Project 03:
Linear Regression

TOÁN ỨNG DỤNG VÀ THỐNG KÊ CHO CÔNG NGHỆ THÔNG TIN

21127006 – Nguyễn Quốc Anh

Contents

I. Giới thiệu

1. Các chức năng đã hoàn thành

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...

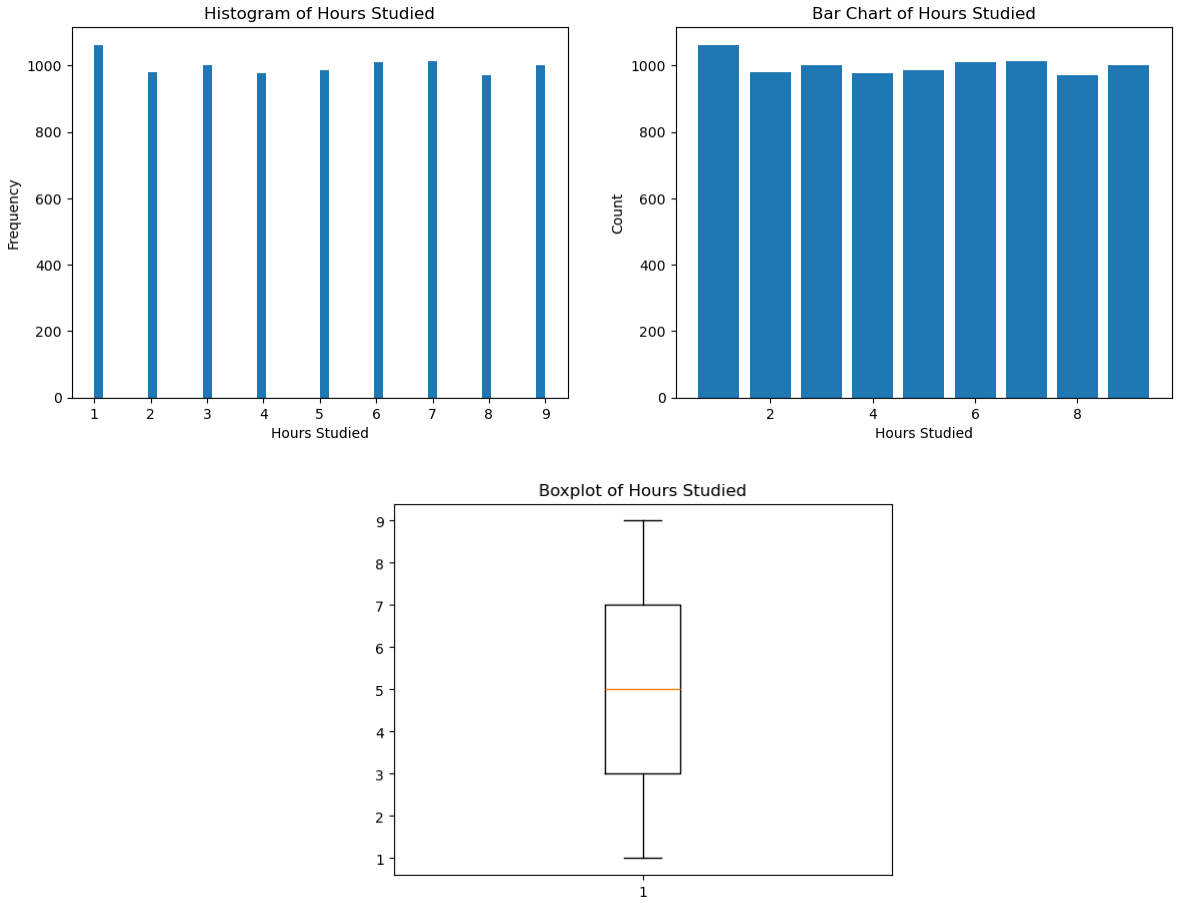
Trong đồ án này, sinh viên được yêu cầu thực hiện:

