TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

Báo cáo đồ án 3: Linear Regression

Thông tin sinh viên:

Họ và tên: Tô Quốc Thanh

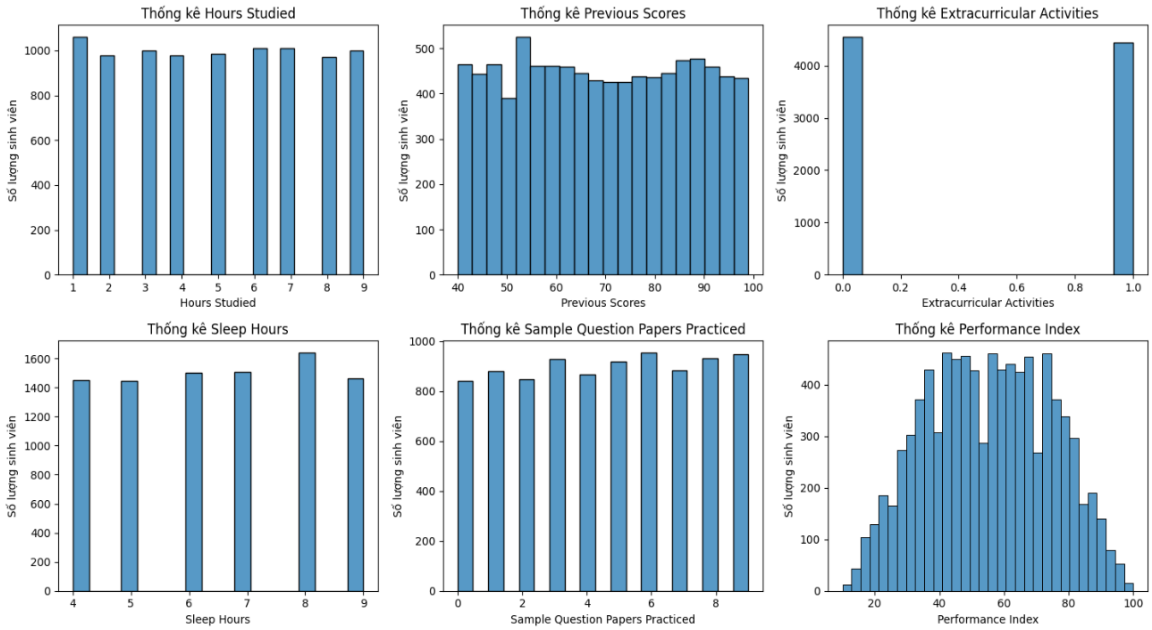
MSSV: 22127388

Lớp: 22CLC10

Mục lục

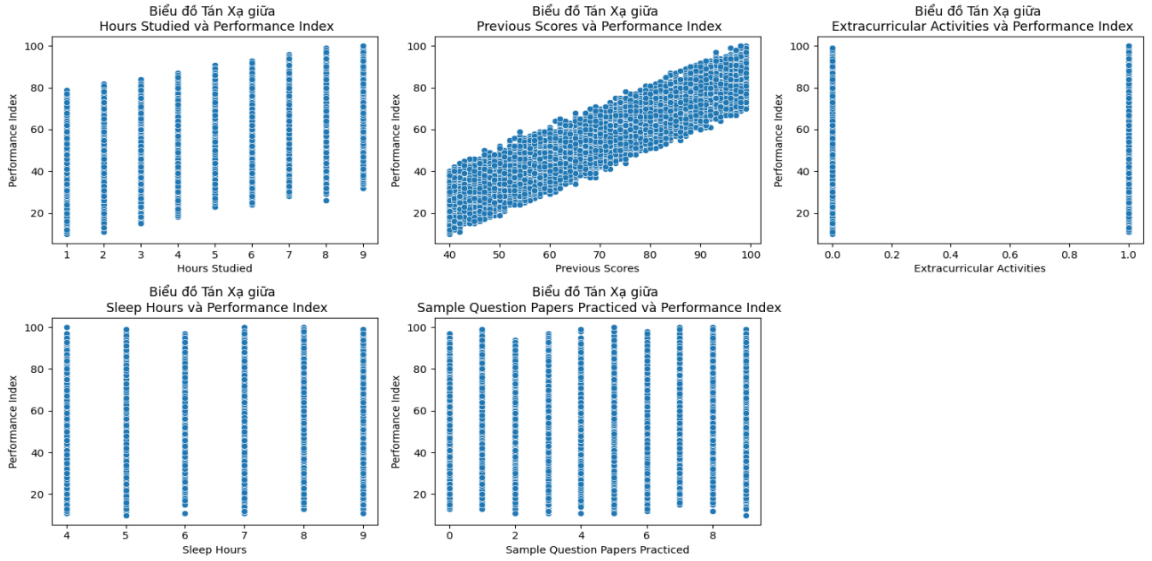
1. Khám phá dữ liệu

- Để có thể nắm bắt tổng quát nhất về các dữ liệu được cung cấp ở tập train, đầu tiên ta dùng biểu đồ cột (histogram) để có dữ liệu tổng quan về số lượng sinh viên ứng với mỗi đặc trưng



Có thể thấy rằng ở hầu hết các đặc trưng, số lượng sinh viên ở các vùng giá trị của mỗi đặc trưng là khá tương đồng, không có sự khác biệt quá lớn. Chỉ có biểu đồ cuối cùng, thống kê về "Performance Index" có sự phân hóa, phần lớn sinh viên có chỉ số Performance Index trong khoảng 40 đến 80, chỉ một số ít thuộc các vùng còn lại.

