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 3:
LINEAR REGRESSION

Môn: Toán ứng dụng và thống kê

Lớp: 22CLC03

Sinh viên thực hiện:

Hoàng Bảo Khanh (22127183)

Giáo viên hướng dẫn:

ThS. Phan Thị Phương Uyên

ThS. Nguyễn Văn Quang Huy

Mục lục

1. Giới thiệu đồ án

Trong đồ án lần này, em sẽ viết chương trình để tìm hiểu các yếu tố ảnh hưởng đến thành tích học tập của sinh viên (Academic Student Performance Index) từ nguồn dữ liệu Student Performance. Các yếu tố ảnh hưởng là các đặc trưng được lưu theo các kiểu dữ liệu theo bảng sau:

STT	Thuộc tính	Mô tả	Kiểu dữ liệu
1	Hours Studied	Tổng số giờ học của mỗi sinh viên	Integer
2	Previous Scores	Điểm số học sinh đạt được trong các bài kiểm tra trước đó	Integer
3	Extracurricular Activities	Sinh viên có tham gia hoạt động ngoại khóa không (Có hoặc Không)	Boolean
4	Sleep Hours	Số giờ ngủ trung bình mỗi ngày của sinh viên	Integer
5	Sample Question Papers Practiced	Số bài kiểm tra mẫu mà học sinh đã luyện tập	Integer
6	Performance Index	Thước đo thành tích tổng thể cho mỗi sinh viên. Chỉ số thể hiện thành tích học tập, nằm trong đoạn [10, 100]. Chỉ số này tỉ lệ thuận với thành tích	Float

Bảng 1: Bảng các yếu tố ảnh hưởng đến thành tích học tập

Dữ liệu trên đã được thực hiện tiền xử lý chuyển đổi kiểu dữ liệu cho thuộc tính Extracurricular Activities, bộ dữ liệu được chia ngẫu nhiên thành 2 tập với tỉ lệ 9:1. Trong đó 9 phần cho tập huấn luyện (train.csv) chứa 9000 mẫu và 1 phần cho tập kiểm tra (test.csv) chứa 1000 mẫu.

Trong đồ án này, em thực hiện các yêu cầu sau:

- Thực hiện phân tích khám phá dữ liệu: sử dụng biểu đồ (bar, box, heatmap, scatter, line, …) để phân tích và quan sát các đặc trưng

- Xây dựng mô hình dự đoán chỉ số thành tích sử dụng mô hình hồi quy tuyến tính

2. Các thư viện sử dụng trong đồ án

Trong đồ án lần này, chương trình sử dụng các thư viện sau:

- Thư viện pandas: Chương trình sử dụng thư viện trên để có thể xử lý trên các tập dữ liệu.
- Thư viện matplotlib: Thư viện trên hỗ trợ trong việc hiển trị cài đặt trục số, tên biểu đồ.
- Thư viện numpy: Thư viện numpy hỗ trợ chương trình trong việc thực hiện các phép tính toán các ma trận.
- Thư viện seaborn: giúp hỗ trợ vẽ các biểu đồ.

3. Các hàm đã cài đặt và đã sử dụng trong đồ án

a. Liệt kê tất cả các hàm

Trong chương trình, các hàm đã được cài đặt là:

- Hàm preprocess: là quá trình tiền xử lý dữ liệu, hàm được sử dụng để thêm 1 cột toàn giá trị vào dữ liệu đầu vào.
- Class OLSLinearRegression:
    - Hàm fit: được dùng để fit dữ liệu bằng phương pháp Ordinary Least Squares.
    - Hàm get_params: hàm lấy các tham số từ mô hình.
    - Hàm predict: hàm được sử dụng để dự đoán đầu ra của mô hình.
- Hàm mae: để tính sai số tuyệt đối trung bình (MAE).
- Hàm k_fold_CrossValidation: hàm thực hiện chia dữ liệu với số lượng tệp dữ liệu cho trước.
- Hàm implement_model: hàm được dùng để xây dựng và thiết kế các mô hình (số lượng mô hình là 3).

Ngoài ra trong đồ án em còn sử dụng các hàm có sẵn từ các thư viện được thống kê quả bảng sau:

Hàm sử dụng từ thư viện	Mô tả
pd.read_csv	hàm đọc dữ liệu từ file .csv.
pandas.DataFrame.iloc	hàm dùng để lấy các đặc trưng.

pandas.DataFrame.describe	Hàm dùng để thống kê dữ liệu, với các thông tin như số mẫu dữ liệu, giá trị trung bình, giá trị lớn nhất và nhỏ nhất và các giá trị tứ phân vị.
plt.figure	hàm dùng để điều chỉnh kích thước của ảnh biểu đồ.
plt.tittle	hàm dùng để biểu thị tên biểu đồ
plt.show	hàm dùng để thể hiện biểu đồ
sns.heatmap, sns.scatterplot	hàm dùng để vẽ lần lượt các biểu đồ nhiệt, phân tán
min	hàm dùng để tìm giá trị nhỏ nhất
pd.concat	dùng để ghép nối các đối tượng (pandas) dọc theo một trục cụ thể.
pandas.DataFrame.sample	hàm dùng để xáo trộn các dòng dữ liệu

Bảng 1: Bảng mô tả các hàm sử dụng từ thư viện

b. Ý tưởng thực hiện và mô tả các hàm

1. Hàm preprocess:

Hàm preprocess là hàm được em tham khảo từ file lab04.ipynb của cô Phương Uyên.