Đồ án sử dụng ngôn ngữ lập trình Python và 3 thư viện được phép sử dụng (NumPy, PIL, matplotlib) thực hiện các chức năng xử lý ảnh cơ bản sau:

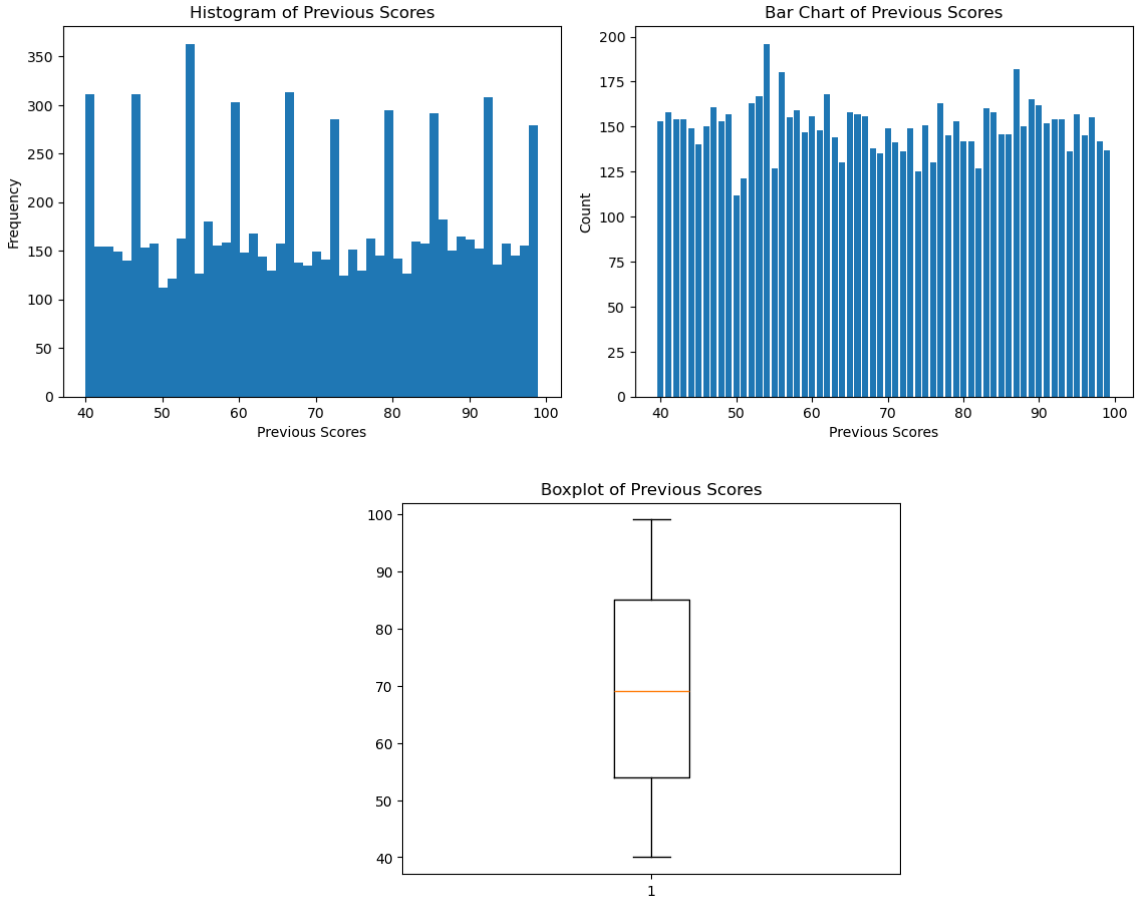
Chức năng	Mức độ hoàn thành
Thay đổi độ sáng cho ảnh	100%
Thay đổi độ tương phản	100%
Lật ảnh (ngang + dọc)	100%
Chuyển đổi ảnh RGB thành ảnh xám/sepia - Ảnh xám - Ảnh sepia	100%
Làm mờ + sắc nét ảnh - Làm mờ - Làm sắc nét	100%
Cắt 1/4 ảnh theo kích thước (cắt ở trung tâm)	100%
Cắt ảnh theo khung: - Khung hình tròn - Khung là 2 hình ellip chéo nhau	0%
Viết hàm process_image để gọi những chức năng xử lý ảnh như trên và hàm main xử lý với các yêu cầu sau: - Cho phép người dùng nhập vào tên tập tin ảnh mỗi khi hàm main được thực thi. - Cho phép người dùng lựa chọn chức năng xử lý ảnh (từ 1 đến 7) và hiển thị ảnh kết quả. Nếu có lựa chọn 0 sẽ cho phép lưu file đầu ra tương ứng với từng chức năng. Ví dụ - Đầu vào: `cat.png` - Chức năng: Làm mờ - Đầu ra: `cat_blur.png	100%

