**TRƯỜNG ĐẠI HỌC KINH TẾ - ĐẠI HỌC ĐÀ NẴNG**

**HỌC PHẦN HỌC MÁY**

A logo with text on it

AI-generated content may be incorrect.

**BÁO CÁO NHÓM GIỮA KÌ**

**ĐỀ TÀI: DỰ ĐOÁN ĐIỂM THI CUỐI KÌ CỦA HỌC SINH**

|  |  |
| --- | --- |
| **Tên học phần** | : Học Máy |
| **Giảng viên** | : Nguyễn Văn Chức |
| **Nhóm thực hiện**  **Thành viên** | : Nhóm 5  : Lê Công Hiếu  Nguyễn Duy Khang  Trần Đình Đạt  Đặng Ngọc Danh |

**MỤC LỤC**

[**I. Giới thiệu bài toán** 4](#_Toc211366308)

[1. Bối cảnh 4](#_Toc211366309)

[2. Tầm quan trọng 4](#_Toc211366310)

[3. Mục tiêu 4](#_Toc211366311)

[4. Phương pháp tiếp cận 5](#_Toc211366312)

[**II. Mô Tả Dữ Liệu** 5](#_Toc211366313)

[1. Nguồn gốc 5](#_Toc211366314)

[2. Tổng quan 6](#_Toc211366315)

[3. Mô tả các trường dữ liệu 6](#_Toc211366316)

[4. Biến mục tiêu 7](#_Toc211366317)

[**III. Phân Tích Dữ Liệu Khám Phá** 7](#_Toc211366318)

[1. Phân tích sơ bộ dữ liệu 7](#_Toc211366319)

[1.1. Thông tin Dataset 7](#_Toc211366320)

[1.2. Kiểm tra dữ liệu thiếu và trùng lặp 8](#_Toc211366321)

[1.3. Thống kê mô tả 9](#_Toc211366322)

[2. Phân tích đơn biến (Univariate Analysis) 12](#_Toc211366323)

[2.1. Biến số (Numerical) 12](#_Toc211366325)

[2.2. Biến phân loại (Categorical) 16](#_Toc211366326)

[3. Phân tích đa biến (Multivariate Analysis) 19](#_Toc211366327)

[3.1. Biểu đồ scatter plot 19](#_Toc211366329)

[3.2. Vẽ ma trận tương quan 26](#_Toc211366330)

[3.3. So sánh phân phối của một biến số bằng boxplot 28](#_Toc211366331)

[3.4. Kết luận chính từ EDA 34](#_Toc211366332)

[**IV. Tiền Xử Lý Dữ Liệu** 35](#_Toc211366333)

[1. Xóa cột không cần thiết 35](#_Toc211366334)

[2. Đưa các cột điểm về thang điểm 10 36](#_Toc211366335)

[3. Tính cột điểm trung bình của G1 và G2 38](#_Toc211366336)

[4. Xử lí dữ liệu thiếu 39](#_Toc211366337)

[5. Xử lí ngoại lai (Outliers) 41](#_Toc211366338)

[6. Mã hóa biến phân loại 42](#_Toc211366339)

[7. Tạo pipeline chuẩn hóa dữ liệu 44](#_Toc211366340)

[8. Lưu các dữ liệu cần thiết 45](#_Toc211366341)

[9. Kiểm tra lại phân phối dữ liệu sau khi đã xử lí 46](#_Toc211366342)

[**V. Lựa Chọn & Huấn Luyện Mô Hình** 48](#_Toc211366343)

[1. Các Mô hình Ứng viên 48](#_Toc211366344)

[2. Thước đo đánh giá 51](#_Toc211366345)

[3. Huấn luyện 51](#_Toc211366346)

[**VI. Phân Tích Kết Quả** 55](#_Toc211366347)

[1. Bảng so sánh kết quả metrics 55](#_Toc211366348)

[2. Các biến ảnh hưởng đến kết quả 57](#_Toc211366349)

[3. Hạn chế và Hướng phát triển: 58](#_Toc211366350)

[**VII. Xây dựng và triển khai mô hình chạy trên Streamlit** 59](#_Toc211366351)

1. **Giới thiệu bài toán**
2. Bối cảnh

- Trong hệ thống giáo dục hiện nay, việc đánh giá kết quả học tập của học sinh không chỉ dừng lại ở việc chấm điểm số, mà còn liên quan đến quá trình học tập, định hướng học tập để nâng cao chất lượng giáo dục. Một trong những yếu tố then chốt là khả năng dự đoán kết quả học tập cuối kỳ của học sinh dựa trên các đặc điểm cá nhân, thông tin học tập, điều kiện gia đình và hành vi học tập trong quá trình học.

- Trong dự án này, chúng tôi tập trung vào việc dự đoán điểm số cuối kỳ G3 của học sinh từ bộ dữ liệu bao gồm nhiều yếu tố như giới tính (sex), độ tuổi (age), thời gian học (studytime), kết quả học tập trước đó (G1, G2), v.v.

1. Tầm quan trọng

Dự đoán chính xác điểm G3 có ý nghĩa quan trọng về giáo dục và quản lý học sinh.

Cụ thể:

* Đối với giáo viên và nhà trường: Họ có thể sớm phát hiện những học sinh có nguy cơ đạt kết quả thấp để đưa ra biện pháp hỗ trợ kịp thời.
* Đối với học sinh và phụ huynh: Kết quả dự đoán giúp điều chỉnh kế hoạch học tập, phân bổ thời gian và nguồn lực hợp lý để cải thiện thành tích.
* Về mặt nghiên cứu: Bài toán giúp hiểu sâu hơn về mối quan hệ giữa các yếu tố ảnh hưởng đến kết quả học tập, từ đó hỗ trợ ra quyết định trong hoạch định chương trình giáo dục.

1. Mục tiêu

Mục tiêu của dự án là xây dựng một mô hình học máy có khả năng dự đoán điểm số cuối kỳ (G3) của học sinh dựa trên các thông tin đầu vào sẵn có trong bộ dữ liệu.

Cụ thể:

* Xác định và phân tích các yếu tố ảnh hưởng đến kết quả học tập.
* Huấn luyện các mô hình dự đoán với độ chính xác cao nhất có thể (đánh giá bằng các thước đo như RMSE và R²). Để đảm bảo kết quả có ý nghĩa thực tế, mô hình được yêu cầu thỏa mãn các ràng buộc sau:

RMSE ≤ 1: sai số trung bình nhỏ hơn 1 điểm, tức dự đoán không lệch quá lớn so với thực tế.

R² ≥ 0.8: mô hình giải thích được ít nhất 80% biến động điểm số.

* Cung cấp một công cụ hỗ trợ giáo viên và nhà trường trong việc dự báo kết quả học tập và đưa ra chiến lược hỗ trợ phù hợp.

1. Phương pháp tiếp cận

Bài toán dự đoán điểm G3 là một bài toán hồi quy (Regression Problem), trong đó biến mục tiêu là điểm số cuối kỳ (giá trị số liên tục).

Chúng tôi sẽ tiến hành:

* Phân tích dữ liệu: Khám phá mối quan hệ giữa các đặc trưng đầu vào và điểm G3, phát hiện các yếu tố ảnh hưởng mạnh nhất.
* Tiền xử lý dữ liệu: Làm sạch dữ liệu, xử lý giá trị thiếu, mã hóa biến phân loại và chuẩn hóa dữ liệu nếu cần.
* Thử nghiệm và so sánh nhiều mô hình học máy như Linear Regression, Decision Tree, Random Forest Regressor,…
* Đánh giá mô hình dựa trên các chỉ số như Root Mean Squared Error (RMSE) và R² score.
* Lựa chọn mô hình tối ưu: Cuối cùng, lựa chọn mô hình có độ chính xác và khả năng dự đoán tốt nhất để triển khai.

Trong phần mở đầu của dự án, chúng tôi đã xây dựng một Project Brief nhằm tóm tắt ngắn gọn nội dung, mục tiêu và phương pháp tiếp cận của bài toán. Phần này giúp người đọc nhanh chóng nắm bắt được bức tranh tổng quan của toàn bộ dự án trước khi đi sâu vào các phân tích chi tiết.

**PROJECT BRIEF :** [Link Project Brief tại đây](https://www.canva.com/design/DAG06BeLGkc/gGDFisH38O8DfLyUET54hA/edit)

1. **Mô Tả Dữ Liệu**
2. Nguồn gốc

Bộ dữ liệu được sử dụng trong dự án xuất phát từ Kaggle. Dữ liệu này là dữ liệu công khai, được các nhà nghiên cứu và người làm học máy thường dùng để phân tích và mô hình hóa kết quả học tập của học sinh.

