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

LINEAR REGRESSION

MÔN HỌC: TOÁN ỨNG DỤNG VÀ THỐNG KÊ CHO CÔNG NGHỆ THÔNG TIN

GIẢNG VIÊN HƯỚNG DẪN : NGUYỄN NGỌC TOÀN

LỚP : 22CLC04

SINH VIÊN THỰC HIỆN : TRẦN TRUNG HIẾU

MSSV : 22127115

HCM, 8/2024

Contents

I.THÔNG TIN SINH VIÊN

Lớp: 22CLC04

MSSV:22127115

Họ và tên: Trần Trung Hiếu

II.GIỚI THIỆU ĐỒ ÁN

Tên đồ án: Linear Regression

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...

Bộ dữ liệu Thành tích sinh viên thu thập từ nguồn Kaggle. Bộ dữ liệu này bao gồm 10,000 dòng và 6 cột, mỗi cột đại diện cho một yếu tố khác nhau ảnh hưởng đến thành tích học tập. Ý nghĩa và kiểu dữ liệu của từng cột được chi tiết hóa trong bảng mô tả dữ liệu.

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 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

Nguồn: Student Performance

Mô hình được sử dụng trong đồ án này là Hồi quy tuyến tính, thuật toán học máy dựa trên Supervised Machine Learning. Bằng cách áp dụng Hồi quy tuyến tính, đồ án này không chỉ nhằm dự đoán thành tích học tập dựa trên các yếu tố đã xác định mà còn cung cấp những hiểu biết sâu hơn về tầm quan trọng tương đối của từng yếu tố đối với thành tính học tập của học sinh.

Phần tiếp theo của báo cáo sẽ trình bày chi tiết định nghĩa vấn đề, thuật toán được sử dụng, cũng như kết quả đánh giá thực nghiệm, qua đó cung cấp những hiểu biết sâu sắc về các yếu tố ảnh hưởng đến thành tích học tập của sinh viên.

III.THỰC HIỆN ĐỒ ÁN

1. Mô tả bài toán

Dự án này hướng đến việc dự đoán Chỉ số Thành tích Học tập của Sinh viên (Academic Student Performance Index) dựa trên các yếu tố như số giờ học tập, thời gian ngủ, tham gia hoạt động ngoại khóa, và số lượng bài kiểm tra mẫu đã luyện tập. Cụ thể, đầu vào của mô hình là một ma trận các đặc trưng X, đại diện cho các yếu tố ảnh hưởng nói trên, và đầu ra y là giá trị của chỉ số thành tích học tập.

2. Thuật toán hồi quy tuyến tính

Thuật toán Hồi quy tuyến tính là một phương pháp phổ biến để xác định mối quan hệ giữa một biến đầu ra liên tục (y) và một hoặc nhiều biến đầu vào (X). Mục tiêu củathuật toán là tìm ra một đường thẳng (hoặc siêu phẳng) sao cho nó thể hiện sự phân bố gần nhất với hầu hết các điểm dữ liệu, tức là giảm thiểu khoảng cách giữa các điểm dữ liệu thực tế và đường thẳng đó.

Công thức hồi quy tuyến tính:

$$y = Xw + w0$$

Trong đó:

- y: Vector cột đại diện cho Chỉ số Thành tích Học tập của Sinh viên (Academic Student Performance Index).
- X: Ma trận đặc trưng, trong đó mỗi hàng là một mẫu dữ liệu và mỗi cột là một đặc trưng.
- w: Vector trọng số (hệ số hồi quy) cần tìm.
- w0: Hệ số tự do (hay còn gọi là hằng số điều chỉnh).

3. Các thư viện sử dụng:

pandas: Đọc file, xử lý và thao tác với dữ liệu dạng bảng (Dataframe)

numpy: Dùng để thực hiện các phép toán số học trên mảng và ma trận.

