TRƯỜNG ĐẠI HỌC VĂN LANG
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN CUỐI KỲ

MÔN HỌC NHẬP MÔN PHÂN TÍCH DỮ LIỆU VÀ HỌC SÂU

Chủ đề:

PHÂN TÍCH KẾT QUẢ THI THPTQG NĂM 2019 TẠI TỈNH HẢI PHÒNG

SVTH: Trần Minh Phúc – 227480201694
Châu Gia Kiệt – 2274802010449
LỚP: 242_71ITDS30203_0103
GVHD: Huỳnh Thái Học

TP. Hồ Chí Minh – 4/2025

MỤC LỤC

DANH MỤC HÌNH ẢNH

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành đến **Th.S Huỳnh Thái Học** – giảng viên môn Phân tích dữ liệu và Học sâu đã tận tình giảng dạy, hướng dẫn và truyền đạt những kiến thức quý báu trong suốt quá trình học tập.

Bên cạnh đó, cũng xin chân thành cảm ơn các thầy cô trong bộ môn đã xây dựng một chương trình học bổ ích, giúp chúng em có cơ hội tiếp cận và ứng dụng những phương pháp, thuật toán quan trọng trong lĩnh vực phân tích dữ liệu và học sâu. Những kiến thức này không chỉ hỗ trợ chúng em trong bài báo cáo mà còn là nền tảng vững chắc cho những nghiên cứu và ứng dụng sau này.

Dù còn nhiều hạn chế, nhưng chúng em đã cố gắng hết mình để hoàn thành bài báo cáo một cách tốt nhất. Chúng em mong nhận được sự góp ý quý báu từ thầy/cô để có thể cải thiện và nâng cao hơn nữa kiến thức cũng như kỹ năng của mình.

Xin chân thành cảm ơn!

PHẦN I: MỞ ĐẦU

1. Tổng Quan Đề Tài

1.1 Bối cảnh hiện tại

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ, dữ liệu ngày càng đóng vai trò quan trọng trong việc phân tích và ra quyết định ở nhiều lĩnh vực, trong đó có giáo dục. Kỳ thi THPT Quốc gia là sự kiện giáo dục có quy mô toàn quốc, phản ánh không chỉ năng lực học sinh mà còn cho thấy bức tranh tổng thể về chất lượng dạy và học của từng địa phương. Việc khai thác và phân tích dữ liệu điểm thi sẽ giúp đánh giá thực trạng giáo dục và cung cấp cơ sở khoa học cho công tác quản lý, điều chỉnh chính sách giáo dục phù hợp.

1.2 Lý do chọn đề tài

Xuất phát từ ý nghĩa đó, nhóm quyết định chọn đề tài: ***"Phân tích dữ liệu điểm thi THPTQG năm 2019 của tỉnh Hải Phòng".*** Hải Phòng là một thành phố cảng trọng yếu ở khu vực phía Bắc, với nền kinh tế – xã hội đang phát triển năng động, tuy nhiên chất lượng giáo dục vẫn còn không ít thách thức. Thông qua việc phân tích dữ liệu điểm thi, nhóm mong muốn nhận diện rõ điểm mạnh, điểm yếu trong kết quả học tập của học sinh địa phương, từ đó đưa ra một số nhận định và đề xuất có giá trị thực tiễn cho ngành giáo dục của thành phố.

Bên cạnh ý nghĩa nghiên cứu, đề tài này còn là cơ hội để nhóm áp dụng các kiến thức đã học về phân tích dữ liệu và machine learning vào xử lý một bài toán thực tế. Việc thu thập, làm sạch, trực quan hóa dữ liệu và phân tích chuyên sâu sẽ giúp nhóm rèn luyện tư duy dữ liệu và nâng cao kỹ năng thực hành, tạo tiền đề tốt cho các nghiên cứu khoa học hoặc dự án phân tích dữ liệu sau này.

1.3 Mục tiêu nghiên cứu

- Phân tích bức tranh chung về điểm thi: Xác định phân bố điểm, điểm trung bình, tỷ lệ đậu – rớt, tỷ lệ các mức điểm theo môn.

- Đánh giá chất lượng giáo dục: Xác định những môn thi có điểm trung bình cao hoặc thấp, từ đó phát hiện điểm mạnh – điểm yếu của học sinh.

- Nghiên cứu mối tương quan giữa các môn thi: Tìm hiểu mức độ liên hệ giữa điểm Toán, Văn, Ngoại ngữ, Khoa học Tự nhiên, Khoa học Xã hội và các môn thành phần khác.

- Đề xuất hướng cải thiện: Dựa trên kết quả phân tích, đề tài gợi ý những đề xuất nhằm nâng cao chất lượng dạy – học, góp phần cải thiện thành tích thi THPTQG trong tương lai.

1.4 Phạm vi và đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài bao gồm toàn bộ thí sinh dự thi Kỳ thi Trung học Phổ thông Quốc gia năm 2019 tại Hải Phòng. Dữ liệu điểm thi được khai thác để làm rõ tình hình học tập, điểm mạnh – điểm yếu của học sinh, đồng thời phục vụ cho các phân tích và đề xuất thiết thực nhằm nâng cao chất lượng giáo dục tại địa phương.

1.5 Phương pháp thực hiện

Để đạt được các mục tiêu nêu trên, đề tài dự kiến áp dụng các phương pháp sau:

1. **Thu thập dữ liệu**
    - Tập hợp dữ liệu điểm thi THPTQG năm 2019 của các thí sinh tại Hải Phòng từ nguồn chính thức
    - Kiểm tra tính đầy đủ và độ tin cậy của dữ liệu.
2. **Tiền xử lý dữ liệu**
    - Chuẩn hóa dữ liệu (định dạng, kiểu dữ liệu).
    - Xử lý các giá trị thiếu (missing), giá trị ngoại lai (outliers) để đảm bảo dữ liệu sạch và nhất quán.
3. **Phân tích và trực quan hóa**
    - Áp dụng các kỹ thuật **thống kê mô tả** (mean, median, mode, tần suất, phân phối điểm).
    - Sử dụng **biểu đồ** (histogram, boxplot, heatmap…) để trực quan hóa kết quả.
    - Tìm hiểu **mối tương quan** giữa các môn thi.
4. **Xây dựng và đánh giá mô hình** (nếu có)
    - Ứng dụng **machine learning** (các mô hình hồi quy, phân cụm…) để phân tích sâu hơn, chẳng hạn dự đoán điểm hoặc nhận diện nhóm học sinh có đặc điểm tương đồng.
    - Đánh giá chất lượng mô hình thông qua các chỉ số như R^2, RMSE, MAE, hoặc một số thước đo khác.
5. **Tổng hợp kết quả và đề xuất**
    - Tóm tắt các phát hiện chính, chỉ rõ điểm mạnh – yếu.
    - Đề xuất giải pháp nhằm cải thiện chất lượng giáo dục dựa trên số liệu phân tích.

2. Tổng Quan Về Dữ Liệu

2.1 Nguồn gốc

- Dữ liệu thô được crawl về từ từ trang web tra cứu điểm thi của TTXVN (TTXVN, 2025).

- Code crawl data: https://github.com/dusgkiet/THPTQG_source_analysis/blob/main/thptqg_data_crawl.py

2.2 Định dạng lưu trữ

- Bộ dữ liệu của đề tài được chuẩn hóa và lưu trữ dưới định dạng **CSV.** Định dạng này giúp thuận tiện trong việc nhập – xuất dữ liệu, cũng như dễ dàng thao tác bằng các công cụ phân tích hoặc lập trình.

PHẦN II: NỘI DUNG

1. Đọc Dữ Liệu

- Sau khi đã thu thập dữ liệu, ta tiến hành đọc file để thực hiện các bước xử lý tiếp theo.

```
# Hiển thị biểu đồ rõ nét hơn
plt.rcParams['figure.dpi'] = 100
df = pd.read_csv('diem_thptqg_2019.csv')
df
```

Hình 1. Code đọc dữ liệu từ file

- Kết quả chạy đoạn code sẽ hiển thị nội dung bảng dữ liệu ban đầu.

	So Bao Danh	Toan	Ngu Van	Ngoai Ngu	Vat Ly	Hoa Hoc	Sinh Hoc	KHTN	Lich Su	Dia Ly	GDCD	KHXH
0	3000001	4.8	6.50	3.8	NaN	NaN	NaN	NaN	4.25	6.50	8.75	6.50
1	3000002	3.0	2.25	NaN	NaN	NaN	NaN	NaN	2.75	3.00	NaN	NaN
2	3000003	3.2	2.50	NaN	NaN	NaN	NaN	NaN	2.25	3.75	NaN	NaN
3	3000004	5.8	3.25	NaN	NaN	NaN	NaN	NaN	2.50	5.50	NaN	NaN
4	3000005	6.8	6.00	8.6	6.0	3.5	4.25	4.58	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...
18480	3018571	7.2	4.75	3.0	NaN	NaN	NaN	NaN	5.00	6.00	7.50	6.17
18481	3018572	4.4	7.00	3.6	NaN	NaN	NaN	NaN	3.50	6.25	6.75	5.50
18482	3018573	5.8	5.00	2.8	NaN	NaN	NaN	NaN	2.75	4.50	5.50	4.25
18483	3018574	5.8	4.00	2.8	NaN	NaN	NaN	NaN	5.50	6.50	8.25	6.75
18484	3018575	7.2	5.25	3.0	5.0	4.0	3.00	4.00	NaN	NaN	NaN	NaN

18485 rows × 12 columns

Hình 2. Dữ liệu sau khi được đọc vào chương trình

- Tiếp theo, ta kiểm tra thông tin tổng quan của tập dữ liệu, bao gồm các thuộc tính, kiểu dữ liệu và số lượng giá trị bị thiếu (missing values).

```
print(df.info())
```

Hình 3. Code kiểm tra cấu trúc và thông tin dữ liệu

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18485 entries, 0 to 18484
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   So Bao Danh  18485 non-null  int64
 1   Toan         18456 non-null  float64
 2   Ngu Van      18256 non-null  float64
 3   Ngoai Ngu    16819 non-null  float64
 4   Vat Ly       6700 non-null   float64
 5   Hoa Hoc      6731 non-null   float64
 6   Sinh Hoc     6645 non-null   float64
 7   KHTN         6571 non-null   float64
 8   Lich Su      11682 non-null  float64
 9   Dia Ly       11601 non-null  float64
 10  GDCD         10229 non-null  float64
 11  KHXH         10229 non-null  float64
dtypes: float64(11), int64(1)
memory usage: 1.7 MB
None
```

Hình 4. Kết quả sau khi kiểm tra dữ liệu

- **Nhận xét:** Các kiểu dữ liệu đã được đọc đúng, không cần thực hiện chuyển đổi kiểu. Có nhiều giá trị bị thiếu do thí sinh đăng ký các tổ hợp môn khác nhau (KHTN hoặc KHXH), dẫn đến không thi một số môn nên không có điểm tương ứng.

2. Tiền Xử Lý Dữ Liệu

- Sau khi thu thập (crawl) dữ liệu thô, bước đầu tiên và cũng là quan trọng nhất chính là làm sạch dữ liệu – hay còn gọi là tiền xử lý dữ liệu.

- Trong giai đoạn này, ta cần rà soát và xử lý các giá trị bị thiếu hoặc bỏ trống để đảm bảo tính đầy đủ và nhất quán cho tập dữ liệu. Việc loại bỏ hoặc thay thế hợp lý các giá trị thiếu giúp nâng cao độ tin cậy và chất lượng phân tích trong các bước xử lý tiếp theo.

2.1 Bổ sung cột "Số Môn Thi"

- Thực hiện bổ sung cột "Số Môn Thi" nhằm xác định rõ mỗi thí sinh tham gia bao nhiêu môn thi. Việc thêm trường thông tin này giúp việc phân tích chi

tiết hơn, đặc biệt khi so sánh tổng điểm trên số môn, hoặc kiểm tra xu hướng chọn môn thi của học sinh.

- Đầu tiên, ta liệt kê danh sách các môn thi có thể xuất hiện trong kỳ thi THPT quốc gia.

```
# Các môn thi
list_of_subjects = ['Toan', 'Ngu Van', 'Ngoai Ngu', 'Vat Ly', 'Hoa Hoc', 'Sinh Hoc', 'Lich Su', 'Dia Ly', 'GDCD']
```

Hình 5. Code liệt kê danh sách môn thi

- Sau đó, ta tiến hành thêm cột "Số Môn Thi" vào DataFrame. Cột này được tính bằng cách đếm số lượng môn thi mà mỗi thí sinh đã tham gia (dựa trên số cột không bị thiếu trong danh sách các môn).

```
df['So Mon Thi'] = df[list_of_subjects].notna().sum(axis=1).tolist()
df
```

Hình 6. Code để thêm cột "Số Môn Thi"

- Sau khi thêm, cột "Số Môn Thi" sẽ nằm ở cuối bảng dữ liệu, thể hiện tổng số môn thi mà từng thí sinh đã dự thi.

	So Bao Danh	Toan	Ngu Van	Ngoai Ngu	Vat Ly	Hoa Hoc	Sinh Hoc	KHTN	Lich Su	Dia Ly	GDCD	KHXH	So Mon Thi
0	3000001	4.8	6.50	3.8	NaN	NaN	NaN	NaN	4.25	6.50	8.75	6.50	6
1	3000002	3.0	2.25	NaN	NaN	NaN	NaN	NaN	2.75	3.00	NaN	NaN	4
2	3000003	3.2	2.50	NaN	NaN	NaN	NaN	NaN	2.25	3.75	NaN	NaN	4
3	3000004	5.8	3.25	NaN	NaN	NaN	NaN	NaN	2.50	5.50	NaN	NaN	4
4	3000005	6.8	6.00	8.6	6.0	3.5	4.25	4.58	NaN	NaN	NaN	NaN	6
...	...	...	...	...	...	...	...	...	...	...	...	...	...
18480	3018571	7.2	4.75	3.0	NaN	NaN	NaN	NaN	5.00	6.00	7.50	6.17	6
18481	3018572	4.4	7.00	3.6	NaN	NaN	NaN	NaN	3.50	6.25	6.75	5.50	6
18482	3018573	5.8	5.00	2.8	NaN	NaN	NaN	NaN	2.75	4.50	5.50	4.25	6
18483	3018574	5.8	4.00	2.8	NaN	NaN	NaN	NaN	5.50	6.50	8.25	6.75	6
18484	3018575	7.2	5.25	3.0	5.0	4.0	3.00	4.00	NaN	NaN	NaN	NaN	6

18485 rows × 13 columns

Hình 7. Kết quả sau khi thêm cột "Số Môn Thi"

- **Nhận xét:** Có sự khác biệt rõ rệt về số môn thi giữa các thí sinh. Nguyên nhân có thể do thí sinh chọn tổ hợp thi khác nhau (KHTN hoặc KHXH) hoặc vắng thi/bỏ thi một số môn, dẫn đến số môn thi không đồng đều.

2.2 Làm Sạch Dữ Liệu

- Trong bộ dữ liệu điểm thi, các giá trị bị thiếu thường xuất hiện do thí sinh **không đăng ký thi môn đó** (thường gặp ở các môn trong tổ hợp KHTN hoặc KHXH).

- Để thuận tiện cho xử lý và trực quan hóa, ta tiến hành **thay thế các giá trị NaN bằng -1**, nhằm biểu diễn rằng thí sinh **không thi môn đó**.

