ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

PROJECT 3

LINEAR REGRESSION

—o0o—

TÁC GIẢ
22127180 Nguyễn Phúc Khang
LỚP
22CLC08

—o0o—

GIẢNG VIÊN HƯỚNG DẪN
Vũ Quốc Hoàng
Nguyễn Văn Quang Huy
Nguyễn Ngọc Toàn
Phan Thị Phương Uyên

THÀNH PHỐ HỒ CHÍ MINH, 8 2024

LỜI CẢM ƠN

Tôi xin gửi lời cảm ơn chân thành đến các giảng viên Vũ Quốc Hoàng, Nguyễn Văn Quang Huy, Nguyễn Ngọc Toàn và Phan Thị Phương Uyên vì sự hỗ trợ và hướng dẫn tận tình trong suốt quá trình thực hiện đồ án này.

Bên cạnh đó, tôi cũng rất biết ơn các tác giả của các tài liệu tham khảo, những người đã đóng góp quan trọng vào việc làm sâu sắc thêm sự hiểu biết của tôi về các thuật toán. Những kiến thức của họ đã đóng vai trò quan trọng trong việc hoàn thành đồ án này một cách suôn sẻ.

MỤC LỤC

MỤC LỤC ẢNH

Nội dung đồ án

1.1 Mục tiêu

Mục tiêu của đồ án là tìm hiểu các yếu tố ảnh hưởng đến thành tích học tập của sinh viên (Academic Student Performance Index). Các yếu tố ảnh hưởng có thể là số giờ học tập/nghiên cứu, hoạt động ngoại khóa, số giờ ngủ, số bài kiểm tra mẫu đã luyện tập.

1.2 Mô tả dữ liệu

Dữ liệu thành tích sinh viên (Student Performance) có 10000 dòng và 6 cột. Ý nghĩa và kiểu dữ liệu của từng cột được thể hiện ở bảng sau:

SST	Thuộc tính	Mô tả	Kiểu dữ liệu
1	Hours Studied	Tổng số giờ học của mỗi sinh	Integer
2	Previous Scores	Điểm số học sinh đạt được trong các bài kiểm tra trước	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	Integer
5	Sample Question Papers Practiced	Số bài kiểm tra mẫu mà học sinh đã luyện	Integer
6	Performance Index	Thước đó 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

Table 1.1: Bảng mô tả kiểu dữ liệu của từng thuộc tính.

Trong đồ án này, dữ liệu trên đã được thực hiện tiền xử lý chuyển đổi kiểu dữ liệu cho thuộc tính Extracurricular Activities. Sau khi tiền xử lý, bộ dữ liệu được chia ngẫu nhiên thành 2 tập với tỉ lệ 9:1. Trong đó, 9 phần cho tập huấn luyện, 1 phần cho tập kiểm tra.

- **train.csv**: Chứa 9000 mẫu dùng để huấn luyện mô hình.
- **test.csv**: Chứa 1000 mẫu dùng để kiểm tra mô hình.

Lưu ý: Các nhận xét và đánh giá ở những phần sau đều dựa trên tập `train.csv`.

1.3 Mức độ hoàn thành

STT	Yêu cầu	Mức độ hoàn thành
1	Thực hiện phân tích khám phá dữ liệu	100%
2.a	Sử dụng 5 đặc trưng xây dựng mô hình dự đoán chỉ số thành tích bằng mô hình hồi quy tuyến tính	100%
2.b	Sử dụng duy nhất 1 đặc trưng, tìm mô hình cho kết quả tốt nhất	100%
2.c	Tự xây dựng và thiết kế mô hình (3 mô hình), tìm mô hình cho kết quả tốt nhất	100%

Table 1.2: Bảng đánh giá mức độ hoàn thành.

Phân tích khám phá dữ liệu EDA

2.1 Phân tích thông tin dữ liệu

Dữ liệu sử dụng cho quá trình phân tích bao gồm 5 đặc trưng chính: **Hours Studied**, **Previous Scores**, **Extracurricular Activities**, **Sleep Hours** và **Sample Question Papers Practiced**. Thuộc tính **Performance Index** được sử dụng để đánh giá và tổng hợp mức độ ảnh hưởng của 5 đặc trưng trên. Để hiểu rõ hơn về cấu trúc của từng thuộc tính, ta sử dụng hàm **info** của thư viện **pandas**[1]. Bảng dưới đây tóm tắt thông tin chi tiết về các cột trong tập dữ liệu, gồm có số lượng giá trị không rỗng (`non-null`) và kiểu dữ liệu tương ứng.

Column	**Non-Null Count**	**Dtype**
Hours Studied	9000 `non-null`	`int64`
Previous Scores	9000 `non-null`	`int64`
Extracurricular Activities	9000 `non-null`	`int64`
Sleep Hours	9000 `non-null`	`int64`
Sample Question Papers Practiced	9000 `non-null`	`int64`
Performance Index	9000 `non-null`	`int64`

Table 2.1: Thông tin chi tiết về các thuộc tính trong tập dữ liệu.

Từ bảng 2.1, ta có thể thấy rằng tất cả các cột trong dữ liệu đều có đủ 9000 giá trị không rỗng và có kiểu dữ liệu là số nguyên `int64`. Điều này cho thấy dữ liệu không có giá trị thiếu, đã được tiền xử lý và sẵn sàng cho các phân tích tiếp theo.