Scikit-learn (sklearn):

- model_selection.KFold: Dùng để chia dữ liệu thành các tập huấn luyện và kiểm tra bằng phương pháp K-Fold Cross-Validation. KFold giúp đảm bảo tính toàn diện trong quá trình kiểm tra và đánh giá mô hình bằng cách chia dữ liệu nhiều lần và giảm thiểu sự phụ thuộc vào cách chia dữ liệu ban đầu.
- StandardScaler: Dùng để chuẩn hóa dữ liệu, đưa các đặc trưng về cùng một thang đo (mean = 0, standard deviation = 1). Điều này giúp cải thiện hiệu suất của các mô hình học máy, đặc biệt là các mô hình nhạy cảm với tỷ lệ dữ liệu.

matplotlib.pyplot: Dùng để vẽ biểu đồ và trực quan hóa dữ liệu.

seaborn: Dùng để trực quan hóa dữ liệu (heatmap).

4. Mô tả các hàm hỗ trợ

4.1. Lớp OLSLinearRegression

Lớp này triển khai thuật toán Hồi quy tuyến tính theo phương pháp bình phương tối thiểu thông thường (Ordinary Least Squares - OLS). Mục tiêu của lớp này là tìm các hệ số hồi quy tốt nhất để mô hình hóa mối quan hệ giữa các đặc trưng đầu vào và biến đầu ra.

Lớp này gồm các phương thức:

a) fit(X,y)

Chức năng: Huấn luyện mô hình bằng cách tính toán các trọng số hồi quy dựa trên dữ liệu đầu vào X và kết quả đầu ra y.

Tham số:

- X: ma trận chứa các đặc trưng cần huấn luyện
- y: vector kết quả (target)

Mô tả:

Tính ma trận nghịch đảo giả của ma trận đặc trưng X. Cụ thể, X.T @ X là tích ma trận của ma trận chuyển vị của X với chính nó, sau đó np.linalg.inv(X.T @ X) tính toán nghịch đảo của ma trận này. Sau đó tính trọng số w

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

b) get_param()

Chức năng: Trả về trọng số của mô hình hồi quy

Mô tả: Hàm trả về w sau khi đã đào tạo mô hình

c) predict(X)

Chức năng: Dự đoán giá trị đầu ra y dựa trên dữ liệu đầu vào X và các trọng số đã tính toán.

Tham số:

- X: Ma trận đặc trưng

Mô tả:

Nhận vào ma trận X và dự đoán giá trị y theo công thức:

$$y\_predict = w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n$$

4.2. Hàm tính sai số trung bình MAE (Mean Absolute Error)

Chức năng: Tính toán lỗi tuyệt đối trung bình (Mean Absolute Error - MAE) giữa giá trị thực tế y và giá trị dự đoán $\hat{y}$

Mô tả:

Hàm nhận vào 2 đối số là y chứa các trị thực tế và $\hat{y}$ do mô hình dự đoán. Sau đó tính giá trị MAE theo biểu thức

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

Trong đó:

- n: Số lượng mẫu dữ liệu.
- y_i: giá trị thực tế của biến đầu ra.
- $\hat{y}_i$: giá trị dự đoán được từ mô hình.

4.3. Hàm preprocess (X)

Chức năng: Tiền xử lý dữ liệu đầu vào. Hàm này thêm một cột các giá trị 1 vào dữ liệu đầu vào X, giúp mô hình tính toán hệ số tự do (intercept).

Mô tả:

np.ones((x.shape[0], 1)) tạo ra một cột chứa các giá trị 1 với số hàng bằng số hàng của x.

np.hstack(...) kết hợp cột này với ma trận x, kết quả là ma trận mới X với một cột 1 ở đầu. Cột 1 này giúp mô hình tính toán hệ số tự do (intercept) trong phương trình hồi quy.

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

4.4. Hàm cross_validate(X, y, model, k=5)

Chức năng: Thực hiện kiểm tra chéo (cross-validation) để đánh giá hiệu suất của mô hình dựa trên giá trị lỗi tuyệt đối trung bình (MAE).