```
df_clean = df.fillna(-1)

df_clean
```

Hình 8. Code để thay thế giá trị NaN bằng -1

	So Bao Danh	Toan	Ngu Van	Ngoai Ngu	Vat Ly	Hoa Hoc	Sinh Hoc	KHTN	Lich Su	Dia Ly	GDCD	KHXH	So Mon Thi
0	3000001	4.8	6.50	3.8	-1.0	-1.0	-1.00	-1.00	4.25	6.50	8.75	6.50	6
1	3000002	3.0	2.25	-1.0	-1.0	-1.0	-1.00	-1.00	2.75	3.00	-1.00	-1.00	4
2	3000003	3.2	2.50	-1.0	-1.0	-1.0	-1.00	-1.00	2.25	3.75	-1.00	-1.00	4
3	3000004	5.8	3.25	-1.0	-1.0	-1.0	-1.00	-1.00	2.50	5.50	-1.00	-1.00	4
4	3000005	6.8	6.00	8.6	6.0	3.5	4.25	4.58	-1.00	-1.00	-1.00	-1.00	6
...	...	...	...	...	...	...	...	...	...	...	...	...	...
18480	3018571	7.2	4.75	3.0	-1.0	-1.0	-1.00	-1.00	5.00	6.00	7.50	6.17	6
18481	3018572	4.4	7.00	3.6	-1.0	-1.0	-1.00	-1.00	3.50	6.25	6.75	5.50	6
18482	3018573	5.8	5.00	2.8	-1.0	-1.0	-1.00	-1.00	2.75	4.50	5.50	4.25	6
18483	3018574	5.8	4.00	2.8	-1.0	-1.0	-1.00	-1.00	5.50	6.50	8.25	6.75	6
18484	3018575	7.2	5.25	3.0	5.0	4.0	3.00	4.00	-1.00	-1.00	-1.00	-1.00	6

18485 rows × 13 columns

Hình 9. Dữ liệu sau khi đã được làm sạch

- Kết quả: Sau khi thực hiện thay thế, các ô trống trong bảng điểm được điền giá trị -1. Điều này giúp dễ dàng nhận biết thí sinh không thi môn nào, đồng thời tránh lỗi khi tính toán, trực quan hóa hay xuất dữ liệu.

2.3 Kiểm tra dữ liệu ngoại lai (Outliers)

- Sau khi xử lý dữ liệu và thống kê cơ bản, bước tiếp theo trong phân tích dữ liệu là phát hiện các giá trị ngoại lai (outliers). Việc xác định outlier giúp làm sạch dữ liệu và hiểu rõ hơn về sự phân bố điểm số bất thường của thí sinh.

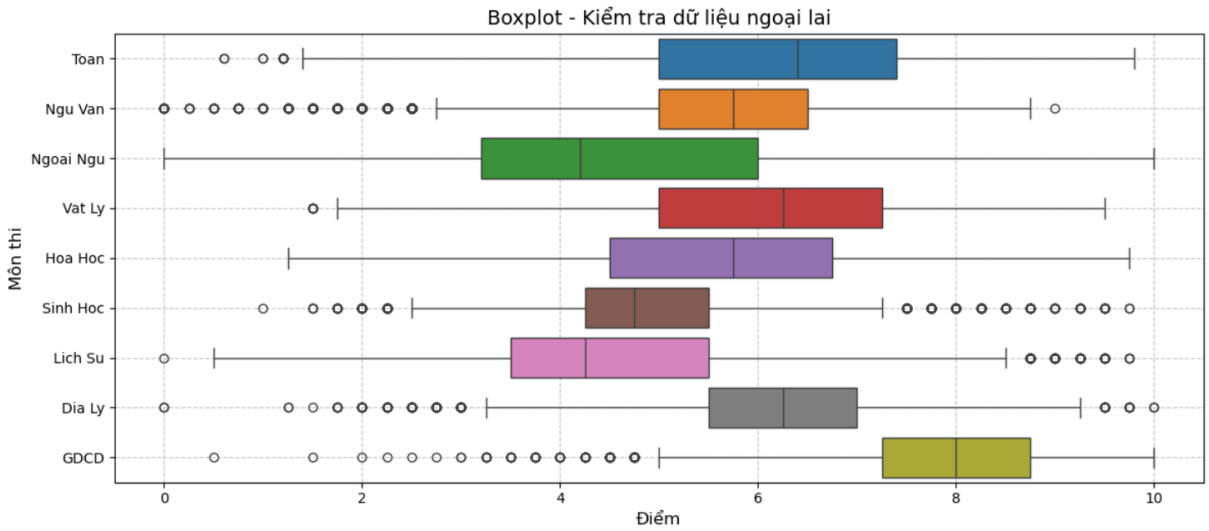
```python
# Kiểm tra dữ liệu ngoại lai (Outliers) bằng phương pháp IQR
def detect_outliers_iqr(data, column):
    Q1 = data[column].quantile(0.25)  # Quartile 1
    Q3 = data[column].quantile(0.75)  # Quartile 3
    IQR = Q3 - Q1  # Interquartile Range
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = data[(data[column] < lower_bound) |
                    (data[column] > upper_bound)]
    return outliers


# Áp dụng kiểm tra ngoại lai cho từng cột trong danh sách các môn học
outliers_dict = {}
# Hiển thị từng môn học tuần tự
for subject, outliers in outliers_dict.items():
    print(f"--- Outliers in {subject} ---")
    if not outliers.empty:
        print(outliers)
    else:
        print("No outliers found.")
    print("\n")  # Thêm dòng trống để dễ đọc hơn
    input("Nhấn Enter để tiếp tục...")
```

Hình 10. Code phát hiện outlier bằng IQR

- Dựng hình

```python
# Vẽ boxplot để trực quan hóa dữ liệu ngoại lai
# Điều chỉnh chiều cao dựa trên số lượng môn học
plt.figure(figsize=(12, len(list_of_subjects) * 0.6))
sns.boxplot(data=df[list_of_subjects], orient="h")
plt.title("Boxplot - Kiểm tra dữ liệu ngoại lai", fontsize=14)
plt.xlabel("Điểm", fontsize=12)
plt.ylabel("Môn thi", fontsize=12)
plt.yticks(fontsize=10)  # Điều chỉnh kích thước nhãn môn học
plt.grid(True, linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

Hình 11. Code vẽ biểu đồ Boxplot



Hình 12. Biểu đồ Boxplot – Kiểm tra dữ liệu ngoại lai

- **Nhận xét:**
  - Phương pháp IQR giúp phát hiện các điểm số cao/thấp bất thường ở mỗi môn.
  - Boxplot là công cụ hữu ích để hình dung phân bố điểm và xác định outlier.
  - Các outlier này có thể được xử lý tùy theo mục đích phân tích:
    - Giữ lại nếu có ý nghĩa đặc biệt.
    - Loại bỏ nếu muốn làm sạch dữ liệu để huấn luyện mô hình.

3. Thao Tác Dữ Liệu

- Sau khi hoàn tất công đoạn làm sạch, dữ liệu đã sẵn sàng để phân tích. Lúc này, ta có thể áp dụng các kỹ thuật thống kê, trực quan hóa hoặc mô hình học máy nhằm khám phá, rút trích thông tin và tạo cơ sở cho việc đưa ra các quyết định quan trọng dựa trên dữ liệu.

3.1 Phần trăm thi mỗi môn của thí sinh

- Sau khi hoàn tất bước xử lý dữ liệu, ta tiến hành thống kê tỷ lệ thí sinh thi ở từng môn học nhằm đánh giá mức độ phổ biến và sự lựa chọn tổ hợp thi của học sinh trong kỳ thi THPT Quốc gia.

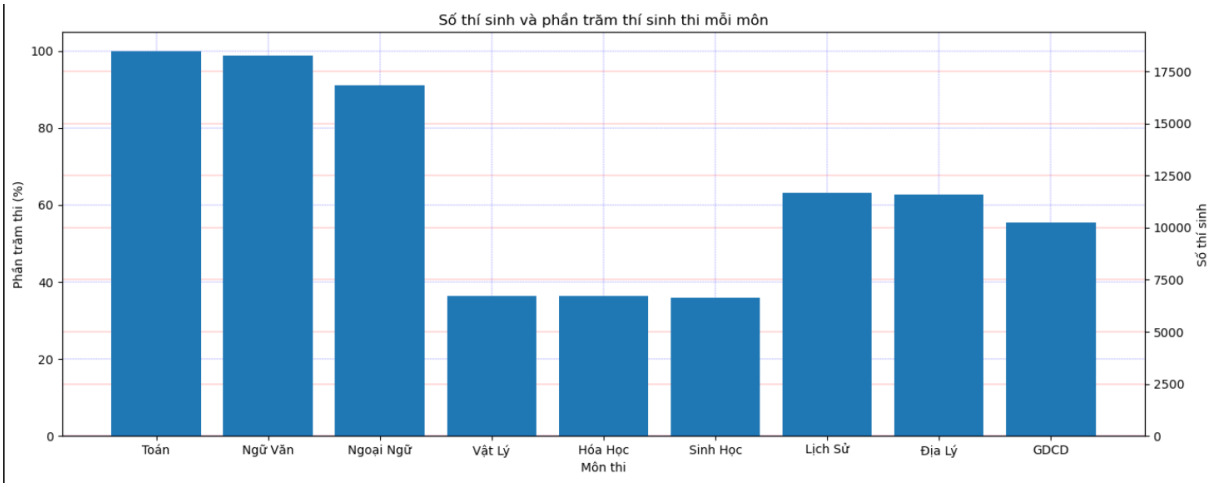
```
thi_toan = df_clean[df_clean['Toan'] >= 0].count()['Toan']
thi_van = df_clean[df_clean['Ngu Van'] >= 0].count()['Ngu Van']
thi_nn = df_clean[df_clean['Ngoai Ngu'] >= 0].count()['Ngoai Ngu']

thi_ly = df_clean[df_clean['Vat Ly'] >= 0].count()['Vat Ly']
thi_hoa = df_clean[df_clean['Hoa Hoc'] >= 0].count()['Hoa Hoc']
thi_sinh = df_clean[df_clean['Sinh Hoc'] >= 0].count()['Sinh Hoc']

thi_su = df_clean[df_clean['Lich Su'] >= 0].count()['Lich Su']
thi_dia = df_clean[df_clean['Dia Ly'] >= 0].count()['Dia Ly']
thi_gdcd = df_clean[df_clean['GDCD'] >= 0].count()['GDCD']

tong_thi_sinh = df_clean.count()['So Bao Danh']

subject = ['Toán', 'Ngữ Văn', 'Ngoại Ngữ', 'Vật Lý', 'Hóa Học', 'Sinh Học', 'Lịch Sử', 'Địa Lý',  'GDCD']

# Số thí sinh thi mỗi môn
no_of_stu = [thi_toan, thi_van, thi_nn, thi_ly, thi_hoa, thi_sinh, thi_su, thi_dia, thi_gdcd]

# Phần trăm thí sinh thi mỗi môn
ptram_thi = no_of_stu / tong_thi_sinh
```

Hình 13. Code tính phần trăm học sinh thi mỗi môn

- Sau khi đã có dữ liệu thống kê, ta tiến hành trực quan hóa bằng biểu đồ cột giúp dễ dàng nhận diện mức độ phổ biến của từng môn.

```
fig = plt.figure(figsize=(16, 6))
ax1 = fig.add_subplot(111)

ax1.bar(subject, ptram_thi * 100)
ax1.set_xlabel('Môn thi')
ax1.set_ylabel('Phần trăm thi (%)')
ax1.grid(color='b', ls='-.', lw = 0.2)
ax1.set_axisbelow(True)

# Biểu diễn chung 1 đồ thị
ax2 = ax1.twinx()

ax2.bar(subject, no_of_stu)
ax2.set_ylabel('Số thí sinh')
ax2.grid(color='r', ls='-', lw = 0.2)
ax2.set_axisbelow(True)

plt.title('Số thí sinh và phần trăm thí sinh thi mỗi môn')
plt.show()
```

Hình 14. Code dựng hình



Hình 15. Biểu đồ Số thí sinh và phần trăm thí sinh thi mỗi môn

- **Nhận xét:**
  - Toán, Ngữ Văn, Ngoại Ngữ có tỷ lệ thí sinh thi gần như 100% (do là môn bắt buộc).
  - Các môn thuộc tổ hợp KHTN (Lý, Hóa, Sinh) có tỷ lệ thí sinh thấp hơn (~35%), phản ánh ít thí sinh chọn tổ hợp này.
  - Các môn KHXH (Sử, Địa, GDCD) có tỷ lệ thi cao hơn nhóm KHTN, cho thấy xu hướng học sinh thiên về khối xã hội.

3.2 Số học sinh thi x môn

- Bước tiếp theo là tìm hiểu mỗi thí sinh đã thi bao nhiêu môn, từ đó xác định xu hướng thi tổ hợp, thi tự do hoặc có trường hợp thi ít hơn do vắng/bỏ thi.

```
# Ensure the 'So Mon Thi' column exists
if 'So Mon Thi' not in df.columns:
    # Create the 'So Mon Thi' column if it doesn't exist
    list_of_subjects = ['Toan', 'Ngu Van', 'Ngoai Ngu', 'Vat Ly', 'Hoa Hoc', 'Sinh Hoc', 'Lich Su', 'Dia Ly', 'GDCD']
    df['So Mon Thi'] = df[list_of_subjects].notna().sum(axis=1).tolist()

cols = ['So Bao Danh', 'So Mon Thi']
thptqg_so_mon = df[cols].groupby('So Mon Thi').count().reset_index()
thptqg_so_mon = thptqg_so_mon.rename(columns={'So Bao Danh': 'So Thi Sinh'})
thptqg_so_mon
```

Hình 16. Code xem số môn thi và số lượng thí sinh tương ứng

	So Mon Thi	So Thi Sinh
0	1	8
1	2	2
2	3	337
3	4	1347
4	5	127
5	6	16631
6	7	5
7	9	28

Hình 17. Kết quả ban đầu

- **Nhận xét:** Phần lớn học sinh thi từ **3–6 môn**. Các nhóm ngoài khoảng này (như 1, 2, 7 môn…) rất ít → nên gom chung thành 1 nhóm là 'Other' để dễ trực quan hóa.

```
other = {'So Mon Thi': 'other', 'So Thi Sinh': thptqg_so_mon[thptqg_so_mon['So Thi Sinh'] < 100].sum()['So Thi Sinh']}
other
```

Hình 18. Code nhóm các trường hợp ít vào nhóm 'Other'

```
{'So Mon Thi': 'other', 'So Thi Sinh': 43}
```

Hình 19. Kết quả sau khi lọc

- Thêm row 'other' và xóa các row có số thí sinh < 100

```
# Xóa các hàng có số thí sinh thấp hơn 100
thptqg_so_mon = thptqg_so_mon.drop(thptqg_so_mon[thptqg_so_mon['So Thi Sinh'] < 100].index)

# Thêm hàng other
other_df = pd.DataFrame([other])  # Convert the dictionary to a DataFrame
thptqg_so_mon = pd.concat([thptqg_so_mon, other_df], ignore_index=True)

thptqg_so_mon
```