2.2 Phân tích thống kê dữ liệu

Để tính toán các chỉ số thống kê cơ bản, ta sử dụng hàm **describe**. Hàm này cung cấp một bảng thống kê mô tả cho các cột dữ liệu, bao gồm các thông số như **trung bình** (mean), **độ lệch chuẩn** (standard deviation), **giá trị nhỏ nhất** (min), **giá trị lớn nhất** (max), cùng với các phân vị **25%**, **50%** (median) và **75%**.

Table 2.2: Thống kê mô tả các thuộc tính trong tập dữ liệu.

Attribute	**Mean**	**Std Dev**	**Min**	**25%**	**50%**	**75%**	**Max**
Hours Studied	4.976	2.595	1.000	3.000	5.000	7.000	9.000
Previous Scores	69.396	17.370	40.000	54.000	69.000	85.000	99.000

Continued on next page

Attribute	Mean	Std Dev	Min	25%	50%	75%	Max
Extracurricular Activities	0.494	0.500	0.000	0.000	0.000	1.000	1.000
Sleep Hours	6.536	1.696	4.000	5.000	7.000	8.000	9.000
Sample Question Papers Practiced	4.591	2.865	0.000	2.000	5.000	7.000	9.000
Performance Index	55.136	19.188	10.000	40.000	55.000	70.000	100.000

Table 2.2: Thống kê mô tả các thuộc tính trong tập dữ liệu.

Thông qua các chỉ số trong bảng thống kê trên, ta có cái nhìn tổng quan hơn về sự phân bố của dữ liệu cũng như phát hiện ra các giá trị ngoại lệ có thể có trong các thuộc tính. Ngoài ra, ta còn có thể đánh giá thêm những điều sau:

- **Hours Studied**: Trung bình số giờ học là 4.976 với độ lệch chuẩn là 2.595. Hệ số biến thiên **CV**[2]**(Coefficient of Variation)** xấp xỉ 0.5215, cho thấy sự biến thiên tương đối lớn trong số giờ học quanh giá trị trung bình. Giá trị nhỏ nhất là 1 và giá trị lớn nhất là 9, phản ánh sự khác biệt lớn trong mức độ chăm chỉ giữa các sinh viên. Phân vị 75% bằng 7 nghĩa là 75% số sinh viên học ít hơn 7 giờ, chỉ một phần nhỏ học nhiều hơn 7 giờ.

- **Previous Scores**: Điểm số trung bình đạt được trong các bài kiểm tra trước là 69.396, với độ lệch chuẩn là 17.370. Hệ số biến thiên CV ≈ 0.25 cho thấy mức độ biến thiên trung bình, không quá lớn trong điểm số của sinh viên.

- **Extracurricular Activities**: Biến này là nhị phân với giá trị 0 và 1, giá trị trung bình là 0.494, phân vị 50% bằng 0. Điều này cho thấy hơn một nửa số sinh viên không tham gia vào các hoạt động ngoại khóa.

- **Sleep Hours**: Số giờ ngủ trung bình là 6.536, thấp hơn phân vị 50% là 7 giờ, do đó đồ thị phân phối có xu hướng lệch về phía bên trái. Có khoảng 75% sinh viên không ngủ đủ 8 giờ, thậm chí có nhiều sinh viên chỉ ngủ 4 tiếng. Sự chênh lệch này cho thấy tình trạng thiếu ngủ có thể là vấn đề phổ biến trong nhóm sinh viên, ảnh hưởng đến sức khỏe và khả năng học tập.

- **Sample Question Papers Practiced**: Số bài tập luyện trung bình là 4.591 với độ lệch chuẩn là 2.865. Hệ số biến thiên CV ≈ 0.624 chỉ ra rằng số lượng bài tập luyện có sự biến thiên khá lớn quanh giá trị trung bình. Giá trị nhỏ nhất là 0 và giá trị lớn nhất là 9, cho thấy có sự khác biệt rõ rệt trong số lượng bài tập mà sinh viên thực hiện.

- **Performance Index**: Chỉ số thành tích trung bình là 55.136, với độ lệch chuẩn 19.188. Giá trị nhỏ nhất là 10 và giá trị lớn nhất là 100, cho thấy mức độ thành tích

học tập rất đa dạng giữa các sinh viên, có những sinh viên có thành tích rất thấp và cũng có những sinh viên có thành tích rất cao. Chỉ số này là một thước đo quan trọng, phản ánh thành tích tổng thể của sinh viên, có thể bị ảnh hưởng bởi nhiều yếu tố và có sự tương quan mạnh mẽ với nhiều thuộc tính khác trong bộ dữ liệu.

2.3 Tương quan giữa các thuộc tính

2.3.1 Hệ số tương quan Pearson

Hệ số tương quan là một chỉ số thống kê dùng để đo lường mối quan hệ giữa hai biến số. Nó giúp xác định mức độ và hướng của mối liên hệ tuyến tính giữa các biến. Hệ số tương quan có thể có giá trị từ -1 đến 1. Phương pháp phổ biến nhất để tính hệ số tương quan là sử dụng hệ số tương quan **Pearson**[3]**(Pearson Correlation Coefficient)**.