Ý tưởng: hàm sẽ thực hiện giai đoạn tiền xử lý, tức là hàm sẽ thêm 1 cột toàn giá trị 1 vào dữ liệu đầu vào, vì đối với trong mô hình hồi quy tuyến tính, việc thực hiện hành động trên giúp mô hình tìm ra được hệ số tự do.

Đầu vào: đầu vào của hàm là một tập dữ liệu (DataFrame).

Đầu ra: đầu ra là một tập dữ liệu sau khi được thêm cột.

Mô tả: hàm sử dụng hàm np.hstack() với tham số đầu vào là ma trận toàn 1 cùng số dòng với dữ liệu đầu vào và giá trị của đầu vào.

2. Class OLSLinearRegression:

Class OLSLInearRegression là class mà em cũng tham khảo từ file lab04.ipynb của cô Phương Uyên.

Bên trong class sẽ bao gồm 3 hàm: hàm fit, get_params và predict.

a. Hàm fit

Ý tưởng: hàm sẽ thực hiện fit các dữ liệu bằng phương pháp Ordinary Least Squares với công thức như sau:

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

Hình 1: Công thức tính trọng số (x) khi có đặc trưng đầu vào A và đầu ra b

Từ công thức trên và sử dụng hàm numpy, ta có được hàm fit.

Đầu vào: dữ liệu input và dữ liệu output

Đầu ra: hàm sẽ trả về đối tượng của class.

Mô tả: hàm sẽ dùng hàm np.linalg.inv() để trợ giúp cho việc tìm ma trận giả nghịch đảo của dữ liệu input, sau đó lấy kết quả trên nhân với dữ liệu output.

b. Hàm get_params

Ý tưởng: hàm được dùng để lấy các trọng số đã được tính sau khi thực hiện hàm fit.

Đầu ra: Một ma trận chứa các trọng số.

Mô tả: hàm sẽ trả về một ma trận với các trọng số.

c. Hàm predict

Ý tưởng: hàm được dùng để dự đoán giá trị đầu ra của mô hình, bằng cách lấy các trọng số nhân với dữ liệu đầu vào.

Đầu vào: dữ liệu input

Đầu ra: kết quả sau khi dự đoán

Mô tả: hàm sẽ nhân ma trận giữa dữ liệu input và mảng gồm các trọng số.

3. Hàm mae:

Hàm mae được sử dụng để ước lượng trung bình của sai số tuyệt đối.

Ý tưởng: hàm sẽ tính giá trị sai số tuyệt đối trung bình từ công thức:

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

Hình 2: Công thức tính MAE

Với n là số lượng mẫu quan sát, y_i là giá trị mục tiêu của mẫu thứ i và $\widehat{yi}$ là giá trị mục tiêu thứ i đã được dự đoán từ mô hình hồi quy tuyến tính.

Đầu vào: dữ liệu output và dữ liệu output đã được dự đoán

Đầu ra: giá trị MAE

Mô tả: hàm sẽ dùng np.mean() để tính giá trị cần tìm sau khi dữ liệu output và dữ liệu dự đoán đã được làm phẳng liên tục qua hàm np.ravel()

4. Hàm k_fold_CrossValidation

Ý tưởng; hàm sẽ chia dữ liệu tập train đầu vào thành 5 tập dữ liệu với kích thước bằng nhau. Tập train ban đầu có 9000 mẫu dữ liệu, do đó 5 tập con sẽ bao gồm 1800 mẫu dữ liệu. Hàm sẽ thực hiện vòng lặp để thêm các dữ liệu với các index tương ứng: index từ 0 đến 1799 cho tập 1, 1800 đến 3599 cho tập 2, 3600 đến 5399 cho tập 3, 5400 đến 7199 cho tập 4 và 7200 đến 8999 cho tập 5.

Đầu vào: là dữ liệu bị chia.

Đầu ra: là một danh sách các tập con

Mô tả: hàm sẽ thực hiện vòng lặp và thêm từng tập con với các giá trị index tương ứng vào danh sách.