[Link Dataset trên Kaggle tại đây](https://www.kaggle.com/code/zeppellinn/student-perfomance/input?fbclid=IwY2xjawNTXgpleHRuA2FlbQIxMABicmlkETFlNko2VXZaTDJiZmZ3Q2pqAR5YL5475FUq5W0Ig-qEyk4orlCr-jOVSXEsR4wMO4Oq-wMHsdVnrZpbtrocZg_aem_ogxdeKkaK3p6tDqVphZsUg)

[File Excel tại đây](https://drive.google.com/drive/folders/1hD7-ZQq8d9GGQEUUNFia3kuL3nF0dAwM?usp=sharing)

1. Tổng quan

+ Số lượng mẫu (số dòng): 395 học sinh.

+ Số lượng biến (số cột): khoảng 33 cột (bao gồm cả biến mục tiêu).

+ Mỗi dòng trong bộ dữ liệu đại diện cho một học sinh, trong khi mỗi cột tương ứng với một đặc trưng mô tả thông tin hoặc kết quả học tập của học sinh đó.

1. Mô tả các trường dữ liệu

Dưới đây là các biến quan trọng được lựa chọn từ bộ dữ liệu, và sẽ được sử dụng trong quá trình phân tích cũng như xây dựng mô hình dự đoán điểm cuối kỳ (G3) của học sinh.

|  |  |
| --- | --- |
| **Tên cột** | **Ý nghĩa** |
| sex | Giới tính của học sinh (M = Nam, F = Nữ) |
| age | Tuổi của học sinh (15 đến 22 tuổi) |
| Pstatus | Tình trạng sống của cha mẹ (T = sống cùng, A = sống riêng) |
| studytime | Thời gian học trung bình mỗi ngày (1–4) |
| failures | Số lần học sinh bị trượt môn (1-3) |
| activities | Học sinh có tham gia hoạt động ngoại khóa hay không (yes/no) |
| higher | Học sinh có dự định học lên cao hơn hay không (yes/no) |
| romantic | Học sinh có đang trong mối quan hệ tình cảm hay không (yes/no) |
| famrel | Mức độ quan hệ gia đình (1 – rất xấu, 5 – rất tốt) |
| freetime | Thời gian rảnh sau giờ học (1 – rất ít, 5 – rất nhiều) |
| goout | Tần suất ra ngoài với bạn bè (1 – rất hiếm, 5 – rất thường xuyên) |
| health | Tình trạng sức khỏe hiện tại (1 – rất kém, 5 – rất tốt) |
| absences | Số buổi học sinh nghỉ học |
| G1 | Điểm học kỳ 1 (0–20) |
| G2 | Điểm học kỳ 2 (0–20) |
| G3 (Target) | Biến mục tiêu – điểm cuối kỳ của học sinh (0–20) |

1. Biến mục tiêu

+ “G3” là biến mục tiêu mà mô hình sẽ dự đoán. Đây là điểm cuối kỳ (0–20) phản ánh kết quả học tập tổng thể của học sinh.

+ Dự đoán chính xác giá trị G3 giúp nhà trường và giáo viên đưa ra các biện pháp hỗ trợ phù hợp để cải thiện kết quả học tập.

1. **Phân Tích Dữ Liệu Khám Phá**
2. Phân tích sơ bộ dữ liệu
   1. Thông tin Dataset

Xem qua 5 dòng đầu của dữ liệu

A screenshot of a computer

Description automatically generated

Dùng df.info() để xem dataset có dữ liệu nào, kiểu dữ liệu của nó, và có dữ liệu thiếu hay không?

A screenshot of a computer

Description automatically generated

* 1. Kiểm tra dữ liệu thiếu và trùng lặp

Kiểm tra lại dữ liệu thiếu:

A screen shot of a computer

Description automatically generated

Kiểm tra dữ liệu trùng lặp

A screen shot of a computer

Description automatically generated

Nhận xét: Có thể thấy ban đầu dữ liệu khá “sạch”, không cần phải xử lí dữ liệu thiếu hay trùng lập.

* 1. Thống kê mô tả

A screenshot of a computer program

Description automatically generated

16 cột kiểu số (int64): age, Medu, Fedu, traveltime, studytime, failures,...

17 cột kiểu object (dạng chuỗi/category):

school, sex, address, famsize, Pstatus, Mjob, Fjob, reason, guardian,...

Các biến yes/no: schoolsup, famsup, paid, activities, nursery,...

A screen shot of a computer

Description automatically generated

Nhận xét:

* Age (Tuổi)
* Phân bố: 15 - 22 tuổi
* Trung bình: ~16.7 tuổi
* Phần lớn học sinh: Từ 16-18 tuổi (25%-75%)
* Đặc biệt: Có một số ít học sinh lớn tuổi (max = 22)
* Studytime (Thời gian học)
* Thang đo: 1-4 (giờ/ngày)
* Trung bình: ~2.04 (mức trung bình)
* Phân bố chính: 1-2 giờ/ngày (50% học sinh)
* Nhận xét: Đa số học sinh dành thời gian học ở mức trung bình-thấp
* Failures (Số lần trượt)

Phạm vi: 0-3 lần

Đa số (75%): Chưa từng trượt (failures = 0)

Trung bình: ~0.33 lần

Ý nghĩa: Phần lớn học sinh có lịch sử học tập khá tốt

* Famrel (Mối quan hệ gia đình)

Thang đo: 1-5 (chất lượng)

Trung bình: ~3.94 (khá tốt)

Phân bố: Tập trung ở mức 4-5 (75% từ 4 trở lên)

Kết luận: Đa số học sinh có mối quan hệ gia đình tích cực

* Freetime (Thời gian rảnh)
* Thang đo: 1-5 (mức độ)
* Trung bình: ~3.24 (mức khá)
* Phân bố đồng đều từ 2-4
* Nhận xét: Học sinh có thời gian rảnh tương đối cân bằng
* Absences (Số buổi vắng)
* Phạm vi rộng: 0-75 buổi
* Trung bình: ~5.71 buổi
* 75% học sinh: Vắng ≤ 8 buổi
* Vấn đề: Có outliers - một số học sinh vắng mặt rất nhiều (max = 75)

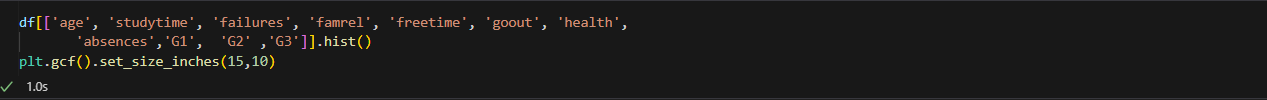
Nhận xét tổng quan: Dataset phản ánh một nhóm học sinh chủ yếu ở độ tuổi 16-18, có mối quan hệ gia đình tốt, ít trượt môn nhưng thời gian học ở mức trung bình. Điểm đáng chú ý là điểm số có xu hướng giảm và biến động tăng (std tăng) qua các thành phần, cùng với một số trường hợp vắng mặt quá mức cần được điều tra thêm.

A screenshot of a computer program

Description automatically generated

1. Phân tích đơn biến (Univariate Analysis)
2. 1. Biến số (Numerical)

Vẽ biểu đồ Histogram và tính hệ số Skewness để xem phân phối của các biến số:



A group of blue and white bars

Description automatically generated

Tính hệ số Skewness:

A black screen with orange and white lights

Description automatically generated

Nhận xét:

 + Lệch mạnh (|Skewness| > 1):

* absences (3.67) – Lệch phải mạnh
  + Phân phối tập trung ở giá trị thấp (0-10 buổi vắng)
  + Một số ít học sinh vắng mặt rất nhiều >30, >50 buổi (outliers)
* failures (2.39) – Lệch phải mạnh
  + Đa số học sinh không trượt môn nào (75% = 0)
  + Rất ít học sinh trượt nhiều lần
  + → Phản ánh đa số học sinh có học lực khá đến tốt

+ Lệch vừa (0.5 < |Skewness| < 1):

* famrel (-0.95) – Lệch trái
  + Phân phối nghiêng về điểm cao (4-5)
  + → Đa số học sinh có mối quan hệ gia đình tốt
* G3 (-0.73) – Lệch trái
  + Điểm số tập trung ở mức khá giỏi
  + → Có thể do đề thi dễ hoặc chất lượng học tập tốt
* studytime (0.63) – Lệch phải
  + Đa số học sinh học 1-2 giờ/ngày
  + Ít học sinh học nhiều giờ
  + → Thời gian học tập ở mức trung bình-thấp

Gần đối xứng (|Skewness| < 0.5)

