Họ và tên: Đỗ Quang Minh

Lớp CNTT 17-11

MSV: 1771020481

**Bài Kiểm Tra Số 2**

Bước 1: Thu thập dữ liệu

- Tạo file dữ liệu dạng Excel và lưu vào Drive: Lấy dữ liệu từ *copilot.microsoft.com* để tạo file Excel và lưu với tên file là: “CNTT1711\_DoQuangMinh\_Phân tích dữ liệu kết quả học tập của sinh viên tại một trường đại học..xlsx”

Bước 2: Làm sạch dữ liệu

2.1. Tạo file mới trong Python

- Import thư viện cần thiết: `pandas`, `numpy`, `matplotlib.pyplot`, `sklearn.model\_selection`, `sklearn.linear\_model`, và `sklearn.metrics`.

2.2. Lưu tên file có định dạng

- Với tên file là: “CNTT\_ĐỗQuangMinh\_BKT2.py”

2.3. Lập chương trình trong Python thực hiện các yêu cầu sau

- Đọc file Excel vừa tạo:

# Đọc dữ liệu từ file Excel

file\_path = "/content/drive/MyDrive/XacXuat/CNTT1711\_DoQuangMinh\_Phân tích dữ liệu kết quả học tập của sinh viên tại một trường đại học..xlsx"

df = pd.read\_excel(file\_path)

- In ra file đó:

print("-----Dữ liệu ban đầu:")

print(df)

- Phát hiện các giá trị lỗi, thiếu, sai định dạng và xử lý:

# 1. Làm sạch dữ liệu

# Chuyển đổi cột 'Ngày sinh' sang kiểu datetime

df['Ngày sinh'] = pd.to\_datetime(df['Ngày sinh'], errors='coerce')

# Xử lý các giá trị bị thiếu trong cột 'Điểm giữa kỳ' và 'Điểm cuối kỳ'

df['Điểm giữa kỳ'].fillna(0, inplace=True)

df['Điểm cuối kỳ'].fillna(0, inplace=True)

# Tính lại cột 'Điểm trung bình'

df['Điểm trung bình'] = (df['Điểm giữa kỳ'] + df['Điểm cuối kỳ']) / 2

# Phát hiện các giá trị lỗi, thiếu, sai định dạng và in ra

error\_rows = df[(df['Điểm giữa kỳ'] < 0) | (df['Điểm cuối kỳ'] < 0) | df.isnull().any(axis=1)]

print("Các dòng bị lỗi hoặc thiếu dữ liệu:")

print(error\_rows)

# Xóa các dòng bị lỗi hoặc có giá trị âm

df = df[(df['Điểm giữa kỳ'] >= 0) & (df['Điểm cuối kỳ'] >= 0) & df.notnull().all(axis=1)]

- In ra bộ dữ liệu mới sau khi đã làm sạch:

# In ra bộ dữ liệu mới sau khi đã làm sạch

print("-----Bộ dữ liệu mới sau khi đã làm sạch:")

print(df)

Bước 3: EDA

- In ra các chỉ số thống kê cơ bản như trung bình, trung vị, phương sai, min, max,... của các biến trong file dữ liệu:

# 2. EDA

# In ra các chỉ số thống kê cơ bản

statistics = df.describe()

print("Các chỉ số thống kê cơ bản:")

print(statistics)

- Vẽ các biểu đồ cột, phân phối để có hiểu biết thông tin từ dữ liệu:

# Vẽ các biểu đồ cột, phân phối để có hiểu biết thông tin từ dữ liệu

# Biểu đồ phân phối điểm giữa kỳ

plt.figure(figsize=(10, 6))

df['Điểm giữa kỳ'].plot(kind='hist', bins=10, alpha=0.7, color='blue')

plt.title('Phân phối điểm giữa kỳ')

plt.xlabel('Điểm giữa kỳ')

plt.ylabel('Số lượng')

plt.show()

# Biểu đồ phân phối điểm cuối kỳ

plt.figure(figsize=(10, 6))

df['Điểm cuối kỳ'].plot(kind='hist', bins=10, alpha=0.7, color='green')

plt.title('Phân phối điểm cuối kỳ')

plt.xlabel('Điểm cuối kỳ')

plt.ylabel('Số lượng')

plt.show()