5. Hàm heat_map

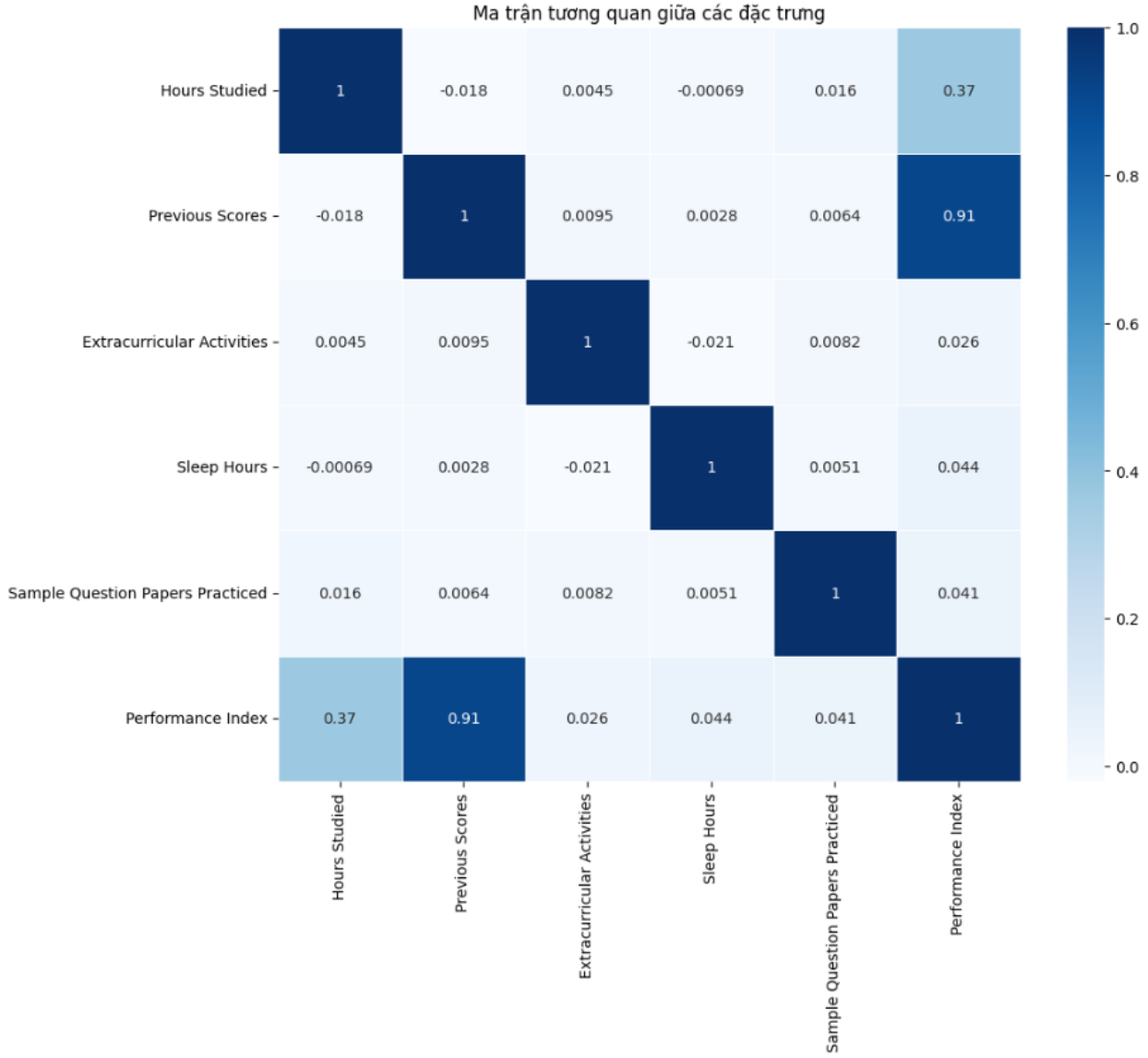
Ý tưởng: hàm dùng để tạo ra ma trận tương quang (được biểu diễn bằng biểu đồ nhiệt), các giá trị trong từng ô sẽ là giá trị tương quan giữa các đặc trưng.

Đầu vào: dữ liệu input

Đầu ra: biểu đồ nhiệt

Mô tả: Hàm trước tiên sẽ tạo kích thước của ảnh, sau đó sử dụng hàm sns.heatmap với các giá trị tương quan được tính bằng hàm corr().

Hình ảnh kết quả sau khi thực hiện hàm:



Hình 3: Biểu độ nhiệt (ma trận tương quan) giữa các đặc trưng

Nhận xét về các đặc trưng từ biểu đồ:

- Hệ số tương quan giữa Previous Scores và Performance Index rất cao, khoảng 0.91. Điều này cho thấy nếu học sinh có điểm số bài kiểm tra trước đó càng cao thì họ có khả năng đạt thành tích học tập tốt.
- Hệ số tương quan giữa Hours Studied và Performance Index khá cao với 0,37. Điều này cho thấy nếu học sinh có số giờ học tập càng nhiều thì họ có thể đạt thành tích tốt.
- Bên cạnh đặc trưng Previous Scores và Hours Studied có sự tương quan tốt với Performance Index, thì các đặc trưng còn lại không có mối quan hệ rõ ràng. Do đó

sinh viên có số giờ ngủ nhiều, hay tham gia hoạt động ngoại khóa hoặc số bài kiểm tra mẫu đã luyện tập không ảnh hưởng đáng kể đến thành tích học tập.

6. Hàm scatter_plot:

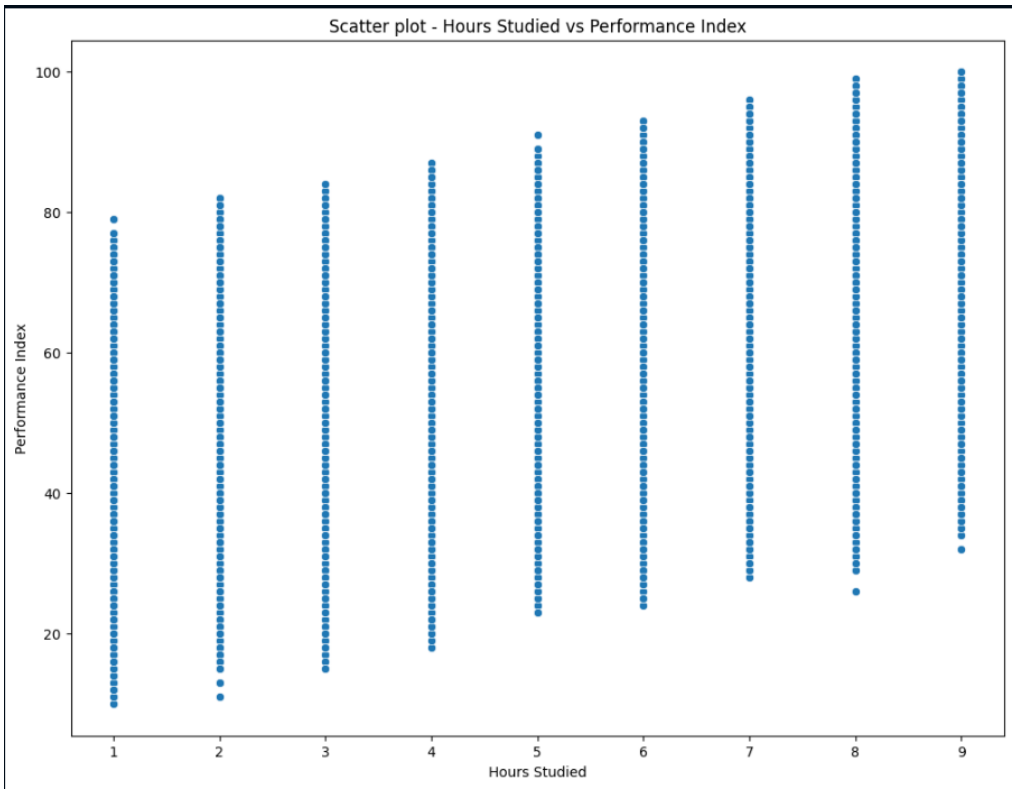
Ý tưởng: hàm dùng để vẽ biểu đồ phân tán dữ liệu của từng đặc trưng đối với Performance Index.