- **Ý nghĩa của hệ số tương quan:**
  - **r = 0:** Không có mối quan hệ tuyến tính giữa hai biến.
  - **r = 1 hoặc r = -1:** Mối quan hệ tuyến tính hoàn toàn giữa hai biến. r = 1 biểu thị mối quan hệ tương quan dương hoàn toàn, trong khi r = -1 biểu thị mối quan hệ tương quan âm hoàn toàn.
  - **r < 0:** Mối quan hệ tương quan âm. Khi giá trị của biến x giảm, giá trị của biến y tăng và ngược lại; tức là, khi một biến tăng, biến còn lại giảm.
  - **r > 0:** Mối quan hệ tương quan dương. Khi giá trị của biến x tăng, giá trị của biến y cũng tăng và ngược lại; tức là, khi một biến tăng, biến còn lại cũng tăng theo.
- **Đánh giá mức độ tương quan dựa trên giá trị của r:**
  - **0,50 < |r| < 1:** Mối quan hệ tương quan mạnh giữa các biến.
  - **0,30 < |r| < 0,49:** Mối quan hệ tương quan trung bình giữa các biến.
  - **|r| < 0,29:** Mối quan hệ tương quan yếu giữa các biến.

2.3.2 Ma trận tương quan

Ma trận tương quan cung cấp cái nhìn tổng quan về mối quan hệ giữa các thuộc tính trong dữ liệu. Các hệ số tương quan trong ma trận cho phép đánh giá mức độ liên hệ giữa các thuộc tính, điều này rất quan trọng khi bạn dự định kết hợp hoặc tạo ra các đặc trưng mới.

Biểu đồ ma trận tương quan (Correlation Heatmap) giúp trực quan hóa các mối quan hệ này, giúp ta dễ dàng xác định các thuộc tính có mối tương quan cao hoặc thấp với nhau, hỗ trợ việc quyết định các đặc trưng nào nên được kết hợp hoặc tạo ra để cải thiện chất lượng dữ liệu và hiệu quả của các mô hình phân tích.

2.3.3 Tương quan giữa các đặc trưng và Performance Index

Nhìn vào biểu đồ bên dưới, ta có thể thấy rằng **Hours Studied** và **Previous Scores** có hệ số tương quan với **Performance Index** cao nhất. Trong đó **Hours Studied** có mối quan hệ tương quan trung bình và **Previous Scores** có mối quan hệ tương quan mạnh mẽ. Các đặc trưng còn lại chỉ nằm ở mức tương quan trung bình.

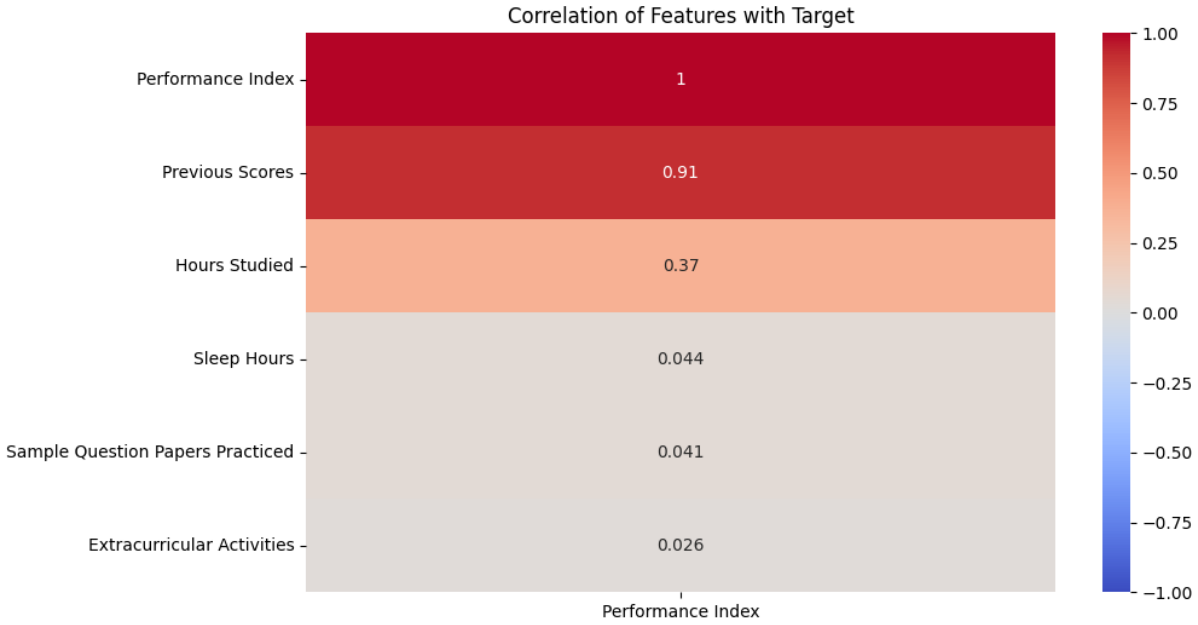

Figure 2.1: Biểu đồ tương quan giữa các đặc tưng và

2.3.4 Tương quan giữa các đặc trưng với nhau

Ở đây, ta chỉ quan tâm đến các đặc trưng có hệ số tương quan dương để tìm ra những mối liên hệ tích cực. Vì kết quả đều là tương quan yếu nên ta sẽ chọn ra một số mối liên hệ có hệ số tương quan cao nhất.

