

# Phân tích điểm thi và ứng dụng ANN

Thời gian: 2.5 giờ - Ngày: 04/04/2025

**Lưu ý:** Sinh viên được phép sử dụng tài liệu, nhưng phải giải thích rõ ràng logic, lý do và ngữ cảnh thực tế cho từng bước. Đáp án chỉ có code hoặc kết quả mà không có giải thích sẽ không được chấm điểm.

## Phần A: Bộ dữ liệu điểm thi sinh viên

Dưới đây là bộ dữ liệu giả định về điểm thi của 30 sinh viên, bao gồm các cột: **Student\_ID**, **Course**, **Score** (0-100), **Attendance (%)** (0-100), **Study\_Hours**, **Gender**, **Campus**. Một số giá trị bị thiếu để bạn thực hành xử lý.

Bảng 1: Dữ liệu điểm thi với giá trị thiếu

Student_ID	Course	Score	Attendance (%)	Study_Hours	Gender	Campus
S001	Math	85	90	5	Male	Campus_A
S002	Physics		85	4	Female	Campus_B
S003	Chemistry	78	70		Male	Campus_A
S004	Math	92	95	6	Female	Campus_C
S005	Physics	65	60	3		Campus_B
S006	Chemistry	88		5	Male	Campus_A
S007	Math		80	4	Female	Campus_C
S008	Physics	75	88	5	Male	Campus_B
S009	Chemistry	90	92		Female	Campus_A
S010	Math	70	65	3	Male	Campus_C
S011	Physics	82	75	4	Female	Campus_B
S012	Chemistry	85		6	Male	Campus_A
S013	Math	95	98	7	Female	Campus_C
S014	Physics	68		3	Male	Campus_B
S015	Chemistry	83	90	5		Campus_A
S016	Math	77	70	4	Female	Campus_C
S017	Physics	89	95		Male	Campus_B
S018	Chemistry	72	80	3	Female	Campus_A
S019	Math	84		5	Male	Campus_C
S020	Physics	91	88	6	Female	Campus_B
S021	Chemistry	66	60	2		Campus_A
S022	Math	87	92	5	Male	Campus_C
S023	Physics		75	4	Female	Campus_B
S024	Chemistry	93	97		Male	Campus_A
S025	Math	79	85	3	Female	Campus_C
S026	Physics	86	90	5	Male	Campus_B
S027	Chemistry	71		4	Female	Campus_A
S028	Math	94	95	6		Campus_C
S029	Physics	80	70	3	Male	Campus_B
S030	Chemistry	88	88	5	Female	Campus_A

