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 03: LINEAR REGRESSION

Môn học: Toán ứng dụng và thống kê cho Công nghệ thông tin - 22CLC10

Giáo viên hướng dẫn:

Thầy Vũ Quốc Hoàng

Thầy Nguyễn Văn Quang Huy

Thầy Nguyễn Ngọc Toàn

Cô Phan Thị Phương Uyên

Sinh viên thực hiện:

22127440 Phan Võ Minh Tuệ

Thành phố Hồ Chí Minh - 2024

MỤC LỤC

MỤC LỤC **2**
LỜI CẢM ƠN **3**
TỔNG QUAN **4**
1. THƯ VIỆN VÀ HÀM SỬ DỤNG **5**
1.1. Thư viện được sử dụng **5**
*1.1.1. Thư viện **Pandas*** *5*
*1.1.2. Thư viện **NumPy*** *5*
*1.1.3. Thư viện **Matplotlib*** *6*
*1.1.4. Thư viện **Seaborn*** *6*
1.2. Hàm được sử dụng **7**
*1.2.1. Hàm từ thư viện **Pandas*** *7*
*1.2.2. Hàm từ thư viện **NumPy*** *9*
*1.2.3. Hàm từ thư viện **Matplotlib*** *11*
*1.2.4. Hàm từ thư viện **Seaborn*** *12*
1.2.5. Hàm tự cài đặt *14*
2. BÁO CÁO PHÂN TÍCH KHAI PHÁ DỮ LIỆU **19**
2.1. Tổng quan về Bộ dữ liệu **19**
2.2. Phân tích Hệ số tương quan Spearman **21**
2.3. Phân tích biểu đồ **23**
3. BÁO CÁO VÀ NHẬN XÉT KẾT QUẢ CỦA CÁC MÔ HÌNH **27**
3.1. Tiền xử lý dữ liệu **27**
3.2. Mô hình sử dụng toàn bộ 5 đặc trưng **27**
3.3. Mô hình sử dụng duy nhất 1 đặc trưng **28**
3.4. Mô hình tự thiết kế **31**
3.4.1. Mô hình tự thiết kế 1 ..... *31*
3.4.2. Mô hình tự thiết kế 2 ..... *32*
3.4.3. Mô hình tự thiết kế 3 ..... *33*
3.4.4. Mô hình tự thiết kế tốt nhất *34*
TỔNG KẾT **37**
TÀI LIỆU THAM KHẢO **38**

LỜI CẢM ƠN

Trước hết, tôi xin bày tỏ lòng biết ơn sâu sắc đến quý thầy cô giáo hướng dẫn, đặc biệt là cô Phan Thị Phương Uyên, người đã trao cho tôi cơ hội thực hiện đồ án này. Đây quả thực là một chủ đề nghiên cứu hết sức thú vị nhưng cũng mang nhiều ý nghĩa quan trọng đối với tôi trên con đường chinh phục bộ môn Khoa học máy tính sau này.

Tôi cũng muốn bày tỏ lòng biết ơn đối với Trường Đại học Khoa học Tự nhiên, Đại học quốc gia Thành phố Hồ Chí Minh đã hỗ trợ, cung cấp không gian nghiên cứu, học tập cho các sinh viên như chúng tôi. Ngoài ra, tôi cũng muốn gửi lời cảm ơn chân thành đến các bạn sinh viên lớp Toán ứng dụng và thống kê cho Công nghệ thông tin_22CLC10 đã cung cấp thông tin, chia sẻ kinh nghiệm và tiếp thêm động lực cho tôi trong suốt thời gian thực hiện đồ án lần này.

Cuối cùng, tôi không thể không nhắc đến gia đình yêu quý của mình, những người luôn ở bên cạnh, hỗ trợ và tạo điều kiện tốt nhất để tôi có thể tập trung vào việc học tập và nghiên cứu.

Lòng biết ơn của tôi cũng dành cho tất cả những ai đã giúp đỡ, trực tiếp hay gián tiếp, trong quá trình hoàn thành tiểu luận này.

Phan Võ Minh Tuệ

TỔNG QUAN

Đồ án 03 - Linear Regression tập trung vào việc phát triển các kỹ năng phân tích dữ liệu và xây dựng mô hình dự đoán sử dụng hồi quy tuyến tính. Mã nguồn được viết bằng ngôn ngữ lập trình Python kết hợp với các thư viện như *NumPy, Pandas, Matplotlib và Seaborn* để thực hiện các chức năng như đọc và xử lý dữ liệu, phân tích khám phá dữ liệu, xây dựng và đánh giá mô hình hồi quy tuyến tính.

Nội dung chi tiết

Báo cáo sẽ trình bày chi tiết về cách thức thực hiện từng phần của đồ án, bao gồm:

- Phân tích khám phá dữ liệu (EDA)
  - Sử dụng thống kê mô tả để phân tích các đặc trưng
  - Sử dụng các biểu đồ để trực quan hóa và phân tích dữ liệu
- Xây dựng mô hình dự đoán chỉ số thành tích
  - Mô hình sử dụng toàn bộ 5 đặc trưng
  - Mô hình sử dụng duy nhất 1 đặc trưng tốt nhất
  - Mô hình tự thiết kế với các biến đổi đặc trưng
- Đánh giá và so sánh hiệu suất của các mô hình
- Giải thích và đưa ra nhận xét về kết quả thu được

Kết quả

Kết quả phân tích và xây dựng mô hình sẽ được minh họa thông qua các bảng, đồ thị và sử dụng sai số tuyệt đối trung bình (MAE: Mean Absolute Error) để đánh giá. Báo cáo này cũng sẽ cung cấp đánh giá chi tiết về hiệu suất của các mô hình hồi quy tuyến tính được xây dựng, so sánh giữa các phương pháp khác nhau và giải thích lý do tại sao một số mô hình hoạt động tốt hơn các mô hình khác. Điều này sẽ cung cấp cái nhìn tổng quan và sâu sắc về khả năng dự đoán của các mô hình hồi quy tuyến tính đối với vấn đề dự đoán các chỉ số kết quả học tập của các sinh viên.

1. THƯ VIỆN VÀ HÀM SỬ DỤNG

1.1. Thư viện được sử dụng

*1.1.1. Thư viện **Pandas***

- **Mục đích:** Pandas là một thư viện mạnh mẽ và linh hoạt dành cho việc thao tác và phân tích dữ liệu. Thư viện này hỗ trợ việc đọc dữ liệu từ các tệp CSV, trích xuất các cột, và cung cấp các phương thức để thống kê, lọc, và thao tác với dữ liệu.
- **Ứng dụng cụ thể:**
  - **Đọc dữ liệu từ tệp CSV:** Pandas cung cấp hàm `pd.read_csv()` giúp đọc dữ liệu từ tệp CSV vào một DataFrame, một cấu trúc dữ liệu linh hoạt cho việc xử lý dữ liệu dạng bảng.
  - **Trích xuất các đặc trưng và giá trị mục tiêu:** Pandas hỗ trợ trích xuất các cột cụ thể từ `DataFrame`, giúp dễ dàng tách các đặc trưng đầu vào (features) và giá trị mục tiêu (target) cho việc huấn luyện mô hình.
  - **Hiển thị thông tin cơ bản về dữ liệu:** Sử dụng hàm `pd.DataFrame.info()` cũng như hàm `pd.DataFrame.describe()` để giúp lập trình viên nắm bắt được cấu trúc và thống kê mô tả cơ bản của dữ liệu, bao gồm số lượng phần tử, kiểu dữ liệu, giá trị trung bình, độ lệch chuẩn, và phân phối dữ liệu.

*1.1.2. Thư viện **NumPy***

- **Mục đích:** NumPy là một thư viện cơ bản cho tính toán khoa học trong Python. Nó cung cấp hỗ trợ cho các mảng và ma trận lớn cùng với các hàm toán học hiệu quả để thao tác và xử lý các dữ liệu số học.
- **Ứng dụng cụ thể:**
  - **Thực hiện các phép toán mảng:** NumPy giúp xử lý các mảng dữ liệu đa chiều một cách nhanh chóng và hiệu quả

hơn. Vì NumPy Array sử dụng ít bộ nhớ hơn và hoạt động nhanh hơn khi sử dụng Python List.

*1.1.3. Thư viện **Matplotlib***

- **Mục đích:** Matplotlib là một thư viện phổ biến cho việc trực quan hóa dữ liệu trong Python. Nó cho phép người dùng tạo các biểu đồ đơn giản đến phức tạp để phân tích và trình bày dữ liệu một cách trực quan.
- **Ứng dụng cụ thể:**
  - **Tạo lưới các trục:** Matplotlib cung cấp các hàm như `plt.subplots()` để tạo ra một lưới các trục (axes) trong cùng một biểu đồ, giúp hiển thị nhiều biểu đồ khác nhau trong cùng một hình ảnh (figure). Điều này rất hữu ích khi cần so sánh hoặc phân tích dữ liệu từ nhiều góc độ hoặc đặc trưng khác nhau.
  - **Hiển thị biểu đồ đã vẽ:**
    - `plt.show()`: Hiển thị biểu đồ đã được vẽ.
    - `plt.title()`: Thiết lập tiêu đề cho biểu đồ.
    - `plt.xticks()`: Thiết lập vị trí và nhãn cho trục x.