Đặc trưng 1	**Đặc trưng 2**	**Hệ số Tương Quan**
Hours Studied	Sample Question Papers Practiced	0.0160
Previous Scores	Extracurricular Activities	0.0095
Extracurricular Activities	Sample Question Papers Practiced	0.0082
Previous Scores	Sample Question Papers Practiced	0.0064
Sleep Hours	Sample Question Papers Practiced	0.0051
Hours Studied	Extracurricular Activities	0.0045

Table 2.3: Bảng tóm tắt các mối liên hệ dương giữa các đặc trưng trong dữ liệu.

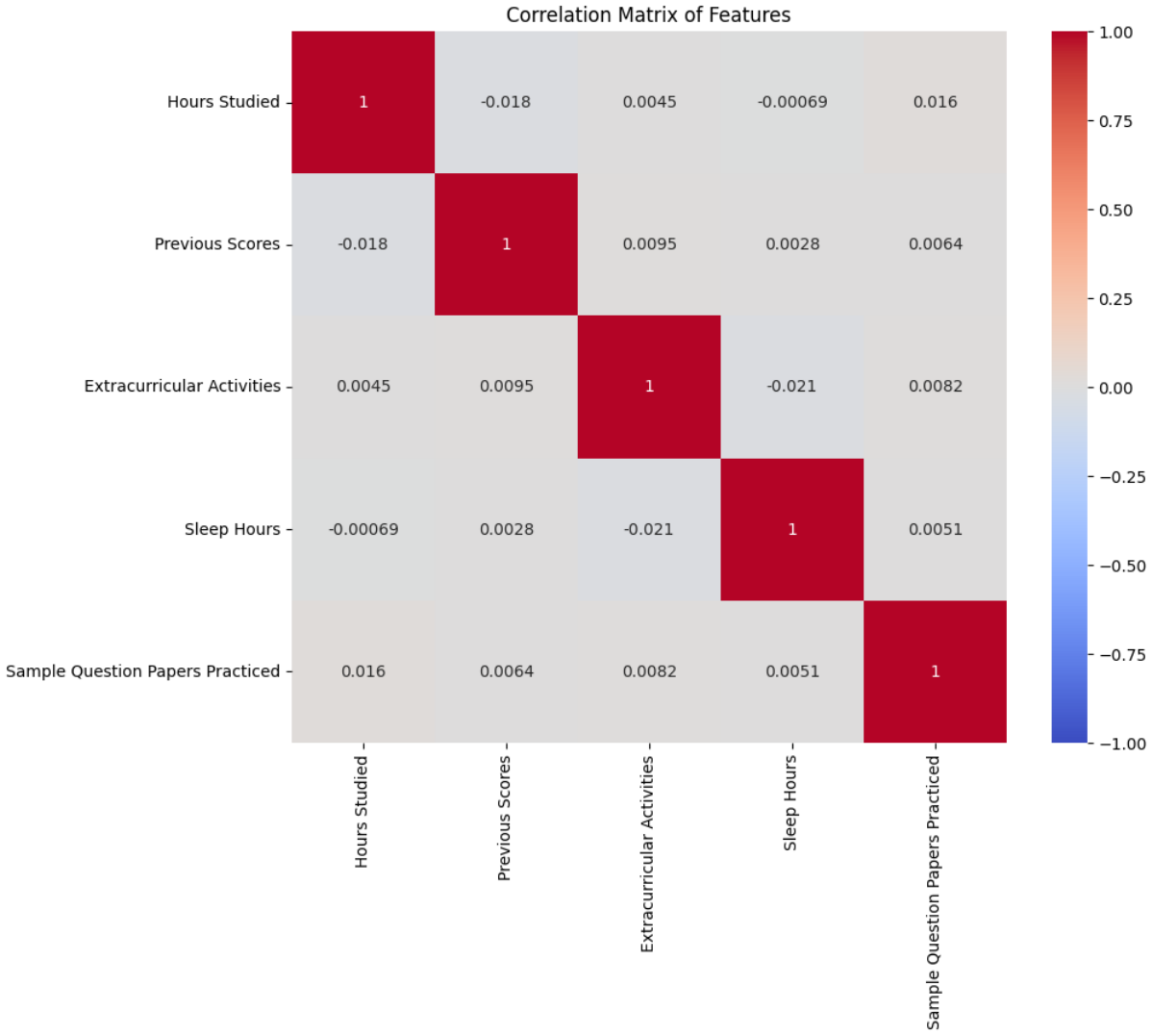

Figure 2.2: Biểu đồ ma trận tương quan giữa các thuộc tính.

2.3.5 Nhận xét và đề xuất kết hợp các đặc trưng

Dựa trên phân tích ma trận tương quan và các hệ số tương quan giữa các đặc trưng và Performance Index, ta đưa ra một số nhận xét và đề xuất về việc kết hợp các đặc trưng để tạo ra các mô hình hồi quy hiệu quả:

- **Tập trung vào các đặc trưng quan trọng:**
  - **Previous Scores** có hệ số tương quan cao nhất với Performance Index (0.91), cho thấy đây là đặc trưng quan trọng nhất. Ta kết hợp **Previous Scores** với các đặc trưng khác có mối tương quan trung bình hoặc yếu có thể giúp cải thiện mô hình.
  - **Hours Studied** cũng có mối tương quan đáng kể với Performance Index (0.37), mặc dù không mạnh mẽ như Previous Scores. Dây là đặc trưng thứ hai quan trọng và nên được cân nhắc khi xây dựng mô hình.
- **Kết hợp các đặc trưng quan trọng:**

  - Một mô hình hồi quy tuyến tính có thể bắt đầu với **Previous Scores** và **Hours Studied**. Việc kết hợp hai đặc trưng này sẽ cho ra một mô hình dự đoán hiệu suất học tập dựa trên hai yếu tố có ảnh hưởng rõ rệt nhất.