Hình 20. Code thêm nhóm "Other" vào dữ liệu thống kê

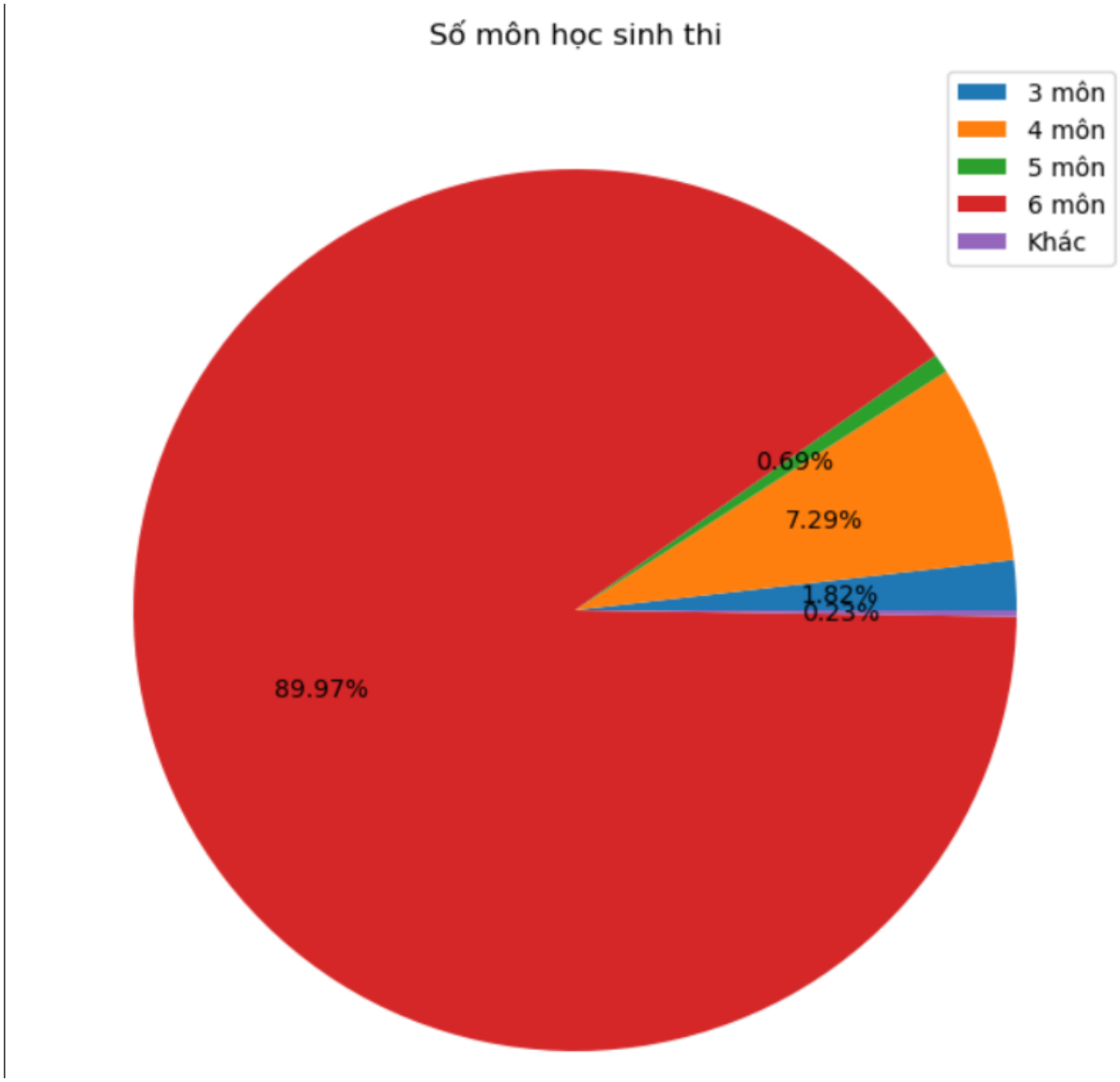
	So Mon Thi	So Thi Sinh
0	3	337
1	4	1347
2	5	127
3	6	16631
4	other	43

Hình 21. Kết quả khi thêm dòng "Other"

- Sau đó, ta đi thực hiện dựng hình

```
fig = plt.figure(figsize=(8, 12))
ax = fig.add_subplot(111)

ax.pie(thptqg_so_mon['So Thi Sinh'], autopct='%.2f%%')

ax.legend(labels=('3 môn', '4 môn', '5 môn', '6 môn', 'Khác'), loc='upper right')
plt.title('Số môn học sinh thi')

plt.show()
```

Hình 22. Code dựng biểu đồ số môn thí sinh thi



Hình 23. Biểu đồ Số môn học sinh thi

- **Nhận xét:** Đa số thí sinh thi đủ 6 môn (89,97%), chỉ ít số thí sinh thi không đủ (0,23%).

3.3 Điểm trung bình theo số lượng môn thi

- Mục tiêu của phần này là phân tích điểm trung bình của thí sinh theo số lượng môn thi. Điều này giúp đánh giá liệu việc thi nhiều môn có ảnh hưởng đến kết quả học tập không.

```
# Copy bảng để tránh thao tác trên dữ liệu gốc
thptqg_avg = df_clean.copy()

# Tạo cột 'điểm trung bình'
thptqg_avg['Diem Trung Binh'] = thptqg_avg[thptqg_avg[list_of_subjects] >= 0].mean(axis=1).round(2).tolist()

thptqg_avg
```

Hình 24. Code lọc điểm trung bình

	So Bao Danh	Toan	Ngu Van	Ngoai Ngu	Vat Ly	Hoa Hoc	Sinh Hoc	KHTN	Lich Su	Dia Ly	GDCD	KHXH	So Mon Thi	Diem Trung Binh
0	3000001	4.8	6.50	3.8	-1.0	-1.0	-1.00	-1.00	4.25	6.50	8.75	6.50	6	5.77
1	3000002	3.0	2.25	-1.0	-1.0	-1.0	-1.00	-1.00	2.75	3.00	-1.00	-1.00	4	2.75
2	3000003	3.2	2.50	-1.0	-1.0	-1.0	-1.00	-1.00	2.25	3.75	-1.00	-1.00	4	2.92
3	3000004	5.8	3.25	-1.0	-1.0	-1.0	-1.00	-1.00	2.50	5.50	-1.00	-1.00	4	4.26
4	3000005	6.8	6.00	8.6	6.0	3.5	4.25	4.58	-1.00	-1.00	-1.00	-1.00	6	5.86
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
18480	3018571	7.2	4.75	3.0	-1.0	-1.0	-1.00	-1.00	5.00	6.00	7.50	6.17	6	5.58
18481	3018572	4.4	7.00	3.6	-1.0	-1.0	-1.00	-1.00	3.50	6.25	6.75	5.50	6	5.25
18482	3018573	5.8	5.00	2.8	-1.0	-1.0	-1.00	-1.00	2.75	4.50	5.50	4.25	6	4.39
18483	3018574	5.8	4.00	2.8	-1.0	-1.0	-1.00	-1.00	5.50	6.50	8.25	6.75	6	5.48
18484	3018575	7.2	5.25	3.0	5.0	4.0	3.00	4.00	-1.00	-1.00	-1.00	-1.00	6	4.58

18485 rows × 14 columns

Hình 25. Kết quả thêm cột Điểm Trung Bình

```
# Cột cần lấy
cols = ['So Mon Thi', 'Diem Trung Binh']

# Lấy điểm trung bình theo số môn thí sinh thi
thptqg_avg = thptqg_avg[cols].groupby('So Mon Thi').mean().round(2).reset_index()

thptqg_avg
```

Hình 26. Code nhóm theo số môn thi

	So Mon Thi	Diem Trung Binh
0	1	1.91
1	2	4.78
2	3	5.79
3	4	3.96
4	5	5.69
5	6	5.80
6	7	4.37
7	9	5.37

Hình 27.Kết quả thống kê

- Do không có thí sinh nào thi 8 môn, thêm một hàng 8 môn có điểm trung bình là 0, biểu thị là không có thí sinh thi 8 môn.

```python
# Tạo df mới có dữ liệu 8 - 0
add_df = pd.DataFrame({'So Mon Thi': 8, 'Diem Trung Binh': 0}, index=[6.5])

# Nối thptqg_avg với add_df
thptqg_avg = pd.concat([thptqg_avg, add_df], ignore_index=False)

# Sort lại thptqg_avg theo index
thptqg_avg = thptqg_avg.sort_index().reset_index(drop=True)

thptqg_avg
```

Hình 28. Tạo dữ liệu cho biểu đồ

	So Mon Thi	Diem Trung Binh
0	1	1.91
1	2	4.78
2	3	5.79
3	4	3.96
4	5	5.69
5	6	5.80
6	7	4.37
7	8	0.00
8	9	5.37

Hình 29. Kết quả bảng điểm trung bình theo số môn

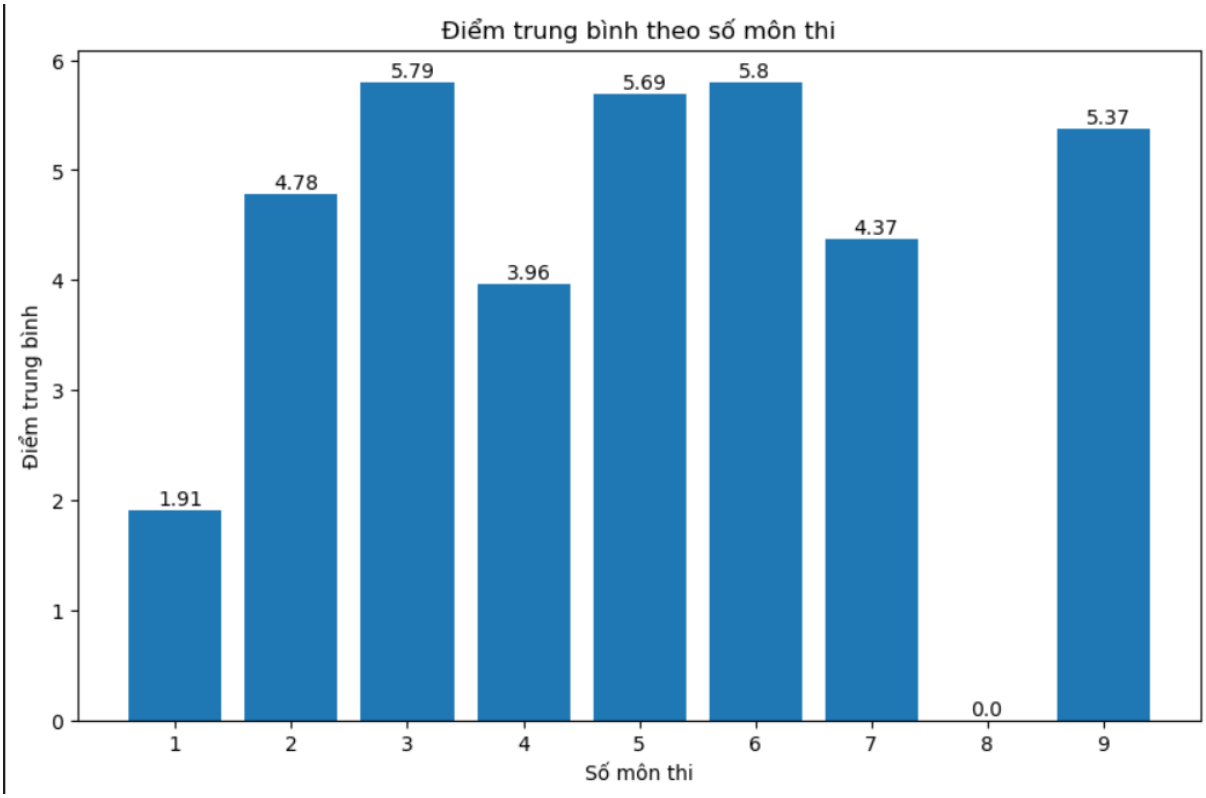
```python
# Thiết lập các tham số cần thiết
so_mon = thptqg_avg['So Mon Thi'].tolist()
diem_trung_binh = thptqg_avg['Diem Trung Binh'].tolist()
```

Hình 30. Thiết lập tham số vẽ biểu đồ

- Sau đó ta thực hiện dựng biểu đồ

```python
fig = plt.figure(figsize=(10, 6))
plt.bar(so_mon, diem_trung_binh)

plt.xticks([1, 2, 3, 4, 5, 6, 7, 8, 9])
plt.xlabel('Số môn thi')
plt.ylabel('Điểm trung bình')
plt.title('Điểm trung bình theo số môn thi')

# Thêm giá trị thực vào phía trên mỗi bar
for index, value in enumerate(diem_trung_binh):
    plt.text(index + 0.85, value + 0.05, str(value))

plt.show()
```

Hình 31. Code dựng biểu đồ



Hình 32. Biểu đồ Điểm trung bình theo số môn thi

3.4 Phổ điểm và Violin plot của từng môn

- Để hiểu rõ hơn về phân bố điểm số của từng môn thi, ta thực hiện hai loại biểu đồ:

- Phổ điểm (Histogram): Thể hiện tần suất điểm số.

- Violin Plot: Kết hợp giữa boxplot và kernel density, giúp hình dung rõ mật độ và sự phân tán điểm.

```
for c in list_of_subjects:
    # Kiểm tra xem có dữ liệu không (all NaN -> bỏ qua)
    if df[c].notna().sum() == 0:
        continue

    # 1. In thống kê mô tả cho môn c
    print(f"=== Thống kê mô tả môn {c} ===")
    print("Số lượng:", df[c].count())
    print("Điểm trung bình:", round(df[c].mean(),2))
    print("Điểm min:", df[c].min())
    print("Điểm max:", df[c].max())
    print("Độ lệch chuẩn:", round(df[c].std(),2))
    print("-"*40)

    # 2. Vẽ Histogram (phổ điểm)
    plt.figure(figsize=(5,4))
    sns.histplot(df[c], bins=20, kde=True)
    plt.title(f"Phổ điểm môn {c}")
    plt.xlabel("Điểm")
    plt.ylabel("Số thí sinh")
    plt.grid(True)
    plt.tight_layout()
    plt.show()

    # 3. Vẽ Violin plot
    plt.figure(figsize=(3,4))
    sns.violinplot(y=df[c])
    plt.title(f"Violin plot - Môn {c}")
    plt.ylabel("Điểm")
    plt.tight_layout()
    plt.show()

    # Thêm khoảng trống hoặc gạch ngang để tách giữa các môn
    print("\n" + "="*60 + "\n")
```

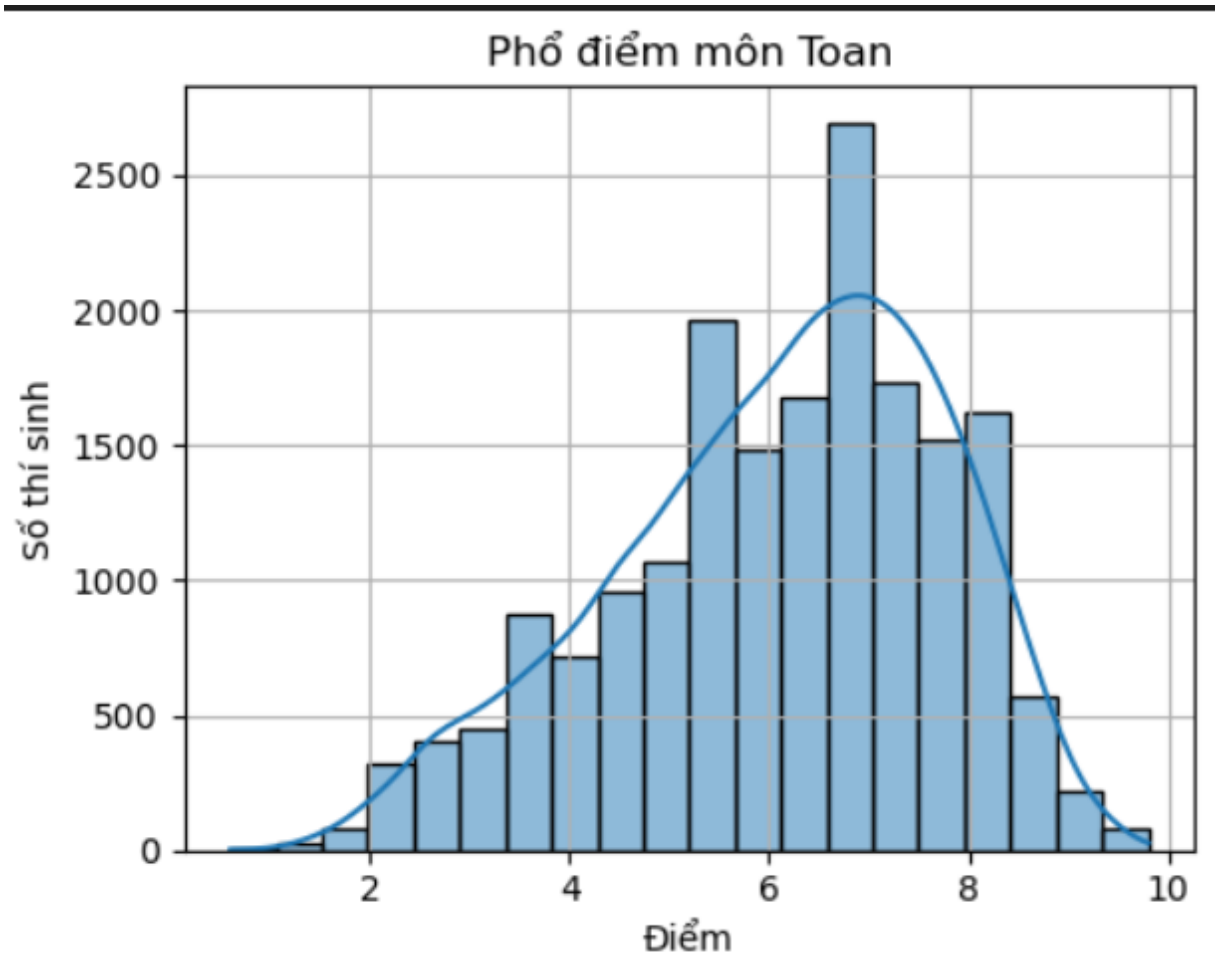
Hình 33. Code tạo phổ điểm và Violin plot