# Biểu đồ phân phối điểm trung bình

plt.figure(figsize=(10, 6))

df['Điểm trung bình'].plot(kind='hist', bins=10, alpha=0.7, color='orange')

plt.title('Phân phối điểm trung bình')

plt.xlabel('Điểm trung bình')

plt.ylabel('Số lượng')

plt.show()

Bước 4: Xây dựng mô hình

4.1. Chuẩn bị dữ liệu

- Chọn các biến đầu vào và biến mục tiêu:

# 3. Xây dựng mô hình hồi quy tuyến tính

# Chuẩn bị dữ liệu cho mô hình

features = df[['Điểm giữa kỳ', 'Điểm cuối kỳ']]

target = df['Điểm trung bình']

4.2. Chia dữ liệu

- Chia dữ liệu thành tập huấn luyện và tập kiểm tra:

# Chia dữ liệu thành tập huấn luyện và tập kiểm tra

X\_train, X\_test, y\_train, y\_test = train\_test\_split(features, target, test\_size=0.2, random\_state=42)

4.3. Xây dựng mô hình

- Xây dựng mô hình hồi quy tuyến tính:

# Xây dựng mô hình hồi quy tuyến tính

model = LinearRegression()

model.fit(X\_train, y\_train)

4.4. Đánh giá mô hình

- Dự đoán trên tập kiểm tra:

# Dự đoán trên tập kiểm tra

y\_pred = model.predict(X\_test)

- Tính toán các hệ số hồi quy và đánh giá mô hình:

# Đánh giá mô hình

r2 = r2\_score(y\_test, y\_pred)

mse = mean\_squared\_error(y\_test, y\_pred)

# Hiển thị kết quả

print(f"Phương trình hồi quy: y = {beta\_0:.2f} + {beta\_1[0]:.2f} \* Điểm giữa kỳ + {beta\_1[1]:.2f} \* Điểm cuối kỳ")

print(f"Hệ số chặn (beta\_0): {beta\_0}")

print(f"Hệ số dốc (beta\_1): {beta\_1}")

print(f"Hệ số xác định (R^2): {r2}")

print(f"Sai số bình phương trung bình (MSE): {mse}")

- Vẽ biểu đồ

# Vẽ biểu đồ

plt.scatter(y\_test, y\_pred, color="blue", label="Dữ liệu thực tế")

plt.plot(y\_test, y\_test, color="red", label="Dự đoán (hồi quy)")

plt.title("Hồi quy tuyến tính: Điểm trung bình")

plt.xlabel("Điểm thực tế")

plt.ylabel("Điểm dự đoán")

plt.legend()

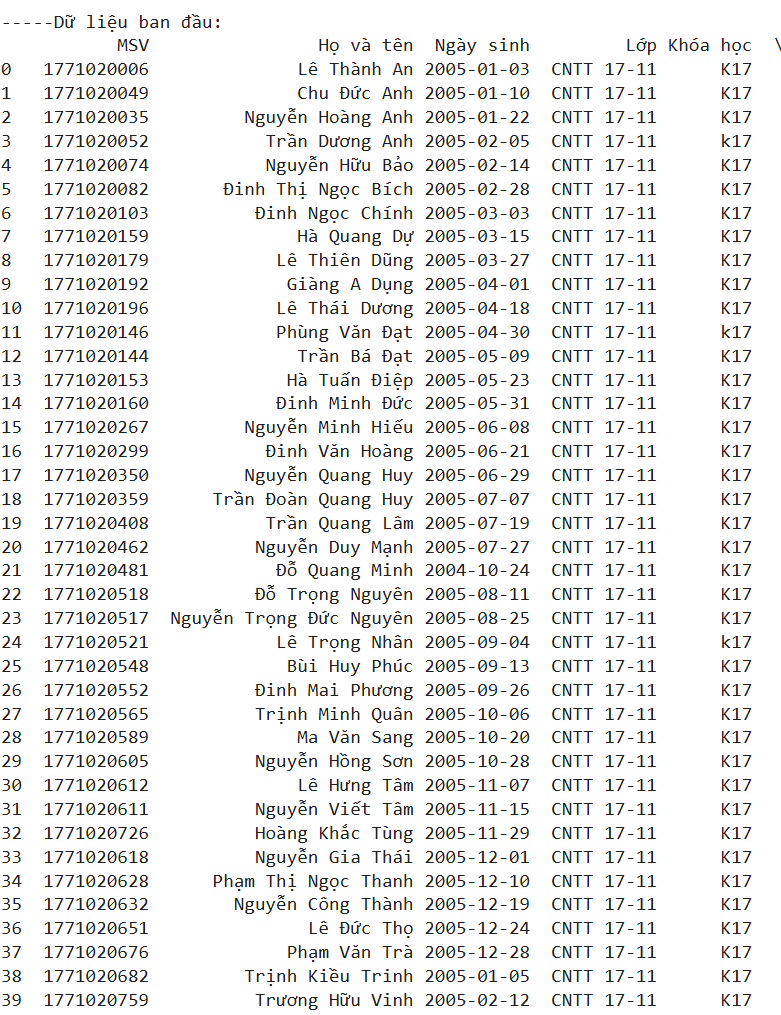
plt.grid(True)