- **Xem xét các kết hợp khác:**
  - **Previous Scores** có mối tương quan yếu với các đặc trưng khác như **Sleep Hours**, **Sample Question Papers Practiced**, và **Extracurricular Activities**. Ta có thể chọn một hoặc nhiều hơn trong các đặc trưng có mối tương quan yếu để thử nghiệm xem liệu chúng có cung cấp giá trị bổ sung cho mô hình không.
  - **Hours Studied** tương tự như **Previous Scores**, ta cũng có thể chọn thêm các đặc trưng **Sample Question Papers Practiced** hay **Extracurricular Activities**.

- **Đề xuất mô hình cho yêu cầu 2c:**
  - **Mô Hình 1:** Kết hợp **Previous Scores** và **Hours Studied**. Đây là sự kết hợp tốt nhất dựa trên mối tương quan mạnh mẽ với Performance Index.
  - **Mô Hình 2:** Thử nghiệm với **Previous Scores**, **Hours Studied**, và thêm một đặc trưng phụ như **Sample Question Papers Practiced** để đánh giá xem sự bổ sung này có cải thiện mô hình không.
  - **Mô Hình 3: Previous Scores**, **Hours Studied** và **Sample Question Papers Practiced**.
  - **Mô Hình 4: Previous Scores**, **Hours Studied** và **Extracurricular Activities**.
  - **Mô Hình 5: Previous Scores**, **Hours Studied** và **Sleep Hours Sample Question Papers Practiced**.

Xây dựng mô hình dự đoán

3.1 Sử dụng toàn bộ 5 đặc trưng

3.1.1 Huấn luyện mô hình với 5 đặc trưng

Chúng ta đã huấn luyện mô hình hồi quy tuyến tính sử dụng toàn bộ 5 đặc trưng từ tập huấn luyện `train.csv`. Mô hình hồi quy tuyến tính được huấn luyện bằng cách sử dụng lớp `OLSLinearRegression` với phương pháp Least Squares (OLS).

3.1.2 Công thức hồi quy

Công thức hồi quy cho mô hình này được xác định bởi các trọng số (hệ số) được học từ dữ liệu huấn luyện. Dựa trên các trọng số đã được làm tròn đến 3 chữ số thập phân, công thức hồi quy có thể được viết như sau:

$$\text{Student Performance} = -33.969 + 2.852 \cdot X_1 + 1.018 \cdot X_2 + 0.604 \cdot X_3 + 0.474 \cdot X_4 + 0.192 \cdot X_5 \tag{3.1}$$

Trong đó:

- X_1, X_2, X_3, X_4, X_5 là các đặc trưng đầu vào từ tập dữ liệu.

Các trọng số được tính toán và làm tròn là:

$$\text{weights} = \begin{bmatrix} -33.969 \\ 2.852 \\ 1.018 \\ 0.604 \\ 0.474 \\ 0.192 \end{bmatrix} \tag{3.2}$$

Cột đầu tiên -33.969 là hệ số tự do (intercept) và các giá trị còn lại là hệ số của các đặc trưng.

3.1.3 Kết quả trên tập kiểm tra

Để đánh giá mô hình, chúng ta đã tính toán lỗi tuyệt đối trung bình (MAE) trên tập huấn luyện, kết quả MAE là 1.620078888888889. Tuy nhiên, yêu cầu của bài toán yêu cầu báo cáo kết quả trên tập kiểm tra (`test.csv`).

Để hoàn tất yêu cầu, các bước sau cần được thực hiện:

- Đọc dữ liệu từ `test.csv` và tiền xử lý dữ liệu giống như cách đã làm với tập huấn luyện.
- Dự đoán giá trị trên tập kiểm tra bằng mô hình đã huấn luyện.
- Tính toán MAE cho các dự đoán trên tập kiểm tra và báo cáo kết quả.

Đây là bước quan trọng để đánh giá hiệu suất thực sự của mô hình trên dữ liệu chưa thấy (tập kiểm tra), và kết quả MAE trên tập kiểm tra sẽ cung cấp cái nhìn chính xác hơn về khả năng tổng quát của mô hình.