Kỹ thuật này gồm các bước như sau:

B1: Xáo trộn dữ liệu

B2: Chia dữ liệu theo k nhóm bằng nhau

B3: Huấn luyện dữ liệu

Quá trình huấn luyện được lặp lại k lần. Ở mỗi lần

- Một nhóm sẽ được chọn làm tập validate (kiểm tra mô hình)
- k-1 nhóm còn lại sẽ được dùng để huấn luyện mô hình
- Sau đó, kiểm tra mô hình trên tập validate.

B4: Tính điểm trung bình hiệu suất (MAE, RMSE, …) của k mô hình.

Mô tả:

Một đối tượng Kfold được thiết lập với các đối số:

- n_splits=k: Chia dữ liệu thành k phần (folds).
- shuffle=True: Xáo trộn dữ liệu trước khi chia.
- random_state=40: Đảm bảo kết quả nhất quán bằng cách thiết lập hạt giống ngẫu nhiên.

```
kf = KFold(n_splits=k, shuffle=True, random_state=40)
```

Tiến hành lặp qua từng nhóm và phân chia dữ liệu thành k nhóm.

```
for train_index, val_index in kf.split(X):
    X_cross_train, X_val = X[train_index], X[val_index]
    y_cross_train, y_val = y[train_index], y[val_index]
```

Trước khi huấn luyện, dữ liệu sẽ được thêm cột với giá trị 1

```
X_cross_train = preprocess(X_cross_train)
X_val = preprocess(X_val)
```

K nhóm dữ liệu được dùng để huấn luyện mô hình, sau đó tính giá trị mae của mô hình đó và lưu vào mảng mae_scores.

```
model.fit(X_cross_train, y_cross_train)
y_pred = model.predict(X_val)
mae_score = mae(y_val, y_pred)
mae_scores.append(mae_score)
```

Giá trị trả về là trung bình của các mae của các mô hình sau k lần lặp

```
return np.mean(mae_scores)
```