*1.1.4. Thư viện **Seaborn***

- **Mục đích:** Seaborn là một thư viện được xây dựng trên nền tảng Matplotlib, cung cấp các hàm cao cấp hơn để tạo các biểu đồ phân tích dữ liệu. Seaborn giúp tạo ra các biểu đồ có tính thẩm mỹ cao và dễ dàng hơn khi thực hiện phân tích khai phá dữ liệu (EDA: Exploratory Data Analysis).
- **Ứng dụng cụ thể:**
  - **Tạo các biểu đồ phân phối (histogram):** Seaborn cung cấp hàm `sns.histplot()` giúp hiển thị phân phối của dữ liệu một cách trực quan và dễ hiểu, đồng thời có thể kết hợp với đường cong mật độ (kde) để hiểu rõ hơn về phân phối xác suất.

- **Tạo biểu đồ cặp (pair plots):** Pair plot hiển thị các biểu đồ scatter plot giữa tất cả các cặp đặc trưng, giúp phân tích các mối quan hệ giữa các biến trong dữ liệu.
- **Tạo biểu đồ hộp (box plots):** Box plot giúp hiển thị phân phối dữ liệu thông qua các thông số như giá trị trung bình, khoảng tứ phân vị, và phát hiện các giá trị ngoại lai (outliers) trong dữ liệu.
- **Tạo biểu đồ heatmap:** Heatmap là một biểu đồ nhiệt giúp hiển thị ma trận tương quan giữa các đặc trưng, qua đó giúp phát hiện các mối quan hệ tuyến tính giữa các biến.

1.2. Hàm được sử dụng

*1.2.1. Hàm từ thư viện **Pandas***

- Hàm: `pd.read_csv()`
  - Đầu vào:
    - Đường dẫn đến tệp CSV (string).
    - Các tham số tùy chọn như `delimiter`, `header`, `encoding`...
  - Đầu ra: Một DataFrame chứa dữ liệu từ tệp CSV.
  - Mô tả chi tiết: Đọc file CSV và chuyển đổi nó thành DataFrame trong Pandas để dễ dàng xử lý trong huấn luyện mô hình.
- Hàm: `pd.DataFrame.iloc[]`
  - Đầu vào: Chỉ số hàng và cột (integer hoặc slice).
  - Đầu ra: Một phần DataFrame dựa trên vị trí được chỉ định.
  - Mô tả chi tiết: Truy cập dữ liệu bất kỳ trong DataFrame thông qua vị trí.
- Hàm: `pd.DataFrame.info()`
  - Đầu ra: In thông tin tổng quan về DataFrame.
  - Mô tả chi tiết: Hiển thị thông tin tổng quan về cấu trúc và nội dung của DataFrame giúp người dùng hiểu đôi nét về dữ liệu, như: số lượng cột, tiêu đề của từng cột, số lượng

dữ liệu non-null (không trống) và kiểu dữ liệu của từng cột, lượng tài nguyên bộ nhớ được sử dụng.

- Hàm: `pd.DataFrame.describe()`
  - Đầu ra: DataFrame chứa thống kê mô tả cho các đặc trưng.
  - Mô tả chi tiết: Tạo bảng thống kê mô tả cho các cột số trong DataFrame, bao gồm: số lượng, giá trị trung bình, giá trị độ lệch chuẩn, giá trị thấp nhất, giá trị cao nhất…
- Hàm: `pd.DataFrame.columns`
  - Đầu ra: Index object chứa tên các cột của DataFrame
  - Mô tả chi tiết: Trả về danh sách tên các cột trong DataFrame.
- Hàm: `pd.DataFrame.copy()`
  - Đầu vào: `deep: (boolean, mặc định là True)`: Nhằm xác định là deep-copy hay shallow-copy.
  - Đầu ra: Một bản sao của DataFrame gốc.
  - Mô tả chi tiết: Tạo một bản sao của DataFrame, có thể là bản sao nông hoặc sâu. Nếu True, tạo ra bản sao sâu (những thay đổi liên quan đến dữ liệu hoặc vị trí của bản sao sẽ không ảnh hưởng đến bản gốc). Nếu False, tạo ra bản sao nông (những thay đổi trên bản sao sẽ ảnh hưởng đến bản gốc).
- Hàm: `pd.Series.mean()`
  - Đầu vào:
    - `axis (int, mặc định là 0)`: Trục để tính trung bình.
    - `skipna (boolean, mặc định là True)`: Bỏ qua giá trị NaN (Not a Number) khi tính toán.
  - Đầu ra: giá trị trung bình của Series (scalar).
  - Mô tả chi tiết: Tính giá trị trung bình của các phần tử trong Series.

*1.2.2. Hàm từ thư viện **NumPy***

- Hàm: `np.array()`
  - Đầu vào:
    - Dữ liệu đầu vào (list, tuple, array...).
    - Kiểu dữ liệu đầu ra mong muốn (tùy chọn).
  - Đầu ra: Mảng NumPy thỏa yêu cầu từ dữ liệu đầu vào.
  - Mô tả chi tiết: Tạo một mảng NumPy từ dữ liệu đầu vào.
- Hàm: `np.copy()`
  - Đầu vào: Mảng cần sao chép.
  - Đầu ra: Bản sao của mảng đầu vào.
  - Mô tả chi tiết: Tạo một bản sao sâu (deep-copy) của mảng đầu vào.
- Hàm: `np.argsort()`
  - Đầu vào:
    - `a: array_like`: Mảng cần sắp xếp.
    - `axis: int (optional)`: Trục để sắp xếp
    - `kind: default 'quicksort'`: Thuật toán sắp xếp, bao gồm: quick-sort, merge-sort, heap-sort và stable.
    - `order: str (optional)`: Thứ tự sắp xếp (ascending hoặc descending, tùy chọn).
  - Đầu ra: Mảng chứa các chỉ số (indices) sắp xếp các phần tử trong mảng đầu vào theo thứ tự.
  - Mô tả chi tiết: Trả về các chỉ số sẽ sắp xếp mảng đầu vào. Chỉ số này có thể được sử dụng để truy xuất các phần tử theo thứ tự sắp xếp.
- Hàm: `np.column_stack()`
  - Đầu vào:
    - `tup: sequence of 1-D or 2-D arrays`: Một dãy các mảng.
  - Đầu ra: Một mảng 2D với các cột là các mảng đầu vào.

- Mô tả chi tiết: Ghép các mảng đầu vào theo chiều dọc, tức là tạo ra một mảng mới bằng cách xếp các mảng đầu vào làm các cột của mảng mới.

- Hàm: `np.random.permutation()`
  - Đầu vào:
    - `x: int or array_like`: Một mảng hoặc một số nguyên.
  - Đầu ra: Mảng có thứ tự ngẫu nhiên của các phần tử trong mảng input hoặc mảng với giá trị từ 0 đến n-1 ngẫu nhiên.
  - Mô tả chi tiết: Trả về một hoán vị ngẫu nhiên của mảng đầu vào hoặc dãy số từ 0 đến n-1 nếu đầu vào là số nguyên.
- Hàm: `np.concatenate()`
  - Đầu vào:
    - `a1,a2,…: sequence of array_like`: Một chuỗi các mảng để nối.
    - `axisint (optional)`: Trục để nối các mảng.
  - Đầu ra: Mảng NumPy mới được nối từ các mảng đầu vào.
  - Mô tả chi tiết: Nối các mảng theo chiều được chỉ định để tạo ra một mảng mới.
- Hàm: `np.ravel()`
  - Đầu vào:
    - `a: array_like`: Mảng đầu vào.
    - `Order: {'C','F', 'A', 'K'} (optional)`: Thứ tự duyệt mảng, bao gồm:
      'C': Hàng-major (mặc định).
      'F': Cột-major.
      'A': Theo kiểu Fortran nếu mảng liên tục trong bộ nhớ, hàng-major nếu không.
      'K': Theo thứ tự trong bộ nhớ, đảo ngược nếu strides âm.
  - Đầu ra: Mảng 1D chứa tất cả phần tử của mảng đầu vào.

- Mô tả chi tiết: Chuyển đổi mảng đầu vào thành mảng 1 chiều theo thứ tự chỉ định.

- Các hàm `np.ones()`, `np.linalg.inv()`, `np.mean()`, `np.random.seed()`, `np.full()`, `np.array_equal()`, `np.abs()` được lược bỏ vì chủ yếu là các hàm dùng để tính toán, xử lý và thao tác mảng đơn giản.

*1.2.3. Hàm từ thư viện **Matplotlib***

- Hàm: `plt.subplots()`
  - Đầu vào:
    - `nrows, ncols`: Số hàng và cột của lưới subplots (mặc định là 1).
    - `figsize`: Kích thước của figure (tùy chọn).
  - Đầu ra:
    - `fig`: Đối tượng Figure.
    - `ax`: Mảng các đối tượng Axes.
  - Mô tả chi tiết: Tạo một figure và một tập hợp các subplots.
- Hàm: `plt.show()`
  - Đầu ra: Hiển thị đồ thị.
  - Mô tả chi tiết: Hiển thị tất cả các figure đang mở.
- Hàm: `plt.figure()`
  - Đầu vào:
    - `figsize`: Kích thước của figure (tùy chọn).
    - `dpi`: Độ phân giải của figure (tùy chọn).
  - Đầu ra: Đối tượng Figure mới.
  - Mô tả chi tiết: Tạo một figure mới hoặc kích hoạt figure hiện có.
- Hàm: `plt.title()`
  - Đầu vào:
    - `label`: Chuỗi tiêu đề được dùng cho Axes hiện tại.

