bố cân bằng: Số lượng sinh viên tham gia và không tham gia hoạt động ngoại khóa là xấp xỉ và gần như bằng nhau.



Hình 4: Biểu đồ về Sleep Hours

Nhận xét:

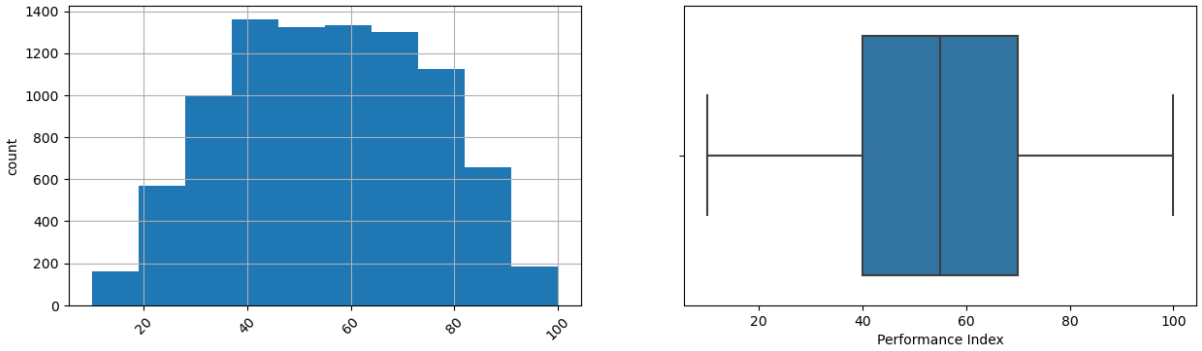
- Phần lớn sinh viên ngủ từ 6 đến 8 giờ mỗi đêm. Đây được coi là khoảng thời gian ngủ lý tưởng cho sức khỏe.
- Mặc dù có một xu hướng chung, nhưng vẫn có sự đa dạng về số giờ ngủ giữa các cá nhân. Một số người cần ngủ ít hơn hoặc nhiều hơn so với trung bình tùy vào yếu tố cá nhân.



Hình 5: Biểu đồ về Sample Question Papers Practiced

Nhận xét:

- Phần lớn học sinh có thói quen ôn luyện bằng cách làm từ 2 đến 6 đề thi.
- Mặc dù có xu hướng chung, nhưng vẫn có sự đa dạng về số lượng đề thi mà học sinh thực hành.