Đầu vào: dữ liệu input

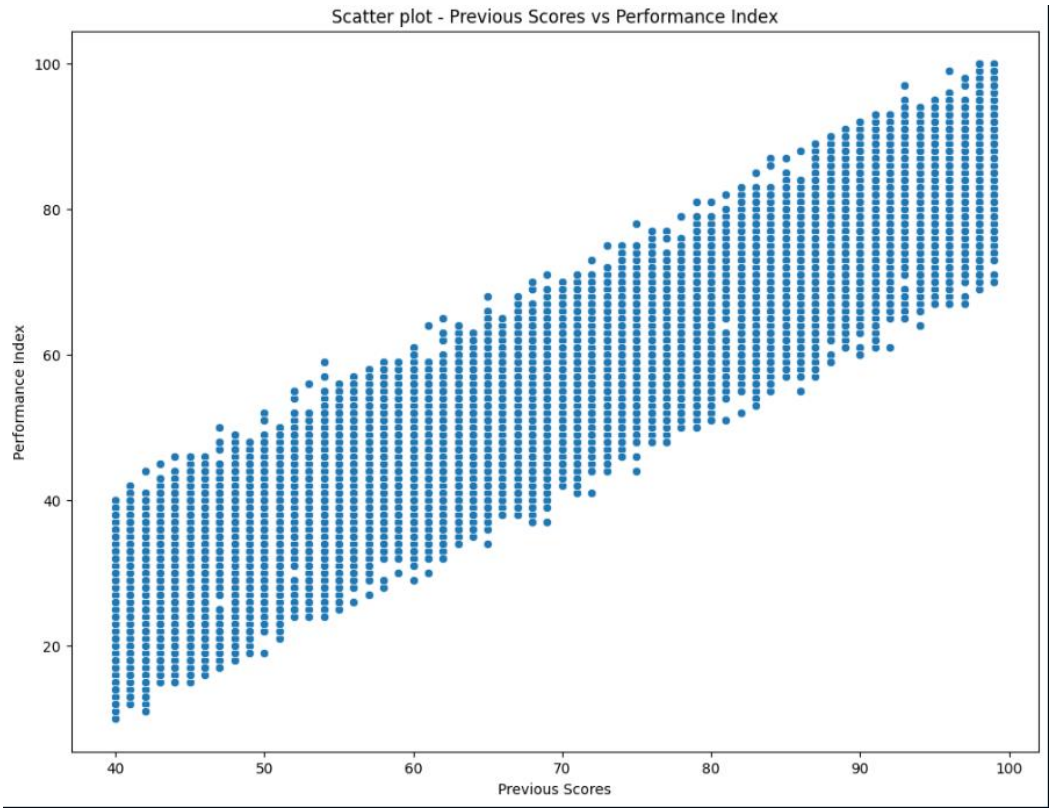
Đầu ra: các biểu đồ phân tán của từng đặc trưng đối với Performance Index

Mô tả: tương tự với hàm bar_plot, hàm sẽ dùng hàm sns.scatterplot để vẽ biểu đồ phân tán của từng đặc trưng được duyệt qua vòng lặp.