        - `fontdict`: Từ điển chứa các thuộc tính font (tùy chọn), bao gồm: `fontsize`, `fontweight`, `color`, `verticalalignment`...
    - Đầu ra: Đối tượng text mô tả đồ thị (tiêu đề).
    - Mô tả chi tiết: Đặt tiêu đề cho axes hiện tại.
- Hàm: `plt.xticks()`
    - Đầu vào:
        - `ticks`: Danh sách các vị trí tick.
        - `labels`: Danh sách nhãn tương ứng (tùy chọn).
    - Đầu ra:
        - `locs`: Danh sách vị trí tick
        - `labels`: Đối tượng list của các nhãn tick
    - Mô tả chi tiết: Lấy hoặc đặt các vị trí và gán nhãn tick trên trục x.

*1.2.4. Hàm từ thư viện **Seaborn***

- Hàm: `sns.histplot()`
    - Đầu vào:
        - `Data: pandas.DataFrame, numpy.ndarray`: mảng dữ liệu đầu vào
        - `x, y`: tên cột hoặc vector dữ liệu
        - `kde: Boolean`: kde (kernel density estimate) là phương pháp xác định hình dạng phân phối của dữ liệu. Ở những nơi có nhiều điểm dữ liệu tập trung thì số lượng các đường cong chồng lấn lên nhau sẽ nhiều hơn và do đó khi tính tổng cộng dồn của nó ta sẽ thu được một giá trị lũy kế kernel density lớn hơn và ngược lại.
        - Các tham số tùy chọn như `hue`, `weights`, `stat`, `bins`...
    - Đầu ra: Đối tượng AxesSubplot.

- Mô tả chi tiết: Vẽ biểu đồ histogram dựa trên dữ liệu đầu vào.

- Hàm: `sns.pairplot()`
  - Đầu vào:
    - `Data: pandas.DataFrame`: mảng dữ liệu đầu vào
    - Các tham số tùy chọn như `hue`, `hue_order`, `palette`...
  - Đầu ra: Đối tượng AxesSubplot.
  - Mô tả chi tiết: Vẽ lưới các biểu đồ scatter và histogram cho các cặp biến nhằm trực quan mối quan hệ của các dữ liệu trên biểu đồ.
- Hàm: `sns.boxplot()`
  - Đầu vào:
    - `Data: pandas.DataFrame, Series`: mảng dữ liệu đầu vào
    - Các tham số tùy chọn như `x, y, hue, order, hue_order, orient, pallete`...
  - Đầu ra: Đối tượng AxesSubplot.
  - Mô tả chi tiết: Vẽ biểu đồ box plot cho một hoặc nhiều biến phân phối.
- Hàm: `sns.heatmap()`
  - Đầu vào:
    - `Data: pandas.DataFrame hoặc ma trận dữ liệu 2D`: mảng dữ liệu đầu vào.
    - `annot: Boolean` : hiển thị giá trị mỗi ô (tùy chọn).
    - `cmap`: Bảng màu (tùy chọn).
    - Các tham số tùy chọn như `x, y, hue, order, hue_order, orient, pallete`...
  - Đầu ra: Đối tượng AxesSubplot.
  - Mô tả chi tiết: Vẽ biểu đồ heatmap cho ma trận đầu vào.

1.2.5. Hàm tự cài đặt

- Hàm: `spearman_correlation(x, y)`
  - Đầu vào:
    - `x: np.array`: Mảng dữ liệu thứ nhất.
    - `y: np.array`: Mảng dữ liệu thứ hai.
  - Đầu ra: Hệ số tương quan Spearman.
  - Mô tả chi tiết: Hàm sẽ thực hiện theo các bước sau:
    - Tính rank của mảng x và y: `np.argsort(array)` trả về chỉ số của các phần tử khi sắp xếp mảng array. Sau đó, `np.argsort(np.argsort(array))` để xác định rank của các phần tử bằng cách sắp xếp các chỉ số đó.
    - Tính số lượng các phần tử n và d (là độ lệch giữa giá trị rank của x, y).
    - Áp dụng công thức tính hệ số tương quan Spearman:

$$r_s = 1 - \frac{6.\sum d^2}{n.(n^2-1)} \in [-1;1]$$

  - Hệ số tương quan Spearman (Spearman rank correlation) là thước đo phi tham số về tương quan thứ hạng. Nó đánh giá mối quan hệ giữa hai biến có thể được mô tả tốt như thế nào bằng cách sử dụng một hàm đơn điệu
- Lớp: `OLSLinearRegression()`
  - Phương thức `fit(self, X, y)`
    - Đầu vào:

      `X: np.array`: Dữ liệu input với các biến độc lập

      `y: np.array`: Dữ liệu đầu ra mục tiêu (target).
    - Đầu ra: Trả về đối tượng của lớp với các tham số tối ưu được lưu trữ trong thuộc tính.
    - Mô tả chi tiết: Thực hiện theo các bước sau:

      1. Thêm một cột đơn vị vào X để tính toán hệ số chặn (intercept).

      2. Tính toán tham số tối ưu bằng phương trình chuẩn:

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

- Phương thức `get_params(self)`
  - Đầu ra: Lấy ra các tham số tối ưu (intercept và coefficients).
  - Mô tả chi tiết: Lấy các tham số (intercept và coefficients) đã học được của mô hình.
- Phương thức `predict(self, X)`
  - Đầu vào: `X (np.array)`: Dữ liệu đầu vào để thực hiện dự đoán.
  - Đầu vào: Các giá trị dự đoán tương ứng với dữ liệu đầu vào X.
  - Mô tả chi tiết:

    1. Thêm một cột đơn vị vào X để tính toán hệ số chặn (intercept).

    2. Tính toán đầu ra dự đoán bằng cách nhân ma trận dữ liệu với vector tham số:

$$y_{pred} = Xw$$

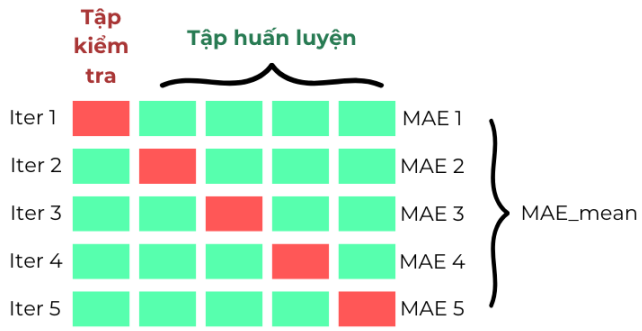
- Hàm: `mean_absolute_error(y, y_hat)`
  - Đầu vào:
    - `y: np.array`: Dữ liệu thực tế.
    - `y_hat: np.array`: Dữ liệu dự đoán.
  - Đầu ra: Giá trị sai số tuyệt đối trung bình (MAE).
  - Mô tả chi tiết: Công thức tìm độ đo sai số tuyệt đối trung bình MAE là:

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

    Trong đó:
    - y_i: giá trị thực tế
    - $\hat{y}_i$: giá trị dự đoán
  - Ghi chú: Ở phần cài đặt hàm, tôi sử dụng `np.ravel()` để đảm bảo rằng phép trừ sẽ thực hiện trên các mảng 1D có

cùng chiều dài, tránh các lỗi liên quan đến broadcast hoặc khác chiều. Ví dụ, nếu y có dạng (n, 1) và y_hat có dạng (n,) thì việc không sử dụng `np.ravel()` có thể gây ra lỗi.

- Hàm: `create_folds(n_samples, k, random_state)`
  - Đầu vào:
    - `n_samples: int`: Số lượng mẫu dữ liệu.
    - `k: int`: Số lượng folds (mặc định là 5).
    - `random_state: int`: Seeds để tái tạo kết quả ngẫu nhiên (mặc định là 42).
  - Đầu ra: Danh sách chứa các chỉ số của mỗi fold.
  - Mô tả chi tiết: Tạo ra các folds được sắp xếp lại chỉ số ngẫu nhiên, chia dữ liệu thành k phần tương đương đảm bảo dữ liệu được đồng nhất cho các lần chạy để sử dụng trong phương pháp k-fold cross-validation.
- Hàm: `k_fold_cross_validation(X, y, folds)`
  - Đầu vào:
    - `X: np.array`: Dữ liệu đầu vào.
    - `y: np.array`: Dữ liệu đầu ra.
    - `folds: list`: Danh sách chứa các chỉ số mỗi fold.
  - Đầu ra: Giá trị trung bình của độ đo sai số tuyệt đối trung bình dựa trên số lần huấn luyện và kiểm tra trên tập huấn luyện.
  - Mô tả chi tiết: Hàm sẽ thực hiện theo các bước sau:
    - Chia tập huấn luyện và kiểm tra: Cho mỗi fold, huấn luyện mô hình trên các fold khác và kiểm tra trên fold hiện tại.
    - Tính toán MAE: Trả về giá trị MAE trung bình, đại diện cho hiệu suất của mô hình đó.
    - Phương pháp k-fold cross-validation có thể được biểu diễn qua hình ảnh sau:



- Hàm: `create_model_1(X)`
  - Đầu vào: `X (pandas.DataFrame)`: Dữ liệu đầu vào chứa các đặc trưng.
  - Đầu ra: DataFrame mới bao gồm hai đặc trưng Hours Studied và Previous Scores.
  - Lý do xây dựng mô hình: Đây là mô hình cơ bản kết hợp hai đặc trưng có tương quan mạnh nhất với Performance Index thông qua hệ số Spearman được tính ở phần EDA lần lượt là Previous Scores (hệ số Spearman 0.9202) và Hours Studied (hệ số Spearman 0.3479).
- Hàm: `create_model_2(X)`
  - Đầu vào: `X (pandas.DataFrame)`: Dữ liệu đầu vào chứa các đặc trưng.
  - Đầu ra: DataFrame mới bao gồm ba đặc trưng Hours Studied, Previous Scores và một đặc trưng tương tác giữa hai biến này (Previous Scores x Hours Studied).
  - Lý do xây dựng mô hình: Đây là mô hình mở rộng từ mô hình 1 bằng cách thêm một đặc trưng là tương tác giữa hai đặc trưng là Previous Scores và Hours Studied nhằm giúp phát hiện các mối quan hệ phi tuyến tiềm ẩn giữa các đặc trưng với Performance Index.
- Hàm: `create_model_3(X)`
  - Đầu vào: `X (pandas.DataFrame)`: Dữ liệu đầu vào chứa các đặc trưng.

- Đầu ra: DataFrame mới, trong đó mỗi đặc trưng đã được chuẩn hóa.

  Mô tả chi tiết: Dữ liệu được chuẩn hóa bằng cách trừ đi giá trị trung bình và chia cho độ lệch chuẩn (standard deviation). Để dễ hình dung, ta có công thức tìm điểm chuẩn z như sau:

$$z = \frac{x - \mu}{\sigma}$$

  Với:

  - μ là giá trị trung bình của dữ liệu;
  - σ là độ lệch chuẩn của tổng thể.

  Kết quả là dữ liệu được chuẩn hóa về phân phối chuẩn (phân phối Gaussian) với giá trị trung bình là 0 và độ lệch chuẩn là 1. Điều này giúp cân bằng tầm quan trọng của các đặc trưng có thang đo khác nhau và cải thiện hiệu suất của nhiều thuật toán học máy.
- Lý do xây dựng mô hình:
  - Công bằng hơn trong huấn luyện mô hình: Các đặc trưng có các đơn vị đo lường khác nhau hoặc có các giá trị nằm trong các khoảng rất khác biệt. Điều này có thể dẫn đến sự chênh lệch trong việc huấn luyện mô hình, khi mà mô hình có thể đặt quá nhiều trọng số vào các đặc trưng có giá trị lớn hơn.
  - Cải thiện hiệu suất mô hình: Chuẩn hóa giúp mô hình tránh tình trạng trọng số của các đặc trưng bị lệch do quy mô dữ liệu, và do đó giúp mô hình hội tụ nhanh hơn trong quá trình huấn luyện.

2. BÁO CÁO PHÂN TÍCH KHAI PHÁ DỮ LIỆU

2.1. Tổng quan về Bộ dữ liệu

- Bộ dữ liệu huấn luyện có tổng cộng 9000 mẫu với mỗi mẫu đại diện cho một sinh viên. DataFrame này bao gồm 5 đặc trưng, bao gồm:
    - **Hours Studied**: Thời gian học (giờ)
    - **Previous Scores**: Điểm số trước đó
    - **Extracurricular Activities**: Hoạt động ngoại khóa (có/không)
    - **Sleep Hours**: Thời gian ngủ (giờ)
    - **Sample Question Papers Practiced**: Số lượng đề đã luyện tập
- Phân tích Thống kê mô tả:
    - **Hours Studied:**
        - Trung bình: 4.98 giờ
        - Độ lệch chuẩn: 2.59 giờ
        - Giá trị nhỏ nhất: 1 giờ
        - Giá trị lớn nhất: 9 giờ
        - Phân vị 25%: 3 giờ
        - Phân vị 50% (median): 5 giờ
        - Phân vị 75%: 7 giờ
        - ➔ Nhận xét: Dữ liệu về thời gian học cho thấy sinh viên có sự phân bố khá rộng về số giờ học, dao động từ 1 đến 9 giờ. Khoảng giữa của phân phối, từ 3 đến 7 giờ, bao gồm 50% sinh viên. Thời gian học trung bình là 4.98 giờ, cho thấy một số sinh viên học rất ít, trong khi một số khác lại dành thời gian học nhiều hơn đáng kể. Độ lệch chuẩn 2.59 giờ cho thấy có sự phân tán tương đối trong thời gian học của các sinh viên.
    - **Previous Scores:**
        - Trung bình: 69.40 điểm
        - Độ lệch chuẩn: 17.37 điểm
        - Giá trị nhỏ nhất: 40 điểm
        - Giá trị lớn nhất: 99 điểm

- Phân vị 25%: 54 điểm
- Phân vị 50% (median): 69 điểm
- Phân vị 75%: 85 điểm

➔ Nhận xét: Điểm số trước đó của sinh viên trải dài từ 40 đến 99 điểm. Điểm trung bình là 69.40, với phân vị 50% là 69 điểm, cho thấy sự phân bố đều của điểm số xung quanh giá trị trung bình. Độ lệch chuẩn cao, 17.37 điểm, cho thấy sự khác biệt đáng kể về năng lực học tập giữa các sinh viên. Điều này có thể là dấu hiệu của một số yếu tố bên ngoài ảnh hưởng đến kết quả học tập, chẳng hạn như phương pháp học tập hoặc môi trường giáo dục.

- **Extracurricular Activities:**
  - Trung bình: 0.49
  - Độ lệch chuẩn: 0.50

  ➔ Nhận xét: Khoảng 49% sinh viên tham gia các hoạt động ngoại khóa, trong khi 51% không tham gia. Điều này cho thấy có sự phân chia rõ rệt trong việc tham gia các hoạt động ngoài giờ học, có thể là do sự khác biệt về thời gian rảnh rỗi, sự quan tâm đến các hoạt động ngoại khóa, hoặc các yếu tố cá nhân khác.

- **Sleep Hours:**
  - Trung bình: 6.54 giờ
  - Độ lệch chuẩn: 1.70 giờ
  - Giá trị nhỏ nhất: 4 giờ
  - Giá trị lớn nhất: 9 giờ
  - Phân vị 25%: 5 giờ
  - Phân vị 50% (median): 6 giờ
  - Phân vị 75%: 8 giờ

  ➔ Nhận xét: Thời gian ngủ trung bình của sinh viên là 6.54 giờ, với phạm vi từ 4 đến 9 giờ. Độ lệch chuẩn 1.70 giờ cho thấy có sự khác biệt tương đối lớn về thời gian ngủ giữa các sinh viên. Mặc dù thời gian ngủ trung bình tương

đối tốt, vẫn có những sinh viên ngủ ít hơn hoặc nhiều hơn mức trung bình đáng kể, điều này có thể ảnh hưởng đến sức khỏe và khả năng học tập của họ.

- **Sample Question Papers Practiced:**
  - Trung bình: 4.59 đề
  - Độ lệch chuẩn: 2.86 đề
  - Giá trị nhỏ nhất: 0 đề
  - Giá trị lớn nhất: 9 đề
  - Phân vị 25%: 2 đề
  - Phân vị 50% (median): 5 đề
  - Phân vị 75%: 7 đề

  ➔ Nhận xét: Sinh viên trung bình luyện tập khoảng 4.59 đề mẫu, với sự phân bố từ 0 đến 9 đề. Điều này cho thấy có sự đa dạng trong việc chuẩn bị cho các kỳ thi giữa các sinh viên, có thể phản ánh sự khác biệt về phương pháp học tập hoặc mức độ chuẩn bị cho kỳ thi của mỗi cá nhân.