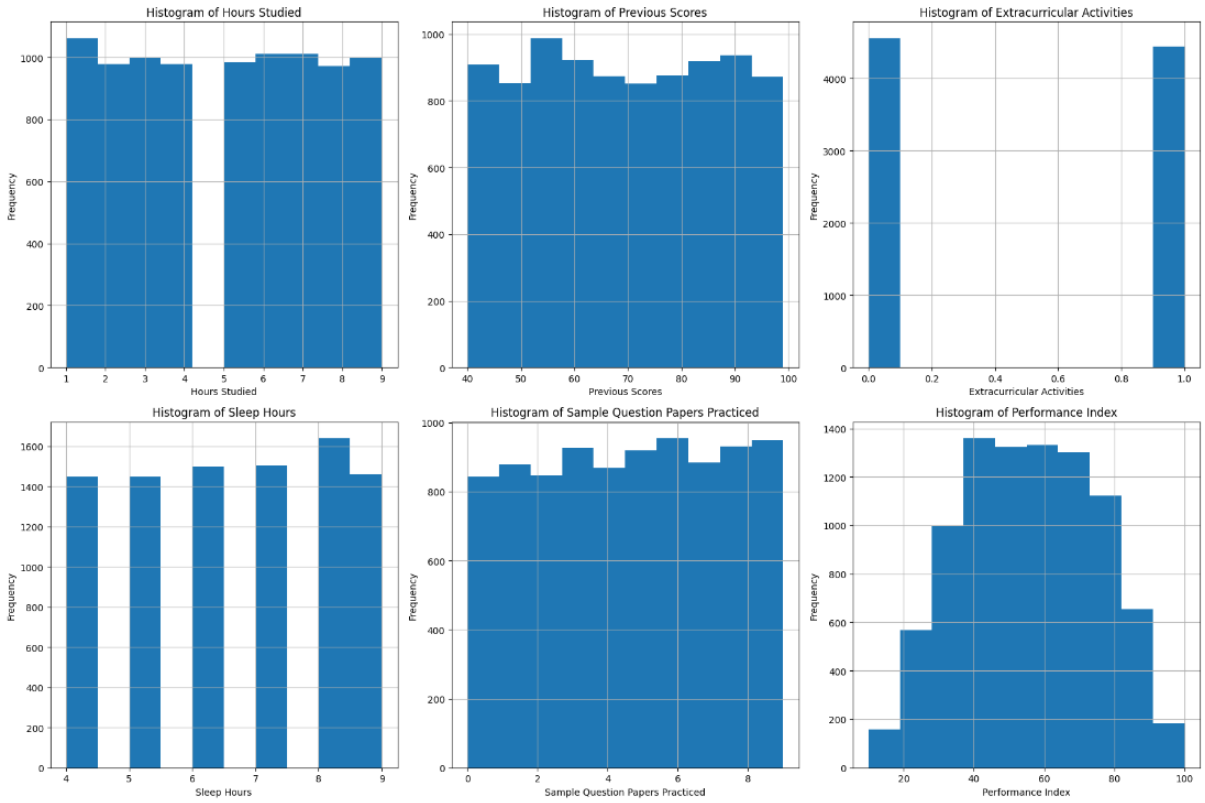
5. Thực hiện yêu cầu đồ án

Yêu cầu 1:

a) Biểu đồ histogram các đặc trưng

Dữ liệu các đặc trưng phân phối khá đồng đều. Các đặc trưng không có các giá trị ngoại lai.

Ở biểu đồ Performance Index, phân phối có hình dạng gần giống phân phối chuẩn, với đỉnh ở giữa (khoảng 50-60) và giảm dần về hai phía. Điều này cho thấy hầu hết sinh viên có chỉ số thành tích ở mức trung bình, với số ít sinh viên có thành tích rất cao hoặc rất thấp.



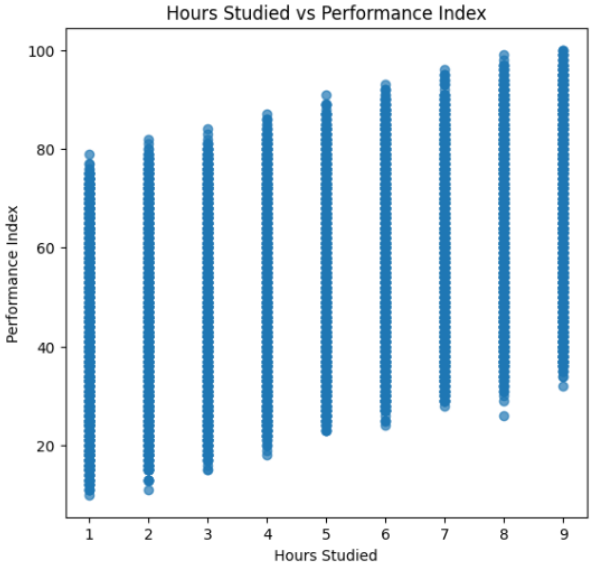
b) Biểu đồ heatmap

Đặc trưng Previous Scores và Hours Studied thể hiện môi tương quan dương rõ ràng với Performance Index (0.91 và 0.37) trong khi mối tương quan của các đặc trưng còn lại với Performance Index yếu.

Điều này cho thấy điểm số đợt trước và số giờ học tác động mạnh đến chỉ số thành tích học tập của học sinh.

c) Biểu đồ scatter

Chúng ta có thể xem rõ mối tương quan giữa Previous Scores với Performance Index và Hours Studied với Performance Index thông qua biểu đồ scatter.