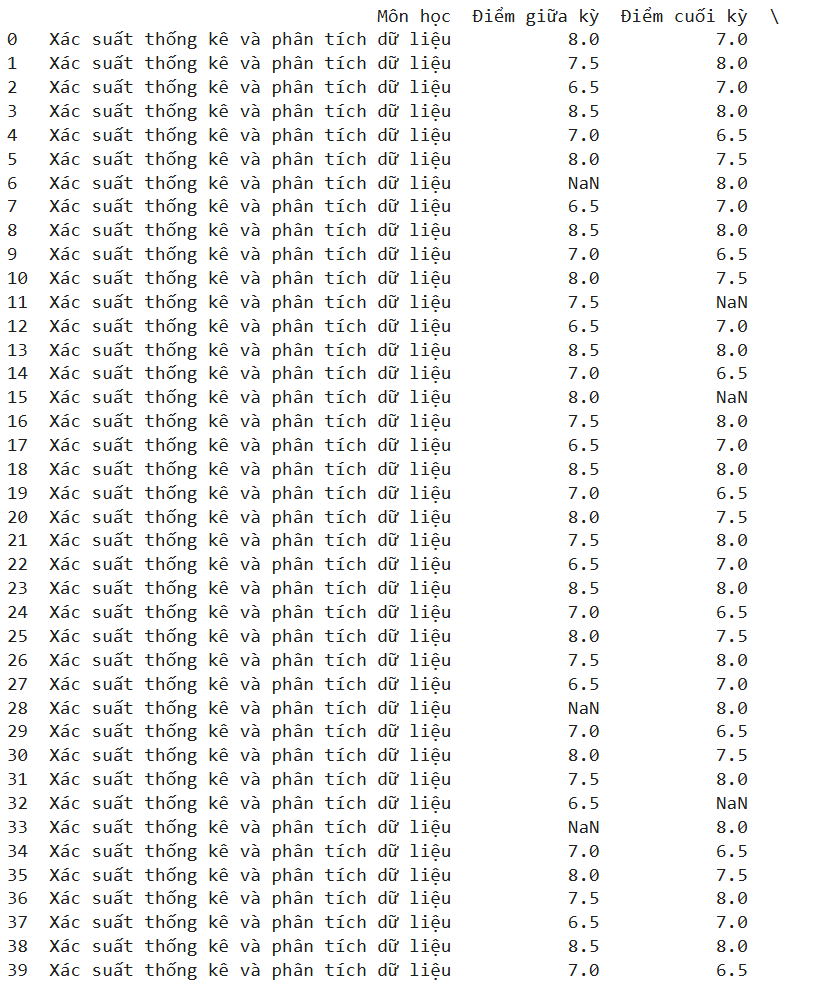
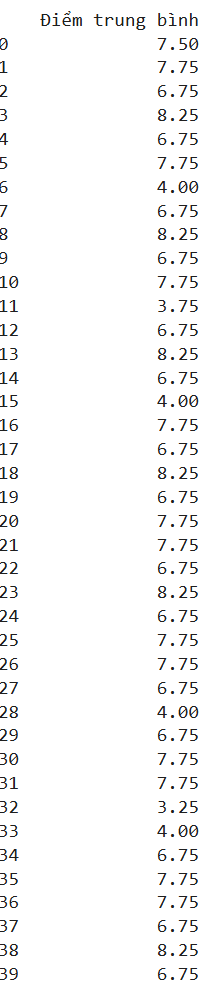
plt.show()