- Môn Toán:

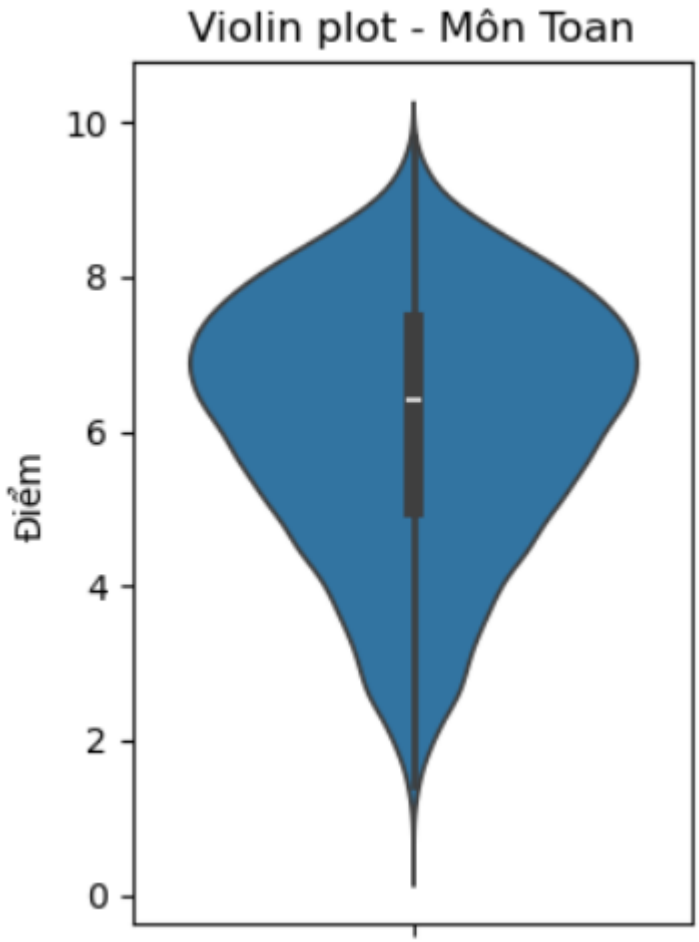
```
=== Thống kê mô tả môn Toan ===
Số lượng: 18456
Điểm trung bình: 6.1
Điểm min: 0.6
Điểm max: 9.8
Độ lệch chuẩn: 1.66
----------------------------------------
```

Hình 34. Thống kê mô tả môn Toán

- **Điểm trung bình:** 6.10 → mức trung bình khá.
- **Khoảng điểm:** 0.6 – 9.8
- **Độ lệch chuẩn:** 1.66 → phân tán rộng.
- **Nhận xét:** Toán là môn bắt buộc nên phổ điểm rộng. Một số học sinh điểm rất thấp, nhưng cũng có nhiều điểm cao → đa dạng năng lực.



Hình 35. Phổ điểm môn Toán



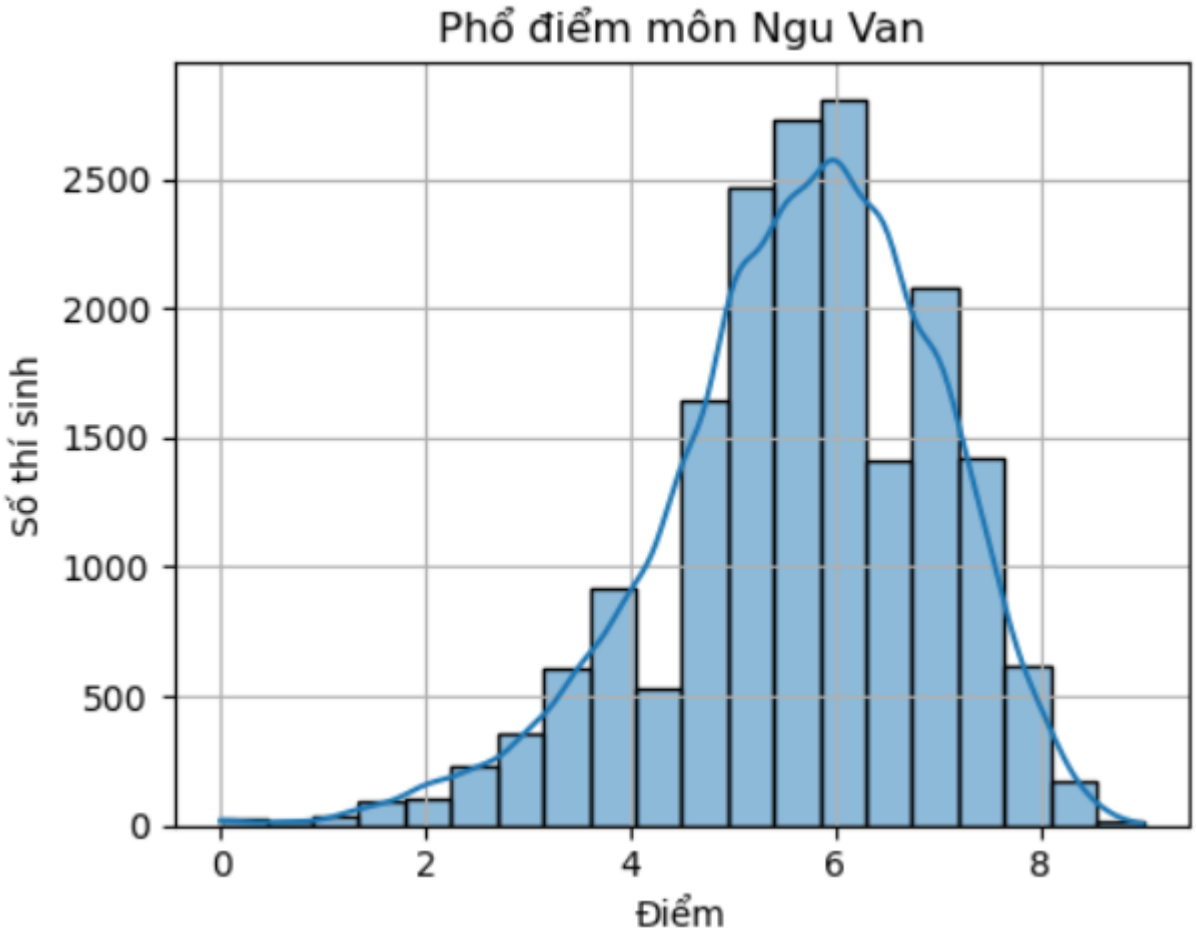
Hình 36. Violin môn Toán

- Môn Ngữ Văn:

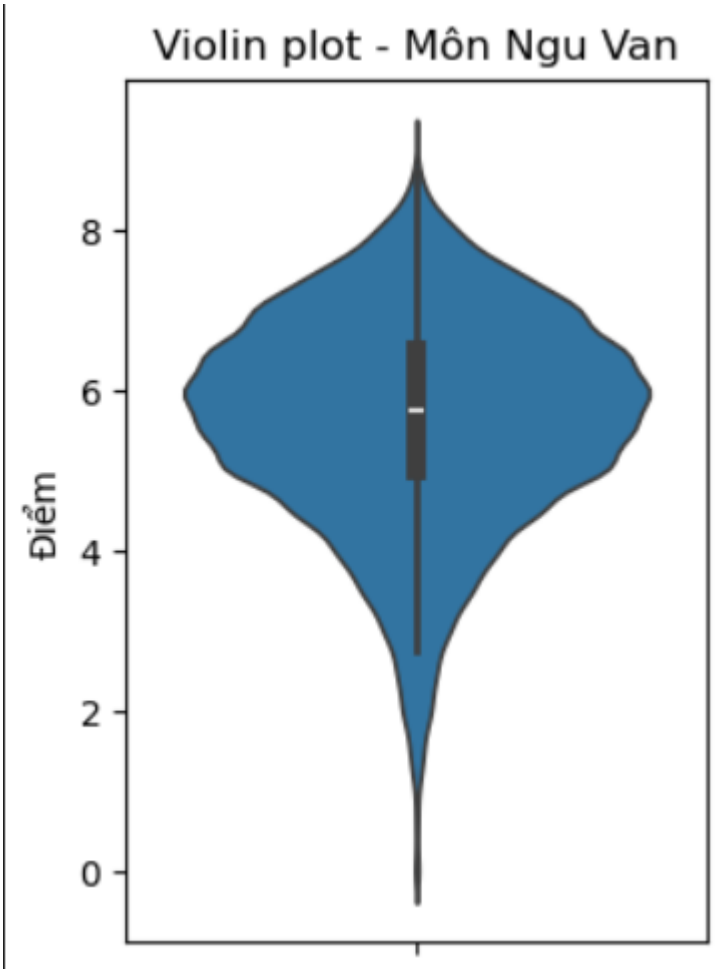
```
=== Thống kê mô tả môn Ngu Van ===
Số lượng: 18256
Điểm trung bình: 5.63
Điểm min: 0.0
Điểm max: 9.0
Độ lệch chuẩn: 1.34
----------------------------------------
```

Hình 37. Thống kê mô tả môn Ngữ Văn

- **Điểm trung bình:** 5.63
- **Điểm min:** 0.0 → có thí sinh không làm bài/vi phạm quy chế.
- **Độ lệch chuẩn:** 1.34
- **Nhận xét:** Mức điểm trung bình thấp, phổ điểm lệch trái nhẹ. Văn thường phụ thuộc kỹ năng trình bày nên khó đạt điểm tuyệt đối.



Hình 38. Phổ điểm môn Ngữ Văn



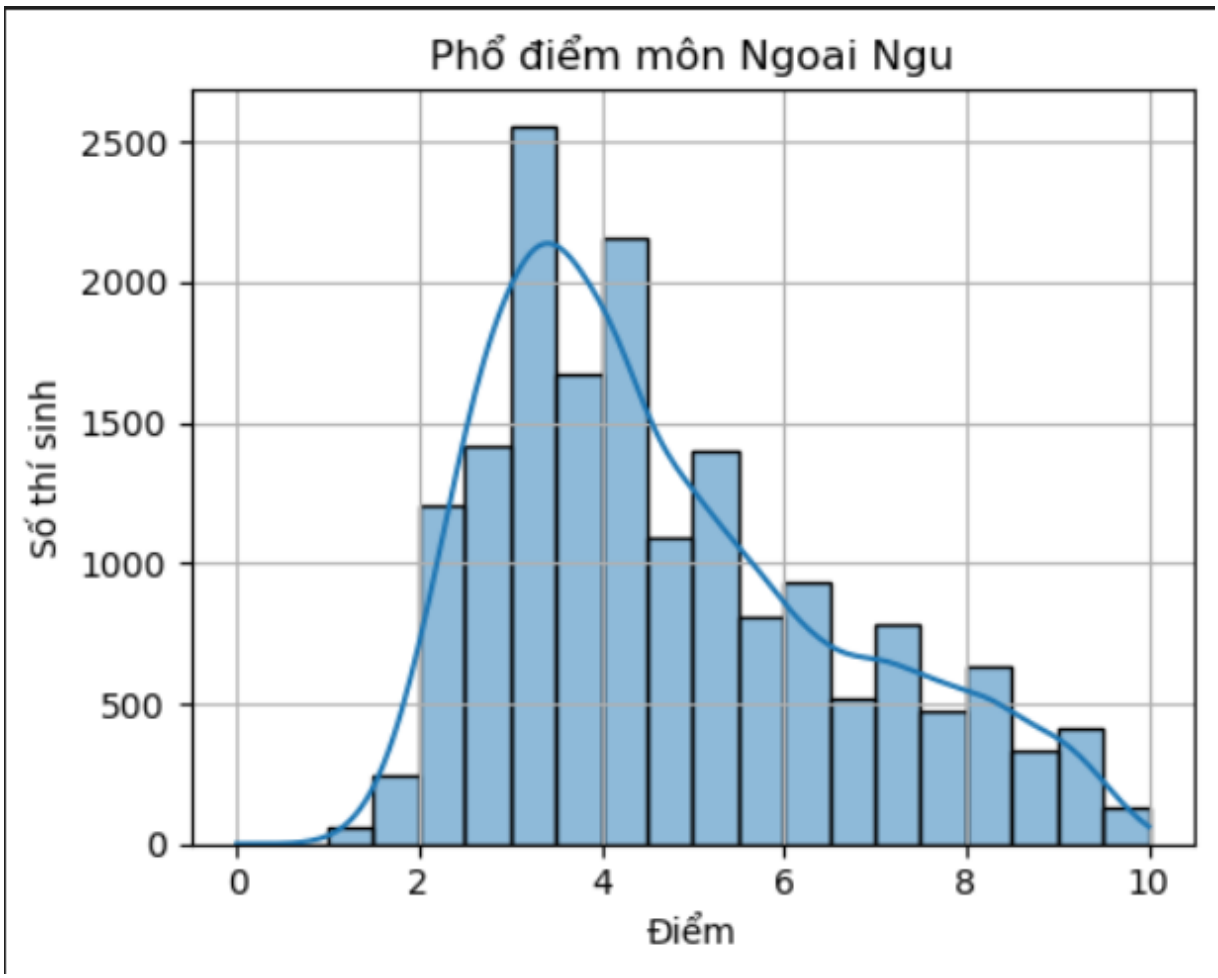
Hình 39. Vionlin môn Ngữ Văn

- Môn Ngoại Ngữ:

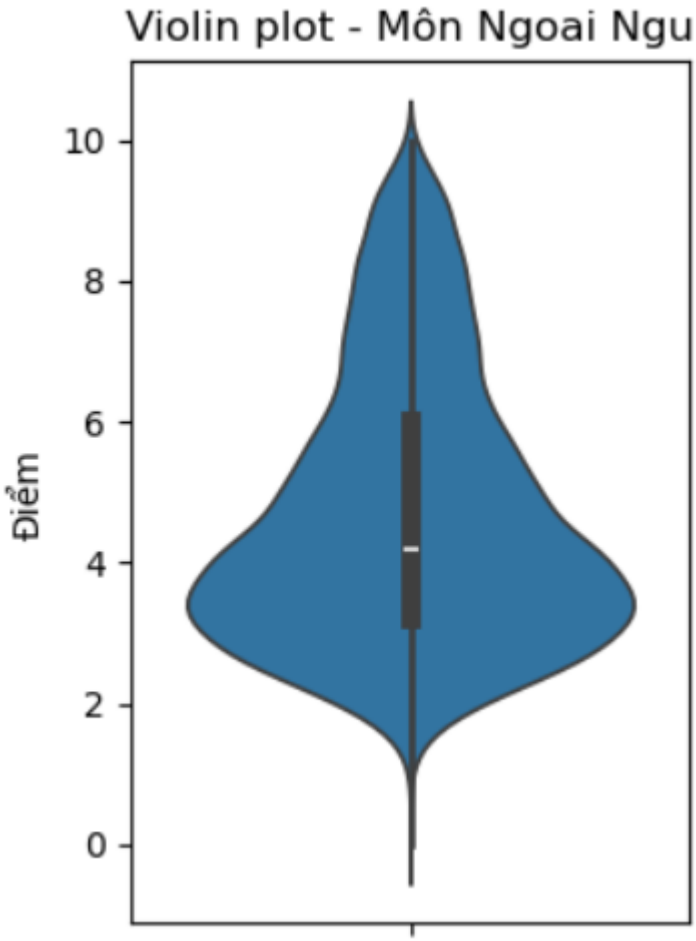
```
=== Thống kê mô tả môn Ngoai Ngu ===
Số lượng: 16819
Điểm trung bình: 4.7
Điểm min: 0.0
Điểm max: 10.0
Độ lệch chuẩn: 1.93
----------------------------------------
```

Hình 40. Thống kê mô tả môn Ngoại Ngữ

- **Điểm trung bình:** 4.70 → thấp nhất trong tất cả các môn.
- **Độ lệch chuẩn:** 1.93 → phân bố rất rộng, chênh lệch lớn giữa học sinh giỏi và yếu.
- **Nhận xét:** Đây là môn có phổ điểm "2 cực", nhiều bạn mất gốc → điểm thấp, trong khi học sinh khá giỏi đạt điểm cao.