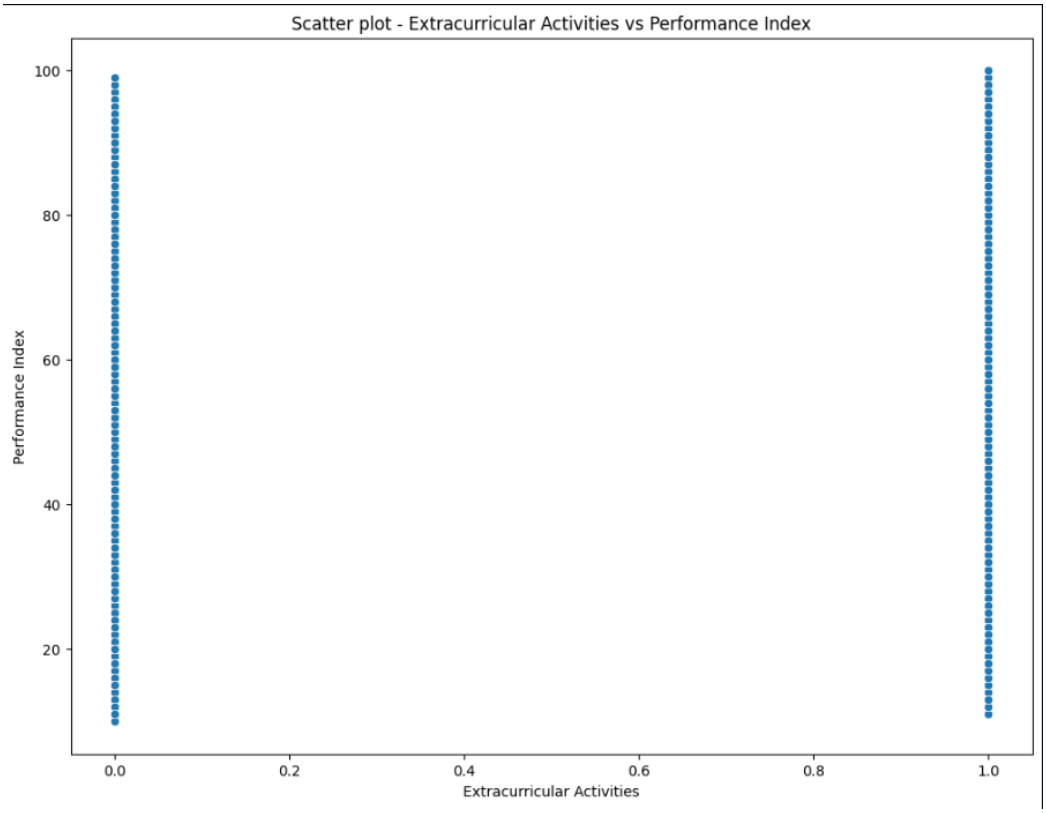
Dưới đây là các biểu đồ phân tán của từng đặc trưng đối với Performance Index:



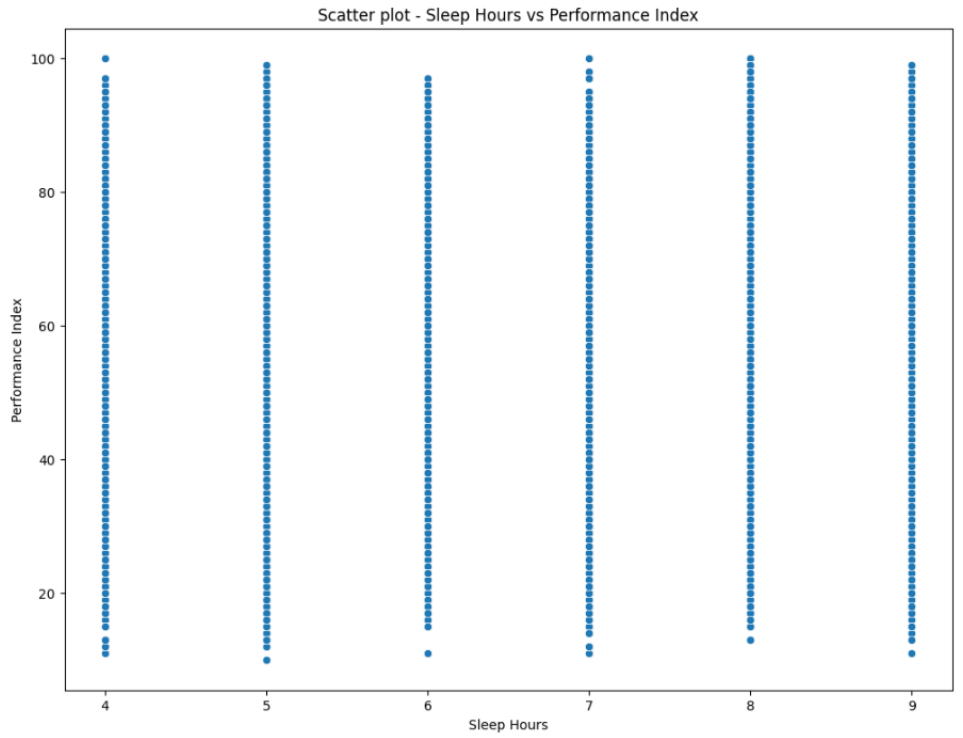
Hình 4: Biểu đồ phân tán của đặc trưng Hours Studied



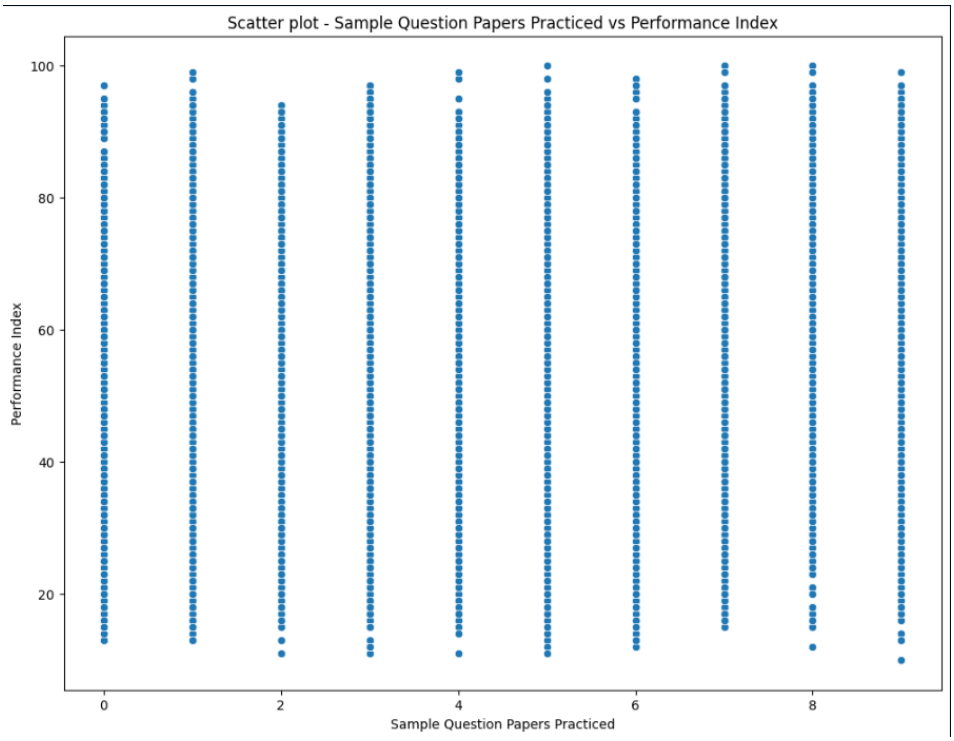
Hình 5: Biểu đồ phân tán của đặc trưng Previous Scores