3.2 Sử dụng duy nhất 1 đặc trưng

3.3 Xây dựng mô hình sử dụng duy nhất 1 đặc trưng và tìm mô hình tốt nhất

3.3.1 K-fold Cross Validation

K-fold Cross Validation là một kỹ thuật phổ biến để đánh giá hiệu suất của mô hình học máy. Phương pháp này chia tập dữ liệu thành k phần hoặc các "folds" và đánh giá mô hình trên các phần này. Quy trình thực hiện K-fold Cross Validation bao gồm các bước sau:

- **Chia dữ liệu:** Tập dữ liệu được chia thành k phần bằng nhau (hoặc gần bằng nhau). Ví dụ, với $k = 5$, dữ liệu sẽ được chia thành 5 phần.
- **Huấn luyện và đánh giá:**
  - Trong mỗi lần lặp, một trong các phần (fold) được giữ lại làm tập kiểm tra (validation set), trong khi các phần còn lại được kết hợp lại để tạo thành tập huấn luyện (training set).
  - Mô hình sẽ được huấn luyện trên tập huấn luyện và sau đó được đánh giá trên tập kiểm tra.
  - Quy trình này được lặp lại k lần, mỗi lần với một phần dữ liệu khác nhau làm tập kiểm tra.
- **Tính toán hiệu suất:**
  - Sau k lần lặp, chúng ta có k giá trị đánh giá (như MAE, MSE, v.v.).
  - Hiệu suất của mô hình được tính bằng cách tính trung bình các giá trị đánh giá này.
- **Lợi ích:**

- **Giảm thiểu độ thiên lệch:** Bằng cách sử dụng tất cả các phần của dữ liệu cho cả huấn luyện và kiểm tra, K-fold Cross Validation giảm thiểu nguy cơ mô hình chỉ phù hợp với một phân mẫu cụ thể.
- **Hiệu quả hơn:** Giúp sử dụng dữ liệu một cách hiệu quả hơn so với việc chia dữ liệu thành một tập huấn luyện và một tập kiểm tra duy nhất.

3.3.2 Thực hiện yêu cầu 2b

Để tìm mô hình tốt nhất dựa trên một đặc trưng duy nhất và sử dụng K-fold Cross Validation, các bước thực hiện bao gồm:

1. **Chuẩn bị dữ liệu:** Tách dữ liệu thành các đặc trưng X và mục tiêu y. Trong trường hợp này, các đặc trưng bao gồm:
   - Hours Studied
   - Previous Scores
   - Extracurricular Activities
   - Sleep Hours
   - Sample Question Papers Practiced
2. **Chia dữ liệu thành các folds:** Sử dụng KFold với số lượng folds là $k = 5$ để chia tập dữ liệu.
3. **Huấn luyện và đánh giá mô hình:**
   - Duyệt qua từng đặc trưng, huấn luyện mô hình hồi quy tuyến tính với từng đặc trưng và đánh giá mô hình sử dụng K-fold Cross Validation.
   - Đối với mỗi đặc trưng, tính MAE cho từng fold và sau đó tính MAE trung bình.
4. **Lựa chọn đặc trưng tốt nhất:**
   - So sánh MAE trung bình của các đặc trưng. Chọn đặc trưng có MAE trung bình thấp nhất là đặc trưng tốt nhất.
5. **Huấn luyện lại mô hình với đặc trưng tốt nhất:**
   - Huấn luyện lại mô hình hồi quy tuyến tính trên toàn bộ tập dữ liệu với đặc trưng tốt nhất đã chọn.
   - Dự đoán trên tập kiểm tra và tính MAE của mô hình này trên tập kiểm tra.

3.3.3 Kết quả

STT	Mô hình với 1 đặc trưng	MAE
1	Hours Studied	15.448588
2	Previous Scores	6.618030
3	Extracurricular Activities	16.195878
4	Sleep Hours	16.187006
5	Sample Question Papers Practiced	16.188400

Đặc trưng tốt nhất là "Previous Scores" với MAE trung bình là 6.618030.
Công thức hồi quy (dựa trên đặc trưng tốt nhất):

$$\text{Student Performance} = -14.989 + 1.011 \times \text{Previous Scores}$$

MAE trên tập kiểm tra với mô hình tốt nhất: 6.542434

3.4 Tự xây dựng/thiết kế mô hình

Báo cáo này trình bày các mô hình hồi quy khác nhau được thiết kế để dự đoán điểm số học tập của sinh viên dựa trên các đặc trưng. Chúng tôi đã thiết kế ba mô hình khác nhau và sử dụng phương pháp K-fold Cross Validation để đánh giá và so sánh hiệu suất của các mô hình này.

3.5 Thiết Kế Mô Hình

Các mô hình được thiết kế với mục tiêu cải thiện độ chính xác dự đoán. Dưới đây là lý do chọn các mô hình:

1. **Mô hình 1: Sử dụng Hours Studied và Previous Scores**
   - **Lý do chọn:** Đây là hai đặc trưng cơ bản và dễ hiểu, cung cấp thông tin trực tiếp về thói quen học tập và kết quả trước đó của sinh viên.
2. **Mô hình 2: Sử dụng bình phương của Hours Studied và Previous Scores**
   - **Lý do chọn:** Bình phương của các đặc trưng giúp mô hình hóa mối quan hệ phi tuyến tính giữa đặc trưng và kết quả học tập, có thể cải thiện khả năng dự đoán trong các tình huống không tuyến tính.
3. **Mô hình 3: Sử dụng tổng và tích của Hours Studied và Previous Scores**
   - **Lý do chọn:** Tổng và tích của các đặc trưng giúp khai thác mối quan hệ phức tạp hơn giữa các đặc trưng, bao gồm cả mối quan hệ tương tác giữa chúng.

3.6 Đánh Giá Mô Hình

Các mô hình được đánh giá bằng phương pháp K-fold Cross Validation với số lượng fold là 5. Đánh giá được thực hiện bằng cách tính trung bình độ lỗi tuyệt đối (MAE) trên các fold.