Hình 41. Phổ điểm môn Ngoại Ngữ



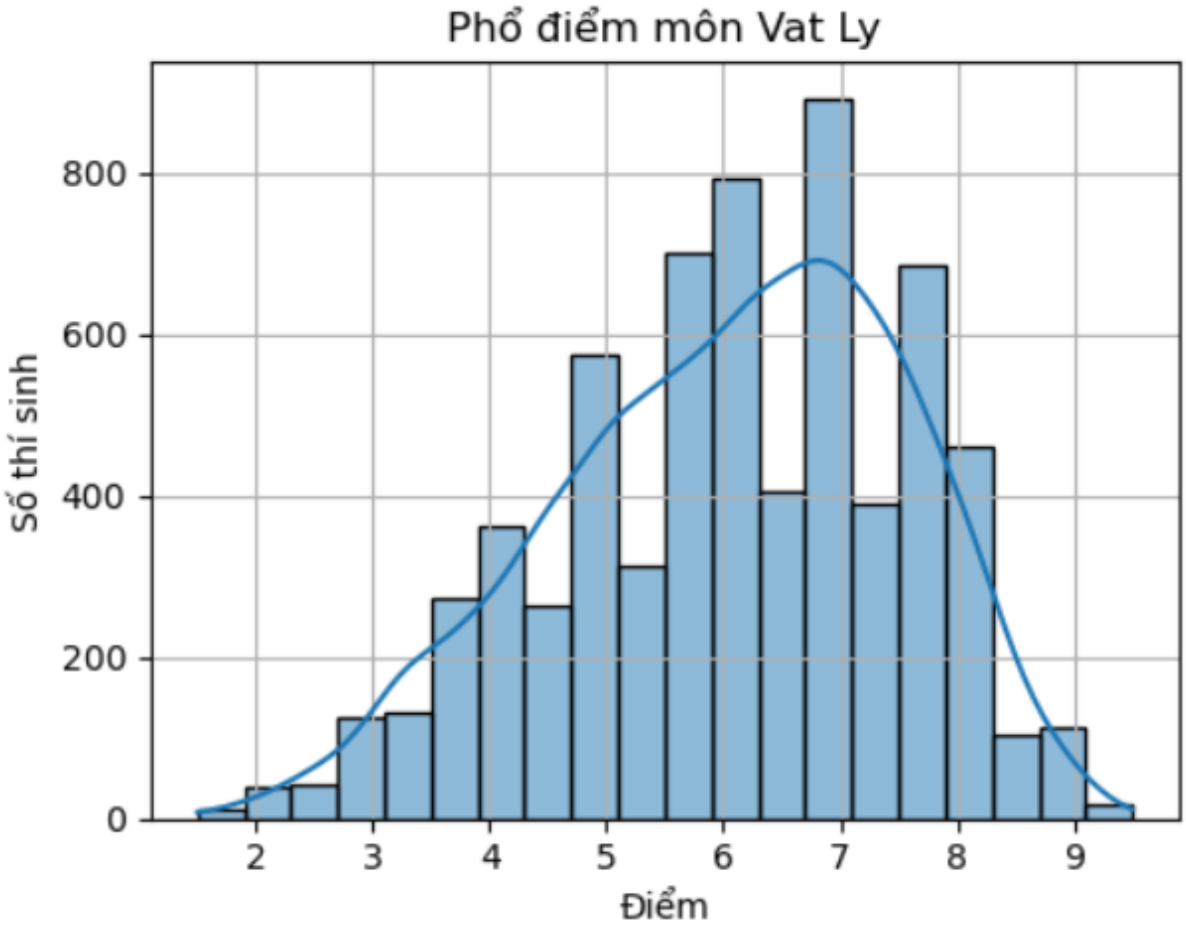
Hình 42. Vionlin môn Ngoại Ngữ

- Môn Vật Lý:

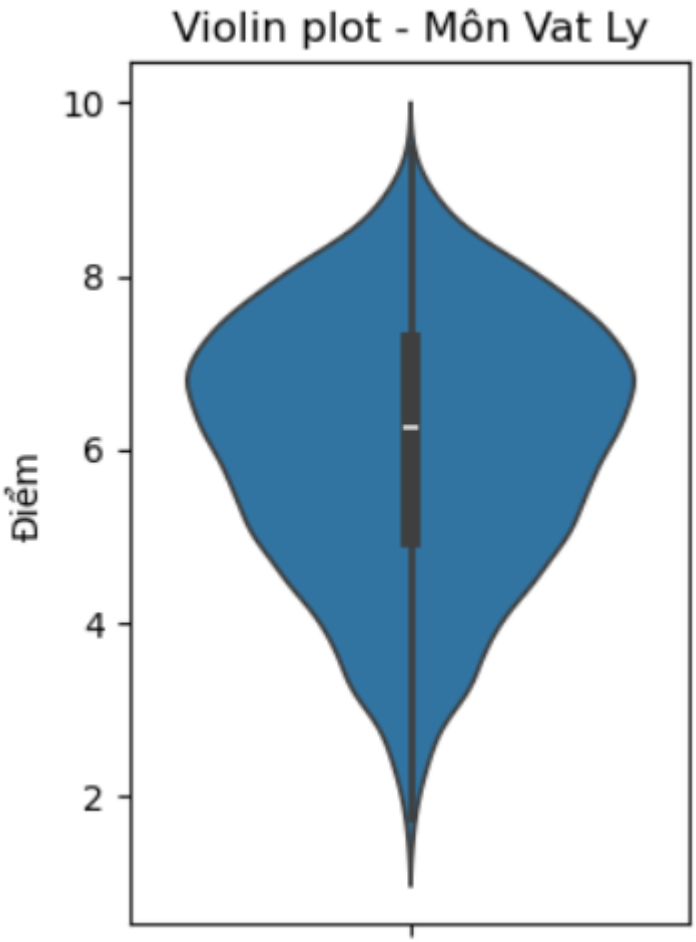
```
=== Thống kê mô tả môn Vat Ly ===
Số lượng: 6700
Điểm trung bình: 6.05
Điểm min: 1.5
Điểm max: 9.5
Độ lệch chuẩn: 1.49
----------------------------------------
```

Hình 43. Thống kê mô tả môn Vật Lý

- **Nhận xét:** Thuộc nhóm KHTN, điểm khá ổn, độ lệch không quá cao. Học sinh có sự chuẩn bị tốt nếu chọn tổ hợp này.



Hình 44. Phổ điểm môn Vật Lý



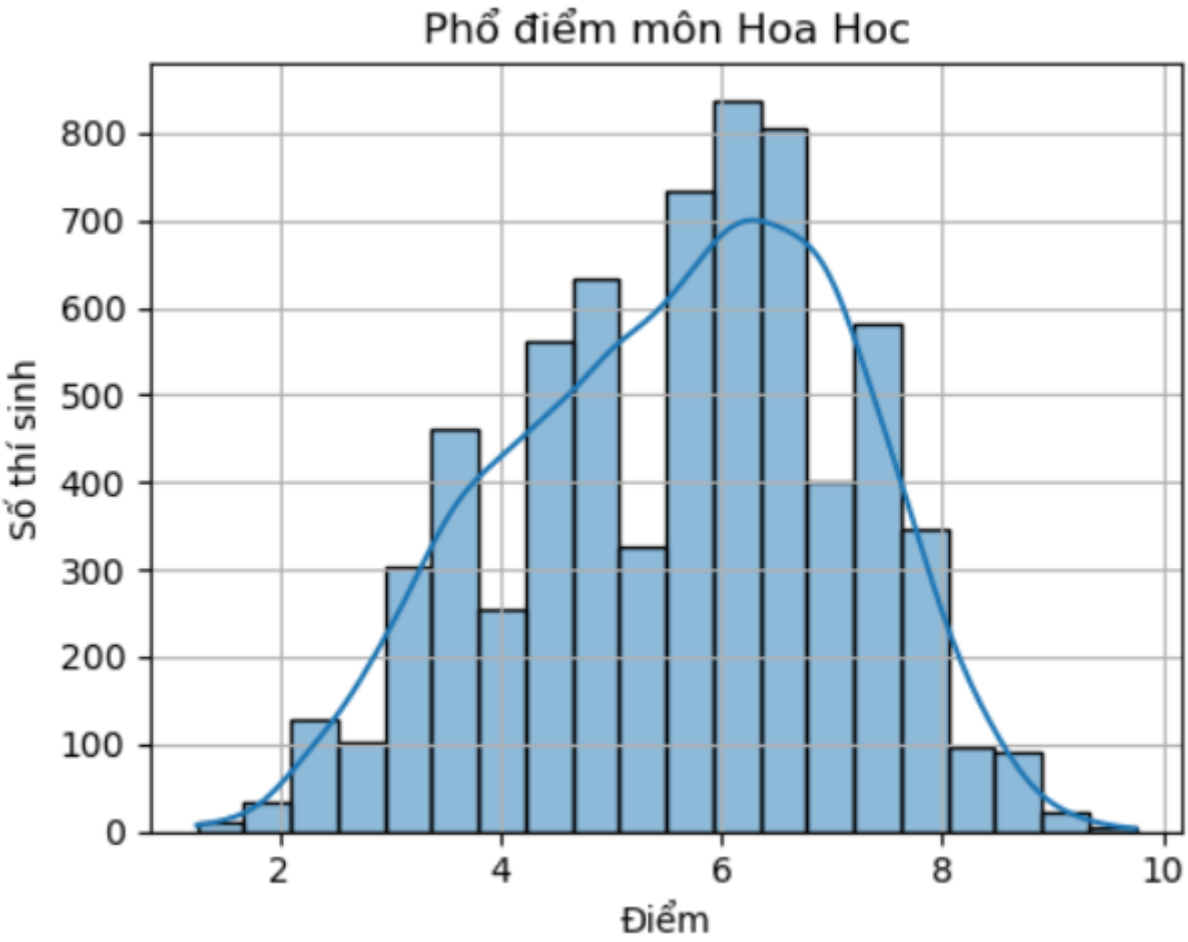
Hình 45. Vionlin môn Vật Lý

- Môn Hóa Học:

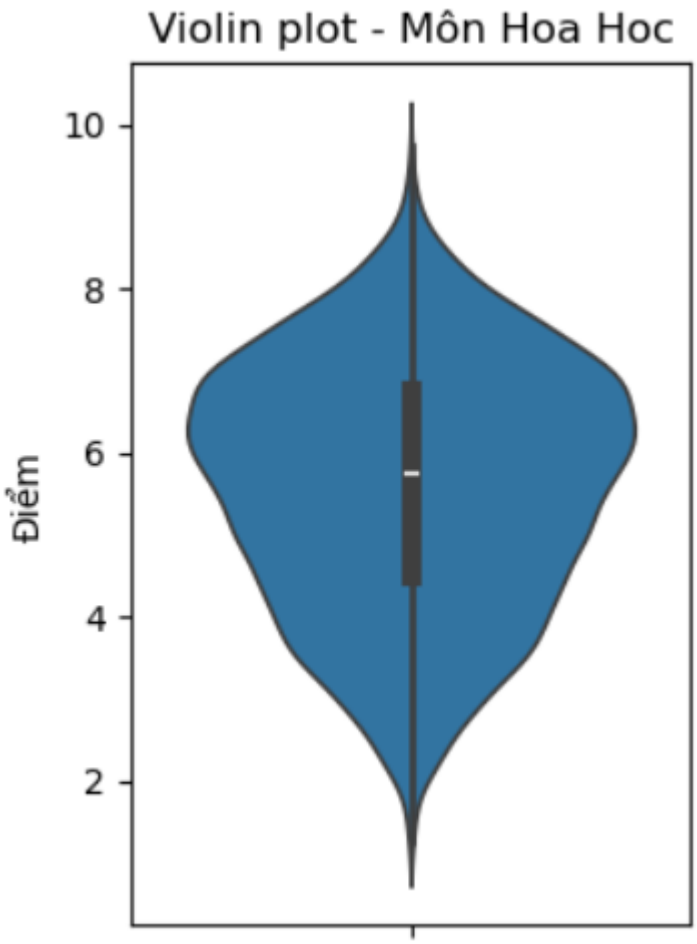
```
=== Thống kê mô tả môn Hoa Hoc ===
Số lượng: 6731
Điểm trung bình: 5.61
Điểm min: 1.25
Điểm max: 9.75
Độ lệch chuẩn: 1.52
----------------------------------------
```

Hình 46. Thống kê mô tả môn Hóa Học

- **Nhận xét:** Phổ điểm hơi lệch trái, trung bình thấp hơn Lý. Hóa có tính phân hóa cao, thể hiện năng lực rõ ràng.