* age (0.47), health (-0.49), G1 (0.24)
  + Phân phối tương đối cân đối
  + → Dữ liệu ổn định
* freetime (-0.16), goout (0.12)
  + Phân phối gần như chuẩn
  + → Phản ánh sự cân bằng trong cuộc sống học sinh

Vẽ biểu đồ Boxplot để kiểm tra giá trị ngoại lai (Outliers):

A screenshot of a computer screen

Description automatically generated

A screenshot of a graph

Description automatically generated

Nhận xét:

* Age (Tuổi):
* Phân phối chuẩn với đỉnh ở 16-17 tuổi
* Đối xứng quanh điểm trung tâm
* Có tồn tại ngoại lai nhưng không đáng kể
* → Độ tuổi điển hình của học sinh trung học
* Studytime (Thời gian học):
* Phân phối lệch phải rõ rệt
* Đa số học sinh học 1-2 giờ/ngày
* Có ngoại lai ở nhóm học nhiều giờ (4h)
* → Thời gian học tập ở mức khiêm tốn
* Failures (Số lần trượt):
* Phân phối cực lệch phải
* ~80% học sinh không trượt môn nào (0)
* Ngoại lai rõ rệt ở nhóm trượt từ 2 lần trở lên
* → Đa số học sinh có học lực tốt
* Health (Sức khỏe):
* Phân phối gần chuẩn với đỉnh ở mức 3-4
* Đối xứng tương đối
* Không có ngoại lai
* → Sức khỏe học sinh ở mức trung bình-khá
* Absences (Số buổi vắng):
* Phân phối cực lệch phải nghiêm trọng
* Đa số vắng dưới 10 buổi
* Nhiều ngoại lai nghiêm trọng vắng trên 30-70 buổi
* → Có vấn đề về chuyên cần với một nhóm nhỏ
* Điểm số (G1, G2, G3)
* G1: Phân phối tương đối đều, hơi lệch trái
* G2: Phân phối chuẩn hơn, tập trung quanh 10-11 điểm, có ngoại lai
* G3: Lệch trái rõ rệt - tập trung ở điểm cao (10-15)
* → Chất lượng học tập cải thiện qua các thành phần

Đề xuất xử lí:

* Các biến có ngoại lai sẽ được xử lí bằng Capping hoặc Biến đổi log/Square,…
* Riêng biến failures thì sẽ không xử lí vì:
  + Phản ánh chính xác thực trạng - đa số học sinh không trượt môn
  + Các giá trị >0 mang thông tin quan trọng về nhóm học sinh yếu
  + Biến đếm rời rạc nên outliers thực chất là các trường hợp thực tế
  1. Biến phân loại (Categorical)

Vẽ biểu đồ cột (bar chart) để xem tần suất của từng danh mục.

A computer screen shot of a code

Description automatically generated

A graph with blue rectangles

Description automatically generated

A graph with blue rectangular bars

Description automatically generated

A graph with blue rectangles

Description automatically generated

A graph with blue rectangles

Description automatically generated

A graph with blue rectangular bars

Description automatically generated with medium confidence

A graph with blue rectangles

Description automatically generated

Nhận xét:

* sex: Tỷ lệ nữ (F: 208) cao hơn nam (M: 187)
* Pstatus: Đa số sống cùng gia đình (T: 354 vs A: 41)
* paid: Đa số không tham gia lớp học thêm (no: 214)
* higher: Gần như tất cả muốn học cao hơn (yes: 375)
* romantic: Đa số không có quan hệ tình cảm (no: 263)

1. Phân tích đa biến (Multivariate Analysis)
2. 1. Biểu đồ scatter plot

A black screen with white text

Description automatically generated

Không có mối quan hệ rõ ràng

A graph with blue dots

Description automatically generated

age vs G3

* Mối quan hệ: Không có mối quan hệ rõ ràng.
* Nhận xét: Ở các độ tuổi phổ biến (15-18), điểm bao trùm toàn bộ phạm vi. Ở độ tuổi lớn hơn (19-22), số lượng học sinh ít hơn và điểm số dường như không đạt mức cao nhất (20), nhưng đây có thể là do kích thước mẫu nhỏ ở các nhóm tuổi này.

A graph of a study time

Description automatically generated

studytime vs G3

* Mối quan hệ: Không có mối quan hệ rõ ràng.
* Nhận xét: Dù có thể kỳ vọng thời gian học tập có tương quan dương, biểu đồ không cho thấy mối quan hệ này. Phạm vi điểm G3 là nhất quán ở tất cả các mức (từ 1 đến 4).

A graph with blue dots

Description automatically generated

famrel vs G3

* Mối quan hệ: Không có mối quan hệ rõ ràng.
* Nhận xét: Mức độ mối quan hệ gia đình từ 1 đến 5 đều có học sinh đạt điểm G3 từ 0 đến 20. Không có bằng chứng cho thấy mối quan hệ gia đình tốt hơn dẫn đến điểm G3 tốt hơn.

A graph of different sizes of dots

Description automatically generated with medium confidence

freetime vs G3

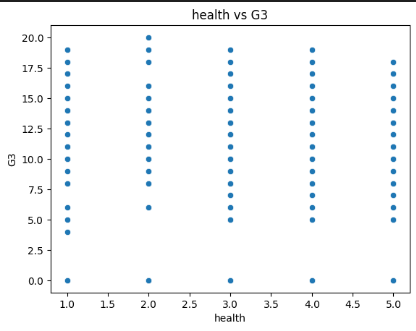
* Mối quan hệ: Không có mối quan hệ rõ ràng.
* Nhận xét: Tương tự, thời gian rảnh từ mức 1 đến 5 đều chứa toàn bộ phạm vi điểm G3 (từ 0 đến 20). Không có xu hướng cụ thể nào chỉ ra rằng nhiều hay ít thời gian rảnh sẽ ảnh hưởng đến điểm số cuối cùng.

A graph with blue dots

Description automatically generated

goout vs G3

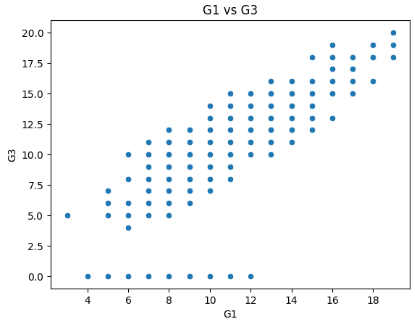
* Mối quan hệ: Không có mối quan hệ rõ ràng.
* Nhận xét: Mức độ đi chơi từ 1 đến 5 cũng không cho thấy ảnh hưởng rõ ràng đến điểm G3. Học sinh đi chơi nhiều hay ít đều có thể đạt được điểm tối đa (20) hoặc điểm thấp nhất (0).



health vs G3

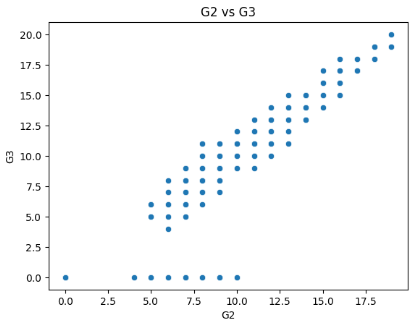
* Mối quan hệ: Không có mối quan hệ rõ ràng.
* Nhận xét: Tình trạng sức khỏe từ mức 1 đến 5 đều bao gồm toàn bộ phạm vi điểm.

Mối Quan hệ Tương quan Mạnh (Strong Correlation)



G1 vs G3

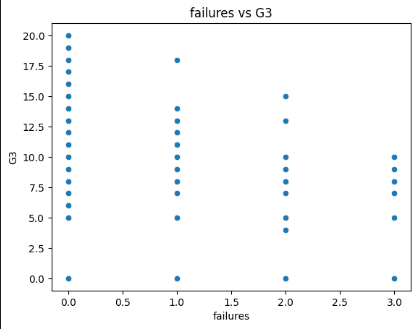
* Mối quan hệ: Tương quan dương mạnh.
* Nhận xét: Mặc dù không chặt chẽ như , G1 cho thấy một xu hướng tăng rõ ràng. Học sinh đạt điểm cao có xu hướng đạt điểm cao.



G2 vs G3

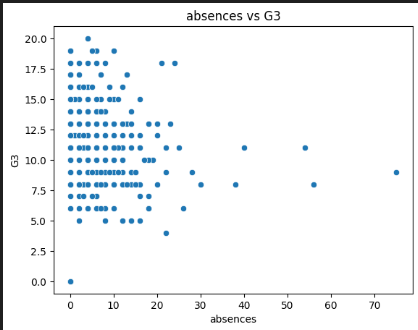
* Mối quan hệ: Tương quan dương rất mạnh và rõ ràng.
* Nhận xét: Các điểm dữ liệu tạo thành một dải hẹp dọc theo một đường chéo đi lên. Điều này cho thấy G2 là yếu tố dự đoán mạnh nhất cho G3. Học sinh đạt điểm G2 cao gần như chắc chắn sẽ đạt điểm G3 cao tương ứng.