II. Chi tiết thực hiện

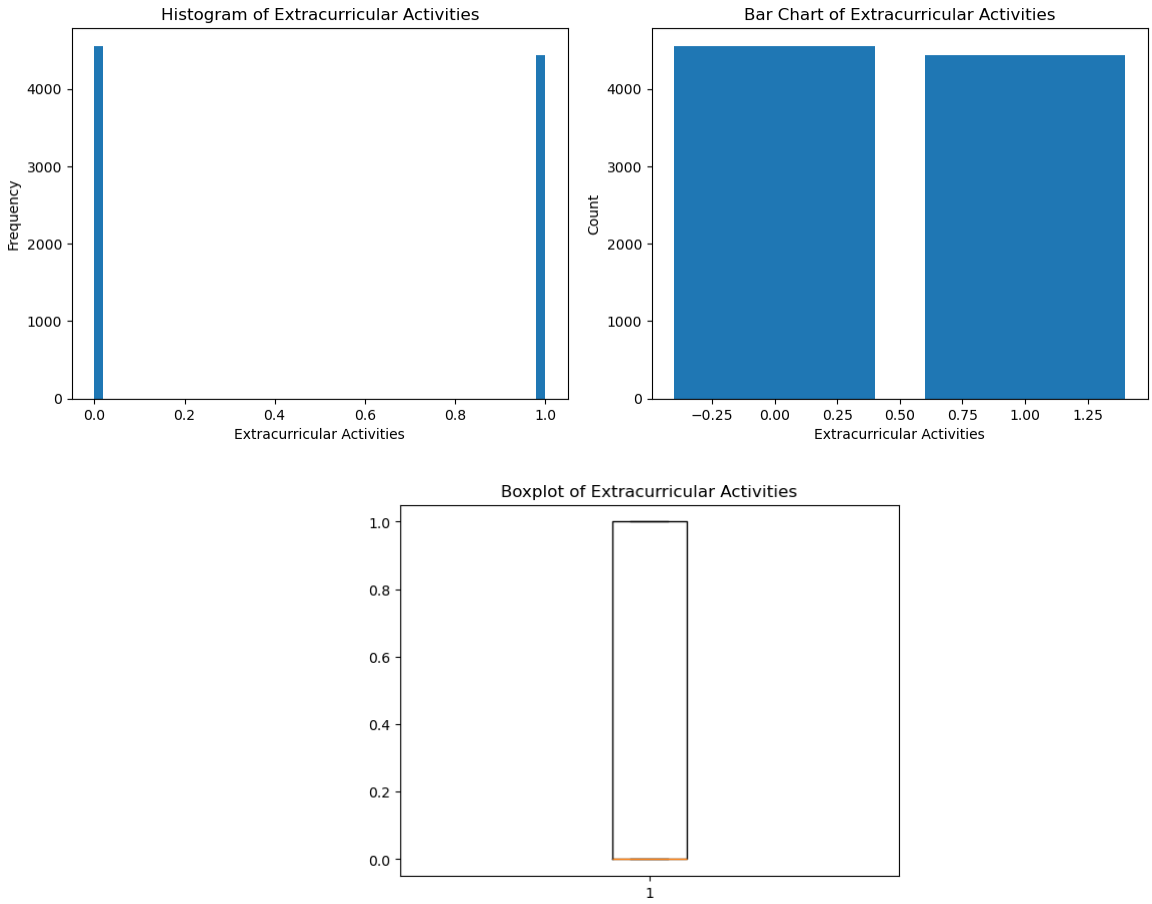
1. Thực hiện phân tích khám phá dữ liệu