2.2. Phân tích Hệ số tương quan Spearman

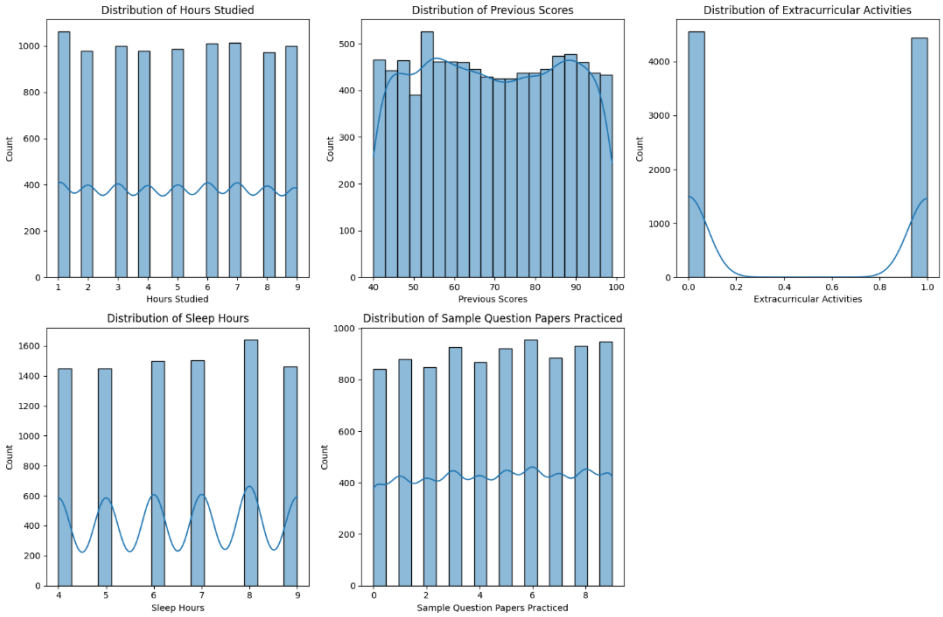
- **Hours Studied và Performance Index** có hệ số tương quan Spearman là 0.3456 → Nhận xét: Mối tương quan dương nhẹ giữa *Hours Studied* và *Performance Index* cho thấy thời gian học có ảnh hưởng đến kết quả học tập, nhưng không phải là yếu tố quyết định. Một số sinh viên có thể học ít hơn nhưng vẫn đạt được kết quả cao nhờ vào phương pháp học tập hiệu quả hoặc sự hỗ trợ từ các nguồn học tập khác.
- **Previous Scores và Performance Index** có hệ số tương quan Spearman là 0.9201 → Nhận xét: *Previous Scores* có mối tương quan rất mạnh với *Performance Index*, cho thấy rằng lịch sử học tập là yếu tố dự báo rất tốt cho kết quả học tập hiện tại. Điều này cũng có nghĩa là sinh viên có thành tích tốt trong quá khứ có nhiều khả năng tiếp tục duy trì kết quả cao trong các kỳ thi tiếp theo.

- **Extracurricular Activities và Performance Index** có hệ số tương quan Spearman là 0.0182 → Nhận xét: Mối tương quan rất yếu giữa *Extracurricular Activities* và *Performance Index* cho thấy rằng việc tham gia vào các hoạt động ngoại khóa có ít hoặc không có ảnh hưởng đáng kể đến kết quả học tập. Điều này có thể là do các hoạt động này không trực tiếp liên quan đến việc học, hoặc do sinh viên cân bằng tốt giữa học tập và các hoạt động ngoài giờ học.
- **Sleep Hours và Performance Index** có hệ số tương quan Spearman là 0.0418 → Nhận xét: *Sleep Hours* và *Performance Index* có mối tương quan rất yếu, cho thấy rằng thời gian ngủ của sinh viên không có tác động đáng kể đến kết quả học tập. Tuy nhiên, điều này không loại trừ khả năng rằng thời gian ngủ có thể ảnh hưởng đến các yếu tố khác như sức khỏe tinh thần hoặc thể chất, từ đó gián tiếp ảnh hưởng đến hiệu suất học tập.
- **Sample Question Papers Practiced và Performance Index** có hệ số tương quan Spearman là 0.0377 → Nhận xét: Mối tương quan rất yếu giữa *Sample Question Papers Practiced* và *Performance Index* cho thấy rằng việc luyện tập đề mẫu không đóng vai trò quan trọng trong việc cải thiện kết quả học tập. Điều này có thể do chất lượng của việc luyện tập, hoặc do sinh viên đã có nền tảng kiến thức tốt và không cần luyện tập nhiều để đạt kết quả cao.
- Như vậy, ta có thể rút ra một vài nhận xét sau cho bộ dữ liệu:
    - **Previous Scores** là yếu tố quan trọng nhất trong việc dự đoán *Performance Index*, thể hiện sự ảnh hưởng mạnh mẽ của lịch sử học tập đến kết quả hiện tại. Do đó, việc theo dõi và cải thiện điểm số qua từng kỳ thi là cần thiết để đảm bảo thành tích học tập ổn định và cao.
    - **Hours Studied** có mối tương quan nhẹ với *Performance Index*, cho thấy rằng mặc dù thời gian học tập quan trọng, nhưng không phải là yếu tố quyết định duy nhất. Sinh viên cần tìm kiếm phương pháp học tập hiệu quả hơn thay vì chỉ tăng thời gian học.

- **Extracurricular Activities**, **Sleep Hours**, và **Sample Question Papers Practiced** có mối tương quan rất yếu với *Performance Index*, cho thấy rằng các yếu tố này không ảnh hưởng đáng kể đến kết quả học tập. Tuy nhiên, chúng có thể đóng vai trò quan trọng trong việc phát triển các kỹ năng mềm, cải thiện sức khỏe tinh thần, và duy trì sự cân bằng trong cuộc sống của sinh viên.

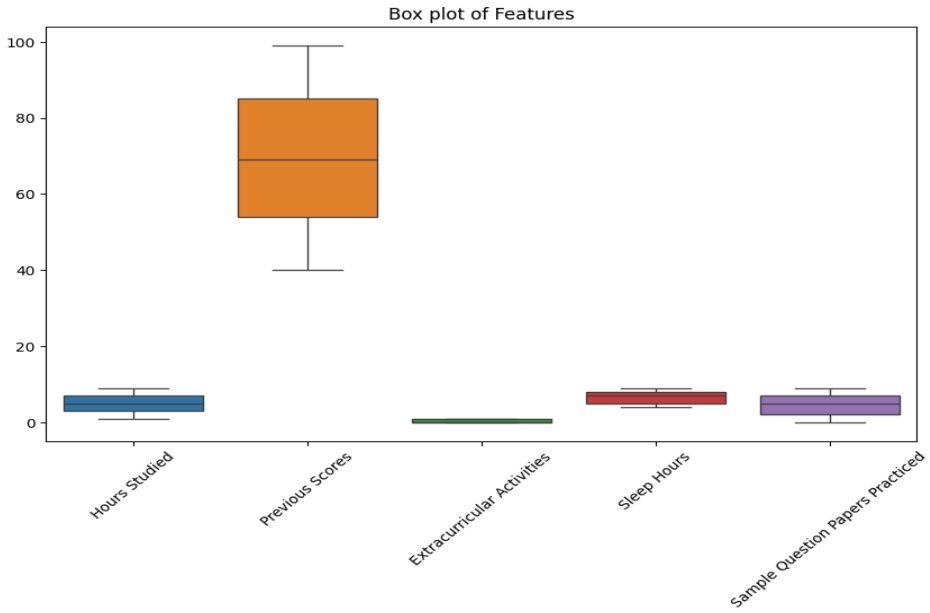
2.3. Phân tích biểu đồ

- Biểu đồ phân phối



- **Hours Studied, Sleep Hours, và Sample Question Papers Practiced** phân bố khá đồng đều, không có sự tập trung đáng kể ở bất kỳ một giá trị cụ thể nào. Chứng tỏ các sinh viên được khảo sát có thói quen học tập, ngủ nghỉ, và luyện tập đề mẫu đa dạng, không tập trung vào một nhóm nhỏ giá trị nào.
- **Previous Scores** phân phối khá đồng đều nhưng có một số đỉnh nhẹ. Điều này có thể phản ánh các ngưỡng điểm số nhất định trong hệ thống đánh giá, nơi nhiều sinh viên đạt được những điểm số tương tự.
- **Extracurricular Activities** phân phối đều giữa 2 giá trị CÓ và KHÔNG tham gia hoạt động ngoại khóa.