- Để thể hiện tác động của mỗi đặc trưng đến Performance Index, ta dùng biểu đồ Tán xạ (Scatter plot), biểu đồ này thể hiện giá trị của mỗi đặc trưng bằng những dấu chấm màu xanh, đồng thời thể hiện giá trị Performance Index tương ứng với giá trị đó.

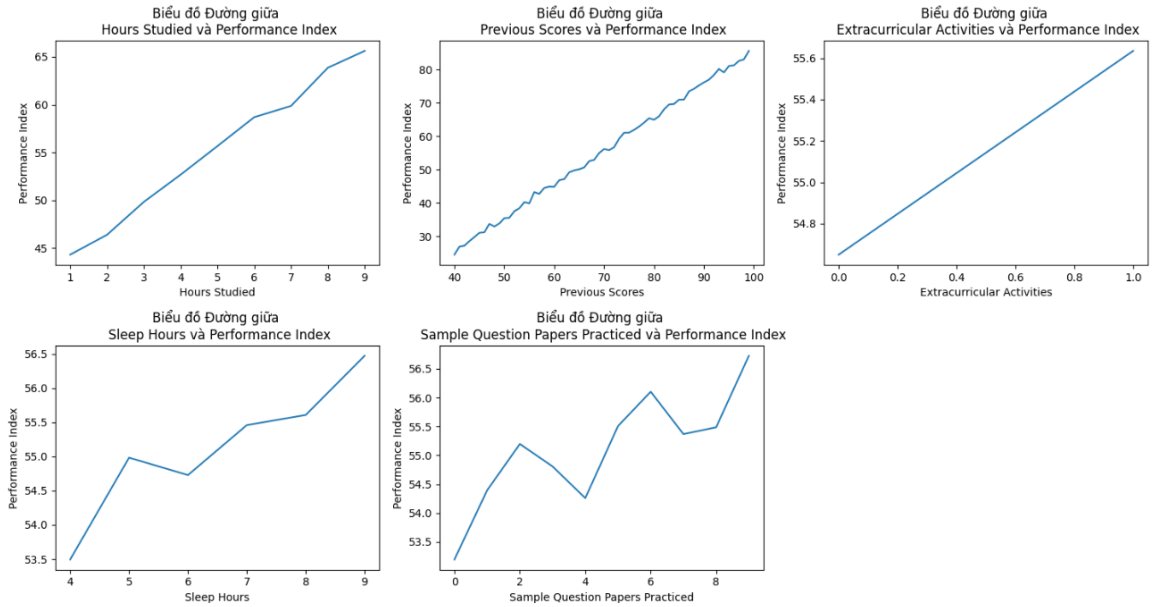


Quan sát các biểu đồ, ta dễ thấy các đặc trưng như Extracurricular Activities,Sleep Hours,Sample Question Papers Practiced không ảnh hưởng nhiều đến Performance Index. Bằng chứng là với mọi giá trị của 'Sleep Hours', Performance Index đều đồng đều đạt các giá trị từ 0 đến 100, không có sự phân hóa quá rõ ràng để ta nhìn ra được mối quan hệ giữa đặc trưng này và kết quả. Điều này cũng tương tự với 2 đặc trưng Extracurricular Activities, Sample Question Papers Practiced còn lại.

Trong khi đó, Hours Studied và Previous Scores lại có ảnh hưởng đến Performance Index.

- Hours Studied: Tuy ở mỗi giá trị của đặc trưng này đều có sự phân bố đồng đều của các giá trị Performance Index, tuy nhiên phạm vi phân bố có sự thay đổi. Dễ thấy rằng với các giá trị Hours Studied càng lớn, phạm vi phân bố của Performance Index càng tăng tịnh tiến, từ khoảng (0, 80) ở Hours Studied = 1 lên thành khoảng (35, 100) ở Hours Studied = 9. Điều này cũng phù hợp với thực tế, rằng thời gian học càng nhiều, thì hiệu quả học tập có xu hướng càng tăng cao.
- Previous Scores: So với Hours Studied, đặc trưng này có sự phân bố các giá trị rõ ràng hơn, vùng phân bố của Performance Index ở mỗi giá trị của đặc trưng hẹp hơn, ta có thể hiểu rằng là nó có mối quan hệ tuyến tính mạnh hơn đối với Performance Index. Có thể thấy từ biểu đồ, rằng Previous Scores càng lớn, thì Performance Index càng cao.

➢ Biểu đồ tán xạ thể hiện mối quan hệ giữa đặc trưng với Performance Index theo phân vùng giá trị của Performance index ở mỗi giá trị của đặc trưng đó, tuy nhiên phân vùng này có khá nhiều giá trị, mỗi giá trị lại có thể xuất hiện nhiều lần nên ta khó để quan sát và có kết luận minh bạch, chính xác. Do đó, ta sẽ biểu diễn các phân vùng này bằng một giá trị duy nhất, đó là giá trị trung bình của nó bằng cách sử dụng biểu đồ đường (Line plot).