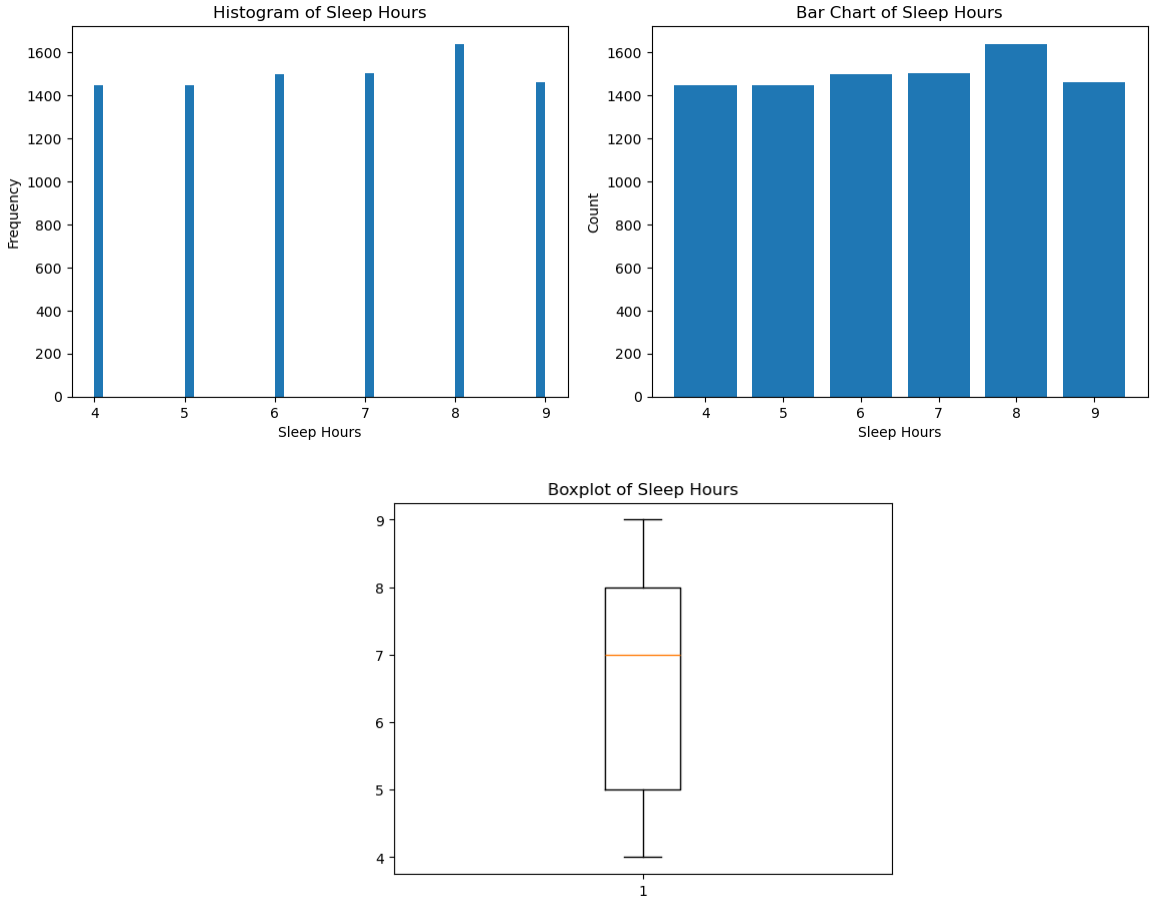
Biểu đồ histogram, cột và boxplot của Hours Studied