Bước 5: Phân tích kết quả

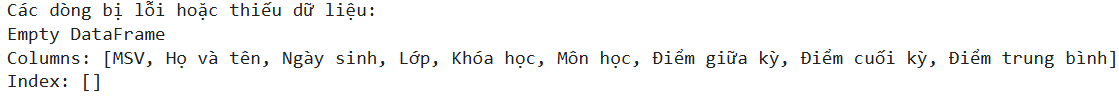
- Tóm tắt kết quả

+ In ra dữ liệu ban đầu:

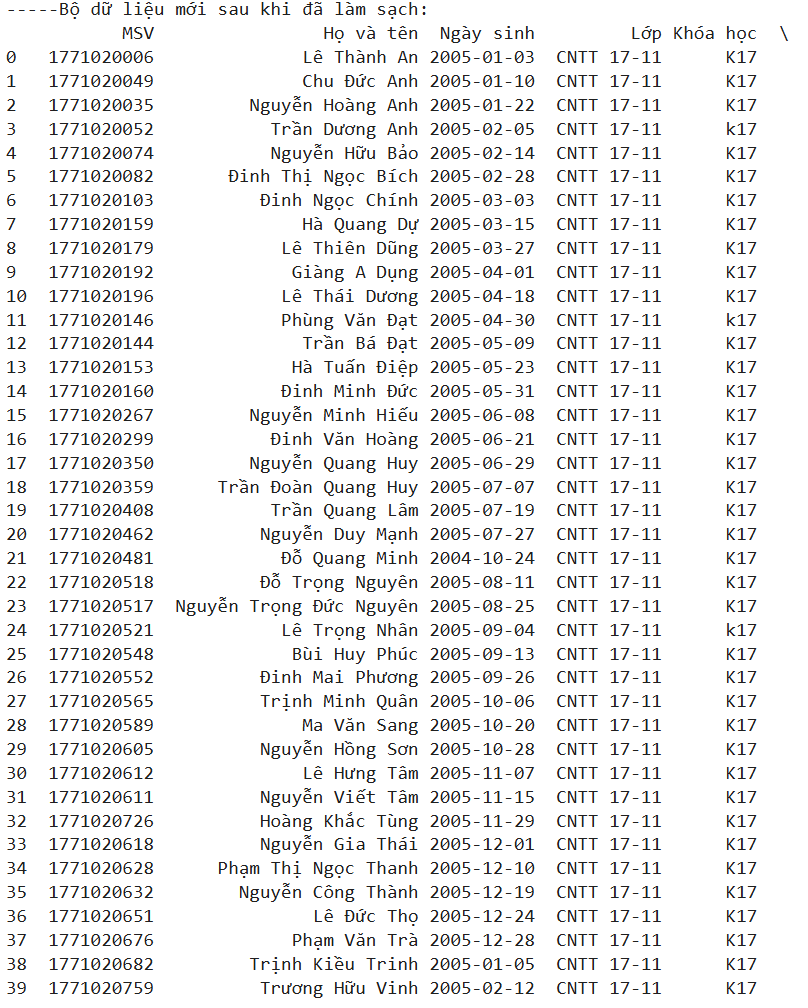
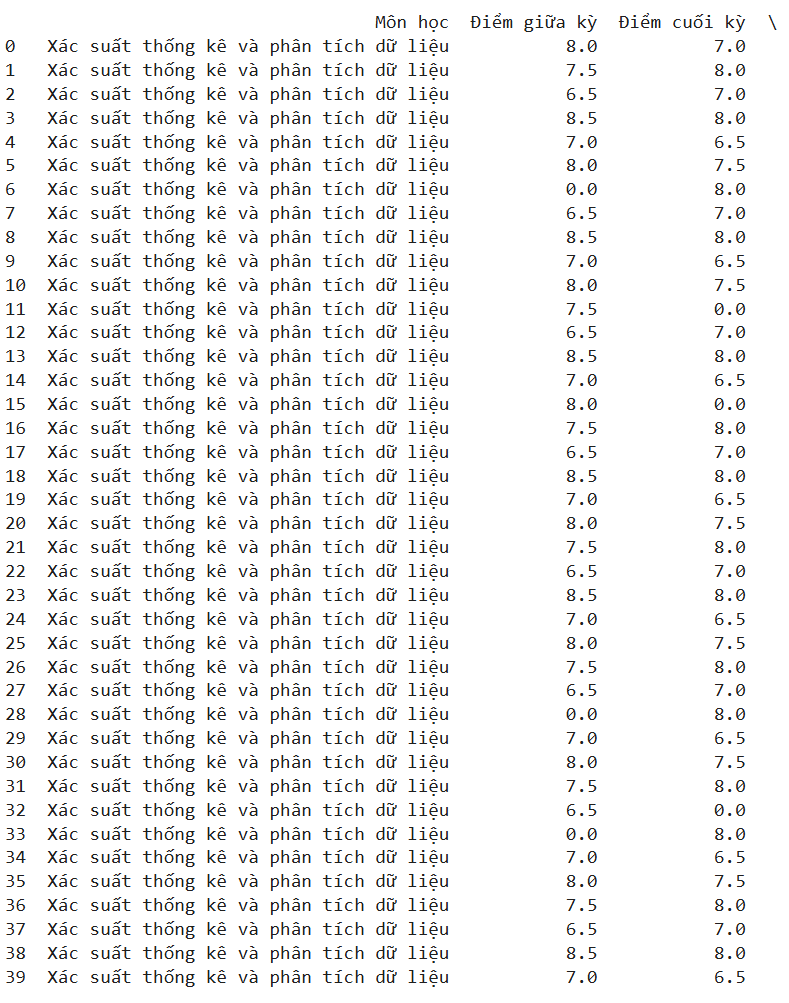
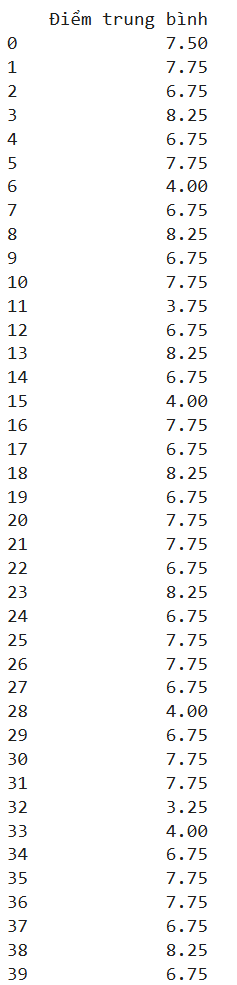