Ghi chú: do sử dụng giá trị trung bình, phạm vi của Performance Index có sự thay đổi, không còn là [0, 100] như các biểu đồ trước đó nữa mà thay đổi theo giá trị trung bình được tính từ tập train.

Tương tự như các kết luận trước đó, quan sát các biểu đồ, ta dễ thấy các đặc trưng như Extracurricular Activities,Sleep Hours,Sample Question Papers Practiced không ảnh

hưởng nhiều đến Performance Index khi mà vùng giá trị của Performance Index là khá nhỏ, điều này thể hiện rằng sự thay đổi giá trị của đặc trưng không làm thay đổi quá nhiều đến Performance Index. Tuy nhiên, vẫn có một số điểm mà ta cần chú ý ở các đặc trưng này:

- Extracurricular Activities: Performance Index có sự thay đổi rõ giữa nhóm các sinh viên có tham gia hoạt động ngoại khóa so với không tham gia, rõ ràng nhóm sinh viên có tham gia hoạt động ngoại khóa có Performance Index cao hơn nhóm còn lại.
- Sleep Hours: : xu hướng của đặc trưng này là với các giá trị càng lớn của đặc trưng, Performance Index sẽ tăng theo.
- Sample Question Papers Practiced: xu hướng của đặc trưng này là với các giá trị càng lớn của đặc trưng, Performance Index sẽ tăng theo, tuy nhiên có những giá trị của đặc trưng mà ở đó xảy ra sự biến thiên tương đối lớn. Trong trường hợp của đặc trưng Sample Question Papers Practiced, ta có thể đoán rằng nguyên nhân là do các bài làm mẫu trước đó mà sinh viên làm có thể quá khác hoặc quá giống so với bài thi/ kiểm tra thật, từ đó dẫn đến sự biến thiên ngẫu nhiên này, không có tính quy luật.

Nếu trước đó ta đã nhận thấy mối quan hệ mạnh giữa Hours Studied và Previous Scores đến Performance Index, thì ở dạng biểu đồ đường này, điều đó càng thể hiện rõ ràng hơn. Chúng đều có xu hướng làm tăng Performance Index khi các giá trị của đặc trưng càng tăng.

- Hours Studied: Đường thẳng của biểu đồ với đặc trưng này 'ổn định' hơn so với Previous Scores, nghĩa là trong phạm vi rất nhỏ, sự biến thiên là ít hơn. Một phần giải thích cho điều này là vì tập giá trị của đặc trưng này nhỏ, chỉ từ khoảng (0, 10), trong khi của Previous Scores là từ (40, 100]. Một phần đến từ kết luận mang tính cá nhân, rằng trong thực tế, việc dành nhiều thời gian ra học tập thường sẽ giúp sinh viên nắm vững kiến thức, từ đó tỷ lệ đạt kết quả tốt trong học tập sẽ cao hơn. Tuy nhiên phân vùng của Performance Index không quá lớn, chỉ nằm trong khoảng (40, 70), điều này chỉ ra rằng thời gian học tập ít hay nhiều sẽ không gây ra sự thay đổi quá lớn, đột ngột trong thành tích học tập.
- Previous Scores: Có dạng giống một đường thẳng dốc lên, thể hiện rằng Performance Index sẽ tăng lên khi đặc trưng này tăng, phạm vi của Performance Index lớn, thuộc khoảng (20, 90), thể hiện mối quan hệ mạnh giữa đặc trưng này đến kết quả. Trong phạm vi nhỏ, đường thẳng có sự biến thiên không đều, thể hiện rằng các kết quả trước đó trong một số trường hợp nhỏ sẽ có sự thay đổi về mức ảnh hưởng đến Performance Index.

- Từ các khám phá dữ liệu trên, ta đã phần nào kết luận được sự ảnh hưởng của các đặc trưng đến Performance Index, tuy nhiên để thể hiện rõ ràng và minh bạch hơn, ta cần sử dụng ma trận tương quan [2]

Ma trận tương quan là một công cụ tốt để thể hiện mối quan hệ giữa các thành phần trong tập dữ liệu (bao gồm cả đặc trưng và kết quả). Với mỗi giá trị trong ma trận thể hiện mối quan hệ tuyến tính của các thành phần tương ứng, trong đó giá trị của mỗi ô được gọi là hệ số tương quan, nếu hệ số tương quan dương thì các thành phần có mối quan hệ tuyến tính dương và ngược lại, hệ số tương quan nằm trong khoảng [-1, 1]

Ta tập trung vào dòng của thành phần Performance Index để khám phá mối quan hệ của các thành phần khác với nó.

Dễ thấy, Previous Scores có mối quan hệ tuyến tính dương mạnh, hệ số tương quan đạt đến 0.915. Kế đó là Hours Studied, có quan hệ tuyến tính dương mạnh, hệ số tương quan đạt 0.369, qua hệ tuyến tính yếu hơn so với Previous Scores. Các đặc trưng còn lại không ảnh hưởng nhiều đến Performance Index, mối quan hệ tuyến tính là dương và rất yếu, tuy nhiên ta có thể kết luận rằng tất cả các đặc trưng của đề bài đều giúp tăng Performance Index khi giá trị của chúng tăng lên, điều này cũng có ý nghĩa trong thực tế vì hầu hết các đặc trưng được chọn đều mang tính 'tích cực' khi giá trị của chúng tăng lên và ngược lại, vì thế chúng làm tăng Performance Index khi giá trị của các đặc trưng đó tăng lên.