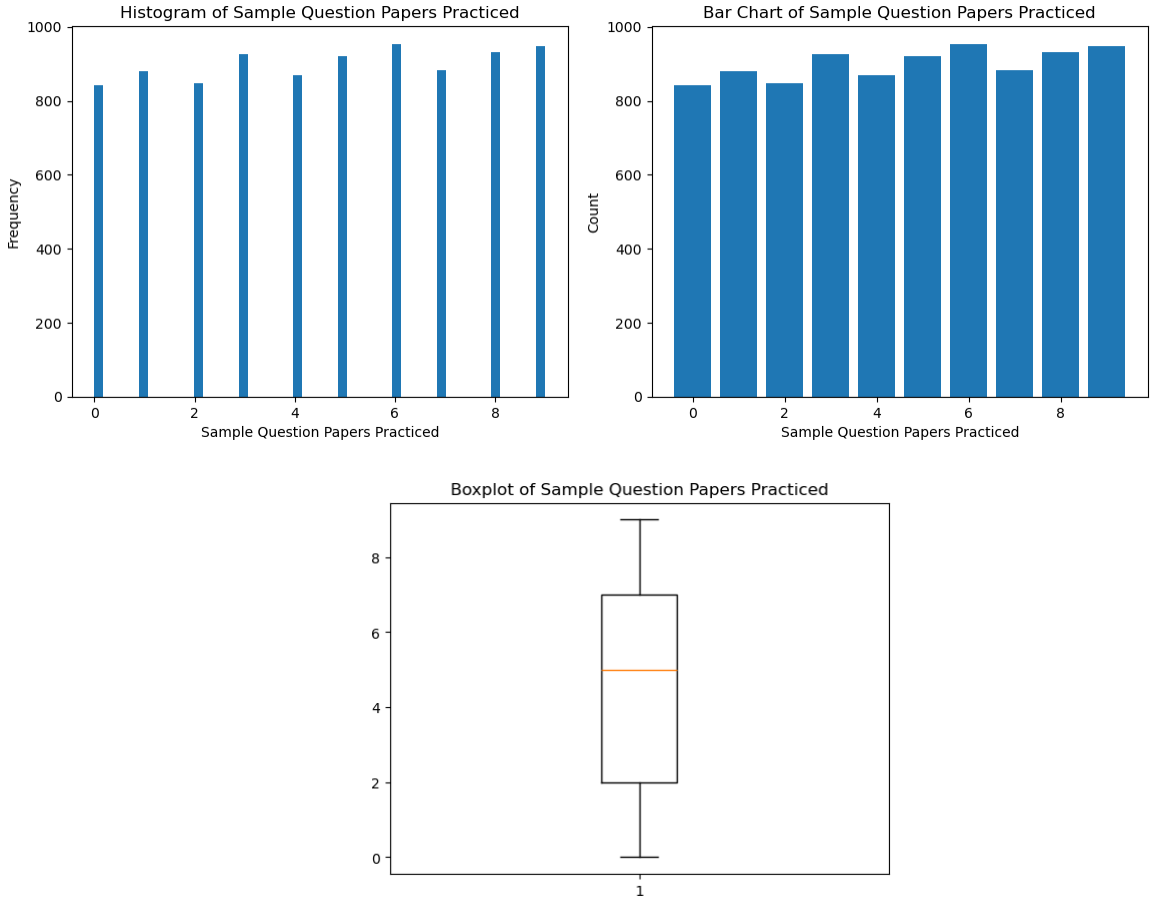
Biểu đồ histogram, cột và boxplot của Previous Scores