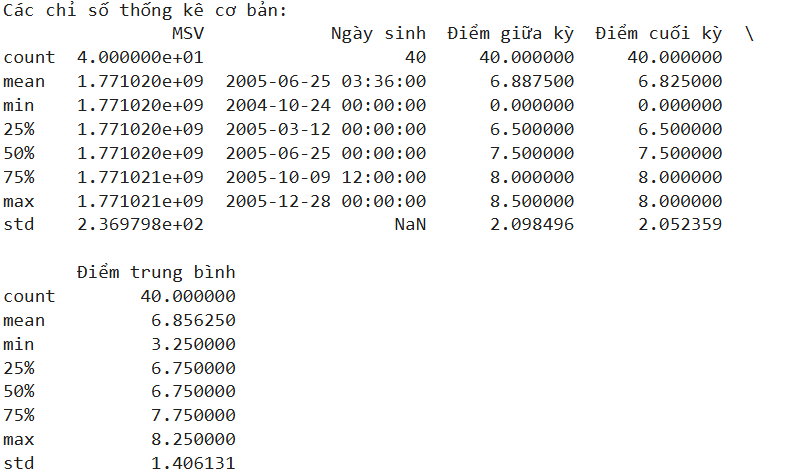
+ Phát hiện các giá trị lỗi, thiếu, sai định dạng và xử lý:



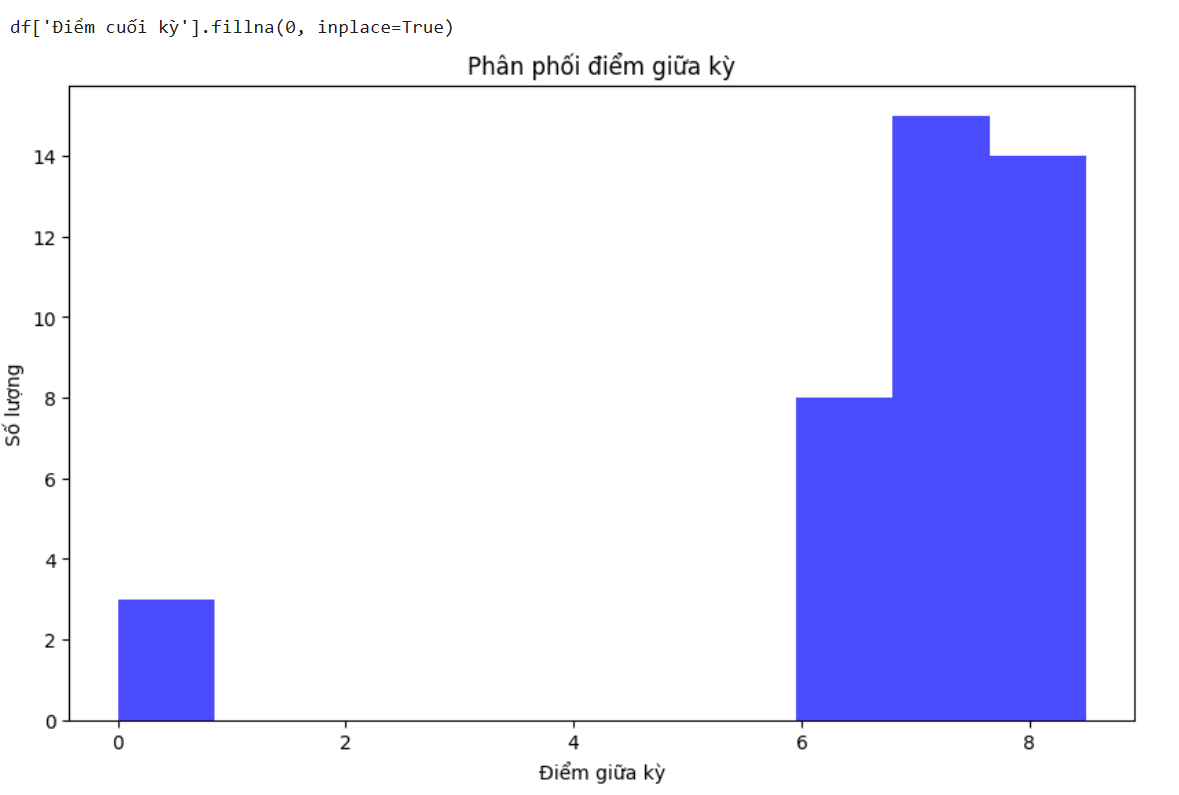
+ In ra bộ dữ liệu sau khi làm sạch

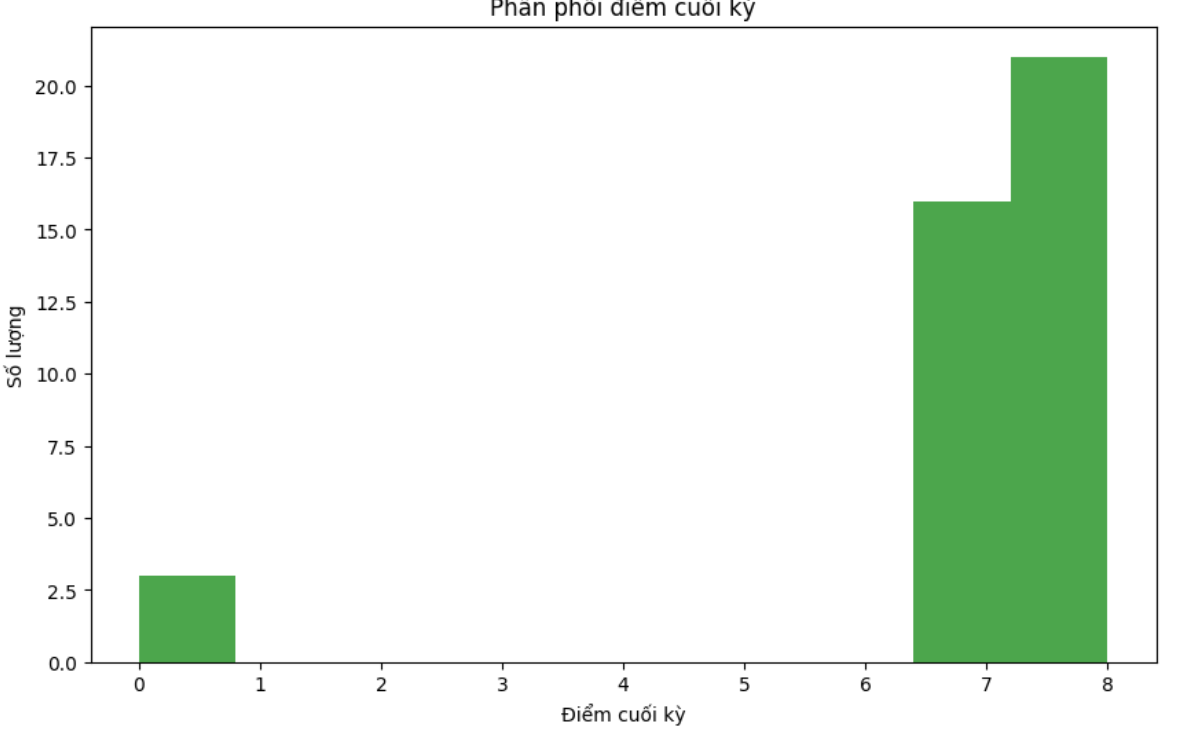