Một số mối quan hệ khác mà ta có thể thấy từ ma trận tương quan:

-Sleep Hours và Hours Studied có hệ số tương quan âm (-0.001)cho thấy rằng thời gian ngủ càng nhiều thì thời gian học càng ít và ngược lại.

-Sleep Hours và Extracurricular Activities có hệ số tương quan âm (-0.021), cho thấy rằng sinh viên dành nhiều thời gian cho việc ngủ thường ít tham gia các hoạt động ngoại khóa.

-Previous Scores và Hours Studied có hệ số tương quan âm (-0.018): có thể do sinh viên gặp áp lực, căng thẳng dẫn đến dù có dành ra nhiều thời gian để học tập, kết quả vẫn không cao. Hoặc sinh viên gặp phải hiệu ứng "Expertise reversal effect", khiến họ giảm năng suất khi dành nhiều thời gian để học hơn [1].

➢ **Kết luận:**

Previous Scores là đặc trưng có ảnh hưởng lớn nhất đến Performance Index (trích từ bài báo khoa học: "prior academic achievement is the most important factor" [3]) , tiếp đó là đến Hours Studied. Các đặc trưng khác tuy có mối quan hệ, tuy nhiên không ảnh hưởng quá nhiều đến Performance Index.

2. Xây dựng mô hình hồi quy tuyến tính

a. Sử dụng toàn bộ 5 đặc trưng

➢ **Mô tả:**

Để có thể sử dụng dữ liệu từ tập train và tập test mà đề bài cung cấp, ta cần chuyển đổi kiểu dữ liệu của chúng từ dataframe và series sang numpy array.

Đối với tập X_train, ta reshape thành 1 mảng có số dòng bằng với số dòng trong file train.csv, mỗi dòng chứa 5 cột tương ứng với 5 đặc trưng trong file đó, tập Y_train cũng có số dòng tương đương với X_train, chỉ chứa 1 cột duy nhất là Performance Index.

Ta thực hiện việc reshape tương tự trên file test.csv để nhận được X_test và Y_test.

Ta gọi X_train_preprocessed là tập X_train sau khi được tiền xử lí theo đoạn mã sau:

```
X = np.hstack((np.ones((x.shape[0], 1)), x))
```

Ta thêm ma trận 1 với số dòng tương ứng với X_train vào để tạo thành tập tiền xử lí, việc này là để cung cấp một tham số tự do cho mô hình.

Dựa vào tập X_train_preprocessed và Y_train nhận được trước đó cùng với các kiến thức về phương pháp Bình phương tối thiểu (OLS), ta dễ dàng tìm được bộ tham số tối ưu (các tham số w) theo dữ liệu tập train.

Từ tham số nhận được, ta viết được mô hình hồi quy tuyến tính. Áp dụng dữ liệu từ tập X_test, ta thu được y_pred là các giá trị y (Performance Index) dự đoán dựa trên mô hình vừa thu được.

So sánh các giá trị trong y_pred với giá trị y thật sự từ tập y_test (đảm bảo các giá trị là tương ứng, cùng index) bằng cách tính độ chênh lệch (sai số) giữa các giá trị y. Trung bình của các độ chênh lệch trên được gọi là MAE (Mean Absolute Error), đây là một chỉ số dùng để đánh giá độ chính xác của mô hình, MAE càng thấp thì độ chính xác của mô hình càng cao.

➢ **Kết quả**

$$\text{Student Performance} = -33.969 + 2.852 * (\textit{HoursStudied}) + 1.018 * (\textit{PreviousScores}) + 0.604 * (\textit{ExtracurricularActivities}) + 0.474 * (\textit{SleepHours}) + 0.192 * (\textit{SampleQuestionPapersPracticed})$$

Chỉ số MAE cho mô hình sử dụng 5 đặc trưng: 1.596

- **Nhận xét:**
  - Mô hình thu được có chỉ số MAE khá nhỏ, chứng minh mô hình xây dựng dựa trên toàn bộ 5 đặc trưng là khá tốt, hầu hết các đặc trưng đều có mối tương quan dương đối với cột mục tiêu, nghĩa là các đặc trưng như Hours Studied, Previous Scores,... đều giúp cho Performance Index tăng lên (dựa vào ma trận tương quan ở câu 1), tuy nhiên mức độ ảnh hưởng của mỗi đặc trưng là khác nhau.
  - Tập train và tập test nhận được có giá trị tốt, không chênh lệch quá nhiều, từ đó cho ta kết quả tốt.

b. Xây dựng mô hình sử dụng duy nhất 1 đặc trưng

- **Mô tả**

Tương tự với câu 2a, để có thể sử dụng các thông tin tự tập train và tập test mà để bài cung cấp, trước tiên ta cần chuyển chúng về dạng numpy array.
Vì đề bài cho ta 5 đặc trưng và yêu cầu tìm ra đặc trưng tốt nhất, ta cần đi tính chỉ số MAE cho từng mô hình. Tuy nhiên, MAE chỉ được tính trên tập test, mà tập test lại là tập được sử dụng cuối cùng, vì thế chúng ta cần tìm cách khác để tính MAE.