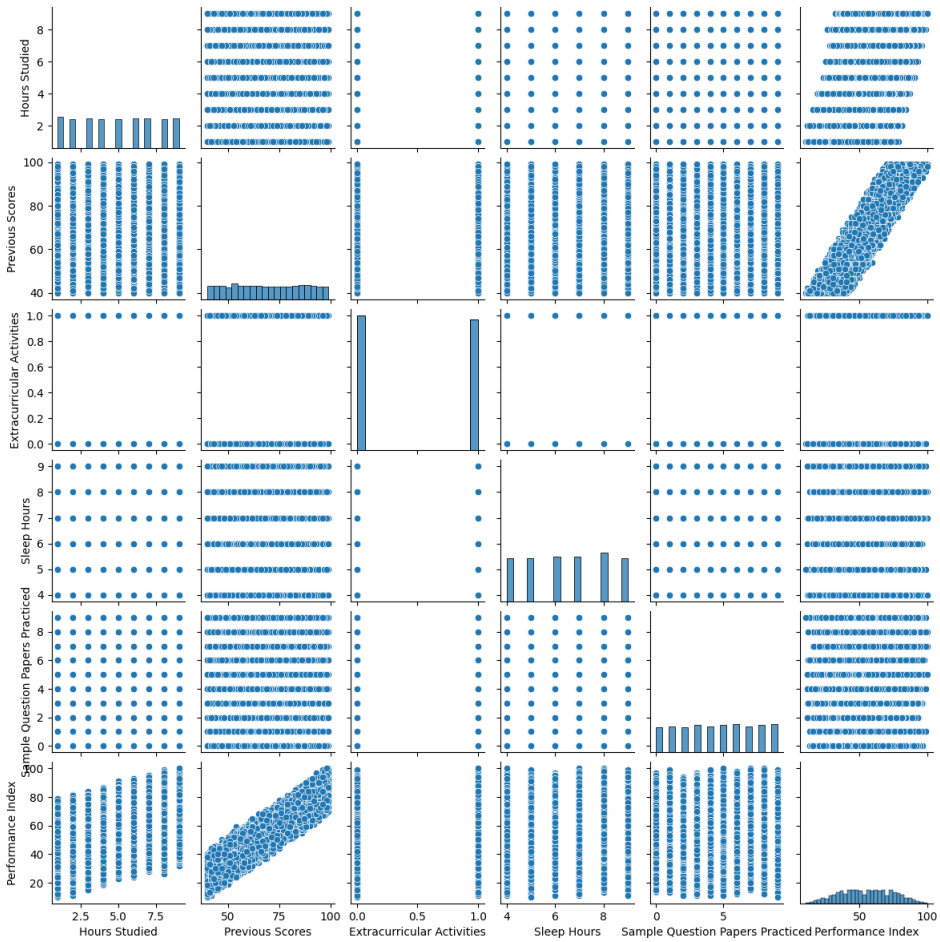
- Biểu đồ hộp



- **Hours Studied** cho thấy sự phân bố hẹp của số giờ học, nằm trong khoảng từ 1 đến 9 giờ. Hầu hết các giá trị tập trung ở giữa khoảng, phản ánh rằng đa phần sinh viên có thời gian học vừa phải. Một vài giá trị ngoại lai xuất hiện, cho thấy có những sinh viên dành thời gian học khác biệt so với đa số.
- **Previous Scores** thể hiện sự phân bố giá trị rộng nhất, từ 40 đến gần 100 điểm. Không có giá trị ngoại lai, chứng tỏ dữ liệu điểm số khá đồng đều và không có sự bất thường nào rõ rệt. Điều này nhấn mạnh tầm quan trọng của điểm số trước đó trong việc dự đoán kết quả học tập.
- **Extracurricular Activities** không mang ý nghĩa phân tích, vì biến số này chỉ bao gồm hai giá trị là 1 (có tham gia) và 0 (không tham gia). Tuy nhiên, nó vẫn cung cấp thông tin về tỷ lệ tham gia hoạt động ngoại khóa của sinh viên.
- **Sleep Hours** phân bố hẹp với khoảng từ 4 đến 9 giờ, cho thấy sinh viên có thời gian ngủ khá đồng đều. Phần lớn giá trị tập trung xung quanh mức trung bình, không có giá trị ngoại lai, điều này cho thấy rằng đa số sinh viên có thói quen ngủ tương đối ổn định.

- **Sample Question Papers Practiced** cho thấy sự phân bố hẹp từ 0 đến 9 đề. Phân bố này đồng đều và không có giá trị ngoại lai, cho thấy sự luyện tập đề mẫu khá tương đồng giữa các sinh viên, không có sự chênh lệch lớn.

- Biểu đồ cặp



- **Previous Scores và Performance Index** có quan hệ tuyến tính mạnh. Điều này nhấn mạnh rằng điểm số trước đó là yếu tố dự đoán mạnh mẽ cho chỉ số hiệu suất hiện tại, với các sinh viên có điểm số cao trước đây thường tiếp tục đạt thành tích cao.
- **Hours Studied và Performance Index** mặc dù cho thấy xu hướng tăng, mối quan hệ này không mạnh như giữa Previous Scores và Performance Index. Sự phân tán lớn hơn của các điểm cho thấy rằng thời gian học tập có ảnh hưởng đến kết quả, nhưng không phải là yếu tố quyết định.
- Các cặp đặc trưng khác không có mối quan hệ tương quan rõ rệt giữa các cặp biến số còn lại, với hầu hết các điểm dữ liệu

được phân bố rải rác. Điều này cho thấy các biến số này không có mối liên hệ trực tiếp với nhau, hoặc tác động của chúng đến nhau là không đáng kể.

- Biểu đồ heatmap

- Heatmap cho thấy tất cả các mối tương quan giữa các biến đều rất yếu, với hệ số tương quan đều dưới 0.02. Điều này cho biết các đặc trưng như số giờ học, hoạt động ngoại khóa, và số giờ ngủ không có mối quan hệ tuyến tính mạnh mẽ với nhau trong bộ dữ liệu này.
- Như vậy, ta cũng có thể suy đoán rằng các yếu tố như số giờ học, hoạt động ngoại khóa, và số giờ ngủ có thể ảnh hưởng rất ít đến nhau hoặc không có mối liên hệ nào đáng kể.

3. BÁO CÁO VÀ NHẬN XÉT KẾT QUẢ CỦA CÁC MÔ HÌNH

3.1. Tiền xử lý dữ liệu

- Cài đặt các thư viện và hàm như đã trình bày.
- Đọc dữ liệu bằng thư viện `pandas` và lấy các đặc trưng X và giá trị mục tiêu y cho các tập huấn luyện và kiểm tra.

3.2. Mô hình sử dụng toàn bộ 5 đặc trưng

- Yêu cầu: Huấn luyện mô hình hồi quy tuyến tính sử dụng toàn bộ 5 đặc trưng cho toàn bộ tập huấn luyện. Sau đó, biểu diễn công thức hồi quy tính toán y (Student Performance) dựa trên 5 đặc trưng này.
- Các bước thực hiện:
    - Bước 1: Huấn luyện mô hình hồi quy tuyến tính sử dụng toàn bộ 5 đặc trưng trên tập huấn luyện.
    - Bước 2: Lấy hệ số chặn và các hệ số tương ứng. Kết quả thu được là:
        - Hệ số chặn (Intercept): -33.969
        - Hệ số của Hours Studied: 2.852
        - Hệ số của Previous Scores: 1.018
        - Hệ số của Extracurricular Activities: 0.604
        - Hệ số của Sleep Hours: 0.474
        - Hệ số của Sample Question Papers Practiced: 0.192
    - Bước 3: In công thức hồi quy tuyến tính sử dụng toàn bộ 5 đặc trưng:

      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
    - Bước 4: Mô hình được sử dụng để dự đoán kết quả Student Performance trên tập kiểm tra.
    - Bước 5: Tính giá trị MAE trên tập kiểm tra: MAE $\approx$ 1.5956.
- Nhận xét:
    - Mô hình sử dụng toàn bộ 5 đặc trưng có kết quả đạt được khá tốt, với MAE khoảng 1.5956 trên tập kiểm tra. Điều này cho

thấy, trung bình, dự đoán của mô hình chỉ sai lệch khoảng 1.6 đơn vị so với giá trị thực tế.

- Hours Studied và Previous Scores có ảnh hưởng lớn nhất đến Student Performance, phản ánh tầm quan trọng của việc học tập và kiến thức nền tảng.
- Extracurricular Activities, Sleep Hours, và Sample Question Papers Practiced có ảnh hưởng tích cực nhưng nhỏ hơn đến kết quả học tập.
- Mô hình này hiệu quả hơn một vài mô hình khác được thử nghiệm trong đồ án, cho thấy việc sử dụng tất cả 5 đặc trưng có thể cung cấp dự đoán chính xác về hiệu suất của học sinh.
- Tuy nhiên, hệ số chặn âm (-33.969) không có ý nghĩa thực tiễn, điều này có thể gợi ý rằng mô hình có thể không phù hợp cho một vài trường hợp cực đoan. Ví dụ, nếu một học sinh có 0 giờ học, 0 điểm trước đó, không tham gia hoạt động ngoại khóa, 0 giờ ngủ và không làm bài tập mẫu, mô hình sẽ dự đoán điểm số là -33.969, điều này không có ý nghĩa trong thực tế.

3.3. Mô hình sử dụng duy nhất 1 đặc trưng

- Yêu cầu: Huấn luyện mô hình hồi quy tuyến tính sử dụng duy nhất một đặc trưng. Yêu cầu sử dụng k-fold Cross Validation (k tối thiểu là 5) để tìm ra đặc trưng tốt nhất. Sau đó, thể hiện công thức cho mô hình hồi quy theo đặc trưng tốt nhất.
- Các bước thực hiện:
    - Bước 1: Xáo trộn tập dữ liệu huấn luyện ở một hàm riêng biệt để đảm bảo dữ liệu đầu vào trước khi thực hiện cross-validation là đồng nhất.
    - Bước 2: Thực hiện cross-validation với k-fold = 5. Sau đó, ta sẽ nhận về một mảng các giá trị MAE trung bình theo từng mô hình. Kết quả thu được như sau:
        - MAE cho đặc trưng Hours Studied: 15.4486
        - MAE cho đặc trưng Previous Scores: 6.6182

- MAE cho đặc trưng Extracurricular Activities: 16.1959
- MAE cho đặc trưng Sleep Hours: 16.187
- MAE cho đặc trưng Sample Question Papers Practiced: 16.1884

➔ Đặc trưng tốt nhất là Previous Scores với MAE ≈ 6.6182 do có giá trị MAE trung bình nhỏ nhất.