Mối quan hệ Tương quan Yếu (Weak Correlation)



failures vs G3

* Mối quan hệ: Tương quan âm yếu.
* Nhận xét: Có xu hướng chung là khi số lần trượt tăng lên (từ 0 lên 3), phạm vi điểm G3 giảm xuống và điểm G3 tối đa cũng giảm xuống (ví dụ: điểm tối đa là 20 khi , nhưng chỉ khoảng 10 khi ). Tuy nhiên, tại mỗi mức , vẫn có sự phân tán lớn.



absences vs G3

* Mối quan hệ: Không rõ ràng/Tương quan rất yếu.
* Nhận xét: Mối quan hệ là một tập hợp các điểm phân tán với không có xu hướng tuyến tính rõ ràng. Tuy nhiên, khi số lần vắng rất cao (khoảng 40 đến 75), chỉ có một vài điểm dữ liệu và chúng nằm ở mức trung bình hoặc thấp.
  1. Vẽ ma trận tương quan

A screenshot of a computer

Description automatically generated

Mối Tương quan với G3

|  |  |  |  |
| --- | --- | --- | --- |
| **Biến** | **Hệ số tương quan ( với )** | **Mức độ tương quan** | **Nhận xét** |
| G1 | 0.8 | Rất mạnh (Dương) | Yếu tố dự đoán mạnh thứ hai. |
| G2 | 0.9 | Rất mạnh (Dương) | Yếu tố dự đoán mạnh nhất cho . |
| failures (Số lần trượt) | −0.4 | Trung bình (Âm | Càng trượt nhiều, càng thấp. |
| Medu (Học vấn mẹ) | 0.2 | Yếu (Dương) | Học vấn mẹ cao hơn có xu hướng đi kèm với G3 cao hơn một chút. |
| Fedu (Học vấn cha) | 0.2 | Yếu (Dương | Học vấn cha cao hơn có xu hướng đi kèm với G3 cao hơn một chút. |

Nhận xét chính:

* Các điểm kiểm tra G1, G2 là yếu tố dự đoán quan trọng nhất của điểm G3.
* Số lần trượt môn failures có mối tương quan âm trung bình, cho thấy đây là một yếu tố tiêu cực có ảnh hưởng đáng kể.
* Các yếu tố khác như age, traveltime, studytime, famrel, freetime, goout, Dalc, Walc, health, absences đều có tương quan rất yếu ( 0.1 hoặc 0.0 ) với G3.
* Kết quả từ ma trận tương quan này phù hợp với những gì đã quan sát được từ các biểu đồ phân tán trước đó:
* Ma trận xác nhận G1, G2 có tương quan mạnh nhất.
* Nó xác nhận rằng các yếu tố hành vi/xã hội như studytime, freetime, goout, health và có tương quan tuyến tính rất yếu với G3.
  1. So sánh phân phối của một biến số bằng boxplot

A black background with colorful text

Description automatically generated

A chart with blue squares

Description automatically generated

sex vs G3

|  |  |  |  |
| --- | --- | --- | --- |
| **Nhóm** | **Trung vị (Median)** | **IQR** | **Nhận xét** |
| F (Nữ) | ≈10 | 8 đến 13 | Điểm thấp hơn |
| M (Nam) | ≈11 | 9 đến 14 | Điểm cao hơn |

Kết luận: Học sinh nam có trung vị điểm cao hơn  so với học sinh nữ . Điều này cho thấy nam sinh có hiệu suất học tập cao hơn một chút.

A graph with blue and white lines

Description automatically generated

Pstatus vs G3

|  |  |  |  |
| --- | --- | --- | --- |
| **Nhóm** | **Trung vị (Median)** | **IQR** | **Nhận xét** |
| **A** (Sống xa nhau) | ≈11 | 9 đến 13 | Phạm vi điểm hẹp hơn |
| **T** (Sống cùng nhau) | ≈11 | 8 đến 14 | Phạm vi điểm rộng hơn |

Kết luận: Cả hai nhóm có trung vị gần như nhau. Nhóm có cha mẹ sống xa nhau có phạm vi hẹp hơn, nhưng không có sự khác biệt đáng kể về hiệu suất học tập.

A graph with blue and black lines

Description automatically generated

paid vs G3

|  |  |  |
| --- | --- | --- |
| **Nhóm** | **Trung vị (Median)** | **IQR** |
| **no** (Không) | ≈11 | ≈8.5 đến 14 |
| **yes** (Có) | ≈11 | ≈9 đến 13 |

Kết luận: Trung vị của cả hai nhóm là như nhau. Nhóm không học thêm (no) có và điểm tối đa cao hơn một chút, cho thấy việc học thêm (paid = yes) không đảm bảo điểm cao hơn.

A graph with blue squares and black text

Description automatically generated

activities vs G3

|  |  |  |
| --- | --- | --- |
| **Nhóm** | **Trung vị (Median)** | **IQR** |
| **no** (Không) | ≈11 | ≈8 đến 13 |
| **yes** (Có) | ≈11 | ≈9 đến 14 |

Kết luận: Nhóm tham gia hoạt động ngoại khóa (yes) có và cao hơn một chút so với nhóm không tham gia, cho thấy phân phối điểm của họ cao hơn một chút. Tuy nhiên, trung vị của cả hai nhóm là gần như nhau.

A graph with blue squares and black text

Description automatically generated

higher vs G3

|  |  |  |  |
| --- | --- | --- | --- |
| **Nhóm** | **Trung vị (Median)** | **IQR** | **Nhận xét** |
| yes | ≈11 | ≈9 đến 14 | Điểm cao hơn |
| no | ≈18 | ≈7.5 đến 10 | Điểm thấp hơn |

Kết luận: Nhóm học sinh có ý định học lên cao hơn (yes) có điểm cao hơn (trung vị là 11 so với 8) và phân phối điểm rộng hơn so với nhóm không có ý định. Sự khác biệt này là rõ ràng nhất trong tất cả các biểu đồ hộp được cung cấp.

A graph with blue squares

Description automatically generated with medium confidence

romantic vs G3

|  |  |  |
| --- | --- | --- |
| Nhóm | Trung vị (Median) | IQR |
| no (Không) | ≈11 | ≈9 đến 14 |
| yes (Có) | ≈11 | ≈8 đến 13 |

Kết luận: Trung vị của cả hai nhóm là như nhau. Nhóm không có mối quan hệ lãng mạn (no) có và điểm tối đa cao hơn một chút, nhưng sự khác biệt này là rất nhỏ và không có ý nghĩa lớn.

* 1. Kết luận chính từ EDA

1. Điểm số các thành phần trước là yếu tố dự đoán mạnh nhất:

* Điểm G1 và G2 có mối tương quan dương rất mạnh với điểm G3( r=0.8 và r=0.9). Điều này cho thấy thành tích học tập trước đó là yếu tố quyết định chính đối với kết quả cuối cùng.

2. Ý chí học tập và số lần trượt môn có tác động lớn:

* Học sinh có ý định học lên cao hơn (higher=yes) có điểm trung vị G3 cao hơn đáng kể (khoảng 11 so với 8) so với nhóm không có ý định.
* Số lần trượt môn (failures) có tương quan âm trung bình (r=-0.4) với . Càng trượt nhiều môn trong quá khứ, điểm cuối cùng càng có xu hướng thấp hơn.

3. Giới tính có sự khác biệt nhẹ; Yếu tố lối sống không có tương quan tuyến tính rõ ràng:

* Học sinh nam (M) có điểm trung vị G3 cao hơn (≈11) so với học sinh nữ (F) (≈10).
* Các yếu tố xã hội và hành vi như thời gian học tập (studytime), mối quan hệ gia đình (freetime), thời gian rảnh (health), sức khỏe (health), và việc học thêm có trả phí (paid) có rất ít hoặc không có tương quan tuyến tính với G3 (r ≈0.0 đến 0.1).
* Tóm lại: Hiệu suất học tập của học sinh được dự đoán chủ yếu bởi thành tích trước đó và động lực cá nhân (higher), trong khi các yếu tố môi trường và lối sống có tương quan tuyến tính rất yếu.

1. **Tiền Xử Lý Dữ Liệu**

Đầu tiên ta cần khai báo các thư viện cần thiết.

A screen shot of a computer program

AI-generated content may be incorrect.