Nếu sử dụng tập train để tính MAE, sẽ xảy ra hiện tượng overfitting vì mô hình đã học được quá nhiều từ dữ liệu trước đó, dẫn đến chỉ số MAE tốt hơn so với thực tế.
Ta sử dụng phương pháp cross-validation, chia các thông tin trong tập train thành k-fold khác nhau (trong phạm vi đồ án, ta chia thành 5 fold). Trong 5 fold đó, ta sử dụng 4 fold để train và 1 fold còn lại để test, ta lặp lại việc này k = 5 lần. Chỉ số MAE sẽ là trung bình của các chỉ số MAE sau mỗi lần lặp. Đặc trưng tốt nhất sẽ là đặc trưng có chỉ số MAE trung bình nhỏ nhất.

- **Cách chia fold**

  Để chia fold, ta tạo ra một mảng chứa các số nguyên từ 0 đến n – 1, với n là số lượng dòng của tập train, từ đó ta có một mảng tạm gọi là Index chứa các index của tập train.
  Ta tiến hành xáo trộn (shuffle) trên mảng Index bằng lệnh np.random.shuffle() của numpy, gọi mảng kết quả thu được là shuffled_index.
  Vì ta chia tập train thành 5 fold, nên mỗi fold sẽ bao gồm n / 5 phần tử (với n là số dòng của tập train) của tập train, với mỗi phần tử được trích xuất từ index tương ứng trong tập shuffled_index, từ đó ta nhận được fold với các phần tử đã được xáo trộn.
  Việc xáo trộn dựa theo index giúp ta dễ dàng trong việc tạo ra tập x_train_fold và y_train_fold tương ứng với nhau (vì dựa theo cùng index trong tập train).
  Ta có thể tiền xử lí rồi mới tiến hành chia fold, hoặc tiền xử lí sau đó, trong đồ án này, ta chọn cách đầu tiên.

- **Cách tìm đặc trưng tốt nhất**

  Lưu ý, với mỗi đặc trưng, ta chỉ được giữ lại tham số w ứng với đặc trưng đó và tham số w tự do, bởi vì ta tiền xử lí 1 lần duy nhất trước đó nên ta cần thực hiện thêm bước này để có bộ tham số đúng để tạo mô hình.
  Để tìm ra đặc trưng tốt nhất, ta cần tính chỉ số MAE trung bình cho từng đặc trưng, chỉ số này được tính 5 lần khác nhau khi ta chọn tập để train và test từ các fold sau khi xáo trộn.

Tập để train bao gồm các fold trong tập train, trừ đi 1 fold bất kì. Fold được loại ra đó sẽ là tập test.

Từ tập train nói trên, ta thu được mô hình với bộ tham số tối ưu. Ta sử dụng tập test để tính y_pred (các giá trị y dự đoán) và so sánh với giá trị y thật (được lấy từ fold bị loại ra trước đó), ta thu được giá trị MAE của lần lặp đó. Sau khi lặp 5 lần, ta tính được MAE trung bình của 1 đặc trưng. Ta kiểm tra xem đó có phải là giá trị MAE nhỏ nhất để kết luận rằng đó là đặc trưng tốt nhất (tính đến lần lặp hiện tại). Sau đó reset list MAE và chuyển sang đặc trưng khác.

Sau khi duyệt qua tất cả đặc trưng, ta tìm được đặc trưng tốt nhất nhờ vào chỉ số MAE nhỏ nhất.

Ta huấn luyện lại mô hình dựa trên đặc trưng tốt nhất, lúc này tập test và tập train sẽ là các tập gốc mà đồ án cung cấp, tuy nhiên trên tập x, ta chỉ giữ lại các giá trị của đặc trưng tốt nhất, không quan tâm đến các đặc trưng khác.

- **Kết quả**

STT	Mô hình với 1 đặc trưng	MAE
1	Hours Studied	15.453
2	Previous Scores	6.618
3	Extracurricular Activities	16.198
4	Sleep Hours	16.192
5	Sample Question Papers Practiced	16.191

Đặc trưng tốt nhất: Previous Scores

Mô hình 1 đặc trưng dựa trên Previous Scores là:

$$\text{Student Performance} = -14.989 + 1.011 * (PreviousScores)$$

Chỉ số MAE cho mô hình sử dụng 1 đặc trưng duy nhất (Previous Scores) là: 6.544

- **Nhận xét:**
  - Các đặc trưng như Extracurricular Activities, Sleep Hours, Sample Question Papers Practiced đều tạo thành mô hình 1 đặc trưng với chỉ số MAE khá lớn, thể hiện rằng các đặc trưng này không có ảnh hưởng lớn đến Performance Index, ta không thể xây dựng mô hình dựa đoán dựa trên các đặc trưng này.
  - Đặc trưng Hours Studied tuy có giá trị MAE nhỏ hơn 3 đặc trưng trên nhưng vẫn còn khá lớn, không thể chỉ dùng đặc trưng này để dự đoán Performance Index.
    - Ta đã chứng minh ở phần khám phá dữ liệu, rằng ảnh hưởng của đặc trưng này là không quá lớn khi chỉ số tương quan chỉ đạt 0.369
    - Bài báo khoa học cũng đã chứng minh điều này, rằng việc tăng/ giảm số giờ học (trong thời gian ngắn) không ảnh hưởng quá lớn đến Performance Index [1]
  - Đặc trưng Previous Scores là đặc trưng tốt nhất, có ảnh hưởng lớn nhất tới Performance Index. Để chứng minh kết quả này đúng, ta có thể dựa vào một số luận điểm sau:

- Các kết quả từ việc khám phá dữ liệu: ta đã kết luận trước đó, rằng Previous Scores là đặc trưng tốt nhất, có ảnh hưởng lớn nhất đến Performance Index.
- Các bài báo, nghiên cứu khoa học: Trích từ bài báo: "prior academic achievement is the most important factor" [3]

c. Tự xây dựng/thiết kế mô hình, tìm mô hình cho kết quả tốt nhất

- **Mô hình 1**

  Vì trước đó ta đã nhận thấy Previous Scores có ảnh hưởng lớn tới Performance Index, nên trong mô hình này, ta chỉ sử dụng đặc trưng này và bỏ qua các đặc trưng khác, đồng thời tăng số chiều của đặc trưng đó lên (bình phương giá trị của đặc trưng đó) để có nhiều thông tin hơn về mối quan hệ giữa đặc trưng này đến tập kết quả (bình phương đặc trưng sẽ nhận được các giá trị mới, từ đó nhận thấy sự thay đổi mới, cho ta góc nhìn/ thông tin mới hơn về dữ liệu).

- **Mô hình 2**

  Không chỉ Previous Scores, Hours Studied, Sleep Hours cũng có ảnh hưởng đến Performance Index, nên ta sử dụng 3 đặc trưng này một cách riêng biệt để xây dựng mô hình trong trường hợp này.

  Phương pháp sử dụng từng đặc trưng riêng lẻ, giúp ảnh hưởng của mỗi đặc trưng đến kết quả đáng kể hơn, nhưng có thể không đạt được độ chính xác cao nếu đặc trưng không đủ mạnh hoặc không đại diện tốt cho dữ liệu.

- **Mô hình 3**

  Bên cạnh sử dụng các đặc trưng riêng lẻ, ta có thể kết hợp các đặc trưng thành một đặc trưng mới để xây dựng mô hình. Trong trường hợp này, ta chọn các đặc trưng có ảnh hưởng nhiều đến Performance Index, đó là Previous Scores và Hours Studied.

  Kết hợp nhiều đặc trưng có thể cung cấp một cái nhìn đầy đủ hơn về dữ liệu và có thể phát hiện ra mối quan hệ phức tạp giữa các đặc trưng.

Trong các mô hình vừa xây dựng, ta cần tìm ra mô hình có chỉ số MAE nhỏ nhất (mô hình dự đoán tốt nhất), nhưng tương tự như câu 2b, ta không thể dùng tập test để tính chỉ số MAE, nên ta cần phải áp dụng phương pháp cross-validation, chia tập test thành 5 fold để train và test.

Để tăng tính tái sử dụng của code, ta cần định nghĩa một object chứa:

- Tên của mô hình: string. Ví dụ: '2c1'
- Các đặc trưng mà mô hình sử dụng: list. Ví dụ ['HS', 'PS']
- Mô tả về mô hình: string. Ví dụ: 'Model created by feature HS and PS'

Bằng cách này, ta chỉ cần xây dựng một dòng lặp duyệt qua list chứa các object mô hình, ở mỗi mô hình ta chọn các cột đặc trưng trong danh sách các đặc trưng mà mô hình sử dụng để xây dựng các tập train và tập test.

Tương tự với câu 2b, ta tạo ra tập train và tập train mới dựa trên tập train gốc được xáo trộn (trộn 1 lần duy nhất cho tất cả mô hình), tính chỉ số MAE sau mỗi vòng lặp và chỉ số MAE trung bình cho mỗi mô hình. Sau khi duyệt qua tất cả các mô hình, ta tìm được mô hình tốt nhất với chỉ số MAE nhỏ nhất.

Trong trường hợp này, mô hình tốt nhất là mô hình số 2, sử dụng riêng lẻ các đặc trưng Previous Scores, Hours Studied, Sleep Hours.

Ta huấn luyện lại mô hình dựa trên toàn bộ tập test dựa trên danh sách các đặc trưng trả về phía trên, sau đó tính chỉ số MAE của mô hình dựa trên tập test.

STT	Mô hình	MAE
1	Bình phương đặc trưng Previous Score	6.619
2	Dùng 3 đặc trưng riêng biệt Hours Studied and Previous Scores, Sleep Hours	1.702
3	Kết hợp 2 đặc trưng Hours Studied và Previous Scores thành đặc trưng mới	1.816

Mô hình 2 có chỉ số MAE nhỏ nhất, vì thế nên ta kết luận mô hình dùng 3 đặc trưng riêng biệt Hours Studied and Previous Scores, Sleep Hours đó là mô hình tốt nhất.