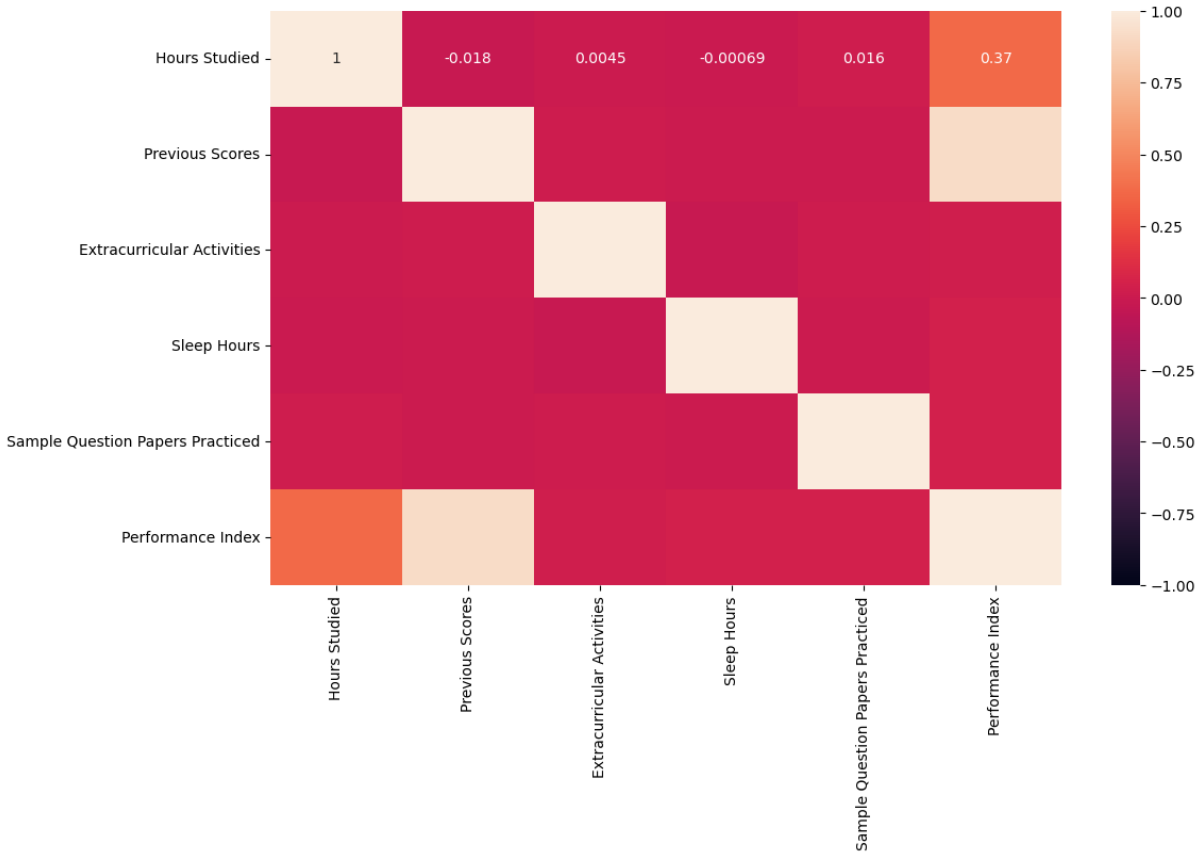


Hình 6: Biểu đồ về Performance Index

Nhận xét:

- Đa số các đối tượng đạt được chỉ số hiệu suất ở mức trung bình, dao động từ 40 đến 60.
- Số sinh viên đạt thành tốt nhất và số sinh viên đạt thành tích không tốt ít hơn so với phần còn lại.

Tiếp đó, ta dùng Heatmap cung cấp một cái nhìn tổng quan về mối quan hệ giữa các biến số trong bộ dữ liệu. Mỗi ô trong biểu đồ thể hiện hệ số tương quan giữa hai biến tương ứng.

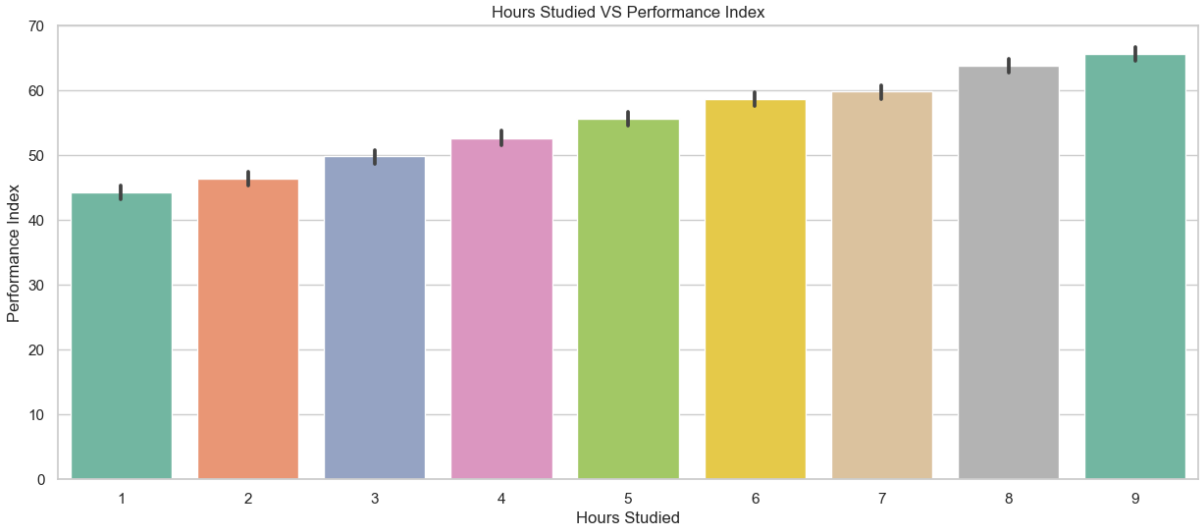


Hình 7: Độ tương quan giữa các đặc trưng

Nhận xét

- Hours Studied và Previous Scores có mối tương quan dương khá mạnh, cho thấy cành học nhiều thì điểm số trước đó càng cao và ngược ngại. Điều này khá hợp lý trong thực tế vì việc học tập thường xuyên sẽ giúp cho ta nhớ lâu và giúp cải thiện kết quả.
- Các đặc trưng còn lại như Extracurricular Activities, Sample Question Papers Practiced, Sleep Hours có mối quan hệ rất yếu hoặc gần như không có mối quan hệ với Performance Index. Điều này cho thấy các yếu tố này không ảnh hưởng đáng kể đế thành tích học tập của sinh viên.