Biểu đồ histogram, cột và boxplot của Extracurricular Activities



Biểu đồ histogram, cột và boxplot của Sleep Hours



Biểu đồ histogram, cột và boxplot của Sample Question Papers Practiced

2. Báo cáo về kết quả, đánh giá và nhận xét các mô hình đã xây dựng

2.1. Liệt kê TẤT CẢ thư viện đã sử dụng và lý do sử dụng chúng.

- pandas:
  - Đọc dữ liệu huấn luyện và thử nghiệm và phân tích dữ liệu đó
  - Ghép dữ liệu lại với nhau
- numpy:
  - Thực hiện các phép tính toán (VD: tính trung bình bằng np.mean, tạo dãy số ngẫu nhiên với np.random).
- matplotlib:
  - Tạo biểu đồ cho dữ liệu.
- scikit-learn:
  - Truy cập và sử dụng các hàm học máy, tiền xử lí dữ liệu, lựa chọn đặc trưng và đánh giá mô hình.

2.2. Liệt kê TẤT CẢ hàm đã sử dụng/đã cài đặt và mô tả các hàm đó (kể cả các hàm từ thư viện). Các hàm tính toán từ NumPy có thể được lược bỏ.

- `def mae_score(y_true, y_pred):`
- Hàm tính toán độ lớn trung bình của lỗi (Mean Absolute Error – MAE) của một tập trong mô hình theo phương trình:

$$MAE = (1/n) * \Sigma|\, true - pred\,|$$

  với:
  - n là số điểm dữ liệu
  - true là giá trị thực
  - pred là giá trị dự đoán
- Input:
  - y_true: giá trị thực
  - y_pred: giá trị dự đoán
- Output: điểm số MAE

- `def train_linear_regression(X_train, y_train):`
- Hàm huấn luyện mô hình hồi quy tuyến tính trên tập dữ liệu được cung cấp:
  - Sử dụng thư viện scikit-learn để tạo mô hình hồi quy tuyến tính.
  - Sau đó fit mô hình với đặc trưng huấn luyện X_train và đặc trưng mục tiêu y_train.
  - Sau khi huấn luyện, hàm trả về mô hình đã được fit để thực hiện các dự đoán trên bộ dữ liệu thử nghiệm.
- Input:
  - X_train: ma trận đặc trưng huấn luyện (mỗi hàng là 1 mẫu, mỗi cột là 1 đặc trưng).
  - Y_train: ma trận mục tiêu, chứa các nhãn/giá trị mục tiêu tương ứng với X_train.
- Output:
  - model: mô hình đã qua huấn luyện

- `def cross_validate_model(X, y, cv_folds=5, random_state=42):`
- Hàm xác thực chéo trong mô hình học máy để đánh giá hiệu suất hoạt động của mô hình bằng cách huấn luyện và thử nghiệm mô hình trên nhiều tập dữ liệu của dữ liệu:
  - Chia dữ liệu thành (cv_folds) nhóm dữ liệu.
  - Với mỗi nhóm dữ liệu, huấn luyện mô hình trên các nhóm dữ liệu còn lại và đánh giá hiệu suất dựa trên nhóm dữ liệu hiện tại.
  - Tính hiệu suất trung bình của mô hình trên các nhóm dữ liệu.
  - Xuất ra danh sách điểm số
- Input:
  - X: ma trận đặc trưng
  - Y: biến mục tiêu.
  - cv_folds=5: số fold thực hiện cho cross-validation (giá trị gốc là 5)
  - random_state=42: số seed ngẫu nhiên sử dụng để tráo dữ liệu (giá trị gốc là 42)
- Output:
  - -scores: điểm MAE