Công thức hồi quy tốt nhất sinh viên tự thiết kế:

$$\text{Student Performance} = -32.820 + 2.856 * (HoursStudied) + 1.018 * (PreviousScores) + 0.472 * (SleepHours)$$

Chỉ số MAE cho mô hình tốt nhất sinh viên tự thiết kế: 1.694

- **Nhận xét:**
  - Mô hình 1 có chỉ số MAE lớn nhất trong số các mô hình được tạo, có nghĩa là việc tăng số chiều không cho ra một mô hình mới tốt hơn các mô hình trước đó, nguyên nhân có thể là do mức ảnh hưởng của đặc trưng đến Performance Index là tương đối lớn (chứng minh ở phần khám phá dữ liệu), vì thế nên việc này không tạo ra sự khác biệt quá lớn.
  - Mô hình 2 có chỉ số MAE nhỏ nhất, thể hiện rằng đây là mô hình tốt nhất, lí do là vì các đặc trưng được lựa chọn có ảnh hưởng riêng lẻ khá lớn đến dữ liệu, nhất là Previous Score – có hệ số tương quan với Performance Index lên đến 0.915.
  - Mô hình 3 có chỉ số MAE lớn hơn mô hình 2, tuy nhiên ta vẫn có thể xem đây là mô hình khá tốt, nguyên nhân làm cho mô hình này có chỉ số MAE lớn có thể là do đặc trưng Hours Studied có hệ số tương quan âm trong mối quan hệ với các đặc trưng còn lại trong mô hình, trong khi đó mô hình này lại dựa trên đặc trưng mới được tạo nên từ các đặc trưng được lựa chọn, có nghĩa là mô hình này không chỉ phụ thuộc vào các đặc trưng được chọn mà còn phụ thuộc vào mối quan hệ của các đặc trưng đó với nhau, vì thế làm cho chỉ số MAE tăng lên.
  - Mô hình hình sinh viên tự thiết kế có chỉ số MAE lớn hơn với mô hình sử dụng 5 đặc trưng ở câu 2a, điều này thể hiện rằng trong trường hợp các tập dữ liệu này, sử dụng càng nhiều đặc trưng riêng lẻ sẽ tạo ra mô hình có khả năng dự đoán tốt hơn (bởi nếu tạo đặc trưng mới từ nhiều đặc trưng, có khả năng chúng sẽ kiềm hãm nhau do mối quan hệ tương quan âm ở các đặc trưng)

3. Các hàm/ thư viện đã sử dụng

- **Các thư viện đã sử dụng**
  - Thư viện numpy, dùng để hỗ trợ việc tính toán, xử lí số liệu
  - Thư viện pandas, dùng để đọc dữ liệu từ file csv
  - Thư viện matplotlib và seaborn dùng để vẽ đồ thị và khám phá dữ liệu
- **Các hàm đã sử dụng/ xây dựng**
  - Đối với yêu cầu 1:
    - setup_plot_grid(num_plots, plot_size): Hàm này dùng để điều chỉnh số lượng biểu đồ, số lượng biểu đồ mỗi hàng và kích thước của mỗi biểu đồ, vị trí của biểu đồ.
    - plot_histograms(data): dùng để vẽ biểu đồ các cột
    - plot_line_plots(data, target_column): dùng để vẽ biểu đồ đường
    - plot_correlation_heatmap(data): dùng để vẽ ma trận tương quan
    - plot_scatter_plots(data, target_column): dùng để vẽ biểu đồ tán xạ

    Trong các hàm vẽ biểu đồ nói trên, ta truyền vào input (data) là dữ liệu từ file train.csv, input có kiểu dữ liệu là Dataframe. Target column là cột mục tiêu, trong trường hợp này nó là 'Performance Index', thể hiện mối liên hệ giữa các đặc trưng với cột này trong một số biểu đồ. Các hàm trên không có kiểu dữ liệu trả về, thay vào đó nó sẽ in ra biểu đồ vừa được tạo.

    Vì các hàm tương tự nhau, ta chỉ mô tả một hàm đại diện, trong trường hợp này, ta chọn hàm plot_scatter_plots(data, target_column):

    data: dữ liệu dạng DataFrame, là toàn bộ thông tin từ file train.csv.

    target_column: 'Performance Index'.

    Đầu tiên, ta điều chỉnh số lượng biểu đồ và kích thước biểu đồ, khu vực tương ứng của nó, chú ý bỏ qua đặc trưng Performance Index khi vẽ biểu đồ tán xạ, vì vậy ta giảm số lượng biểu đồ cần vẽ bằng cách lấy tổng số cột trừ đi cột mục tiêu:

    ```
    fig, axes = setup_plot_grid(len(data.columns) - 1)
    ```

    Vẽ biểu đồ với các thông số tương ứng với từng đặc trưng:

    ```
    for i, column in enumerate(data.columns[:-1]):
        sns.scatterplot(data=data, x=column, y=target_column, ax=axes[i])
    ```

    Trong đó, data là dữ liệu từ file train.csv, column là đặc trưng hiện tại, target_column là cột mục tiêu = 'Performance Index', ax là khu vực để vẽ biểu đồ.