1. Xóa cột không cần thiết

Bộ dữ liệu ban đầu gồm 33 biến (cột), bao gồm cả thông tin định tính và định lượng về học sinh như nhân khẩu học, hoàn cảnh gia đình, thói quen học tập và kết quả học tập. Tuy nhiên, không phải tất cả các biến đều cần thiết hoặc phù hợp để đưa vào mô hình dự đoán.

Tiêu chí lựa chọn biến:

+ Mức độ liên quan (Relevance):

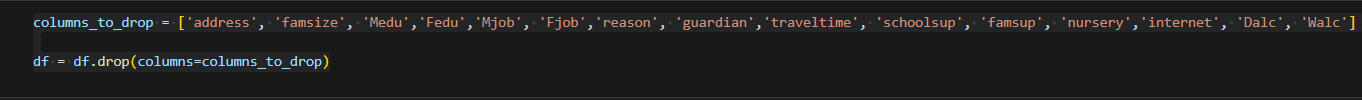
* Chỉ giữ lại những biến có tác động tiềm năng đến biến mục tiêu (G3) theo lý thuyết hoặc qua kiểm tra tương quan ban đầu.

+ Loại bỏ biến dư thừa hoặc trùng lặp thông tin:

* Các biến có nội dung tương đồng hoặc thể hiện cùng một đặc điểm (ví dụ: Medu và Fedu cùng mô tả trình độ học vấn cha mẹ) được loại bỏ để tránh đa cộng tuyến.

+ Loại bỏ biến không mang tính dự báo:

* Một số biến như địa chỉ, lý do chọn trường, người giám hộ, nghề nghiệp phụ huynh không thể hiện mối liên hệ rõ ràng với kết quả học tập và có thể gây nhiễu mô hình.
* Lợi ích của việc chọn 22 biến:
* Giúp mô hình gọn nhẹ, dễ huấn luyện và dễ diễn giải hơn.
* Giảm hiện tượng đa cộng tuyến giữa các biến.
* Tập trung vào những đặc trưng thực sự ảnh hưởng đến điểm cuối kỳ (G3).
* Cải thiện hiệu suất mô hình thông qua việc loại bỏ thông tin thừa.



Đây là các biến mang tính mô tả hoặc ít ảnh hưởng trực tiếp đến kết quả học tập, nên được loại bỏ nhằm giảm nhiễu và tăng hiệu quả mô hình.

1. Đưa các cột điểm về thang điểm 10

Để chuẩn hóa dữ liệu điểm số, bọn em đã tạo các cột điểm mới theo thang điểm 10 từ các cột điểm gốc theo thang điểm 20 ('G1', 'G2', 'G3'). Việc này giúp các điểm số dễ dàng được so sánh.

Code chuẩn hóa điểm về thang 10:

A computer screen with text and numbers

AI-generated content may be incorrect.

Quy trình thực hiện:

Hàm divide\_grades(X):

* df\_copy = X.copy(): Tạo một bản sao của DataFrame đầu vào (X) để thao tác, đảm bảo rằng dữ liệu gốc không bị thay đổi (giúp tránh các lỗi không mong muốn trong quá trình xử lý dữ liệu).
* Vòng lặp for g in ["G1", "G2", "G3"]: Lặp qua tên ba cột điểm cần xử lý.
* Điều kiện if g in df\_copy.columns and (g + "\_10") not in df\_copy.columns: Đảm bảo rằng cột điểm gốc (ví dụ: 'G1') tồn tại trong DataFrame và cột điểm mới (ví dụ: 'G1\_10') chưa được tạo. Điều kiện này giúp hàm có thể chạy lại an toàn mà không bị lỗi.
* Thực hiện chuyển đổi: df\_copy[g + "\_10"] = df\_copy[g] / 2.0
* Dòng này tạo ra một cột mới với tên thêm hậu tố \_10 (ví dụ: 'G1\_10') bằng cách lấy giá trị từ cột gốc và chia cho 2.0 (chuyển từ thang 20 sang thang 10).

Xóa cột cũ (Phòng ngừa):

* df = df.drop(columns=["G1\_10", "G2\_10", "G3\_10"], errors="ignore")
* Đoạn mã này đảm bảo xóa các cột thang điểm 10 đã tạo trước đó (nếu có), giúp quá trình chuẩn hóa được thực hiện lại chính xác. Tham số errors="ignore" đảm bảo chương trình không báo lỗi nếu các cột này chưa từng tồn tại.

Chạy hàm chuyển đổi:

* df = divide\_grades(df)
* Áp dụng hàm vừa định nghĩa vào DataFrame (df).

Kiểm tra kết quả:

* Lệnh cuối cùng hiển thị các cột điểm gốc và các cột điểm mới (thang 10) để xác nhận việc chuyển đổi đã thành công.
* Kết quả sau khi chạy code:
* A black and white screen with numbers

  AI-generated content may be incorrect.

1. Tính cột điểm trung bình của G1 và G2

Trong quá trình xử lý biến đầu vào, nhóm tạo thêm cột G\_Avg là điểm trung bình của G1\_10 và G2\_10 nhằm đại diện cho kết quả học tập trước đó của học sinh. Việc này giúp giảm hiện tượng đa cộng tuyến (multicollinearity) giữa các biến G1\_10, G2\_10 và biến mục tiêu , vì hai biến G1\_10 và G2\_10 có tương quan cao với nhau cũng như với G3\_10

A black screen with colorful text

Description automatically generated

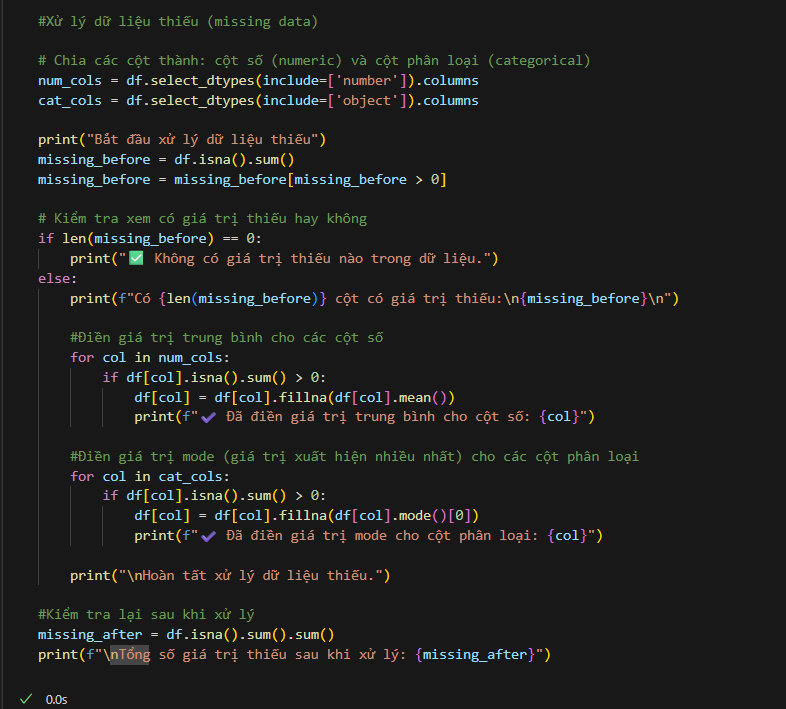
Kết quả sau khi chạy code

A screenshot of a computer

AI-generated content may be incorrect.

1. Xử lí dữ liệu thiếu

Bộ dữ liệu không có giá trị thiếu, tuy nhiên trong quy trình xử lý dữ liệu, nhóm vẫn tiến hành kiểm tra và trình bày phương pháp xử lý dữ liệu thiếu để đảm bảo tính đầy đủ của quy trình.



Quy trình thực hiện:

Phân loại các cột dữ liệu:

* Sử dụng lệnh select\_dtypes để tách riêng các cột kiểu số (num\_cols) và kiểu chuỗi/phân loại (cat\_cols).
* Việc phân loại giúp áp dụng cách xử lý phù hợp với từng loại dữ liệu.

Kiểm tra giá trị thiếu trước khi xử lý:

* Dùng df.isna().sum() để đếm số lượng giá trị bị thiếu ở mỗi cột.
* Nếu không phát hiện giá trị thiếu, chương trình in ra thông báo: “Không có giá trị thiếu nào trong dữ liệu.”
* Nếu có, chương trình sẽ hiển thị số lượng cột có giá trị thiếu và chi tiết từng cột.

Điền giá trị thay thế:

* Với các cột kiểu số, giá trị bị thiếu được thay thế bằng giá trị trung bình (mean) của cột đó.
* Lệnh: df[col] = df[col].fillna(df[col].mean())
  + Đồng thời in thông báo: “Đã điền giá trị trung bình cho cột số: [tên cột]”
  + Với các cột phân loại (chuỗi), giá trị bị thiếu được thay thế bằng mode (giá trị xuất hiện nhiều nhất) của cột.
* Lệnh: df[col] = df[col].fillna(df[col].mode()[0])
  + In thông báo: “Đã điền giá trị mode cho cột phân loại: [tên cột]”
  + Sau khi xử lý xong, chương trình in: “Hoàn tất xử lý dữ liệu thiếu.”