- Bước 4: Huấn luyện lại mô hình với đặc trưng tốt nhất trên toàn bộ tập huấn luyện.
- Bước 5: Lấy hệ số chặn và hệ số tương ứng. Kết quả thu được là:
  - Hệ số chặn (Intercept): -14.989
  - Hệ số của Hours Studied: 1.011
- Bước 6: In công thức hồi quy tuyến tính tương ứng:

  Student Performance = -14.989 + 1.011*Previous Scores
- Bước 7: Mô hình được sử dụng để dự đoán kết quả Student Performance trên tập kiểm tra.
- Bước 8: Tính giá trị MAE trên tập kiểm tra: MAE ≈ 6.5443

- Giải thích cho kết quả mô hình sử dụng đặc trưng Previous Scores là mô hình sử dụng duy nhất một đặc trưng tốt nhất có thể dựa trên một số điểm quan trọng rút ra từ quá trình phân tích dữ liệu ban đầu:
  - Tương Quan Mạnh Mẽ giữa Previous Scores và Performance Index: Previous Scores có hệ số tương quan Spearman là 0.9201 với Performance Index. Điều này chỉ ra rằng kết quả học tập trước đó là một yếu tố dự báo cực kỳ mạnh mẽ cho kết quả hiện tại. Trong ngữ cảnh giáo dục, điều này hoàn toàn hợp lý vì thành tích học tập của sinh viên thường có xu hướng ổn định và liên tục. Những sinh viên có kết quả tốt trong quá khứ thường tiếp tục thể hiện tốt trong các kỳ thi tiếp theo do nền tảng kiến thức vững chắc và kỹ năng học tập tốt.
  - Previous Scores Phản Ánh Quá Trình Học Tập Tích Lũy: Previous Scores không chỉ đại diện cho một bài kiểm tra hay kỳ thi duy nhất mà là kết quả của nhiều kỳ thi trước đó, phản

ánh quá trình học tập và tích lũy kiến thức trong thời gian dài. Do đó, nó cung cấp một cái nhìn toàn diện và chính xác hơn về khả năng học tập thực sự của sinh viên. Điều này giúp mô hình hồi quy tuyến tính sử dụng đặc trưng này có khả năng dự báo tốt hơn so với các đặc trưng khác.

- Sự Ảnh Hưởng Của Các Yếu Tố Khác Đến Hiệu Quả Học Tập: Mặc dù các yếu tố như Hours Studied, Extracurricular Activities, Sleep Hours, và Sample Question Papers Practiced cũng đóng vai trò trong việc hình thành kết quả học tập, nhưng chúng không thể hiện mối tương quan mạnh mẽ như Previous Scores. Điều này có thể do những yếu tố này thường mang tính tạm thời và biến động theo từng học kỳ hoặc từng thời điểm cụ thể. Ví dụ, một sinh viên có thể học nhiều hơn trong một kỳ thi cụ thể nhưng không đạt kết quả như mong muốn nếu không có phương pháp học tập phù hợp hoặc không có sự chuẩn bị tốt trước đó.

- Nhận xét:
    - Khi xây dựng các mô hình hồi quy tuyến tính sử dụng duy nhất một đặc trưng, kết quả cho thấy Previous Scores là đặc trưng tốt nhất, với MAE xấp xỉ 6.6182 trong quá trình k-fold cross-validation. Đây là mô hình có độ chính xác cao nhất, và khi huấn luyện lại trên toàn bộ tập huấn luyện, mô hình này tiếp tục cho kết quả tốt với MAE khoảng 6.5443 trên tập kiểm tra. Điều này khẳng định Previous Scores là yếu tố dự báo mạnh mẽ nhất cho kết quả học tập của sinh viên, nhờ vào sự liên hệ chặt chẽ giữa thành tích học tập trong quá khứ và hiện tại.
    - Các mô hình còn lại, sử dụng các đặc trưng như Hours Studied, Extracurricular Activities, Sleep Hours, và Sample Question Papers Practiced, đều có MAE cao hơn, dao động từ 15.4486 đến 16.1959. Điều này cho thấy các đặc trưng này có mức độ dự báo yếu hơn nhiều so với Previous Scores, chứng tỏ rằng

chúng không đóng vai trò quan trọng trong việc dự báo kết quả học tập khi xét riêng lẻ.

- Mặc dù Previous Scores là đặc trưng tốt nhất khi xét riêng lẻ, nhưng việc sử dụng toàn bộ các đặc trưng kết hợp trong một mô hình đa đặc trưng (như trong mô hình đã xây dựng ở phần trước) vẫn có thể cung cấp dự đoán chính xác hơn với MAE $\approx$ 1.5956. Điều này nhấn mạnh rằng mỗi đặc trưng có thể đóng góp một phần vào việc dự báo, dù ít hay nhiều, và khi kết hợp lại, mô hình có thể tận dụng được sự bổ sung lẫn nhau giữa các đặc trưng.

3.4. Mô hình tự thiết kế

- Yêu cầu: Sinh viên tự xây dựng/thiết kế m mô hình, tìm mô hình cho kết quả tốt nhất. Yêu cầu sử dụng phương pháp k-fold Cross Validation (k tối thiểu là 5) để tìm ra mô hình tốt nhất trong m mô hình mà sinh viên xây dựng/thiết kế.

3.4.1. Mô hình tự thiết kế 1

- Mô tả: Đây là mô hình cơ bản kết hợp hai đặc trưng có tương quan Spearman mạnh nhất với Performance Index, bao gồm: Previous Scores ($r_s = 0.9202$) và Hours Studied ($r_s = 0.3479$).
- Quá trình thiết kế: Bước đầu tiên trong việc thiết kế mô hình này là lựa chọn các đặc trưng có khả năng dự đoán cao nhất. Dựa trên phân tích hệ số tương quan Spearman, hai đặc trưng Previous Scores và Hours Studied đã được chọn vì chúng thể hiện mối tương quan mạnh mẽ nhất với chỉ số hiệu suất học tập. Việc chọn chỉ hai đặc trưng giúp mô hình trở nên đơn giản và dễ hiểu, đồng thời giảm thiểu khả năng quá khớp (overfitting) do số lượng biến đầu vào ít hơn.
- Lý do thiết kế: Việc sử dụng hai đặc trưng có tương quan cao với chỉ số hiệu suất học tập nhằm mục đích tạo ra một mô hình đơn giản nhưng hiệu quả. Previous Scores là một chỉ số rõ ràng về khả năng học tập trước đó của học sinh, trong khi Hours

Studied đại diện cho nỗ lực hiện tại. Sự kết hợp của hai đặc trưng này sẽ giúp mô hình có thể dự đoán chính xác hiệu suất học tập hiện tại của học sinh, mà không bị phân tán bởi các yếu tố ít liên quan khác.

- Các bước thực hiện:
    - Bước 1: Lựa chọn các đặc trưng Hours Studied và Previous Scores từ tập dữ liệu ban đầu.
    - Bước 2: Sử dụng hàm k-fold Cross-Validation để đánh giá mô hình dựa trên MAE trung bình.

3.4.2. Mô hình tự thiết kế 2

- Mô tả: Đây là mô hình mở rộng từ mô hình tự thiết kế 1 bằng cách thêm một đặc trưng mới Hours Studied * Previous Scores để kiểm tra tác động kết hợp giữa thời gian học tập và kết quả học tập trước đó.
- Quá trình thiết kế:
    - Sau khi xây dựng mô hình cơ bản với hai đặc trưng, quá trình thiết kế tiếp theo tập trung vào việc xác định liệu có tồn tại mối quan hệ tương tác giữa Previous Scores và Hours Studied hay không. Đặc trưng mới Hours Studied * Previous Scores được tạo ra bằng cách nhân hai đặc trưng này với nhau, mục tiêu là để mô hình có thể bắt được các hiệu ứng kết hợp mà có thể không được nắm bắt rõ ràng khi chỉ sử dụng từng đặc trưng riêng lẻ.
    - Tiếp theo, quá trình huấn luyện mô hình được thực hiện trên tập dữ liệu đã mở rộng để kiểm tra xem liệu việc thêm đặc trưng tương tác này có cải thiện độ chính xác dự đoán hay không.
- Lý do thiết kế: Việc thêm vào đặc trưng tương tác Hours Studied * Previous Scores dựa trên giả định rằng thời gian học tập có thể ảnh hưởng đến hiệu suất học tập khác nhau tùy thuộc vào kết quả học tập trước đó. Ví dụ, học sinh có điểm số cao

trước đó có thể hưởng lợi nhiều hơn từ việc học tập thêm, trong khi học sinh có điểm số thấp có thể không thấy sự cải thiện tương tự. Mô hình này được thiết kế để kiểm tra giả thuyết này và đánh giá xem mối quan hệ phức tạp này có cải thiện khả năng dự đoán của mô hình hay không.

- Các bước thực hiện:
    - Bước 1: Từ mô hình tự thiết kế 1, ta cần thêm đặc trưng Hours Studied * Previous Scores vào tập dữ liệu.
    - Bước 2: Sử dụng hàm k-fold Cross-Validation để đánh giá mô hình dựa trên MAE trung bình.