Để tăng tính trực quan, ta tiến hành đặt tên cho biểu đồ cũng như các cột của nó

```
axes[i].set_title(f'Biểu đồ Tán Xạ giữa \n {column} và {target_column}')
axes[i].set_xlabel(column)
axes[i].set_ylabel(target_column)
```

Vì setup_plot_grid sẽ tạo ra số lượng biểu đồ là bội số của số dòng thể hiện biểu đồ (có thể hiểu là mỗi dòng đều được mặc định được lấp đầy bởi các biểu đồ, kể cả biểu đồ rỗng), ta cần xóa bỏ các biểu đồ không có giá trị:

```
for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])
```

Cuối cùng, ta hiển thị các biểu đồ lên màn hình bằng hàm show() có sẵn của matplotlib, có thể dùng thêm hàm tight_layout() để điều chỉnh, tối ưu vị trí các biểu đồ, đảm bảo các cột, tên biểu đồ,... không bị chồng chéo lên nhau.

```
plt.tight_layout()
plt.show()
```

- Đối với yêu cầu 2
  - preprocess(x): Hàm này dùng để tiền xử lí dữ liệu, input là một numpy array và output là numpy array sau khi được tiền xử lí. Dữ liệu sau khi đi vào hàm này, được biến đổi cho phù hợp với định dạng, tham số của mô hình mà ta xây dựng, đồng thời bổ sung thêm tham số tự do cho mô hình.
  - Class OLSLinearRegression: class dùng để thực hiện hồi quy tuyến tính sử dụng phương pháp Bình phương tối thiểu và quản lý các thông số của mô hình. Các hàm và thực thi hàm dựa trên kiến thức đã học về bình phương tối thiểu và ý tưởng của code ví dụ trong lab04 – Lớp Toán ứng dụng và thống kê cho Công nghệ thông tin 22CLC10.
    Các hàm thuộc class trên là:
    - fit(self, X, y): Hàm này dùng để tìm bộ tham số tối ưu theo dữ liệu. Input là các numpy array được tạo nên từ tập đặc trưng và tập mục tiêu (tập đặc trưng cần thông qua tiền xử lí trước đó). Kết quả trả về là bản thân mô hình, nhưng các tham số w được thay thế bởi hằng số, tìm được dựa trên dữ liệu X và y.

      ```
      X_pinv = np.linalg.inv(X.T @ X) @ X.T
      self.w = X_pinv @ y
      ```

    - get_params(self): Dùng để lấy các tham số của mô hình sau khi tìm được bộ tham số tối ưu theo dữ liệu.
    - predict(self, X): Dùng để dự đoán các giá trị y trên tập test sau khi có mô hình với các tham số tối ưu. Input truyền vào là numpy array chứa giá trị của các đặc trưng, hàm trả về output là các giá trị y dự đoán bằng cách thế giá trị của các đặc trưng trong input vào mô hình.

- calc_MAE(y_true, y_pred):Dùng để tính chỉ số MAE. Input truyền vào là 2 numpy array, tương ứng với giá trị y từ tập test và giá trị y dự đoán ứng với tập test được tính toán từ mô hình trước đó. Hàm trả về giá trị chênh lệch trung bình giữa 2 input, là một số thực dương.

```
MAE = np.mean(np.abs(y_true - y_pred))
```

- shuffle_index(size: int): Hàm này dùng để xáo trộn mảng, nhưng không phải mảng test hay train mà là mảng các index, hàm này được dùng cho yêu cầu 2b và 2c. Input truyền vào là size - một số nguyên, biểu diễn kích thước của mảng cần xáo trộn. Ta tạo một mảng chứa các số nguyên từ 0 đến size – 1, xem đây là mảng chứa các index của mảng cần xáo trộn. Tiếp đó, ta dùng hàm shuffle random có sẵn trong numpy để xáo trộn các phần tử trong mảng index, ouput của hàm này là mảng Index sau khi xáo trộn.

```
shuffled_index = np.arange(size) #tạo mảng từ 0 đến size-1
np.random.shuffle(shuffled_index)
```

- Đối với yêu cầu 2c
  - preprocess_2c(x, name_model): chức năng tương tự với preprocess trước đó, tuy nhiên hàm này giúp ta dễ dàng quản lý việc tiền xử lí từng model hơn, vì có thêm tham số name_model – tên của model. Từng trường hợp model ta có cách tiền xử lí riêng biệt

```
if name_model == '2c1':
    X = np.hstack((np.ones((x.shape[0], 1)), x**2))
if name_model == '2c2':
    X = np.hstack((np.ones((x.shape[0], 1)), x))
if name_model == '2c3':
    X = np.zeros((x.shape[0], 1))
    for i in range(x.shape[1]):
        X[:, 0] = X[:, 0] + x[:, i]
    X = np.hstack((np.ones((x.shape[0], 1)), X))
```

Nếu là model ‘2c1’, ta tăng số chiều của đặc trưng và thêm tham số tự do w0.
Nếu là model ‘2c2’ (sử dụng 3 đặc trưng riêng lẻ), ta chỉ cần thêm tham số tự do w0, sau đó truyền mảng X – chứa giá trị của đặc trưng tương ứng vào.
Nếu là model ‘2c3’, ta cộng giá trị của các đặc trưng lại để tạo thành đặc trưng mới, sau đó kết hợp với tham số tự do để tạo mô hình.

4. Tài liệu tham khảo

[1] The expertise reversal effect. (apa.org) (Ngày truy cập: 09/08/2024).

[2] Khái niệm và đặc điểm của ma trận tương quan (Ngày truy cập: 10/08/2024).

[3] Predicting academic success in higher education: literature review and best practices (Ngày truy cập: 09/08/2024).