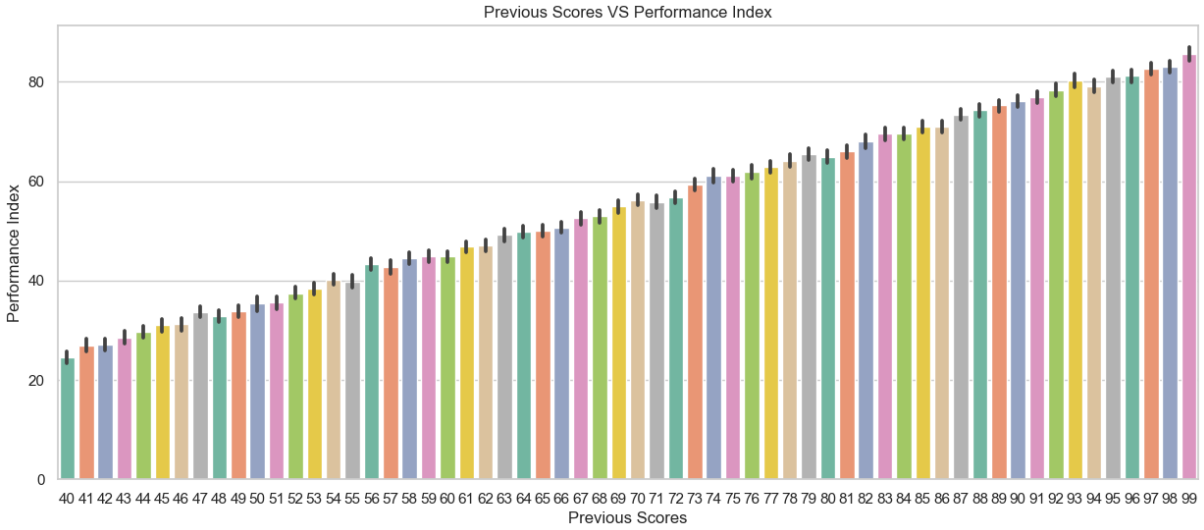
Do đó, ta sẽ phân tích kĩ hơn về hai đặc trưng Hours Studied và Previous Scores để làm rõ hơn về sự ảnh hưởng của chúng đến Performance Index.



Hình 8: Biểu đồ thể hiện mối quan hệ giữa Hours Studied và Performance Index

Nhận xét:

- Nhìn chung, biểu đồ cho thấy một xu hướng tăng khi số giờ học tăng lên thì chỉ số hiệu suất cũng có xu hướng tăng theo. Điều này cho thấy rằng việc tăng thời gian học tập có thể dẫn đến cải thiện kết quả học tập.
- Các thanh lỗi (error bars) cho thấy sự khác biệt trong chỉ số hiệu suất giữa các cá nhân ở cùng một mức độ giờ học. Điều này cho thấy có nhiều yếu tố khác ngoài thời gian học tập ảnh hưởng đến kết quả học tập.



Hình 9: Biểu đồ thể hiện mối quan hệ giữa Previous Scores và Performance Index

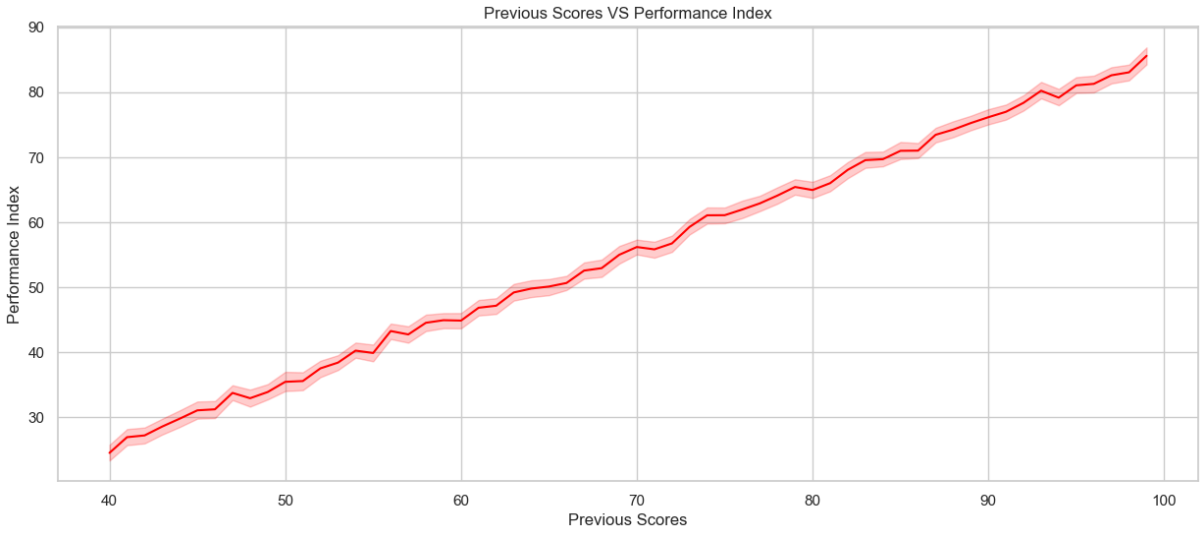
Nhận xét:

- Khi điểm số trước đó tăng lên, chỉ số hiệu suất cũng có xu hướng tăng theo. Điều này cho thấy một mối quan hệ trực tiếp và tích cực giữa hai biến số. Nghĩa là điểm số trước đó tác động đến thành tích của sinh viên rất nhiều. Nếu điểm số tốt sẽ tạo động lực giúp sinh viên cố gắng hơn. Ngược lại, điểm số không tốt làm cho sinh viên chán nản dẫn đến thành tích xấu đi.
- Các cột biểu đồ có xu hướng tăng dần từ trái sang phải, cho thấy sự tương quan cao giữa điểm số trước đó và chỉ số hiệu suất.

Hình 10: Biểu đồ thể hiện mối quan hệ giữa Hours Studied và Performance Index

Nhận xét:

- Nhìn chung, biểu đồ cho thấy một xu hướng tăng khi số giờ học tăng lên thì chỉ số hiệu suất cũng có xu hướng tăng theo. Điều này gợi ý rằng việc tăng thời gian học tập có thể dẫn đến cải thiện kết quả học tập.
- Vùng màu đỏ xung quanh đường biểu diễn cho thấy khoảng tin cậy của chỉ số hiệu suất ở mỗi mức độ giờ học. Điều này giúp chúng ta đánh giá mức độ chắc chắn của mối quan hệ giữa hai biến.
- Có thể thấy đặc trưng này có độ tương quan tốt với Performance Index. Đây có thể là một trong những đặc trưng tốt để xây dựng mô hình hồi quy.



Hình 11: Biểu đồ thể hiện mối quan hệ giữa Previous Scores và Performance Index

Nhận xét:

- Nhìn chung, biểu đồ cho thấy một xu hướng tăng khi điểm số trước đó tăng lên thì chỉ số hiệu suất cũng có xu hướng tăng theo. Điều này cho thấy một mối quan hệ trực tiếp và tích cực giữa hai biến số.
- Đường biểu diễn màu đỏ có xu hướng tăng dần từ trái sang phải, tuy không tăng đều và liên tục như Hình 10, nhưng nó cũng cho thấy sự tương quan cao giữa điểm số trước đó và chỉ số hiệu suất. Điểm số trước đó cũng đóng vai trò quan trọng trong thành tích của sinh viê.
- Đây cũng là một đặc trưng tốt để xây dựng mô hình hồi quy.

6 Các mô hình xây dựng được và nhận xét

6.1 Kết quả câu 2a

Đặc trưng	**Trọng số**
Intercept	-33.969283681727944
Hours Studied	2.8520200719067246
Previous Scores	1.0178695738370305
Extracurricular Activities	0.6042817401213944
Sleep Hours	0.47356583475097425
Sample Questions Papers Practiced	0.19237624280136512

Bảng 1: Kết quả trọng số tương ứng với từng đặc trưng của mô hình (Câu 2a)

6.2 Kết quả câu 2b

STT	**Mô hình với 1 đặc trưng**	**MAE**
1	Hours Studied	21.429857651403907
2	Previous Scores	7.11217107398639
3	Extracurricular Activities	35.693211951447246
4	Sleep Hours	18.996130845044924
5	Sample Questions Papers Practiced	27.97482077945479

Bảng 2: Kết quả tương ứng cho 5 mô hình từ k-fold Cross Validation (Câu 2b)

Nhận xét:

- Nhìn vào bảng kết quả trên, có thể thấy MAE cho mô hình với Previous Scores có giá trị nhỏ nhất và có một khoảng cách lớn so với các đặc trưng còn lại. Do đó, mô hình với Previos Scores là mô hình tốt nhất từ k-fold Cross Validation.