Hình 47. Phổ điểm môn Hóa Học



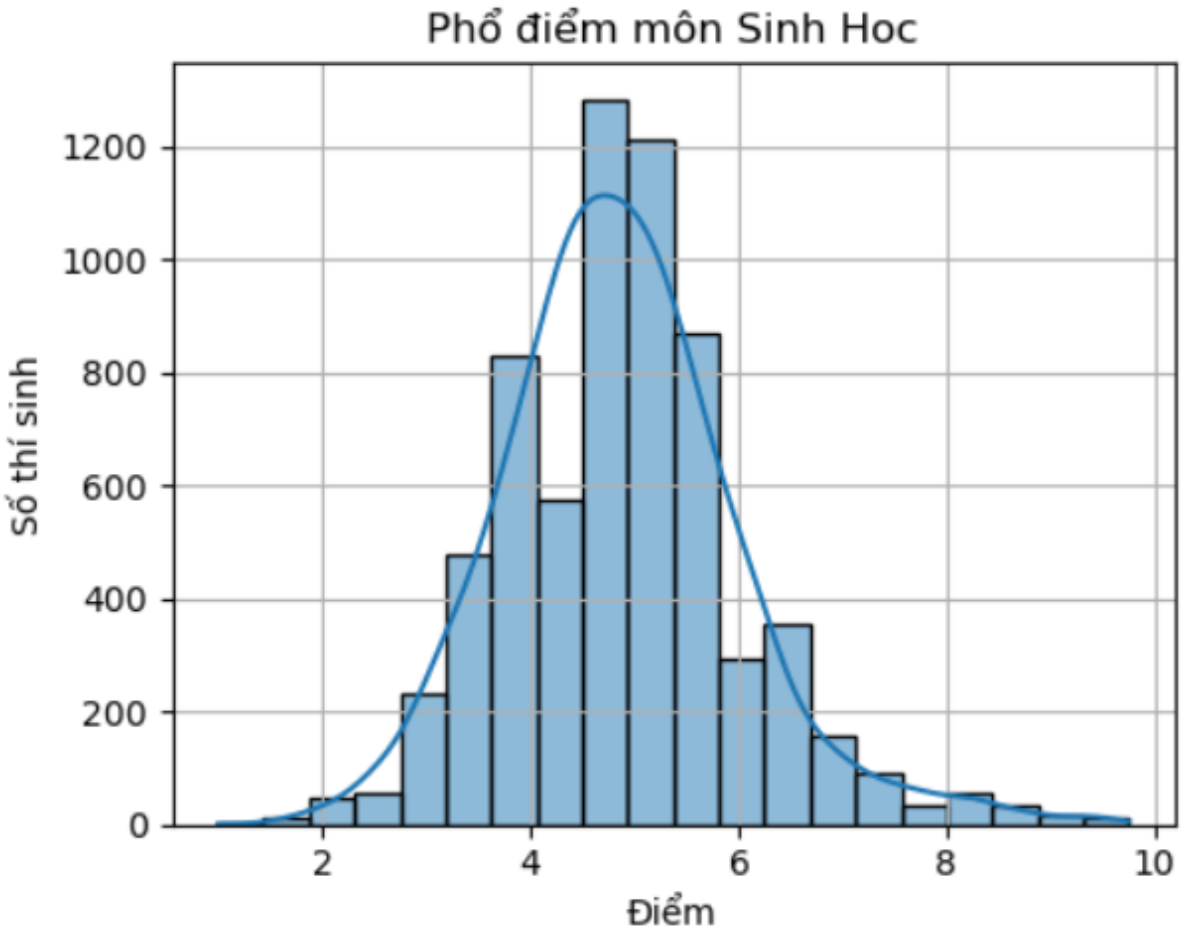
Hình 48. Vionlin môn Hóa Học

- Môn Sinh Học:

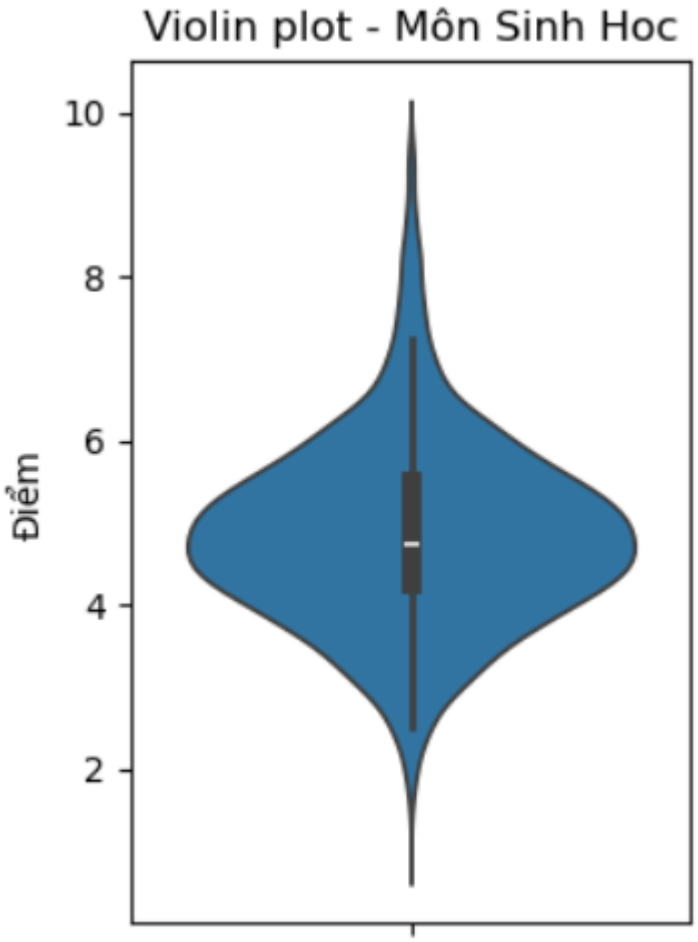
```
=== Thống kê mô tả môn Sinh Hoc ===
Số lượng: 6645
Điểm trung bình: 4.86
Điểm min: 1.0
Điểm max: 9.75
Độ lệch chuẩn: 1.14
----------------------------------------
```

Hình 49.Thống kê mô tả môn Sinh Học

- **Độ lệch chuẩn:** 1.14 → phân tán thấp
- **Nhận xét:** Điểm trung bình thấp, tuy nhiên độ lệch chuẩn nhỏ → đa số học sinh nằm ở mức dưới trung bình.



Hình 50. Phổ điểm môn Sinh Học



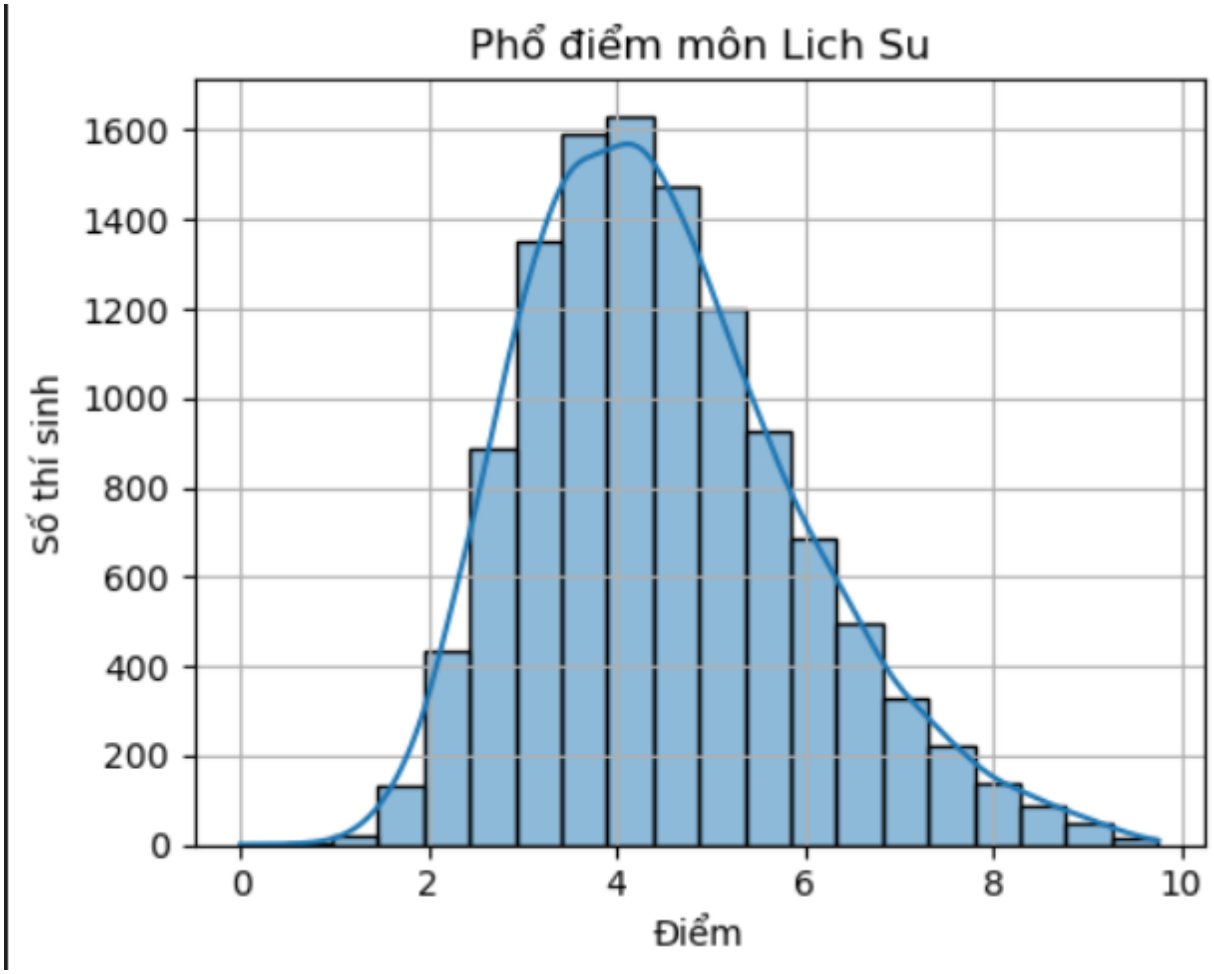
Hình 51. Violin môn Sinh Học

- Môn Lịch Sử:

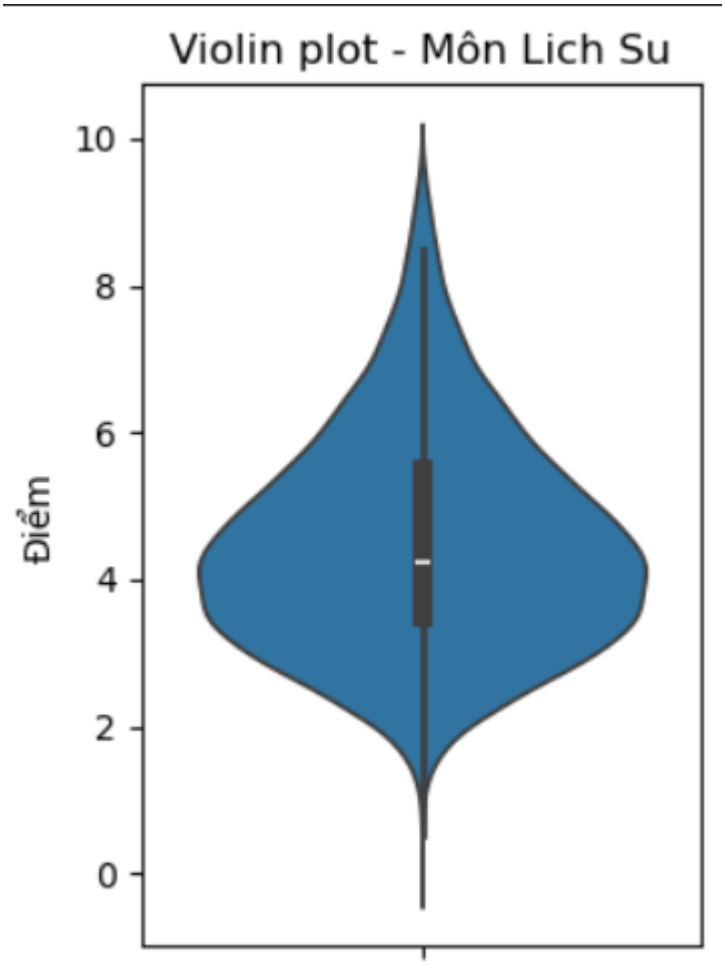
```
=== Thống kê mô tả môn Lich Su ===
Số lượng: 11682
Điểm trung bình: 4.49
Điểm min: 0.0
Điểm max: 9.75
Độ lệch chuẩn: 1.48
----------------------------------------
```

Hình 52. Thống kê mô tả môn Lịch Sử

- **Điểm trung bình:** 4.49 → thuộc nhóm **thấp** nhất
- **Điểm min – max:** 0.0 – 9.75
- **Độ lệch chuẩn:** 1.48
- **Nhận xét:** Phổ điểm lệch trái. Nhiều học sinh đạt điểm thấp, khó đạt điểm cao → cho thấy đây là môn gây "khó" với nhiều học sinh.



Hình 53. Phổ điểm môn Lịch Sử



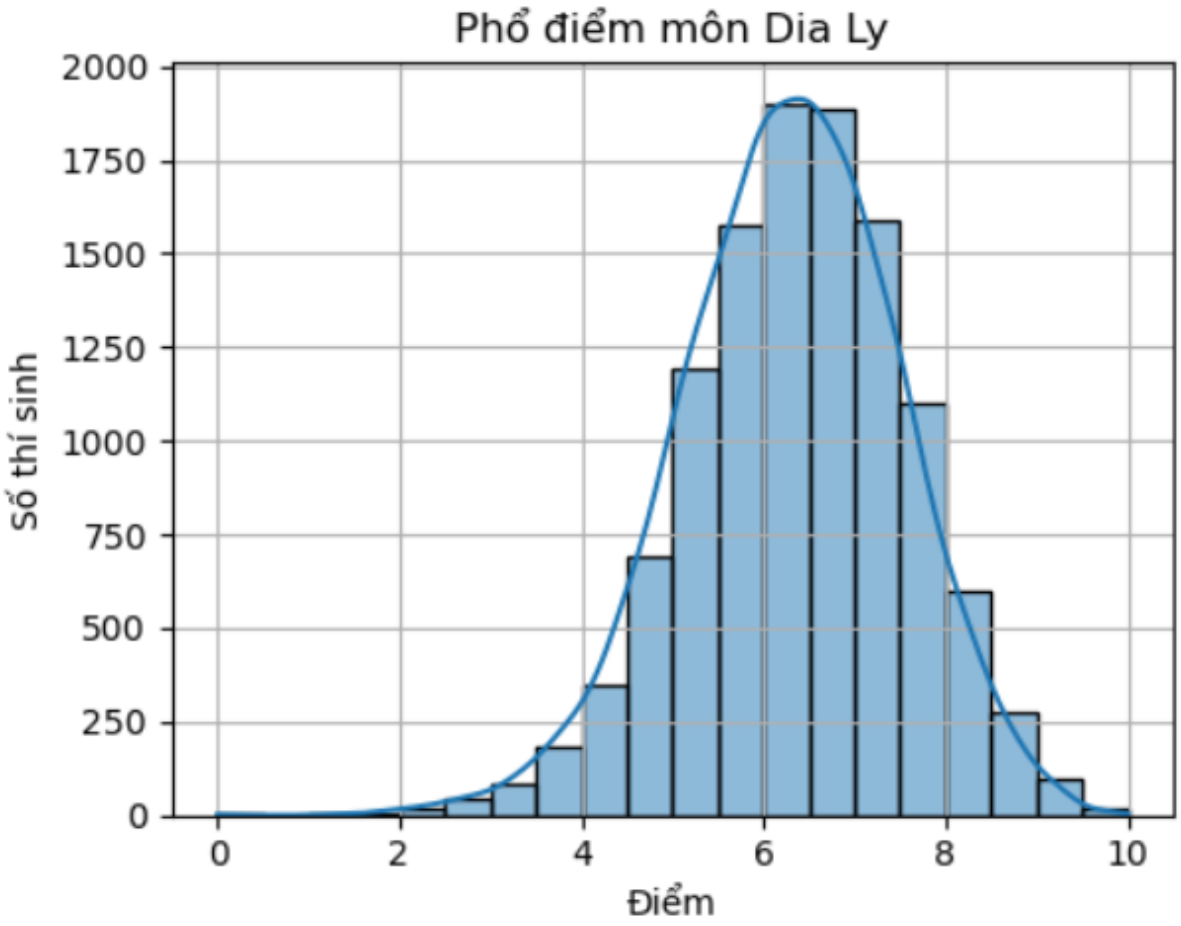
Hình 54. Violin môn Lịch Sử

- Môn Địa Lý:

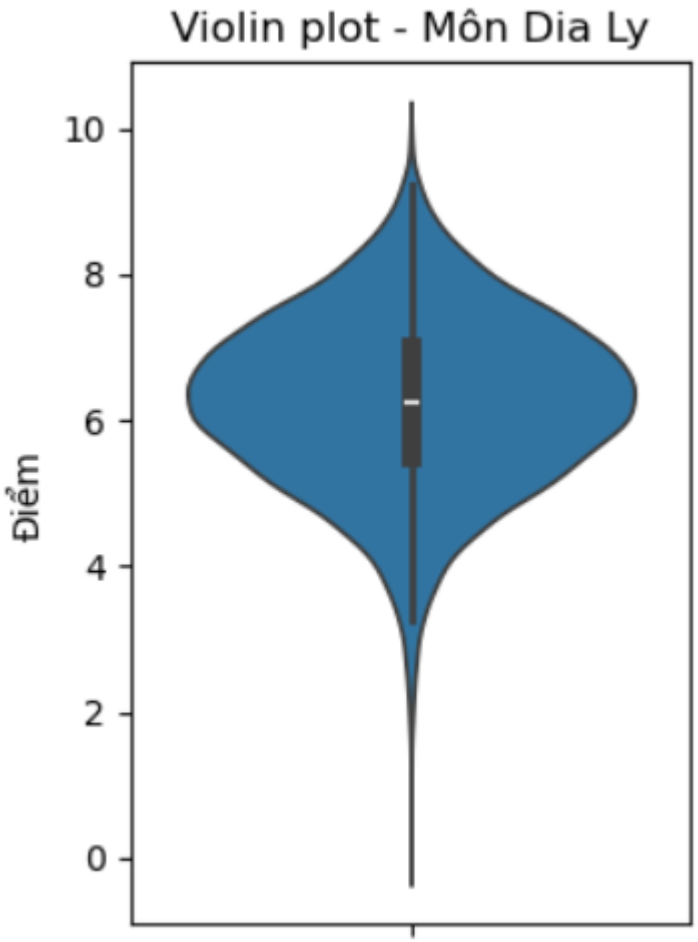
```
=== Thống kê mô tả môn Dia Ly ===
Số lượng: 11601
Điểm trung bình: 6.27
Điểm min: 0.0
Điểm max: 10.0
Độ lệch chuẩn: 1.2
----------------------------------------
```

Hình 55. Thống kê mô tả môn Địa Lý

- **Điểm trung bình:** 6.27
- **Điểm min – max:** 0.0 – 10.0
- **Độ lệch chuẩn:** 1.20
- **Nhận xét:** Điểm trung bình khá cao, phổ điểm ổn định. Phù hợp với nhóm thí sinh chọn tổ hợp KHXH.