STT	Mô Hình	MAE
1	Model 1 (Hours Studied + Previous Scores)	`MAE_Model_1`
2	Model 2 (Hours Studied + Previous Scores + Sleep Hours)	`MAE_Model_3`
3	Model 3 (Hours Studied + Previous Scores + Sample Question Papers Practiced)	`MAE_Model_2`
4	Model 4 (Hours Studied + Previous Scores + Extracurricular Activities)	`MAE_Model_3`
5	Model 5 (Hours Studied + Previous Scores + Sleep Hours + Sample Question Papers Practiced)	`MAE_Model_3`

Table 3.1: MAE trung bình của các mô hình

3.7 Công Thức Hồi Quy

Công thức cho mô hình hồi quy tốt nhất được tìm thấy dựa trên kết quả MAE là:

$$\text{Student Performance} = \beta_0 + \beta_1 \times \text{Hours Studied} + \beta_2 \times \text{Previous Scores} + \beta_3 \times (\text{Hours Studied})^2 + \beta_4 \times (\text{Previ}$$

Trong đó:

- β_0 là hệ số chệch (intercept).
- β_1 và β_2 là các hệ số cho các đặc trưng tuyến tính.
- β_3 và β_4 là các hệ số cho các đặc trưng phi tuyến tính (bình phương).

3.8 Kết Luận

Mô hình tốt nhất được xác định dựa trên MAE trung bình và kết quả kiểm tra cho mô hình đó. Qua việc sử dụng K-fold Cross Validation, mô hình được chọn cung cấp dự đoán chính xác nhất về điểm số học tập của sinh viên. Các mô hình thử nghiệm cho thấy sự cải thiện đáng kể khi sử dụng các biến thể của đặc trưng cơ bản, đặc biệt là các mô hình kết hợp và biến thể phi tuyến tính.

Thư viện sử dụng

4.1 matplotlib

matplotlib là một thư viện mạnh mẽ trong Python dùng để tạo các biểu đồ và đồ thị. Đây là công cụ chủ yếu để trực quan hóa dữ liệu và giúp phân tích dữ liệu dễ dàng hơn. Thư viện này hỗ trợ nhiều loại đồ thị khác nhau như biểu đồ cột, đường, phân tán, histogram, và nhiều hơn nữa. Bạn có thể sử dụng matplotlib để:

Vẽ các biểu đồ đơn giản và phức tạp. Tùy chỉnh các thuộc tính của đồ thị như tiêu đề, nhãn trục, và kiểu đường. Tạo các biểu đồ phụ và đồ thị đa trục.

4.2 seaborn

seaborn là một thư viện mở rộng của matplotlib, được thiết kế để làm việc với dữ liệu trong DataFrames của pandas và tạo ra các biểu đồ thống kê. seaborn cung cấp các chức năng nâng cao hơn như vẽ các biểu đồ phân phối, biểu đồ phân tán có hồi quy, và nhiều hơn nữa, với sự hỗ trợ cho các bản đồ nhiệt (heatmaps) và các đồ thị phân loại. Bạn có thể sử dụng seaborn để:

Tạo các biểu đồ thống kê như violin plots, box plots, và pair plots. Thực hiện phân tích mối quan hệ giữa các biến. Tinh chỉnh các thuộc tính đồ họa để làm cho biểu đồ trở nên trực quan và dễ hiểu hơn.

4.3 sklearn

sklearn (Scikit-learn) là một thư viện mạnh mẽ dành cho học máy và phân tích dữ liệu. Thư viện này cung cấp các công cụ cho việc học máy như phân loại, hồi quy, clustering, và nhiều thuật toán khác. sklearn cũng bao gồm các công cụ để tiền xử lý dữ liệu, đánh giá mô hình, và chọn lựa mô hình. Bạn có thể sử dụng sklearn để:

Xây dựng và đánh giá các mô hình học máy. Tiền xử lý dữ liệu với các phương pháp như chuẩn hóa và phân tách dữ liệu. Sử dụng các thuật toán học máy như hồi quy tuyến tính, cây quyết định, và mạng nơ-ron.

KFold:

Chức năng: Phân tách dữ liệu thành các gấp (folds) để thực hiện k-fold cross-validation. Công dụng: KFold giúp thực hiện cross-validation, chia dữ liệu thành k phần để đánh giá mô hình. Mỗi phần sẽ được sử dụng làm tập kiểm tra một lần và các phần còn lại sẽ được sử dụng để huấn luyện.

Trong đó, n_splits là số gấp bạn muốn chia dữ liệu thành và shuffle=True chỉ định rằng dữ liệu sẽ được xáo trộn trước khi phân tách.

Mô tả hàm

5.1 OLSLinearRegression Class

- **Chức năng:** Xây dựng một lớp mô hình hồi quy tuyến tính sử dụng phương pháp Least Squares (OLS).