- Điều này khá hợp lý trong thực tế do điểm số trước đó là một trong những điều ảnh hưởng lớn đến điểm số học tập của sinh viên. Một sinh viên đạt điểm số tốt trong bài kiểm tra trước đó sẽ tìm thấy niềm vui và động lực để học tập.

Đặc trưng	Trong số
Intercept	-14.988645779316068
Previous Scores	1.0105030093166387

Bảng 3: Kết quả trọng số tương ứng với từng đặc trưng của mô hình với đặc trưng tốt nhất (Câu 2b)

6.3 Kết quả câu 2c

Ba mô hình xây dựng được:

- Mô hình 1: Sử dụng 2 đặc trưng **(Hours Studied, Previous Scores)**. Mô hình này được xây dựng từ 2 đặc trưng có độ tương quan tốt sau khi phân tích EDA ở câu 1.
- Mô hình 2: Sử dụng 3 đặc trưng **(Hours Studied * Hours Studied, Previous Scores, Sample Question Papers Practiced)**. Tuy trong câu 1, Sample Question Papers Practiced có độ tương quan khá thấp nhưng trong thực tế nó lại có độ ảnh hưởng tốt để thành tích học tập của sinh viên. Khi sinh viên đạt được số điểm tốt trước đó, nó sẽ là nguồn động lực để họ bỏ ra nhiều thời gian nhiều để học tập, và khi ấy họ sẽ tiếp xúc và làm được nhiều dạng đề, nó sẽ giúp cho họ có được một phản xạ và dần hiểu rõ được vấn đề hơn. Có thể thấy rằng điểm số, thời gian học và rèn luyện nhiều là ba yếu tố gắn kết mật thiết đến thành tích của một sinh viên. Bên cạnh đó ta sẽ tăng cường đặc trưng Hours Studied.
- Mô hình 3: Sử dụng 3 đặc trưng **(Hours Studied, Previous Scores * Previous Scores, Sample Question Papers Practiced)**. Tương tự như mô hình 2, ta sẽ sử dụng thêm đặc trưng Sample Question Papers Practiced. Tuy nhiên ở mô hình này, ta sẽ tăng cường đặc trưng Previous Scores.

STT	Mô hình	MAE
1	Mô hình 1	5.679061864983116
2	Mô hình 2	4.99168478034849
3	Mô hình 3	2.4901415994773246

Bảng 4: Kết quả tương ứng cho 3 mô hình từ k-fold Cross Validation (Câu 2c)

Nhận xét:

- Mô hình được xây dựng từ 3 đặc trưng Hours Studied, Previous Scores * Previous Scores, Sample Question Papers Practiced là mô hình tốt nhất trong ba mô hình.

- MAE của mô hình 3 nhỏ hơn một nửa so với hai mô hình còn lại.
- Việc tăng cường đặc trưng Previous Scores đã đóng góp đáng kể cho việc cải thiện hiệu suất của mô hình 3. Điều này càng khẳng định hơn Previous Score là đặc trưng ảnh hưởng đến Performance Index nhất (ta đã chứng minh ở câu 2b).
- Tóm lại, Hours Studied, Previous Scores, Sample Question Papers Practiced có là 3 yếu tố chính ảnh hưởng thành tích học tập của sinh viên (dựa trên dữ liệu được cung cấp).

Feature	**Weight**
Intercept term	3.0115121308340598
Hours Studied	2.8518750054993802
Previous Scores * Previous Scores	0.007240430399352047
Sample Question Papers Practiced	0.19161993294427915

Bảng 5: Kết quả trọng số tương ứng với từng đặc trưng của mô hình tốt nhất (Câu 2c)

Tài liệu

[1] Giới thiệu về k-fold Cross Validation, Ngày truy cập: 08/08/2024.

[2] Linear Regression Project, Ngày truy cập: 08/08/2024.

[3] Pandas python tutorial, Ngày truy cập: 07/08/2024.

[4] Student Performance using deep learining, Ngày truy cập: 10/08/2024.

[5] Step-by-Step Exploratory Data Analysis (EDA) using Python, Ngày truy cập: 10/08/2024.

[6] Tài liệu và mã nguồn cung cấp bởi giáo viên.