Hình 56. Phổ điểm môn Địa Lý



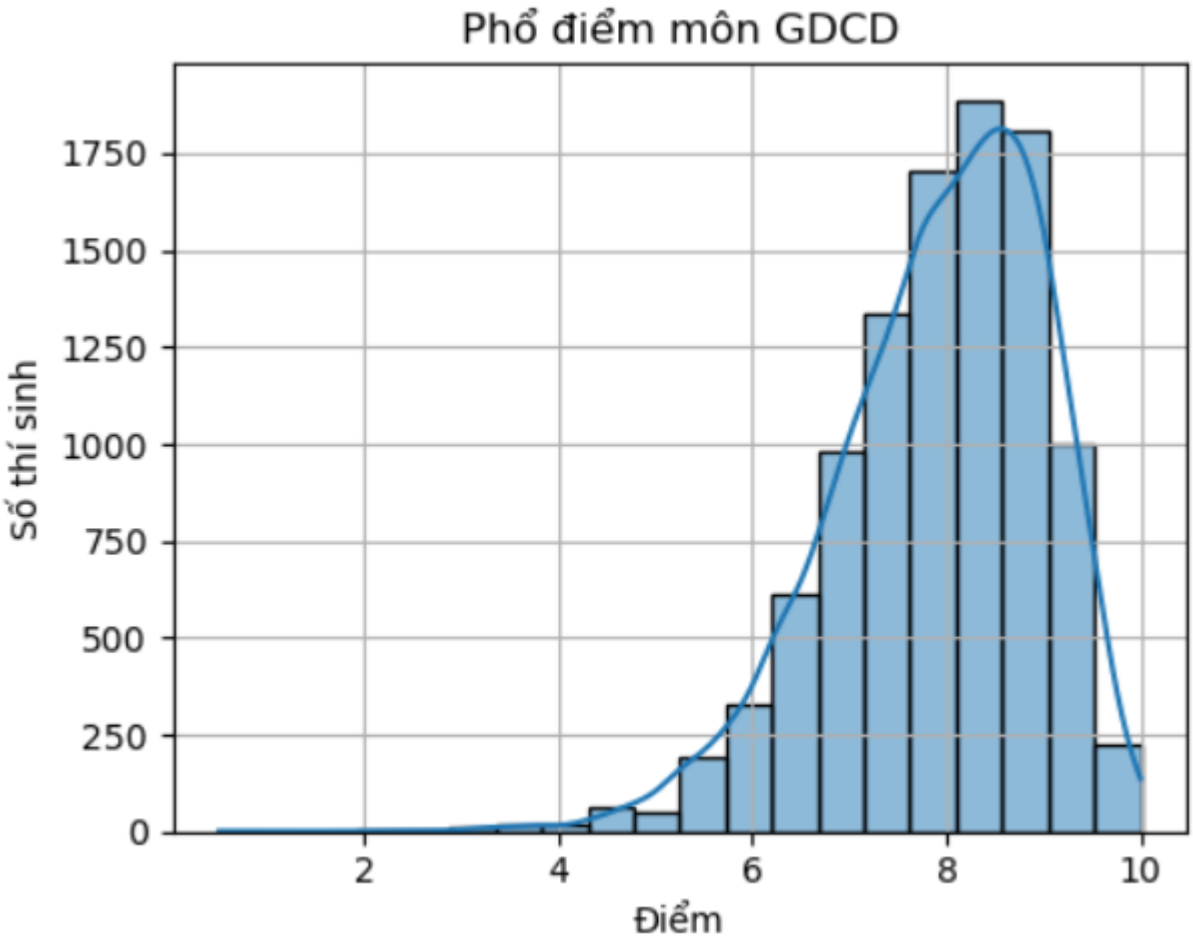
Hình 57. Violin môn Đia Lý

- Môn GDCD:

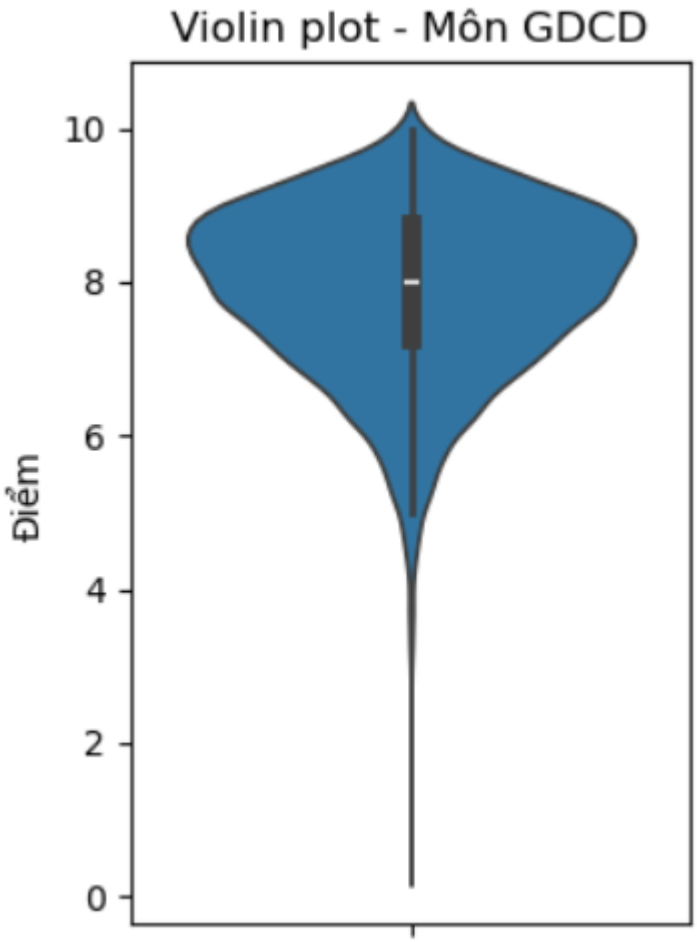
```
=== Thống kê mô tả môn GDCD ===
Số lượng: 10229
Điểm trung bình: 7.92
Điểm min: 0.5
Điểm max: 10.0
Độ lệch chuẩn: 1.11
----------------------------------------
```

Hình 58. Thống kê mô tả môn GDCD

- **Điểm trung bình:** 7.92 → **cao nhất**
- **Điểm min – max:** 0.5 – 10.0
- **Độ lệch chuẩn:** 1.11
- **Nhận xét:** Đây là môn có kết quả tốt nhất trong toàn bộ kỳ thi. Đa số học sinh đạt 8–10 điểm, phổ điểm cao, ít học sinh yếu.



Hình 59. Phổ điểm môn GDCD



Hình 60. Violin môn GDCD

4. Phân Tích Tương Quan Dựa Vào Các Phương Pháp Machine Learning

- Trong phần này, ta sử dụng các mô hình học máy để đánh giá tầm ảnh hưởng (tương quan) của các môn học khác đến điểm thi môn Toán. Hai thuật toán phổ biến được sử dụng là:

- Random Forest Regressor
- Gradient Boosting Regressor

- Phương pháp này cho phép đo độ quan trọng của từng biến đầu vào (feature importance) trong việc dự đoán điểm Toán.

4.1 Phân tích tương quan bằng Random Forest

- Sử dụng RandomForestRegressor từ thư viện sklearn để huấn luyện mô hình dự đoán điểm **Toán** dựa trên các môn khác.

- Tính **feature importance** để xem môn nào có ảnh hưởng mạnh nhất đến điểm Toán.

```
# Biến mục tiêu (target) và các đặc trưng (features)
target = 'Toan'  # Ví dụ: phân tích tương quan với điểm Toán
features = [col for col in list_of_subjects if col != target]

# Loại bỏ NaN và chuẩn bị dữ liệu
df_clean = df_clean.dropna(subset=[target])  # Loại bỏ các hàng có NaN ở cột mục tiêu
X = df_clean[features]
y = df_clean[target]

# Huấn luyện mô hình Random Forest
model = RandomForestRegressor(random_state=42)
model.fit(X, y)

# Lấy giá trị feature importance
feature_importances = model.feature_importances_
importance_df = pd.DataFrame({
    'Feature': features,
    'Importance': feature_importances
}).sort_values(by='Importance', ascending=False)

# Hiển thị kết quả
print(importance_df)
```

Hình 61. Code phân tích tương quan bằng Random Forest

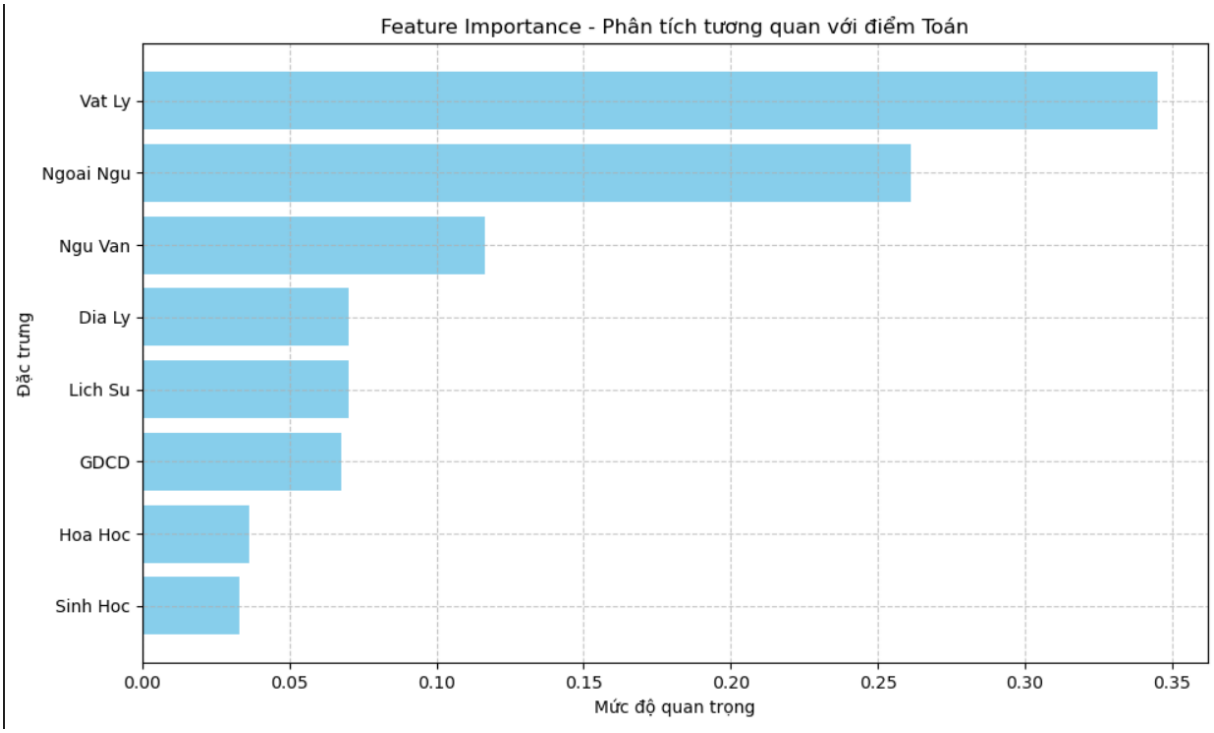
	Feature	Importance
2	Vat Ly	0.344873
1	Ngoai Ngu	0.261246
0	Ngu Van	0.116390
6	Dia Ly	0.070211
5	Lich Su	0.070178
7	GDCD	0.067802
3	Hoa Hoc	0.036193
4	Sinh Hoc	0.033107

Hình 62. Kết quả

- **Nhận xét:** Trả về danh sách độ quan trọng của từng biến đầu vào. Có thể thấy các môn KHTN như **Vật Lý**, **Hóa Học** thường có hệ số quan trọng cao.

```python
# Vẽ biểu đồ mức độ quan trọng của các đặc trưng
plt.figure(figsize=(10, 6))
plt.barh(importance_df['Feature'], importance_df['Importance'], color='skyblue')
plt.title("Feature Importance - Phân tích tương quan với điểm Toán")
plt.xlabel("Mức độ quan trọng")
plt.ylabel("Đặc trưng")
plt.gca().invert_yaxis()  # Đảo ngược trục y để đặc trưng quan trọng nhất ở trên cùng
plt.grid(True, linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

Hình 63. Code dựng hình



Hình 64. Biểu đồ Phân tích tương quan với điểm Toán

- **Nhận xét:** Biểu đồ thể hiện mức ảnh hưởng của từng môn đến điểm Toán. Các thanh dài hơn → ảnh hưởng nhiều hơn đến khả năng dự đoán điểm Toán.

4.2 Phân tích tương quan bằng Gradient Boosting

- Tương tự như Random Forest nhưng sử dụng GradientBoostingRegressor, một mô hình Boosting có độ chính xác cao hơn trong nhiều trường hợp.