+ In ra các chỉ số thống kê cơ bản:



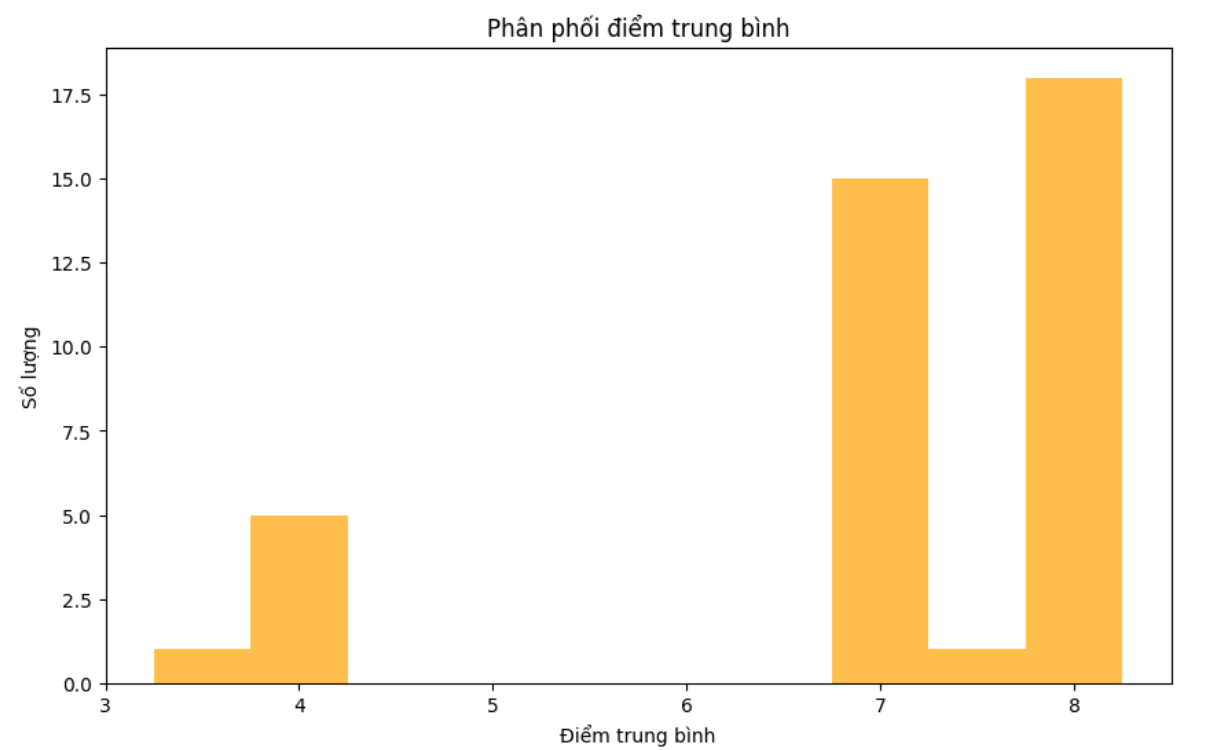
+ Biểu đồ phân phối điểm giữa kỳ:



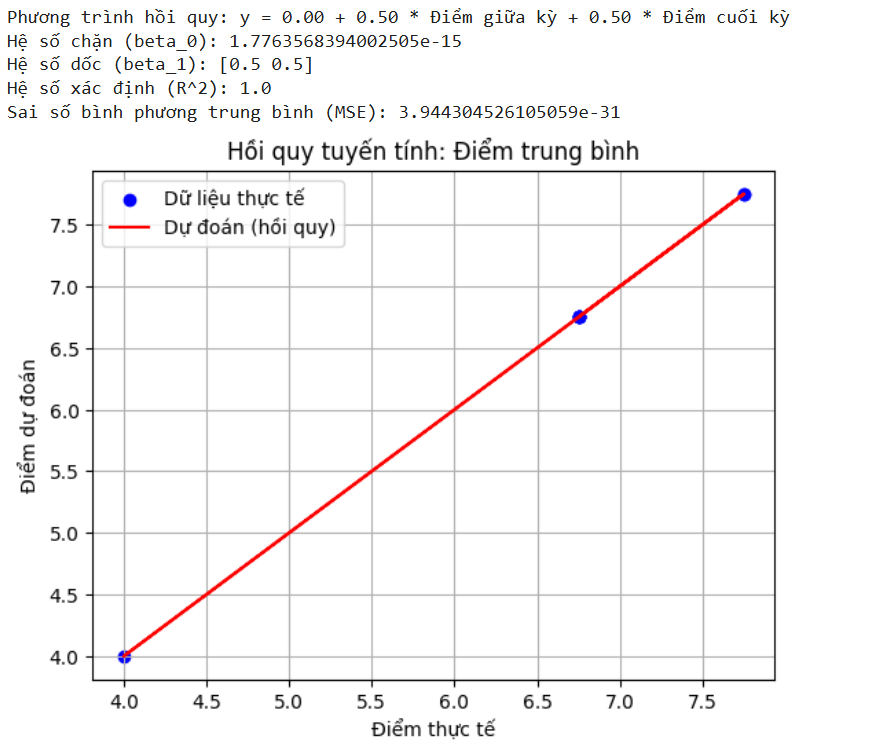
+ Biểu đồ phân phối điểm cuối kỳ:



+ Biểu đồ phân phối điểm trung bình:



- Đánh giá mô hình



- Mô hình có dự đoán tốt không?

+ Có. Với R² = 1.0 và MSE gần bằng 0, mô hình hồi quy này có độ chính xác cao và giải thích hoàn toàn sự biến động của biến phụ thuộc.

- Những điểm cần cải thiện là gì?

+ Mặc dù mô hình hiện tại có độ chính xác rất cao, điều này có thể do mô hình đang bị quá khớp (overfitting) với dữ liệu hiện tại. Điều này có thể xảy ra khi số lượng quan sát ít hoặc dữ liệu không đa dạng.

- Đề xuất các cách cải thiện

+ Thu thập thêm dữ liệu: Thu thập thêm các quan sát mới để kiểm tra tính ổn định của mô hình.

+ Xử lý tốt hơn các biến đầu vào: Kiểm tra và loại bỏ các biến không cần thiết, hoặc thêm các biến mới có thể cải thiện mô hình.

+ Thử nghiệm các thuật toán khác: Sử dụng các phương pháp hồi quy khác như hồi quy Ridge, Lasso, hoặc hồi quy cây quyết định (Decision Trees) để kiểm tra tính ổn định của mô hình.