## Phần B: Câu hỏi kiểm tra

### Câu hỏi EDA (60 điểm)

1. (4 điểm) Tính tỷ lệ phần trăm giá trị thiếu trong từng cột bằng Pandas. Dựa trên kết quả, đề xuất một quy trình thu thập dữ liệu cụ thể cho trường học để giảm thiểu dữ liệu thiếu trong tương lai, giải thích tại sao quy trình này phù hợp với từng môn học (Math, Physics, Chemistry).
2. (4 điểm) Điền giá trị thiếu trong **Score** bằng trung bình của **Course** tương ứng, sau đó đề xuất một phương pháp điền giá trị khác (không dùng thư viện tự động) dựa trên đặc điểm của từng **Campus**. Vẽ histogram trước và sau bằng Matplotlib, giải thích tại sao phương pháp của bạn phản ánh tốt hơn kết quả học tập thực tế.
3. (4 điểm) Tính độ lệch chuẩn của **Study\_Hours** bằng NumPy. Đề xuất một chính sách khuyến khích học tập cho sinh viên dựa trên phân tích độ lệch chuẩn và trung vị, giải thích tại sao chính sách này có thể cải thiện điểm số tổng thể.
4. (4 điểm) Tính correlation giữa **Attendance (%)** và **Score** cho từng **Campus**. Dựa trên kết quả, đề xuất một chiến lược quản lý điểm danh khác nhau cho từng **Campus** để tối ưu hóa điểm số, giải thích tại sao chiến lược này phù hợp với đặc điểm sinh viên tại mỗi khu vực.
5. (4 điểm) Vẽ boxplot của **Score** theo **Course** và **Gender** (kết hợp) bằng Seaborn. Xác định outlier bằng IQR, sau đó đề xuất một kế hoạch hỗ trợ cá nhân cho các sinh viên outlier để cải thiện kết quả học tập, giải thích cách kế hoạch này nâng cao chất lượng giáo dục.
6. (4 điểm) Tạo cột mới **Efficiency = Score / Study\_Hours**. Tìm sinh viên có **Efficiency** cao nhất, sau đó đề xuất một phần thưởng hoặc chương trình học bổng dựa trên chỉ số này, giải thích tác động của nó đến động lực học tập của sinh viên khác.
7. (4 điểm) Tính tỷ lệ sinh viên nữ (**Gender = Female**) trong từng **Course** sau khi điền giá trị thiếu bằng mode. Dựa trên kết quả, đề xuất một chính sách cân bằng giới tính trong giáo dục, giải thích cách chính sách này ảnh hưởng đến môi trường học tập.
8. (4 điểm) Vẽ scatter plot giữa **Attendance (%)** và **Score**, tô màu theo **Campus**. Đề xuất một kế hoạch cải thiện điểm danh cho từng **Campus** dựa trên phân bố dữ liệu, giải thích tại sao kế hoạch này tối ưu hóa kết quả học tập.
9. (4 điểm) Tính trung bình **Score** của từng **Course** sau khi điền dữ liệu thiếu. Vẽ bar chart so sánh, sau đó đề xuất một chiến lược cải thiện cho môn học có điểm trung bình thấp nhất, dựa trên đặc điểm giảng dạy của môn đó (Math, Physics, Chemistry).
10. (4 điểm) Tìm các sinh viên có **Attendance (%)** dưới 70% nhưng **Score** trên 85 bằng Pandas. Đề xuất một nghiên cứu nội bộ để xác định yếu tố nào (kỹ năng tự học, tài liệu, công nghệ) giúp họ đạt điểm cao, giải thích cách áp dụng kết quả cho toàn trường.
11. (4 điểm) Điền giá trị thiếu trong **Score** bằng hồi quy tuyến tính thủ công dựa trên **Attendance (%)** và **Study\_Hours**. So sánh với phương pháp trung bình, sau đó đề xuất một cách tiếp cận lại (kết hợp hồi quy và trung bình) để cải thiện độ chính xác, giải thích lý do.
12. (4 điểm) Tính skewness của **Score** bằng SciPy. Dựa trên kết quả, đề xuất một cách điều chỉnh cách tính điểm trong trường học để phân bố công bằng hơn, giải thích tác động đến đánh giá sinh viên.
13. (4 điểm) Vẽ pairplot bằng Seaborn cho **Score**, **Attendance (%)**, **Study\_Hours**. Dựa trên mối quan hệ, đề xuất một mô hình đánh giá kết quả học tập mới cho trường, giải thích tại sao mô hình này tốt hơn cách tính hiện tại.
14. (4 điểm) Nhóm dữ liệu theo **Campus**, tính tỷ lệ sinh viên có **Score** trên 80. Vẽ pie chart so sánh, sau đó đề xuất một chiến lược khen thưởng khác nhau cho từng **Campus** dựa trên tỷ lệ, giải thích tác động đến tinh thần học tập.
15. (4 điểm) Tạo hàm Python xác định sinh viên có **Score** ngoài 2 độ lệch chuẩn. Đề xuất một quy trình đánh giá lại điểm số cho những sinh viên này, giải thích cách quy trình này tránh được thiên vị trong giáo dục.

## Câu hỏi ANN (40 điểm)

Xây dựng một ANN để dự đoán **Score** dựa trên **Attendance (%)**, **Study\_Hours**, và **Course** (mã hóa one-hot encoding).

16. (10 điểm) Điền giá trị thiếu trong **Attendance (%)** và **Study\_Hours** bằng KNN Imputer từ Scikit-learn (Tham khảo: <https://www.geeksforgeeks.org/handling-missing-data-with-knn-imputer/>). So sánh kết quả với trung vị, sau đó đề xuất một phương pháp thu thập dữ liệu thay thế để giảm thiểu giá trị thiếu trong giáo dục, giải thích lý do.
17. (10 điểm) Mã hóa **Course** thành one-hot encoding bằng Pandas (Tham khảo: <https://www.geeksforgeeks.org/ml-one-hot-encoding/>). Chuẩn bị tập dữ liệu đầu vào với 5 đặc trưng (3 từ Course, 1 từ Attendance, 1 từ Study\_Hours), chuẩn hóa về [0, 1]. Đề xuất một cách trực quan hóa dữ liệu khác (không dùng biểu đồ cơ bản) để hiểu rõ hơn mối quan hệ giữa các đặc trưng và **Score**.
18. (15 điểm) Xây dựng ANN bằng PyTorch với kiến trúc như hình sau:
  - Input Layer: 5 nơ-ron.
  - Hidden Layer 1: 32 nơ-ron, ReLU.
  - Hidden Layer 2: 16 nơ-ron, ReLU.
  - Hidden Layer 3: 8 nơ-ron, ReLU.
  - Output Layer: 1 nơ-ron (**Score**).

Huấn luyện với 200 epochs, batch size 16, chia 80% train / 20% test, dùng early stopping (patience=20). Vẽ biểu đồ loss, sau đó đề xuất một kiến trúc ANN khác (thay đổi số nơ-ron hoặc tầng) để cải thiện dự đoán, giải thích lý do dựa trên đặc điểm dữ liệu giáo dục.

19. (5 điểm) Đánh giá mô hình bằng MSE và  $R^2$  trên tập test. Nếu  $R^2$  dưới 0.8, phân tích nguyên nhân sai lệch dự đoán dựa trên đặc trưng đầu vào, đề xuất một cách cải thiện mô hình dựa trên ngữ cảnh giáo dục (không chỉ dùng dropout hay thay đổi optimizer).

## Hình: Kiến trúc ANN