```python
# Huấn luyện mô hình Gradient Boosting
gb_model = GradientBoostingRegressor(random_state=42)
gb_model.fit(X, y)

# Lấy giá trị feature importance
gb_feature_importances = gb_model.feature_importances_
gb_importance_df = pd.DataFrame({
    'Feature': features,
    'Importance': gb_feature_importances
}).sort_values(by='Importance', ascending=False)

# Hiển thị kết quả
print(gb_importance_df)
```

Hình 65. Code phân tích tương quan bằng Gradient Boosting

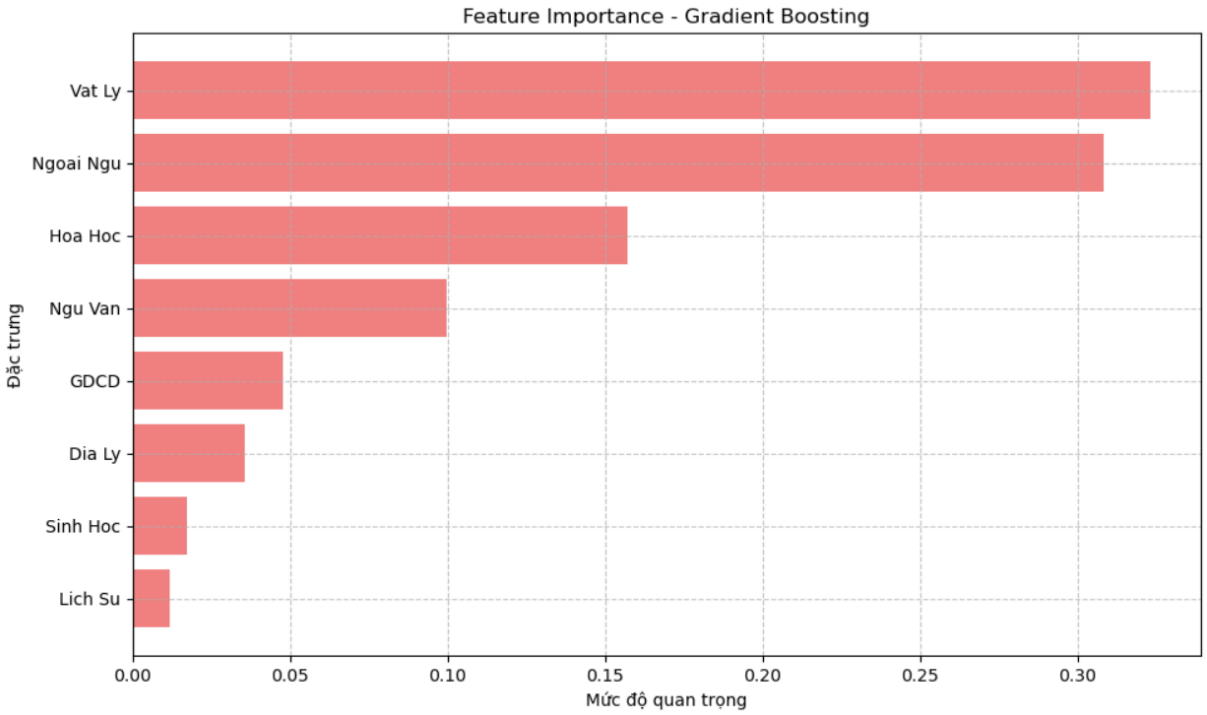
	Feature	Importance
2	Vat Ly	0.322994
1	Ngoai Ngu	0.308196
3	Hoa Hoc	0.157100
0	Ngu Van	0.099595
7	GDCD	0.047756
6	Dia Ly	0.035397
4	Sinh Hoc	0.017366
5	Lich Su	0.011595

Hình 66. Kết quả

- **Nhận xét:** Các hệ số trọng số được tính lại, có thể hơi khác với Random Forest do cách huấn luyện khác nhau.

```python
# Vẽ biểu đồ mức độ quan trọng của các đặc trưng
plt.figure(figsize=(10, 6))
plt.barh(gb_importance_df['Feature'], gb_importance_df['Importance'], color='lightcoral')
plt.title("Feature Importance - Gradient Boosting")
plt.xlabel("Mức độ quan trọng")
plt.ylabel("Đặc trưng")
plt.gca().invert_yaxis()  # Đảo ngược trục y để đặc trưng quan trọng nhất ở trên cùng
plt.grid(True, linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

Hình 67. Code dựng hình



Hình 68. Biểu đồ

- **Nhận xét:** Phân bố ảnh hưởng giữa các môn học tương tự, nhưng có thể nhấn mạnh hơn vào những môn có mối quan hệ phức tạp và phi tuyến với Toán.

- Nhận xét:

- Random Forest và Gradient Boosting đều cho kết quả đáng tin cậy về ảnh hưởng giữa các môn học.
- Các môn có tương quan cao nhất với Toán thường là:
  - Vật Lý, Hóa Học, Ngoại Ngữ
- Các môn như GDCD, Lịch Sử thường có ảnh hưởng thấp nhất đến việc dự đoán điểm Toán.

5. Phân Tích Các Cycles/Frequencies Sử Dụng Thư Viện Stumpy Hoặc Kĩ Thuật Clustering

- Mục tiêu của phần này là khám phá các chu kỳ lặp lại (cycles) hoặc mẫu phổ biến (patterns) trong điểm thi của học sinh, có thể giúp:

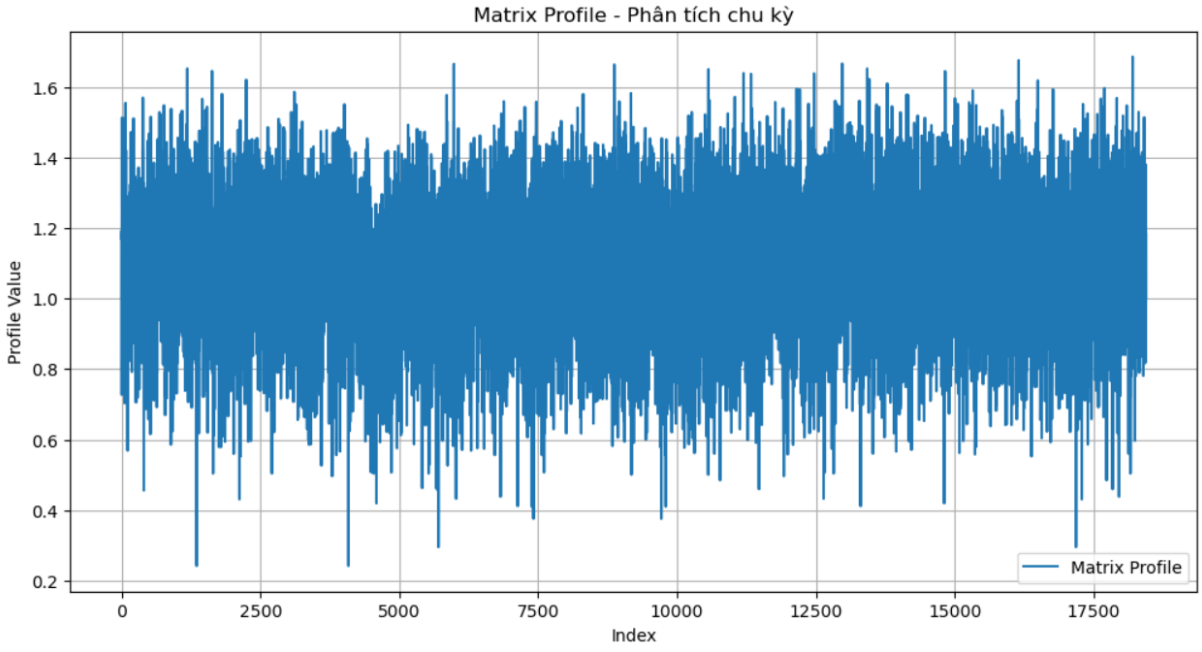
- Xác định xu hướng học tập lặp lại trong dữ liệu.
- Phát hiện nhóm thí sinh có kiểu điểm giống nhau.
- Phục vụ phân tích chuyên sâu hoặc gợi ý cải tiến phương pháp giảng dạy.

5.1 Sử dụng Stumpy để phân tích chu kì

- **STUMPY** là thư viện mạnh mẽ giúp tìm chuỗi con (subsequences) giống nhau trong dữ liệu thời gian – ở đây ta ứng dụng để tìm chu kỳ điểm số lặp lại.

```
time_series = df['Toan'].dropna().values  # Loại bỏ giá trị NaN

# Tính toán Matrix Profile với độ dài chuỗi con (window size)
window_size = 10  # Độ dài chuỗi con, bạn có thể điều chỉnh
matrix_profile = stumpy.stump(time_series, m=window_size)

# Trích xuất giá trị Matrix Profile và vị trí
profile_values = matrix_profile[:, 0]  # Giá trị ma trận khoảng cách
profile_indices = matrix_profile[:, 1]  # Vị trí chuỗi con tương tự

# Vẽ biểu đồ Matrix Profile
plt.figure(figsize=(12, 6))
plt.plot(profile_values, label="Matrix Profile")
plt.title("Matrix Profile - Phân tích chu kỳ")
plt.xlabel("Index")
plt.ylabel("Profile Value")
plt.legend()
plt.grid(True)
plt.show()

# Xác định các chu kỳ lặp lại
min_index = np.argmin(profile_values)  # Vị trí chu kỳ lặp lại tốt nhất
print(f"Chu kỳ lặp lại tốt nhất bắt đầu tại index {min_index} và tương tự với index {profile_indices[min_index]}")
```

Hình 69. Code sử dụng stumpy



Hình 70. Biểu đồ Phân tích chu kì

```
Chu kỳ lặp lại tốt nhất bắt đầu tại index 1351 và tương tự với index 4083
```

Hình 71. Kết quả

5.2 Phân tích clustering để tìm patterns

- Ngoài STUMPY, ta có thể dùng các kỹ thuật phân cụm (clustering) như KMeans để nhóm các học sinh có điểm tương đồng trên nhiều môn.

```
# Chuẩn hóa dữ liệu
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df_clean[list_of_subjects].dropna())

# Áp dụng K-Means clustering
n_clusters = 3  # Số cụm, bạn có thể điều chỉnh
kmeans = KMeans(n_clusters=n_clusters, random_state=42)
clusters = kmeans.fit_predict(scaled_data)

# Thêm kết quả clustering vào DataFrame
df_clean['Cluster'] = clusters

# Hiển thị số lượng mẫu trong mỗi cụm
print(df_clean['Cluster'].value_counts())
```

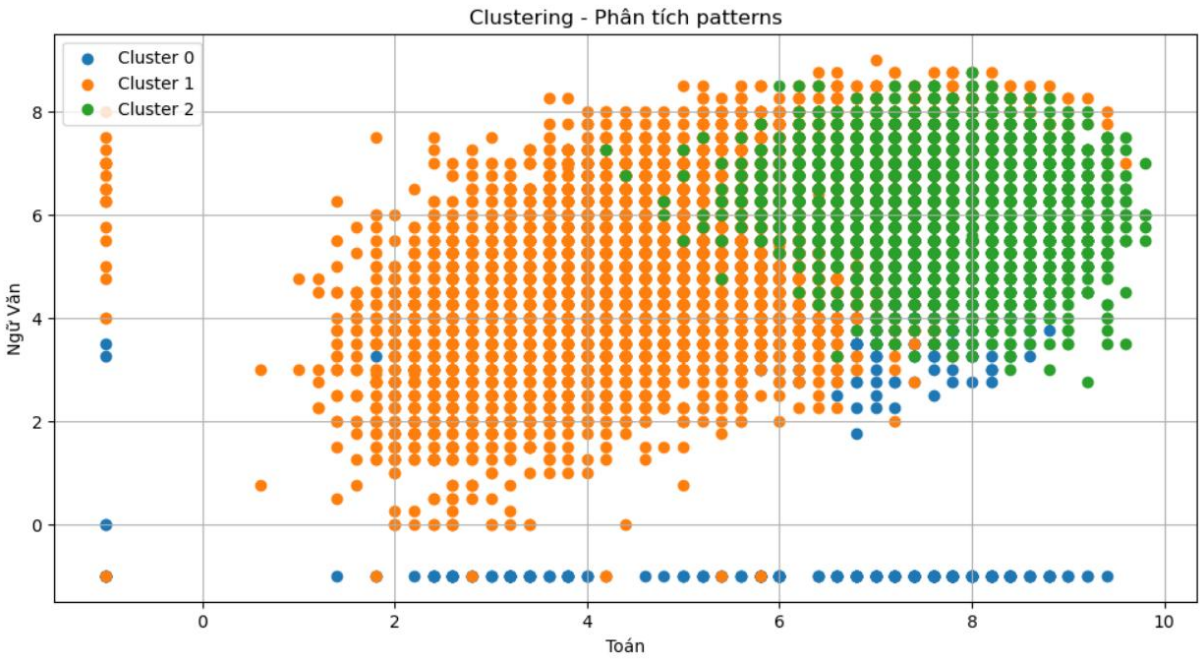
Hình 72. Code phân tích clustering

```
Cluster
1    11680
2     4316
0     2489
Name: count, dtype: int64
```

Hình 73. Kết quả

```
# Vẽ biểu đồ phân cụm
plt.figure(figsize=(12, 6))
for cluster in range(n_clusters):
    cluster_data = df_clean[df_clean['Cluster'] == cluster]
    plt.scatter(cluster_data['Toan'], cluster_data['Ngu Van'], label=f"Cluster {cluster}")
plt.title("Clustering - Phân tích patterns")
plt.xlabel("Toán")
plt.ylabel("Ngữ Văn")
plt.legend()
plt.grid(True)
plt.show()
```

Hình 74. Code dựng hình



Hình 75. Biểu đồ Phân tích patterns

PHẦN III: KẾT LUẬN

- Từ quá trình thu thập, tiền xử lý và phân tích dữ liệu điểm thi THPTQG năm 2019 tại Hải Phòng, có thể rút ra các nhận định chính sau:

- **Phân bố điểm thi** cho thấy sự phân hóa tương đối rõ rệt giữa các môn: Toán, Văn, Ngoại Ngữ đều có độ lệch chuẩn lớn, trong đó Ngoại Ngữ là môn có điểm trung bình thấp nhất. Về khối KHTN, Vật Lý và Hóa Học nhỉnh hơn Sinh Học, còn khối KHXH thì Lịch Sử thấp, Địa Lý ổn định, GDCD cao nhất.
- **Thống kê số môn thi** khẳng định phần lớn thí sinh thi từ 3–6 môn, phản ánh xu hướng chọn khối tập trung hoặc môn bắt buộc. Điểm trung bình khi thi nhiều môn không giảm đáng kể, cho thấy năng lực tổng quát tốt ở nhóm thí sinh này.
- **Phân tích tương quan** (bằng Random Forest, Gradient Boosting) chỉ ra Toán có mối liên hệ chặt chẽ với Lý, Hóa; trong khi Lịch Sử và GDCD ít ảnh hưởng đến dự đoán điểm Toán. Điều này nhấn mạnh tầm quan trọng của nền tảng KHTN đối với kết quả môn Toán.
- **Kết quả** phản ánh chất lượng giáo dục Hải Phòng ở mức khá, nhưng vẫn còn hạn chế ở một số môn như Ngoại Ngữ và Lịch Sử. Do vậy, cần ưu tiên cải tiến phương pháp dạy – học, tăng thực hành và ôn tập sát nhu cầu, nhất là với môn có phổ điểm thấp. Việc khai thác các kỹ thuật phân cụm hoặc phân tích chu kỳ cũng góp phần hỗ trợ ra quyết định chính sách giáo dục địa phương một cách hiệu quả và khoa học hơn.

PHẦN IV: PHỤ LỤC

1. Phụ Lục

Link GitHub: https://github.com/dusgkiet/THPTQG_source_analysis

2. Phân Công

Tên Thành Viên	Phân Công Tuần 1	Deadline
Trần Minh Phúc	Phân tích tài liệu	11/03/2025
Châu Gia Kiệt	Tìm kiếm tài liệu, Viết tài liệu mô tả	11/03/2025

Tên Thành Viên	Phân Công Tuần 2	Deadline
Trần Minh Phúc	Outlier, Boxplot	18/03/2025
Châu Gia Kiệt	Làm sạch dữ liệu	18/03/2025

Tên Thành Viên	Phân Công Tuần 3	Deadline
Trần Minh Phúc	Làm Machine Learning	25/03/2025
Châu Gia Kiệt	Viết lại tài liệu và tóm tắt	25/03/2025

TÀI LIỆU THAM KHẢO

TTXVN. (2025, 03 11). *TRA CỨU ĐIỂM THI TỐT NGHIỆP THPT 2019*. (TTXVN, Editor) Retrieved from TTXVN: https://diemthi.vnanet.vn/diem-thi/2019

--- HẾT---