<u>Yêu cầu 2a:</u>

Các bước thực hiện:

B1: Thêm cột 1 vào dữ liệu train và huấn luyện mô hình

```
# Huấn luyện mô hình
model = OLSLinearRegression()
X_train_1=preprocess(X_train)
model.fit(X_train_1, y_train)
```

B2: Thêm cột 1 vào dữ liệu test và kiểm tra MAE trên tập test

```
X_test_1=preprocess(X_test)
y_pred1 = model.predict(X_test_1)
error = mae(y_test, y_pred1)
print("Mean Absolute Error (MAE) trên tập kiểm tra:", error)
```

Với mô hình được xây dựng bằng việc sử dụng toàn bộ 5 đặc trưng sẽ cho kết quả MAE trên tập test là: 1.5956486884762893

Công thức hồi quy của mô hình này là:

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

Yêu cầu 2b:

Các bước thực hiện:

Lặp qua từng đặc trưng và áp dụng hàm cross_validate cho từng đặc trưng.

```
for feature in features:
    X_single = X_train[[feature]].values
    model = OLSLinearRegression()
    mae_result = cross_validate(X_single, y_train, model)
    mae_results.append(mae_result)
```

Ở mỗi lần lặp, dữ liệu sẽ được xáo trộn ngẫu nhiên và chia thành k nhóm như đã trình bày ở trên (hàm cross_validate). Sau đó dữ liệu được huấn luyện với duy nhất 1 đặc trưng và tính toán giá trị MAE trung bình cho từng đặc trưng.
Kết quả MAE của các đặc trưng:

STT	Mô hình với 1 đặc trưng	MAE
1	Hours Studied	15.448665
2	Previous Scores	6.618581
3	Extracurricular Activities	16.197449
4	Sleep Hours	16.188195
5	Sample Question Papers Practiced	16.188763

Công thức hồi quy của mô hình tốt nhất (đặc trưng Previous Score) là:

Student Performance = -14.989 + 1.011 * Previous Scores

MAE trên tập kết quả của mô hình đó là: 6.5442772934525015

Nhận xét: Đặc trưng Previous Score cho mô hình tốt nhất trong các mô hình sử dụng duy nhất 1 đặc trưng vì mối tương quan giữa "Previous Scores" và "Performance Index" rất cao (0.91). Khi hệ số tương quan cao, điều đó có nghĩa là khi giá trị của "Previous Scores" thay đổi, giá trị của "Performance Index" cũng thay đổi theo một cách có thể dự đoán được. Vì vậy, mô hình hồi quy chỉ cần dựa vào "Previous Scores" cũng đã có thể dự đoán "Performance Index" với độ chính xác cao.

Yêu cầu 2c:

Các bước thực hiện:

B1: Chuẩn bị dữ liệu cho 3 mô hình

B2: Xây dựng các đặc trưng cho từng mô hình

Mô hình 1: Sử dụng 2 đặc trưng 'Hours Studied', 'Previous Scores'

```
# Chuẩn bị dữ liệu cho mô hình đầu tiên
X_train_transformed = X_train.copy()
features = ['Hours Studied', 'Previous Scores']
X1 = X_train_transformed[features].values
```

Mô hình 2: Sử dụng 4 đặc trưng ('Hours Studied', 'Previous Scores', 'Sleep Hours', 'Sample Question Papers Practiced')

```
# Chuẩn bị dữ liệu cho mô hình thứ hai
X_train_transformed = X_train.copy()
features = ['Hours Studied', 'Previous Scores', 'Sleep Hours', 'Sample Question Papers Practiced']
X2 = X_train_transformed[features].values
```

Mô hình 3: Bình phương các đặc trưng ('Extracurricular Activities', 'Sleep Hours', 'Sample Question Papers Practiced')

```
# Chuẩn bị dữ liệu cho mô hình thứ ba
features = ['Hours Studied', 'Previous Scores', 'Extracurricular Activities', 'Sleep Hours', 'Sample Question Papers Practiced']
X_train_transformed = X_train.copy()

# Bình phương từng feature
X_train_transformed['Extracurricular Activities'] = X_train_transformed['Extracurricular Activities'] ** 2
X_train_transformed['Sleep Hours'] = X_train_transformed['Sleep Hours'] ** 2
X_train_transformed['Sample Question Papers Practiced'] = X_train_transformed['Sample Question Papers Practiced'] ** 2


# Chọn lại các feature sau khi đã bình phương
X3 = X_train_transformed[features].values
scaler = StandardScaler()
X3 = scaler.fit_transform(X3)
```

Ở mô hình 3, các đặc trưng đã được biến đổi và chuẩn hóa chúng bằng phương pháp StandardScaler đảm bảo rằng các đặc trưng có quy mô tương đương, giúp giảm trọng số trong quá trình tối ưu hóa mô hình.