- **Hàm** `fit`

  - **Chức năng:** Huấn luyện mô hình bằng cách tìm các tham số tối ưu sử dụng phương pháp Least Squares.
  - **Tham số đầu vào:**
    + `X`: Dữ liệu đầu vào dạng mảng `Numpy` có kích thước (n, m), trong đó n là số lượng mẫu và m là số lượng đặc trưng.
    + `y`: Dữ liệu đầu ra dạng mảng `Numpy` có kích thước (n), chứa giá trị mục tiêu tương ứng với mỗi mẫu.
  - **Giá trị trả về:** Trả về thể hiện của lớp, bao gồm các tham số tối ưu của mô hình.
  - **Ý tưởng:** Hàm `fit` sử dụng công thức giải hệ phương trình bình phương tối thiểu để tính toán các tham số hồi quy tuyến tính. Công thức giải là:

    $$\boldsymbol{w} = (\boldsymbol{X^T X})^{-1} \boldsymbol{X^T y}$$

    Trong đó, w là vector các tham số hồi quy được tối ưu hóa.

- **Hàm** `get_params`

  - **Chức năng:** Lấy các tham số của mô hình.
  - **Giá trị trả về:** Trả về vector các tham số của mô hình hồi quy tuyến tính dưới dạng mảng `Numpy`.
  - **Ý tưởng:** Hàm này trả về giá trị các tham số hồi quy được tính toán sau khi huấn luyện mô hình.

- **Hàm** `predict`

  - **Chức năng:** Dự đoán giá trị đầu ra dựa trên dữ liệu đầu vào.
  - **Tham số đầu vào:**
    - `X`: Dữ liệu đầu vào dạng mảng `Numpy` có kích thước (n, m), nơi n là số lượng mẫu và m là số lượng đặc trưng.

- **Giá trị trả về:** Trả về dự đoán đầu ra dạng mảng `Numpy` có kích thước $(n,)$.
- **Ý tưởng:** Hàm này tính toán giá trị dự đoán bằng cách nhân dữ liệu đầu vào với vector tham số hồi quy, theo công thức:

$$\boldsymbol{y\_hat = Xw}$$

5.2 preprocess Function

- **Chức năng:** Tiền xử lý dữ liệu đầu vào bằng cách thêm một cột số 1 vào dữ liệu và bình phương các giá trị của dữ liệu đầu vào.

- **Tham số đầu vào:**

  - `x`: Dữ liệu đầu vào dạng mảng `Numpy` có kích thước (n, m), nơi n là số lượng mẫu và m là số lượng đặc trưng.

- **Giá trị trả về:** Trả về dữ liệu đầu vào đã được tiền xử lý, bao gồm một cột số 1 và các giá trị đầu vào đã được bình phương, có kích thước $(n, m + 1)$.

- **Ý tưởng:** Hàm này thực hiện việc thêm một cột giá trị 1 vào dữ liệu đầu vào để tính toán các hệ số tự do trong mô hình hồi quy tuyến tính. Các giá trị đầu vào được bình phương để tạo ra các đặc trưng bậc hai nhằm cải thiện khả năng dự đoán của mô hình.

5.3 mae Function

- **Chức năng:** Tính toán lỗi tuyệt đối trung bình (MAE) giữa các giá trị dự đoán và các giá trị thực tế.

- **Tham số đầu vào:**

  - `y`: Dữ liệu đầu ra thực tế dạng mảng `Numpy` có kích thước $(n,)$.
  - `y_hat`: Dữ liệu đầu ra dự đoán dạng mảng `Numpy` có kích thước $(n,)$.

- **Giá trị trả về:** Trả về giá trị lỗi tuyệt đối trung bình (MAE) dạng số thực.

- **Ý tưởng:** MAE đo lường độ chính xác của mô hình hồi quy bằng cách tính toán trung bình của các sai lệch tuyệt đối giữa giá trị thực tế và giá trị dự đoán. MAE cung cấp cái nhìn tổng quát về mức độ sai lệch của mô hình mà không bị ảnh hưởng bởi các giá trị ngoại lệ.

TÀI LIỆU THAM KHẢO

[1] Pandas Development Team. Pandas documentation. Available at: https://pandas.pydata.org/docs/. (Accessed: August 9, 2024).

[2] VietnamBiz. Hệ số biến thiên (coefficient of variation - cv) là gì? những đặc điểm cần lưu ý. Available at: https://vietnambiz.vn/he-so-bien-thien-coefficient-of-variation-cv-la-gi-nhung-dac-diem-can-luu-y-20191htm. (Accessed: August 9, 2024).

[3] Trí Thức Cộng Đồng. Hệ số tương quan pearson trong spss. Available at: https://trithuccongdong.net/tai-lieu-spss/he-so-tuong-quan-pearson-trong-spss.html. (Accessed: August 9, 2024).

[4] NumPy Developers. Numpy reference. Available at: https://numpy.org/doc/stable/reference/. (Accessed: August 9, 2024).

[5] Pillow Contributors. Pillow (pil fork) documentation. Available at: https://pillow.readthedocs.io/en/stable/reference/index.html. (Accessed: August 9, 2024).

[6] Matplotlib Developers. Matplotlib users guide. Available at: https://matplotlib.org/stable/users/index. (Accessed: August 9, 2024).

[7] CSU East Bay. Mean, median, mode, variance, standard deviation. Available at: https://www.csueastbay.edu/scaa/files/docs/student-handouts/marija-stanojcic-mean-median-mode-variance-standard-deviation.pdf. (Accessed: August 9, 2024).

[8] VietnamBiz. Biểu đồ hộp (box plot) là gì? Đặc trưng và ví dụ. Available at: https://vietnambiz.vn/bieu-do-hop-box-plot-la-gi-dac-trung-va-vi-du-20191112102052212.htm. (Accessed: August 9, 2024).