3.4.3. Mô hình tự thiết kế 3

- Mô tả: Đây là mô hình chuẩn hóa dữ liệu, áp dụng phương pháp chuẩn hóa để đưa các đặc trưng về cùng một thang đo nhằm cải thiện hiệu suất mô hình.
- Quá trình thiết kế:
    - Quá trình thiết kế của mô hình này bắt đầu với việc nhận diện vấn đề: các đặc trưng trong tập dữ liệu có thể có đơn vị và thang đo khác nhau, dẫn đến việc mô hình có thể bị ảnh hưởng bởi các đặc trưng có giá trị lớn hơn. Để khắc phục vấn đề này, các đặc trưng được chuẩn hóa, tạo ra một tập dữ liệu có giá trị trung bình là 0 và độ lệch chuẩn là 1.
    - Mô hình sau đó được huấn luyện trên tập dữ liệu đã được chuẩn hóa để đảm bảo rằng tất cả các đặc trưng đều được đánh giá trên cùng một thang đo, giúp mô hình tập trung vào mối quan hệ thực sự giữa các đặc trưng và chỉ số hiệu suất học tập.
- Lý do thiết kế: Chuẩn hóa dữ liệu là một bước quan trọng trong nhiều bài toán hồi quy tuyến tính, đặc biệt khi các đặc trưng có đơn vị khác nhau. Quá trình này giúp tránh việc mô hình bị "thiên vị" bởi các đặc trưng có giá trị lớn, đồng thời cải thiện tính ổn định và hiệu suất của mô hình. Bằng cách chuẩn hóa,

mô hình có thể tối ưu hóa các trọng số một cách công bằng cho tất cả các đặc trưng, từ đó giúp cải thiện độ chính xác dự đoán và giảm thiểu sai số.

- Các bước thực hiện:
  - Bước 1: Chuẩn hóa các đặc trưng trong tập dữ liệu (trung bình = 0 và độ lệch chuẩn = 1).
  - Bước 2: Sử dụng hàm k-fold Cross-Validation để đánh giá mô hình dựa trên MAE trung bình.

3.4.4. Mô hình tự thiết kế tốt nhất

- Tiếp tục thực hiện từ các bước trên:
  - Bước 3: Kết quả thu được sau khi k-fold Cross-validation:
    - MAE cho mô hình tự thiết kế 1: 1.8162
    - MAE cho mô hình tự thiết kế 2: 1.8164
    - MAE cho mô hình tự thiết kế 3: 1.6215

  ➔ Mô hình tốt nhất là mô hình tự thiết kế 3 (Mô hình huẩn hóa dữ liệu) do có MAE trung bình thấp nhất.
  - Bước 4: Huấn luyện lại mô hình tốt nhất trên toàn bộ tập huấn luyện.
  - Bước 5: Lấy hệ số chặn và hệ số tương ứng. Kết quả thu được là:
    - Hệ số chặn (Intercept): 55.136
    - Hệ số của Hours Studied: 7.4
    - Hệ số của Previous Scores: 17.68
    - Hệ số của Extracurricular Activities: 0.302
    - Hệ số của Sleep Hours: 0.803
    - Hệ số của Sample Question Papers Practiced: 0.551
  - Bước 6: In công thức hồi quy tuyến tính tương ứng:
    Student Performance = 55.136 + 7.400*Hours Studied + 17.680*Previous Scores + 0.302*Extracurricular Activities + 0.803*Sleep Hours + 0.551*Sample Question Papers Practiced

    - Bước 7: Mô hình được sử dụng để dự đoán kết quả Student Performance trên tập kiểm tra.
    - Bước 8: Tính giá trị MAE trên tập kiểm tra: MAE $\approx$ 1.7695
- Giả thuyết cho mô hình đạt kết quả tốt nhất:
    - Chuẩn hóa dữ liệu giúp cân bằng ảnh hưởng của các đặc trưng: Trước khi chuẩn hóa, các đặc trưng có đơn vị và thang đo khác nhau có thể dẫn đến việc mô hình bị "thiên vị" bởi các đặc trưng có giá trị lớn hơn. Chuẩn hóa giúp giảm thiểu vấn đề này bằng cách đưa tất cả các đặc trưng về cùng một thang đo (giá trị trung bình là 0 và độ lệch chuẩn là 1). Điều này cho phép mô hình tối ưu hóa các trọng số một cách công bằng cho tất cả các đặc trưng.
    - Giảm thiểu khả năng quá khớp (overfitting): Chuẩn hóa giúp giảm thiểu khả năng quá khớp của mô hình bằng cách làm mịn sự phân phối của các đặc trưng. Điều này cho phép mô hình học được mối quan hệ thực sự giữa các đặc trưng và chỉ số hiệu suất học tập, thay vì bị ảnh hưởng bởi các giá trị cực đoan hoặc biến đổi không đều.
    - Tăng cường tính ổn định của mô hình: Khi các đặc trưng có đơn vị và thang đo khác nhau, mô hình có thể trở nên không ổn định và khó khăn trong việc hội tụ đến nghiệm tối ưu. Chuẩn hóa giúp cải thiện tính ổn định của mô hình bằng cách đảm bảo rằng quá trình tối ưu hóa trọng số diễn ra mượt mà và hiệu quả hơn.
    - Hiệu quả của các đặc trưng chính: Các đặc trưng Hours Studied, Previous Scores, Extracurricular Activities, Sleep Hours, và Sample Question Papers Practiced sau khi được chuẩn hóa đã thể hiện một mối quan hệ rõ ràng hơn với chỉ số hiệu suất học tập. Điều này dẫn đến sự gia tăng độ chính xác của dự đoán, vì mô hình có thể nhận diện và khai thác tốt hơn mối quan hệ giữa các đặc trưng và kết quả học tập.
- Nhận xét:

- Mô hình với các đặc trưng đã được chuẩn hóa có kết quả đạt được khá tốt, với MAE khoảng 1.6215 trên tập kiểm tra. Điều này cho thấy, trung bình, dự đoán của mô hình chỉ sai lệch khoảng 1.6 đơn vị so với giá trị thực tế.
- Mô hình này tuy có hiệu quả kém hơn một ít so với mô hình sử dụng toàn bộ 5 đặc trưng ở trên, nhưng lại có các hệ số phù hợp với phân tích khai phá dữ liệu (EDA) thực hiện ở phần 2 (Previous Scores có ảnh hưởng lớn nhất, kế tiếp là Hours Studied và Sleep Hours, Sample Question Papers Practiced, Extracurricular Activities lần lượt xếp ở phía sau) cũng như là hệ số chặn dương (55.136) giúp cho mô hình này có ý nghĩa hơn khi áp dụng vào thực tiễn, phù hợp với hầu hết các trường hợp, kể cả các trường hợp cực đoan ở ví dụ trên.

TỔNG KẾT

Đồ án 03 - Linear Regression đã thành công trong việc phân tích dữ liệu và xây dựng mô hình dự đoán sử dụng hồi quy tuyến tính. Thông qua việc sử dụng ngôn ngữ lập trình Python và các thư viện hỗ trợ mạnh mẽ như NumPy, Pandas, Matplotlib, và Seaborn, đồ án đã giúp tôi thực hành các bước cơ bản trong việc khám phá, trực quan hóa dữ liệu, và xây dựng các mô hình dự đoán.

Đầu tiên, phần phân tích khám phá dữ liệu (EDA) đã giúp làm rõ mối quan hệ giữa các đặc trưng trong bộ dữ liệu và chỉ số thành tích của sinh viên. Thông qua các phân tích thống kê và biểu đồ trực quan, đồ án đã xác định được những đặc trưng quan trọng nhất, tạo nền tảng cho việc xây dựng mô hình sau này.

Trong quá trình xây dựng mô hình dự đoán, đồ án đã thử nghiệm nhiều cách tiếp cận khác nhau, bao gồm mô hình sử dụng toàn bộ 5 đặc trưng, mô hình sử dụng duy nhất 1 đặc trưng tốt nhất, và các mô hình tự thiết kế với những biến đổi đặc trưng riêng biệt. Kết quả từ các mô hình này đã được so sánh và đánh giá chi tiết, với trọng tâm là chỉ số MAE để đo lường hiệu suất của từng mô hình.

Tóm lại, đồ án đã hoàn thành mục tiêu ban đầu là xây dựng và đánh giá các mô hình hồi quy tuyến tính, đồng thời cung cấp những phân tích sâu sắc và nhận xét giá trị về hiệu suất của các mô hình. Những kết quả đạt được không chỉ minh họa rõ ràng vai trò của hồi quy tuyến tính trong việc dự đoán mà còn khẳng định tầm quan trọng của việc lựa chọn và kết hợp các đặc trưng phù hợp trong quá trình xây dựng mô hình dự đoán.

TÀI LIỆU THAM KHẢO

[1] Tài liệu Pandas, https://pandas.pydata.org/pandas-docs/stable/reference/index.html, ngày truy cập 11/8/2024.

[2] Tài liệu Matplotlib.pyplot, https://matplotlib.org/stable/api/pyplot_summary.html, ngày truy cập 11/8/2024.

[3] Tài liệu Seaborn, https://seaborn.pydata.org/api.html, ngày truy cập 11/8/2024.

[4] Pham Dinh Khanh, "Visualization trong python" – DeepAIKhanhBlog, https://phamdinhkhanh.github.io/2019/09/16/VisualizationPython.html, ngày truy cập 11/8/2024.

[5] Duy Sang, "Thống kê mô tả trong nghiên cứu – Các đại lượng về sự tương quan", https://thongke.cesti.gov.vn/dich-vu-thong-ke/tai-lieu-phan-tich-thong-ke/861-thong-ke-mo-ta-trong-nghien-cuu-dai-luong-tuong-quan, ngày truy cập 12/8/2024.