B3: Áp dụng cross_validate cho từng mô hình

```
model1 = OLSLinearRegression()
mae_model1 = cross_validate(X1_scaled, y_train, model1)

# Mô hình thứ hai:
model2 = OLSLinearRegression()
mae_model2 = cross_validate(X2, y_train, model2)

# Mô hình thứ ba:
model3 = OLSLinearRegression()
mae_model3 = cross_validate(X3, y_train, model3)
```

B4: Tính mae trung bình cho từng mô hình

Kết quả mae cho từng mô hình

STT	Mô hình	MAE
1	Sử dụng 2 đặc trưng ('Hours Studied', 'Previous Scores')	1.816596
2	Sử dụng 4 đặc trưng ('Hours Studied', 'Previous Scores', 'Sleep Hours', 'Sample Question Papers Practiced')	1.641818
3	Bình phương 3 đặc trưng ('Extracurricular Activities', 'Sleep Hours', 'Sample Question Papers Practiced')	1.625333

Công thức hồi quy của mô hình tốt nhất (bình phương 3 đặc trưng 'Extracurricular Activities', 'Sleep Hours', 'Sample Question Papers Practiced' là:

$$\text{Student Performance} = 55.136 + 7.403 \times \text{Hours Studied} + 17.680 \times \text{Previous Scores} + 0.304 \times (\text{Extracurricular Activities})^2 + 0.801 \times (\text{Sleep Hours})^2 + 0.530 \times (\text{Sample Question Papers Practiced})^2$$

MAE trên tập kết quả của mô hình đó là: 1.6000666895617723

Nhận xét:

Mô hình thứ nhất sử dụng 2 đặc trưng có mối tương quan với Performance Index cao, tuy nhiên mô hình đó thiếu tính bổ sung của các đặc trưng còn lại. Có thể xem ở mô hình thứ hai, bổ sung thêm 2 đặc trưng với hệ số tương quan với Performance Index 0.04, cho kết quả tốt hơn mô hình đầu.

Mô hình thứ ba áp dụng bình phương đối với các đặc trưng không thể hiện rõ mối tương quan với target vì em giả định rằng mối quan hệ giữa các đặc trưng này và thành tích học tập có tính phi tuyến, cụ thể là mối quan hệ dạng bậc hai. Có thể vì lí do đó với việc đặc trưng Previous Score và Hours Studied thể hiện mối quan hệ tuyến tính nên mô hình cho kết quả tốt nhất.

6. Quá trình, lý do thiết kế/xây dựng 3 mô hình ở yêu cầu 2c

Mô hình thứ nhất: Dựa vào hệ số tương quan của các đặc trưng đối với biến target, em chọn 2 đặc trưng ‘Previous Scores’ và ‘Hours Studied’ (0.91 và 0.37) để huấn luyện mô hình vì đặc trưng này có khả năng cung cấp thông tin dự đoán đáng kể cho mô hình. Việc chọn những đặc trưng có mối tương quan cao giúp mô hình tập trung vào các yếu tố chính, loại bỏ những đặc trưng không liên quan hoặc ít liên quan, từ đó tối ưu hóa hiệu quả dự đoán.