Hình 6: Biểu đồ phân tán của đặc trưng Extracurricular Activities



Hình 7: Biểu đồ phân tán của đặc trưng Sleep Hours



Hình 8: Biểu đồ phân tán của đặc trưng Sample Question Papers Practiced

Nhận xét các đặc trưng từ biểu đồ:

- Từ 5 biểu đồ trên, ta có thể nhận thấy được rằng với các giá trị của đặc trưng Hours Studied và Previous Scores càng tăng thì giá trị Performance Index cũng tăng. Do đó, hai đặc trưng Hours Studied và Previous Scores có sự ảnh hưởng rõ ràng đến thành tích học tập của sinh viên.
- Ngoài ra, 3 biểu đồ còn lại Extracurricular Activities, Sleep Hours và Sample Question Papers Practiced cho thấy giá trị của Performance Index có sự dao động nhưng không đáng kể. Khi đó, 3 đặc trưng này sẽ ít sự ảnh hưởng hơn đến thành tích học tập.

7. Hàm implement_model

Ý tưởng: hàm này được cài đặt với mục đích xây dựng và thiết kế mô hình. Trong chương trình em sẽ tạo ra 3 mô hình chính. Nhận xét từ các biểu đồ đã vẽ ở trên, ta thấy được rằng hai đặc trưng có mối quan hệ rõ ràng nhất đối với Performance Index là Previous Scores và Hours Studied. Do đó việc thiết kế và tạo mô hình của chương trình sẽ dựa vào chủ yếu hai đặc trưng này. Chi tiết các mô hình sẽ được trình bày ở mục báo cáo và nhận xét kết quả mô hình xây dựng.

Đầu vào: dữ liệu từ tập train và có một giá trị index nhằm để nhận biết sử dụng mô hình nào.

Đầu ra: đầu ra của hàm là một DataFrame sau khi thực hiện kết hợp các đặc trưng.

Mô tả: Hàm sẽ thực hiện ứng với mỗi giá trị index. Khi index là 1 thì hàm sẽ thực hiện thiết kế mô hình cho mô hình 1, tương tự với mô hình 2, 3 với index lần lượt là 2 và 3. Sau đó, hàm sẽ dùng lệnh pd.concat() để ghép nối các đặc trưng cần huấn luyện lại thành 1 DataFrame hoàn chỉnh.

4. Báo cáo và nhận xét kết quả các mô hình xây dựng

1. Yêu cầu 2a: Huấn luyện 1 lần duy nhất cho 5 đặc trưng cho toàn bộ tập huấn luyện

Để thực hiện yêu cầu 2a, trước tiên ta sẽ thực hiện quá trình tiền xử lý của tập dữ liệu X_train (là tập bao gồm các đặc trưng trong file train.csv). Sau đó, chương trình sẽ tiến hành huấn luyện tập dữ liệu X_train và y_train để tìm ra các trọng số.

Sau khi huấn luyện, ta sẽ có được phương trình hồi quy tuyến tính như sau:

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

Sau đó, để tìm giá trị MAE, ta phải tìm được dữ liệu dự đoán từ mô hình trên. Trước tiên, chương trình sẽ thực hiện tiền xử lý dữ liệu X_test (là tập bao gồm các đặc trưng trong file test.csv). Sau đó, chương trình thực hiện hàm predict và mae để tìm sai số tuyệt đối trung bình.

Kết quả sai số tuyệt đối trung bình sau khi thực hiện huấn luyện 1 lần duy nhất cả 5 đặc trưng là xấp xỉ 1.59565.

Nhận xét kết quả: Với giá trị sai số là 1.59565, ta nhận xét được rằng sai số trên là sai số nhỏ, do đó mô hình huấn luyện cả 5 đặc trưng 1 lần duy nhất lý tưởng trong việc nhận xét thành tích học tập của sinh viên.

2. Yêu cầu 2b: Xây dựng mô hình duy nhất 1 đặc trưng, tìm mô hình cho kết quả tốt nhất

Tại yêu cầu 2b, chương trình sẽ xây dựng mô hình cho duy nhất 1 đặc trưng, rồi tìm ra mô hình cho kết quả tốt nhất.

Trước khi thực hiện huấn luyện cho từng đặc trưng, cụ thể là 5 đặc trưng Hours Studied, Previous Scores, Extracurricular Activities, Sleep Hours, Sample Question Papers Practiced, chương trình sẽ tiến hành xáo trộn tập dữ liệu train khi dùng hàm sample.