- `def print_regression_formula(model, feature_names, target_name="Performance Index"):`
- Hàm in ra công thức hồi quy sau khi tính toán mô hình hồi quy tuyến tính
  VD output:

$$Performance\ Index\ =\ -32.82\ +\ 2.856 * Hours\ Studied$$

$$+\ 1.018 * Previous\ Scores\ +\ 0.472 * Sleep\ Hours$$

- Input:
  - model: mô hình hồi quy tuyến tính cần tính toán
  - feature_names: list tên các đặc trưng
  - value="Performance Index": tên của đặc trưng mục tiêu cần tính toán (nếu không có giá trị khác thì giá trị sẵn có sẽ là "Performance Index"
- Output:
  - formula: string chứa công thức hồi quy tuyến tính của mô hình

- `def plot_histogram(feature_name):`
- Hàm tạo biểu đồ histogram từ 1 đặc trưng trong dữ liệu
- Input:
  - feature_name: đặc trưng cần vẽ biểu đồ
- Output:
  - Biểu đồ histogram

- `def plot_bar_chart(feature_name):`
- Hàm tạo biểu đồ cột từ 1 đặc trưng trong dữ liệu
- Input:
  - feature_name: đặc trưng cần vẽ biểu đồ
- Output:
  - Biểu đồ cột

- `def plot_boxplot(feature_name):`
- Hàm tạo biểu đồ boxplot từ 1 đặc trưng trong dữ liệu
- Input:
  - feature_name: đặc trưng cần vẽ biểu đồ
- Output:
  - Biểu đồ cột

- `def cubed_train(features, int):`
- Hàm trả về giá trị lập phương của 1 đặc trưng trong dữ liệu huấn luyện
- Input:

  - features: tập dữ liệu chứa đặc trưng
  - int: vị trí của cột đặc trưng
- Output:
  - Cột đặc trưng với các giá trị đã được bình phương lên

```
- def cubed_test(features, int):
```

- Hàm trả về giá trị lập phương của 1 đặc trưng trong dữ liệu thử nghiệm
- Input:
  - features: tập dữ liệu chứa đặc trưng
  - int: vị trí của cột đặc trưng
- Output:
  - Cột đặc trưng với các giá trị đã được bình phương lên

```
- def main():
```

- Nhận thông tin đầu vào từ người dùng và thực hiện các lệnh có sẵn.

2.3. Báo cáo và nhận xét kết quả từ TOÀN BỘ các mô hình xây dựng được (có $1 + (5 + 1) + (m + 1)$ kết quả)

- Mô hình 2a: cả 5 đặc trưng

$$\begin{aligned} Performance\ Index\ =\ & -33.969 \\ & + 2.852 * Hours\ Studied \\ & + 1.018 * Previous\ Scores \\ & + 0.604 * Extracurricular\ Activities \\ & + 0.474 * Sleep\ Hours \\ & + 0.192 * Sample\ Question\ Papers\ Practiced \end{aligned}$$

  Có thể thấy trong mô hình 2a:
  - Đặc trưng $Hours\ Studied$ chiếm ảnh hưởng lớn nhất tới hiệu suất học tập
  - Đặc trưng $Previous\ Scores$ ảnh hưởng lớn nhì nhưng chênh lệnh rõ rệt với ảnh hưởng của đặc trưng 1
  - 3 đặc trưng còn lại có mức độ ảnh hưởng thấp tương đương nhau, trong đó ảnh hưởng thấp nhất là $Sample\ Question\ Papers\ Practiced$

2.4. Với yêu cầu 2b và 2c: Giải thích hoặc nêu giả thuyết (có logic) cho mô hình đạt kết quả tốt nhất ở mỗi yêu cầu.

- Mô hình 2b: mỗi mô hình chỉ có 1 đặc trưng
  - Mô hình $Hours\ Studied$:
    - MAE = 15.449 (±0.123)
    - Cross-validation scores: [15.54, 15.481, 15.487, 15.208, 15.527]
  - Mô hình $Previous\ Scores$:
    - MAE = 6.618 (±0.112)
    - Cross-validation scores: [6.573, 6.598, 6.836, 6.522, 6.561]