Kiểm tra lại sau khi xử lý:

* Dùng df.isna().sum().sum() để đếm lại tổng số giá trị thiếu trong toàn bộ dữ liệu sau khi xử lý.
* In ra kết quả: “Tổng số giá trị thiếu sau khi xử lý là: [số lượng].”

Đây là kết quả sau khi chạy code trên:

**A black screen with white text

Description automatically generated**

1. Xử lí ngoại lai (Outliers)

Khi phân tích các biến số được trực quan hóa bằng biểu đồ Boxplot để nhận diện các giá trị ngoại lệ (outliers). Kết quả cho thấy một số biến như age, absences, failures, famrel, studytime, freetime, G2\_10 (tương đương với G2) có xuất hiện các điểm nằm ngoài khoảng tứ phân vị (IQR). Tuy nhiên, nhóm không loại bỏ hay biến đổi các giá trị này mà sử dụng phương pháp Capping đây là kỹ thuật giới hạn các giá trị ngoại lệ ở mức ngưỡng nhất định (thường là theo percentile hoặc IQR rule).

Cụ thể:

Các giá trị quá thấp được thay bằng giới hạn dưới (Lower Cap).

Các giá trị quá cao được thay bằng giới hạn trên (Upper Cap).

Ví dụ:

Phương pháp này giúp giảm ảnh hưởng của outliers mà không làm mất dữ liệu, rất hữu ích trong các trường hợp dữ liệu có nhiễu lớn hoặc phạm vi biến động cao.

A screen shot of a computer program

AI-generated content may be incorrect.

Kết quả sau khi chạy code:

A screen shot of a computer

Description automatically generated

1. Mã hóa biến phân loại

Code mã hóa

A screen shot of a computer program

AI-generated content may be incorrect.

Quy trình thực hiện

+ Import công cụ LabelEncoder của thư viện scikit-learn — dùng để chuyển giá trị dạng chuỗi (như "yes", "no", "F", "M") thành số nguyên (như 0, 1).

* from sklearn.preprocessing import LabelEncoder

+ Danh sách các biến phân loại cần mã hóa:

* pre\_encoding = ['school', 'sex', 'Pstatus', 'paid', 'activities', 'higher', 'romantic']

+ Một số cột có thể bị loại trước đó → dòng này dùng để kiểm tra cột nào vẫn còn tồn tại trong DataFrame trước khi mã hóa (tránh lỗi khi cột không tồn tại).

* pre\_encoding\_present = [c for c in pre\_encoding if c in df.columns]

+ Tạo một dictionary để lưu lại toàn bộ thông tin mã hóa:

* encoders = {}

+ Chuẩn hóa dữ liệu:

* Ép kiểu thành str để đảm bảo mọi giá trị đều là chuỗi (tránh lỗi khi có NaN, None hoặc số).
* Dùng .str.strip() để xóa khoảng trắng dư thừa, đảm bảo "yes " và "yes" được xem là cùng một giá trị
* for col in pre\_encoding\_present:
* df[col] = df[col].astype(str).str.strip()

+ Khởi tạo và huấn luyện bộ mã hóa (fit\_transform) để chuyển đổi các giá trị duy nhất trong cột thành số nguyên. Kết quả được lưu vào cột mới có hậu tố \_encoded.

* Ví dụ: sex → sex\_encoded
* le = LabelEncoder()
* df[col + '\_encoded'] = le.fit\_transform(df[col])

+ Tạo bảng ánh xạ giữa nhãn gốc và giá trị mã hóa. Dòng này dùng để hiển thị hoặc lưu lại giúp giải thích kết quả sau này.

* Ví dụ: {'F': 0, 'M': 1}
* mapping = {k: int(v) for v, k in enumerate(le.classes\_)}

+ Lưu lại encoder và mapping của từng cột trong dictionary encoders

* encoders[col] = {"label\_encoder": le, "mapping": mapping}

+ In ra thông tin từng cột đã mã hóa

* print(f"Encoded {col} -> {col + 'encoded'} (classes: {list(le.classes\_)})")

Kết quả sau khi chạy code:

A computer screen with white text

AI-generated content may be incorrect.

1. Tạo pipeline chuẩn hóa dữ liệu

Code tạo pipeline chuẩn hóa

A screen shot of a computer

AI-generated content may be incorrect.

Pipeline giúp gộp nhiều bước xử lý liên tiếp (ở đây gồm chia điểm, cắt ngoại lai).

FunctionTransformer được dùng để tích hợp hàm tự định nghĩa (divide\_grades, cap\_outliers\_iqr\_np\_all) vào pipeline.

Cấu trúc này giúp dễ tái sử dụng, bảo trì và triển khai trong thực tế.

Sau này, chỉ cần gọi pipeline.transform (df\_moi) là có dữ liệu xử lý y hệt như tập train.

Chạy code tạo pipeline



Sau đó fit pipeline lên dữ liệu

A black background with white text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

fit\_transform vừa học các đặc điểm cần thiết (nếu có, ví dụ median, IQR...) vừa áp dụng biến đổi dữ liệu.

Kết quả fixed là DataFrame đã được chuẩn hóa sẵn (có thể dùng để train model).

1. Lưu các dữ liệu cần thiết

A screen shot of a computer program

AI-generated content may be incorrect.

1. Kiểm tra lại phân phối dữ liệu sau khi đã xử lí

Vẽ lại histogram:

A black screen with text

AI-generated content may be incorrect.

Vẽ lại boxplot:

A black screen with text on it

Description automatically generated

A group of blue squares

AI-generated content may be incorrect.

A group of blue squares

AI-generated content may be incorrect.

Kiểm tra luôn G\_Avg:

A computer screen with text

Description automatically generated

A diagram with blue lines

AI-generated content may be incorrect.

1. **Lựa Chọn & Huấn Luyện Mô Hình**
2. Các Mô hình Ứng viên

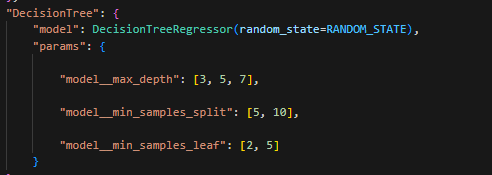
+ Linear Regression

A grey background with white text

Description automatically generated

* Mô tả: Mô hình hồi quy tuyến tính tìm đường thẳng tốt nhất mô tả mối quan hệ giữa các biến đầu vào (tuổi, số lần trượt, thời gian học, v.v.) và biến mục tiêu (điểm G3\_10)
* Là mô hình nền tảng, phổ biến và dễ diễn giải
* Dùng để kiểm tra xem dữ liệu có quan hệ tuyến tính với biến đầu ra hay không
* Là cơ sở để so sánh với các mô hình phi tuyến phức tạp hơn như Decision Tree hoặc Random Forest
* Ưu điểm:
  + Dễ hiểu, kết quả có thể diễn giải bằng hệ số hồi quy.
  + Chạy nhanh, ít yêu cầu tính toán.
  + Thích hợp với dữ liệu có quan hệ tuyến tính

+ Decision Tree Regressor



* Mô tả: Decision Tree chia dữ liệu thành các nút (nodes) dựa trên giá trị của đặc trưng sao cho sai số trong mỗi nhóm nhỏ nhất
* Quá trình này được lặp lại cho đến khi đạt được cây phân chia tối ưu hoặc đạt giới hạn độ sâu (max\_depth)
* Có thể mô hình hóa mối quan hệ phi tuyến và tương tác phức tạp giữa các đặc trưng
* Dễ trực quan hóa và giải thích quy tắc phân chia.
* Ưu điểm:
  + Không cần chuẩn hoá dữ liệu
  + Dễ hiểu, dễ vẽ thành sơ đồ cây

+ Random Forest Regressor

A computer screen with text

AI-generated content may be incorrect.