Sau khi xáo trộn các dòng dữ liệu trong tập train, chương trình sẽ sử dụng k-fold Cross Validation để chia tập dữ liệu train lớn ban đầu thành k tập con với kích thước bằng nhau. Như đã đề cập ở trên, chương trình sẽ mặc định chọn số tập con là 5, do đó sau khi thực hiện k-fold Cross Validation, ta sẽ có được 5 tập con với kích thước 1800 mẫu dữ liệu.

Chương trình sẽ duyệt qua từng đặc trưng, mỗi đặc trưng sẽ được huấn luyện với 5 tập dữ liệu con đã được chia, từ đó giá trị sai số tuyệt đối trung bình của đặc trưng được tính bằng trung bình cộng các sai số tuyệt đối trung bình ở cả 5 tập.

Sau khi huấn luyện với 5 tập con, ta có được kết quả sau:

STT	Mô hình với 1 đặc trưng	MAE
1	Hours Studied	15.451
2	Previous Scores	6.619
3	Extracurricular Activities	16.195
4	Sleep Hours	16.188

5	Sample Question Papers Practices	16.185

Bảng 2: Bảng kết quả MAE ứng với từng đặc trưng

Dựa vào kết quả, ta nhận xét được rằng với mô hình 1 đặc trưng của Previous Scores là mô hình tốt nhất, với sai số là 6.619, thấp nhất đáng kể so với các đặc trưng còn lại. Lí do để có được sai số thấp như trên là vì đặc trưng Previous Scores có hệ số tương quan rất cao so với Performance Index. Do đó việc huấn luyện đặc trưng Previous Scores sẽ tốt hơn so với các đặc trưng khác và xảy ra ít sai số hơn.

Sau khi tìm được đặc trưng tốt nhất là Previous Scores, chương trình sẽ tiến hành huấn luyện trên toàn bộ tập huấn luyện (tập train). Để tìm đặc trưng tốt nhất, tương ứng với giá trị mae nhỏ nhất, chương trình sẽ dùng hàm min. Từ đó, chương trình sẽ tiến hành giai đoạn tiền xử lý dữ liệu và huấn luyện mô hình để có được các trọng số.

Ta có được phương trình hồi quy tuyến tính sau khi huấn luyện như sau:

Student Performance = -14,989 + 1,011.Previous Scores

Tương tự như yêu cầu 2a, chương trình sẽ tìm sai số tuyệt đối trung bình khi huấn luyện trên toàn bộ tập huấn luyện. Kết quả của sai số là xấp xỉ 6.54428.

Nhận xét kết quả: Tuy sai số tuyệt đối trung bình khi huấn luyện 1 đặc trưng tốt nhất (Previous Scores) khác đi đáng kể so với các đặc trưng còn lại, nhưng sai số trên vẫn là quá lớn so với mô hình ở yêu cầu 2a. Do đó ta thấy được rằng, việc có càng nhiều đặc trưng (cụ thể là 5) thì mô hình sẽ có nhiều cơ sở, căn cứ và thông tin hơn trong việc dự đoán dữ liệu đầu ra, từ đó sự chính xác trong việc dự đoán của mô hình sẽ càng được cải thiện.

3. Yêu cầu 2c: Tự xây dựng, thiết kế mô hình, tìm mô hình cho kết quả tốt nhất

Tại yêu cầu 2c, chương trình sẽ tự xây dựng, thiết kế mô hình và tìm ra mô hình cho kết quả tốt nhất. Việc xây dựng được thực hiện qua hàm implement_model.

Về ý tưởng xây dựng mô hình, em nhận xét được từ ma trận tương quan (biểu đồ nhiệt) giữa các đặc trưng có nêu ở phần trên, thì đặc trưng Hours Studied và Previous Scores có mối quan hệ rõ ràng hơn với Performance Index so với các đặc trưng còn lại. Do đó, việc xây dựng mô hình sẽ xoay quanh hai đặc trưng này là chính.

Đối với mô hình 1, em sẽ tạo ra tập dữ liệu với các đặc trưng lần lượt là bình phương của 5 đặc trưng ban đầu. Lí do em tạo ra mô hình này bởi em muốn nhận xét xem việc bình phương các giá trị đặc trưng có tác động mạnh mẽ đến biến mục tiêu (Performance Index) hay không, đồng thời có thể so sánh được mô hình 1 (với mối quan hệ bậc 2) và mô hình ở yêu cầu 2a (với mối quan hệ bậc 1), mô hình nào lý tưởng hơn đến huấn luyện dữ liệu.

Đối với mô hình 2, em sẽ kết hợp hai đặc trưng có tính tương quan mạnh nhất là Hours Studied và Previous Scores. Chương trình sẽ thực hiện huấn luyện mô hình 2 với 2 đặc trưng trên cùng 1 lần để nhận xét xem việc lược bỏ các đặc trưng có sự tương quan yếu có ảnh hưởng như thế nào đến sự chính xác của mô hình.

Đối với mô hình 3, em vẫn sẽ kết hợp hai đặc trưng Hours Studied và Previous Scores. Tuy nhiên lần này mô hình sẽ tạo ra hai đặc trưng mới, đối với đặc trưng mới thứ nhất sẽ được tính bằng cách nhân 2 đặc trưng trên lại với nhau, và đặc trưng mới thứ hai sẽ là cộng của 2 đặc trưng trên. Việc lựa chọn trên là nhằm để kiểm tra xem giữa Hours Studied và Previous Scores có tính cộng hưởng lẫn nhau hay không, và kiểm tra xem việc kết hợp như vậy có thể cải thiện giá trị MAE so với các mô hình trước hay không.