- Mô hình $Extracurricular\ Activities$:
  - MAE = 16.196 (±0.150)
  - Cross-validation scores: [16.409, 16.045, 16.313, 16.021, 16.191]
- Mô hình $Sleep\ Hours$:
  - MAE = 16.187 (±0.145)
  - Cross-validation scores: [16.388, 16.047, 16.286, 16.001, 16.213]
- Mô hình $Sample\ Question\ Papers\ Practiced$:
  - MAE = 16.188 (±0.141)
  - Cross-validation scores: [16.378, 16.012, 16.307, 16.051, 16.194]

➔ Đặc trưng tốt nhất: Previous Scores

Lí do:

- Như có thể thấy ở yêu cầu 2a, đặc trưng $Hours\ Studied$ có ảnh hưởng lớn nhất tới kết quả $Performance\ Index$.
- Điểm MAE của mô hình $Hours\ Studied$ thấp nhất trong cả 5 mô hình và giá trị này cũng thấp đáng kể so với các giá trị trong cột đặc trưng $Hours\ Studied$.

2.5. Với yêu cầu 1d: Trình bày toàn bộ quá trình và lý do trích chọn/thiết kế các đặc trưng cho `m` mô hình mà sinh viên xây dựng. Sinh viên có thể sử dụng các thuật toán/phương pháp có sẵn nhưng phải trình bày lại phương pháp đó trong báo cáo.|

- Mô hình 2c: 3 mô hình tự tạo
  - Model 1: Study Time & Play Time:
    - ['Cubed Hours Studied', 'Previous Scores', 'Cubed Extracurricular Activities']
  - Model 2: Study Time & Sleep Time:
    - ['Cubed Hours Studied', 'Previous Scores', 'Cubed Sleep Hours']
  - Model 3: Study Time & Work Done:
    - ['Cubed Hours Studied', 'Previous Scores', 'Cubed Sample Question Papers Practiced']

Do các giá trị trong các đặc trưng $Hours\ Studied$, $Sleep\ Hours$ và $Sample\ Question\ Papers\ Practiced$ đều nhỏ hơn đáng kể với cột đặc trưng $Previous\ Scores$, đồng thời do cột đặc trưng $Extracurricular\ Activities$ chỉ có các giá trị 0 hoặc 1, nên trong 3 tập dữ liệu tự tạo trên, các cột dữ liệu vị trí 0 và 2 đều được lập phương lên để dễ dàng tính toán hơn.

Từ đó ta có kết quả sau:

- Model 1: Study Time & Play Time:
  - Features: ['Hours Studied', 'Previous Scores', 'Extracurricular Activities']
  - Cross-validation scores: [2.898, 2.869, 2.903, 2.867, 2.83]
  - MAE = 2.873 (±0.026)
- Model 2: Study Time & Sleep Time:
  - Features: ['Hours Studied', 'Previous Scores', 'Sleep Hours']
  - Cross-validation scores: [2.864, 2.843, 2.868, 2.811, 2.765]
  - MAE = 2.830 (±0.038)

- Model 3: Study Time & Work Done:
  - Features: ['Hours Studied', 'Previous Scores', 'Sample Question Papers Practiced']
  - Cross-validation scores: [2.888, 2.852, 2.871, 2.855, 2.829]
  - MAE = 2.859 (±0.020)

➔ Mô hình tốt nhất: Model 2: Study Time & Sleep Time

Lí do:

- Điểm MAE của mô hình 2 thấp nhất trong cả 3 mô hình.

III. Tài liệu tham khảo

- def train_linear_regression():

What is fit() method in Python's Scikit-Learn? - GeeksforGeeks

- def cross_validate_model():

3.1. Cross-validation: evaluating estimator performance — scikit-learn 1.7.1 documentation

python - Why does shuffling training data for cross validation increase performance? - Stack Overflow