* Mô tả: Random Forest là mô hình tập hợp gồm nhiều cây quyết định.
* Mỗi cây được huấn luyện trên một tập con ngẫu nhiên của dữ liệu và đặc trưng.
* Kết quả cuối cùng là trung bình dự đoán của toàn bộ các cây
* Giải quyết các nhược điểm của Decision Tree (giảm overfitting, tăng ổn định).
* Có khả năng mô hình hoá quan hệ phi tuyến, tương tác đặc trưng và nhiễu trong dữ liệu rất tốt.
* Thường cho hiệu suất cao nhất trong các mô hình dạng tabular (bảng dữ liệu).
* Ưu điểm:
  + Hiệu suất cao, ổn định, ít bị overfitting.
  + Làm việc tốt với dữ liệu chứa đặc trưng cả số và phân loại.
  + Có thể đo được tầm quan trọng của đặc trưng (feature importance)

1. ****Thước đo đánh giá****

Vì bài toán thuộc loại hồi quy và biến đầu ra là điểm số thực liên tục, nên nhóm không thể dùng các chỉ số phân loại như Accuracy, Precision, Recall

Thay vào đó, nhóm sử dụng các thước đo sai số hồi quy, bao gồm RMSE, và R², trong đó RMSE và R² là hai chỉ số chính được dùng trong báo cáo

**+ RMSE**

* RMSE được sử dụng để đo sai lệch trung bình giữa giá trị dự đoán và giá trị thực tế.
* Chỉ số này có cùng đơn vị với biến mục tiêu, do đó giúp việc diễn giải kết quả trở nên dễ hiểu và trực quan hơn.
* RMSE nhạy cảm với các sai lệch lớn vì sai số được bình phương trước khi tính trung bình. Điều này khiến mô hình phải tối ưu sao cho không chỉ chính xác với đa số dữ liệu mà còn hạn chế sai lệch nghiêm trọng đối với một số trường hợp.
* RMSE càng nhỏ thể hiện mô hình dự đoán càng sát với giá trị thực tế.

**+ R²**

* R² thể hiện tỷ lệ phần trăm phương sai của biến mục tiêu được mô hình giải thích thông qua các đặc trưng đầu vào.
* Chỉ số này cho biết mô hình có nắm bắt được xu hướng và cấu trúc thực tế trong dữ liệu hay không.
* R² càng cao chứng tỏ mô hình càng hiểu rõ mối quan hệ giữa các biến đầu vào và đầu ra, đồng thời có khả năng khái quát tốt hơn.
* R² giúp đánh giá mức độ phù hợp tổng thể của mô hình và cho phép so sánh hiệu quả giữa các mô hình hồi quy khác nhau.

1. ****Huấn luyện****

**A screenshot of a computer program

Description automatically generated**

**Ở phần đầu của chương trình, import các thư viện cần thiết để xử lý dữ liệu và xây dựng mô hình dự đoán điểm thi cuối kỳ cho học sinh.**

**Đầu tiên là NumPy. Thư viện này giúp làm việc với các phép tính toán học và mảng số học — ví dụ như khi cần tính sai số trung bình bình phương căn (RMSE), sẽ dùng NumPy để tính căn bậc hai.**

**Tiếp theo là Pandas,. Pandas giúp xử lý dữ liệu ở dạng bảng, dùng Pandas để đọc dữ liệu từ file CSV, lọc các cột đặc trưng, và xem vài dòng đầu tiên để kiểm tra dữ liệu có hợp lý hay chưa.**

**Sau đó là Joblib. Đây là thư viện giúp em lưu lại mô hình sau khi đã huấn luyện xong, ví dụ như lưu mô hình tốt nhất vào file best\_model.pkl. Nhờ vậy, khi cần dự đoán lại thì không phải train lại từ đầu nữa, tiết kiệm rất nhiều thời gian.**

**Tiếp theo, em import các hàm trong thư viện scikit-learn, cụ thể là từ module model\_selection.**

**Trong đó có:**

**train\_test\_split dùng để chia dữ liệu thành tập huấn luyện và tập kiểm tra.**

**KFold để thực hiện Cross-Validation, tức là chia dữ liệu huấn luyện thành nhiều phần nhỏ để đánh giá mô hình ổn định hơn.**

**Và GridSearchCV để tự động tìm tham số tối ưu cho mô hình bằng cách thử nhiều giá trị khác nhau và chọn ra cấu hình tốt nhất.**

**Tiếp đến, import bốn mô hình học máy sẽ dùng để so sánh:**

* **DummyRegressor**
* **LinearRegression**
* **DecisionTreeRegressor**
* **RandomForestRegressor**

**Cuối cùng, import hai hàm đánh giá là mean\_squared\_error và r2\_score. Hai hàm này giúp đo xem mô hình dự đoán có tốt không.**

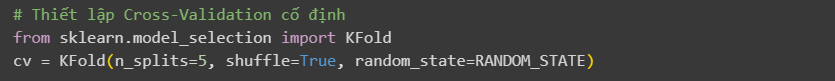
****

Đọc file dữ liệu đã được tiền xử lý là **processed.csv. Việc tiền xử lý nằm ở bước trước đó, phần này tập trung vào huấn luyện mô hình**

****

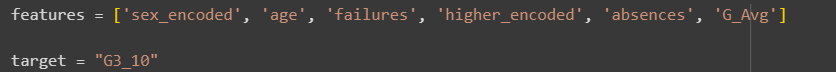
**đặt RANDOM\_STATE = 42 để cố định ngẫu nhiên, giúp quá trình huấn luyện và đánh giá mô hình có thể tái lập được kết quả.**

**Nếu không đặt, mỗi lần chạy mô hình có thể cho ra kết quả hơi khác nhau do ngẫu nhiên trong việc chia dữ liệu hoặc huấn luyện rừng cây.**

****

**Dùng KFold Cross Validation với 5 phần, có xáo trộn dữ liệu và cố định seed để đảm bảo chia ngẫu nhiên nhưng tái lập được.**

**Cách này giúp đánh giá mô hình ổn định và khách quan hơn, giảm rủi ro do ngẫu nhiên khi chia train/test.**

****

Ở bước này chọn các đặc trưng đầu vào gồm thông tin cá nhân và những yếu tố có ảnh hưởng rõ rệt đến điểm cuối kỳ.

****

Chọn tỷ lệ 80/20 vì đây là mức phổ biến giúp cân bằng giữa việc có đủ dữ liệu để huấn luyện và đủ mẫu để kiểm tra độ tổng quát.

Ngoài ra đặt random\_state=42 để đảm bảo mỗi lần chạy code đều cho kết quả chia dữ liệu giống nhau, giúp tái lập kết quả khi cần.

**A computer screen shot of text

Description automatically generated**

**Bắt đầu khởi tạo mô hình baseline và các mô hình ứng viên**

**Đầu tiên là mô hình Baseline, dùng DummyRegressor với chiến lược mean.**

**Nghĩa là mô hình này chỉ đơn giản đoán trung bình điểm của toàn bộ dữ liệu.**

**Mục đích của nó là tạo ra một mốc tham chiếu cơ bản – để kiểm tra xem các mô hình học máy phức tạp hơn có thực sự tốt hơn việc chỉ đoán trung bình hay không.**

**Tiếp theo là mô hình Linear Regression, hay hồi quy tuyến tính.**

**Mô hình này giả định rằng điểm thi cuối kỳ có mối quan hệ tuyến tính với các yếu tố đầu vào như tuổi, số lần trượt môn, hay điểm trung bình trước đó.**

**Với Linear Regression thì không có tham số cần tối ưu, nên phần params của nó để trống.**

**Mô hình thứ ba là Decision Tree, hay cây quyết định.**

**Đây là mô hình có khả năng học được các mối quan hệ phi tuyến giữa các đặc trưng.**

**Ví dụ như: “Nếu học sinh vắng nhiều buổi và có nhiều lần trượt môn thì điểm thi có thể thấp hơn.”**

**Đặt các tham số cần tìm là:**

* **max\_depth: độ sâu tối đa của cây, thử các giá trị 3, 5 và 7.**
* **min\_samples\_split: số lượng mẫu tối thiểu để chia một nút, thử 5 và 10.**
* **min\_samples\_leaf: số lượng mẫu tối thiểu ở mỗi lá, thử 2 và 5.**

**Việc này giúp tránh overfitting, tức là mô hình học quá kỹ trên dữ liệu train mà mất khả năng tổng quát hóa.**

**Cuối cùng là mô hình Random Forest, hay còn gọi là rừng ngẫu nhiên.**

**Đây là mô hình tổng hợp nhiều cây quyết định nhỏ, mỗi cây học trên một phần dữ liệu khác nhau, sau đó tất cả cùng dự đoán và lấy trung bình.**

**Mô hình này thường cho kết quả ổn định và chính xác cao hơn so với một cây đơn lẻ.**

**Với Random Forest, cũng khai báo các tham số cần tối ưu:**

* **max\_depth: giới hạn độ sâu của mỗi cây, thử 3 và 7.**
* **min\_samples\_leaf: số mẫu tối thiểu trên mỗi lá, thử 5, 10 và 20.**
* **n\_estimators: số lượng cây trong rừng, thử 50 và 100.**