Mô hình thứ hai: Em đã bổ sung thêm 2 đặc trưng 'Sleep Hours', 'Sample Question Papers Practiced' vì hệ số tương quan của 2 đặc trưng này đối với biến target tốt hơn đặc trưng còn lại và việc bổ sung thêm 2 đặc trưng giúp cải thiện mô hình. Việc kết hợp các đặc trưng có tính đa dạng, ngay cả khi tương quan với target không cao, có thể giúp mô hình học được những mối quan hệ tiềm ẩn và nâng cao độ chính xác của dự đoán.

Mô hình thứ ba: Qua biểu đồ scatter, em nhận thấy mối quan hệ giữa đặc trưng Previous Scores và Hours Studied với thành tính học tập có tính tuyến tính, trong khi các đặc trưng còn lại thể hiện mối quan hệ phi tuyến, cụ thể mối quan hệ dạng bậc 2 nên em đã áp dụng bình phương đối với các đặc trưng có hệ số tương quan với target thấp. Đặc biệt ở mô hình này, em đã áp dụng Standard Scaler cho tập dữ liệu nhằm giảm trọng số trong quá trình tối ưu hóa mô hình.

IV.BÁO CÁO VÀ NHẬN XÉT KẾT QUẢ CÁC MÔ HÌNH

Mô hình sử dụng 5 đặc trưng ở yêu cầu 2a

Kết quả MAE trên tập test tốt nhất

MAE = 1.5956486884762893

Các mô hình duy nhất 1 đặc trưng ở câu 2b

Đặc trưng có mối tương quang càng mạnh với chỉ số thành tích của học sinh thì sẽ có kết quả tốt hơn. Giá trị MAE của các thuộc tính 'Extracurricular Activities', 'Sleep Hours', 'Sample Question Papers Practiced' đều lớn hơn 15. Đây là kết quả MAE kém.

STT	Mô hình với 1 đặc trưng	MAE
1	Hours Studied	15.448665
2	Previous Scores	6.618581
3	Extracurricular Activities	16.197449
4	Sleep Hours	16.188195
5	Sample Question Papers Practiced	16.188763

Mô hình duy nhất 1 đặc trưng tốt nhất (Previous Scores) thể hiện tốt ở tập test với giá trị MAE trên tập test:

MAE = 6.5442772934525015

Các mô hình học sinh xây dựng ở yêu cầu 2c

Sai số của cả 3 mô hình đều tốt, không chênh lệch nhiều. Đặc biệt ở mô hình đầu tiên, sau khi bổ sung thêm đặc trưng Hours Studied đã cải thiện mô hình đáng kể với sai số giảm từ 6.5442772934525015 xuống 1.816596. Sau khi bổ sung thêm 2 đặc trưng nữa ở mô hình 2, sai số được cải thiện nhưng không đáng kể.

Mô hình thể hiện tốt nhất là ở mô hình 3 với phương pháp bình phương các đặc trưng có mối quan hệ phi tuyến với target.

STT	Mô hình	MAE
1	Sử dụng 2 đặc trưng ('Hours Studied', 'Previous Scores')	1.816596
2	Sử dụng 4 đặc trưng ('Hours Studied', 'Previous Scores', 'Sleep Hours', 'Sample Question Papers Practiced')	1.641818
3	Bình phương 3 đặc trưng ('Extracurricular Activities', 'Sleep Hours', 'Sample Question Papers Practiced')	1.625333

Mối hình 3 cũng thể hiện tốt ở tập test với giá trị MAE:

MAE = 1.6000666895617723

V.TÀI LIỆU THAM KHẢO

[1] Trang web Builtin. "Polynomial Regression: A Comprehensive Guide to Polynomial Regression and How It Works". Truy cập từ: https://builtin.com/machine-learning/polynomial-regression

Sử dụng để tham khảo trong quá trình xây dựng và thiết kế mô hình.
[2] Medium. "Feature Scaling with Scikit-learn for Data Science". Truy cập từ: https://hersanyagci.medium.com/feature-scaling-with-scikit-learn-for-data-science-8c4cbcf2daff
Tham khảo để áp dụng chuẩn hóa cho dữ liệu.