Sau khi thiết kế được 3 đặc trưng, thì chương trình sẽ tiến hành xáo trộn dữ liệu, và chia ra thành 5 tập dữ liệu con tương tự như yêu cầu 2b. Các giá trị MAE cuối cùng ứng với từng mô hình sẽ là trung bình cộng của các giá trị sai số sau khi huấn luyện trên toàn bộ 5 tập con của dữ liệu.

Dưới đây là kết quả MAE có được tương ứng với từng mô hình:

STT	Mô hình	MAE
1	Bình phương từng đặc trưng	2.6488
2	Kết hợp 2 đặc trưng Hours Studied và Previous Scores	1.81628
3	Tạo ra 2 đặc trưng mới là tổng và tích của 2 đặc trưng Hours Studied và Previous Scores	2.0667

Bảng 3: Bảng kết quả ở yêu cầu 2c ứng với từng mô hình

Từ bảng trên, ta có thể dễ dàng nhận thấy rằng, độ lỗi thấp nhất có được là của mô hình 2. Tuy nhiên, ta có thể nhận thấy được rằng giá trị MAE ở cả 3 mô hình đều khá khiêm tốn, đặc biệt là so với sai số của từng đặc trưng ở yêu cầu 2b khi huấn luyện trên 5 tập dữ liệu con.

Từ đó ta rút ra được, trong 3 mô hình trên, mô hình có kết quả tốt nhất là mô hình 2, kết hợp huấn luyện 2 đặc trưng Hours Studied và Previous Scores. Tiếp theo, chương trình sẽ thực hiện huấn luyện mô hình 2 trên toàn bộ tập huấn luyện để tìm ra các trọng số. Các bước thực hiện tương tự như yêu cầu 2b.

Từ đó ta có được phương trình hồi quy tuyến tính ở mô hình 2 là như sau:

Student Performance = -29,747 + 2,856.Hours Studied + 1,018.Previous Scores

Sau đó, chương trình sẽ tìm sai số tuyệt đối để kiểm tra độ chính xác của mô hình. Kết quả sai số thu được là 1.83944.

Nhận xét kết quả: Sai số ở mô hình 2a không quá lớn, từ đó cho thấy mô hình 2 có độ chính xác cao hơn và lý tưởng hơn so với 2 mô hình còn lại.

❖ **Nhận xét kết quả trên toàn bộ mô hình:**

Sau khi huấn luyện mô hình 2 trên toàn bộ tập huấn luyện, ta nhận thấy được rằng sai số ở mô hình trên cũng không quá cao. Từ đó ta rút ra được những nhận xét như sau:

- Ở cả 3 mô hình mà em tự thiết kế, thì cả 3 mô hình trên đều lý tưởng hơn so với mô hình huấn luyện 1 đặc trưng ở yêu cầu 2b, tuy nhiên lại không chính xác hơn so với mô hình huấn luyện toàn bộ 5 đặc trưng ở yêu cầu 2a. Do đó về tổng thể tất cả các mô hình đã trình bày trong báo cáo này, thì mô hình ở yêu cầu 2a cho ra kết quả chính xác hơn so với các mô hình còn lại.
- Việc lược bỏ đi các đặc trưng có sự tương quan yếu không làm giảm đi độ lỗi của mô hình, thậm chí còn tăng lên. Qua đó em nhận thấy được việc huấn luyện càng nhiều đặc trưng khác nhau sẽ dẫn tới mô hình sẽ có được nhiều dữ liệu, thông tin hơn, từ đó có thể xây dựng được mô hình có độ chính xác cao hơn.
- Việc bình phương từng đặc trưng và huấn luyện như mô hình 1 không cải thiện được sai số so với mô hình ở yêu cầu 2a, do đó em nhận thấy mối quan hệ giữa 5 đặc trưng so với biến mục tiêu là Performance Index là tuyến tính, tức cả 5 đặc trưng đều có sự ảnh hưởng nhất định đến thành tích học tập cuối cùng.

5. Tài liệu tham khảo

[1]. Lab04.ipynb, Ms Phan Thị Phương Uyên

[2]. searborn, statistical data visualization,

https://seaborn.pydata.org/index.html, truy cập vào ngày 08/08/2024

[3]. pandas, pandas.concat,
https://pandas.pydata.org/docs/reference/api/pandas.concat.html, truy cập vào ngày 10/08/2024

[4]. pandas, pandas.DataFrame.corr,
https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.corr.html, truy cập vào ngày 11/08/2024

[5] matplotlib, matplotlib.pyplot,
https://matplotlib.org/3.5.3/api/_as_gen/matplotlib.pyplot.html, truy cập vào ngày 09/08/2024

[6] pandas, pandas.DataFrame.sample,
https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sample.html, truy cập vào ngày 11/08/2024

[7]. Shanthababu Pandian, Analytics Vidhya, K-Fold Cross Validation Technique and its Essentials

https://www.analyticsvidhya.com/blog/2022/02/k-fold-cross-validation-technique-and-its-essentials/, truy cập vào ngày 10/08/2024

[8]. IBM, What is Linear Regression?

https://www.ibm.com/topics/linear-regression#:~:text=IBM-,What%20is%20linear%20regression%3F,is%20called%20the%20independent%20variable., truy cập vào ngày 09/08/2024