**Như vậy, đoạn code này giúp định nghĩa toàn bộ các mô hình cùng với bộ tham số cần thử nghiệm.**

**Sau đó, đưa tất cả những mô hình này vào GridSearchCV để tự động tìm ra mô hình tốt nhất và tham số tối ưu nhất dựa trên quá trình cross-validation.**



Danh sách này được dùng để **lưu lại kết quả huấn luyện tốt nhất của từng mô hình**

**A black rectangular object with white lines

Description automatically generated**

**Danh sách này được dung để lưu lại thước đo đánh giá của từng mô hình**

A screen shot of a computer

Description automatically generated

**Sau khi đã khai báo xong các mô hình cùng với các tham số cần tối ưu, ở đoạn code này bắt đầu huấn luyện từng mô hình một cách tự động.**

**Cụ thể, dùng một vòng for để lần lượt duyệt qua từng mô hình trong dictionary models.**

**Ở mỗi lượt, in ra tên mô hình đang được huấn luyện, ví dụ như Training LinearRegression hay Training RandomForest.**

**Bên trong vòng lặp, sử dụng hàm GridSearchCV để tìm tham số tốt nhất cho mô hình đó.**

**Truyền vào ba thứ:**

* **Mô hình (config["model"]),**
* **Tập tham số cần thử (config["params"]),**
* **Và biến cv, chính là 5-Fold Cross Validation đã định nghĩa trước đó.**

**Chọn thước đo đánh giá là neg\_mean\_squared\_error.**

**Đây là sai số trung bình bình phương (MSE) nhưng lấy giá trị âm vì thư viện scikit-learn mặc định muốn điểm càng cao càng tốt.**

**Như vậy, MSE càng nhỏ thì điểm này càng lớn.**

**Khi chạy lệnh grid.fit(X\_train, y\_train), chương trình sẽ tự động:**

**Chia tập huấn luyện thành 5 phần nhỏ,**

**Lần lượt huấn luyện và kiểm tra mô hình trên từng phần,**

**Tính sai số trung bình qua 5 lần,**

**Rồi chọn ra bộ tham số tối ưu nhất giúp sai số nhỏ nhất.**

**Sau khi có mô hình tốt nhất từ cross-validation, dùng grid.predict(X\_test) để dự đoán trên tập test – đây là tập dữ liệu mà mô hình chưa từng thấy.**

**Tiếp theo, tính hai chỉ số đánh giá quan trọng:**

* **rmse – sai số trung bình bình phương căn, giúp hiểu trung bình mô hình dự đoán lệch bao nhiêu điểm so với thực tế.**
* **r2 – hệ số xác định, cho biết mô hình giải thích được bao nhiêu phần trăm sự biến thiên của điểm số học sinh.**

**Lưu lại các kết quả này vào hai nơi:**

**Một danh sách best\_models để tiện so sánh mô hình nào tốt nhất,**

**Và một từ điển models\_metrics để lưu chi tiết hơn, gồm tham số tốt nhất (best\_params), RMSE, R² và mô hình đã huấn luyện (best\_estimator).**

**Nhờ cách làm này có thể so sánh khách quan tất cả các mô hình trên cùng một tập dữ liệu, dưới cùng điều kiện cross-validation.**

**Kết quả cuối cùng sẽ cho biết mô hình nào hoạt động tốt nhất – ví dụ như Random Forest có RMSE thấp nhất và R² cao nhất – và có thể lưu lại để triển khai hoặc dự đoán các điểm thi mới.”**

A screenshot of a computer program

Description automatically generated

Sau khi huấn luyện xong thì in ra kết quả của các mô hình

A screenshot of a computer program

Description automatically generated

Chuyển sang dạng bảng để dễ dàng quan sát hơn

A screenshot of a computer program

Description automatically generated

In ra mô hình tối ưu và các siêu tham số của mô hình

A screen shot of a computer

Description automatically generated

Lưu lại mô hình tối ưu bằng thư viện joblib

1. **Phân Tích Kết Quả**
2. ****Bảng so sánh kết quả metrics****

A screenshot of a graph

AI-generated content may be incorrect.

A graph with blue squares

Description automatically generated

Chỉ số RMSE nêu ra nếu RMSE càng thấp thì tương đương giá trị dự đoán gần bằng giá trị thực tế,mang tính đúng đắn cao.

A graph of different colored squares

Description automatically generated with medium confidence

Chỉ số R^2 chỉ ra rằng nếu R^2 càng bằng 1 thì mô hình càng giải thích được sự biến thiên của biến mục tiêu (G3) .

Cho thấy kết quả rằng: model Decision Tree là model tốt nhất với chỉ số RMSE thấp nhất ( 0,793607 ) đồng thời cả chỉ số R^2 là cao nhất ( 0,87714 ) .

1. ****Các biến ảnh hưởng đến kết quả****

A screenshot of a computer

Description automatically generated

A graph with a bar graph

Description automatically generated

Kết quả cho thấy : có 4 giá trị ảnh hưởng tới kết quả là G\_AVG(điểm trung bình kỳ trước),absences (số ngày nghỉ), age (tuổi), sex (giới tính). Trong đó biến có tác động cao nhất là biến G\_AVG với giá trị 0.876681 hay 87,67% tác động mạnh mẽ lên G3,phản ánh trực tiếp năng lực học tập hiện tại của học sinh.

1. ****Hạn chế và Hướng phát triển:****

Hạn chế:

* Phụ thuộc quá nhiều vào G\_Avg
* → Mô hình gần như chỉ dựa vào điểm trung bình quá khứ mà bỏ qua yếu tố hành vi hoặc động lực học tập.
* Dữ liệu có thể mất cân đối hoặc ít mẫu
* → Nếu số lượng học sinh ít hoặc phân bố điểm lệch, mô hình dễ bị overfitting (học thuộc dữ liệu cũ).
* Thiếu các biến mô tả đặc tính học sinh khác
* → Chưa có dữ liệu về phương pháp học, mức độ tham gia hoạt động, điều kiện gia đình,... → mô hình chưa toàn diện.

Phát triển:

* Bổ sung và làm giàu dữ liệu đầu vào
* → Thu thập thêm các yếu tố về hành vi học tập, thái độ, phương pháp học, môi trường gia đình, hoặc kết quả kiểm tra khác.
* Cân bằng và mở rộng tập dữ liệu
* → Thu thập thêm dữ liệu từ nhiều học kỳ hoặc nhiều trường khác nhau để tăng khả năng tổng quát của mô hình.
* Mở rộng thử nghiệm trên nhiều model khác
* → Đánh giá khách quan hơn cho các biến độc lập.

1. **Xây dựng và triển khai mô hình chạy trên Streamlit**

Code Python dùng để xây dựng và chạy ứng dụng web bằng Streamlit.

A screen shot of a computer program

Description automatically generated

**A screen shot of a computer program

Description automatically generated**

Sau đó lưu file, đặt tên file là app.py

Bấm chạy code và nhập: “Streamlit run app.py”



Sau khi chạy lệnh: streamlit run app.py, hệ thống hiển thị thông báo như hình.

A screen shot of a computer

AI-generated content may be incorrect.

Điều này cho biết ứng dụng đã được khởi động thành công và có thể truy cập qua trình duyệt web tại:

* Local URL: http://localhost:8501 – dùng để chạy trên máy cá nhân.
* Network URL: http://192.168.1.234:8501 – cho phép truy cập từ các thiết bị khác trong cùng mạng nội bộ.

Ảnh này thể hiện giao diện ứng dụng Streamlit sau khi chạy thành công

**A screenshot of a computer screen

Description automatically generated**

Mô tả:

* Ứng dụng được xây dựng bằng Streamlit, cho phép người dùng nhập các thông tin của học sinh như: Giới tính (Gender), tuổi (Age), số môn trượt (Past Failures), số buổi nghỉ (Absences), điểm trung bình (Average Grade)
* Sau khi nhấn Predict Score, hệ thống sẽ dự đoán điểm môn cuối kỳ (thang 0–10) dựa trên mô hình học máy

**A screenshot of a computer screen

Description automatically generated**

Kết quả dự đoán điểm thi (Student Exam Score Predictor)

Mô tả:

* Người dùng nhập thông tin học sinh gồm: giới tính, tuổi, số môn trượt, số buổi nghỉ, có học thêm hay không, điểm trung bình.
* Sau khi nhấn Predict Exam Score, hệ thống hiển thị:
  + Kết quả dự đoán: điểm thi cuối kỳ được mô hình dự đoán — ví dụ 8.7/10 trong hình.
* Ứng dụng giúp giáo viên, phụ huynh hoặc nhà quản lý giáo dục ước lượng trước kết quả học tập của học sinh dựa trên dữ liệu thực tế