

TRƯỜNG ĐẠI HỌC KỸ THUẬT – CÔNG NGHỆ CẦN THƠ

KHOA CÔNG NGHỆ THÔNG TIN

PHÂN TÍCH HIỆU XUẤT HỌC TẬP CỦA SINH VIÊN

NGÀNH: KHOA HỌC DỮ LIỆU

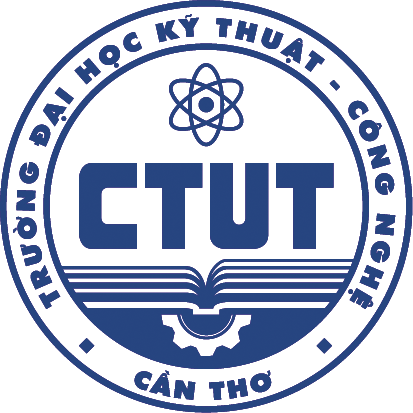
GIẢNG VIÊN HƯỚNG DẪN: Lê Anh Nhã Uyên

SINH VIÊN THỰC HIỆN:

Huỳnh Chí Phi Thuận MSSV: KHDL2211038

Cần Thơ, tháng 8 năm 2024

ĐỒ ÁN 3



PHÂN TÍCH HIỆU XUẤT HỌC TẬP CỦA SINH VIÊN

NGÀNH: KHOA HỌC DỮ LIỆU

GIẢNG VIÊN HƯỚNG DẪN: Lê Anh Nhã Uyên

SINH VIÊN THỰC HIỆN:

Huỳnh Chí Phi Thuận MSSV: KHDL2211038

TRƯỜNG ĐẠI HỌC KỸ THUẬT – CÔNG NGHỆ CẦN THƠ

KHOA CÔNG NGHỆ THÔNG TIN

Cần Thơ, tháng 8 năm 2024

ĐỒ ÁN 3

PHIẾU NHẬN XÉT GIẢNG VIÊN HƯỚNG DẪN

**Tên đề tài:**

PHƯƠNG PHÁP NGHIÊN CỨU PHÂN LOẠI ÂM THANH BẰNG CNN

Tên SVTH: HUỲNH CHÍ PHI THUẬN MSSV: KHDL2211038

**Họ và tên GVHD:** ThS. HÀ LÊ NGỌC DUNG

**Nhận xét của giảng viên hướng dẫn:**

·································································································

·································································································

·································································································

·································································································

·································································································

·································································································

·································································································

·································································································

·································································································

·································································································

Cần thơ, ngày...tháng...năm 2024

**GIẢNG VIÊN HƯỚNG DẪN**

ThS. HÀ LÊ NGỌC DUNG

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

Giảng viên phản biện:

**Nhận xét của giảng viên phản biện:**

·································································································

·································································································

·································································································

·································································································

·································································································

·································································································

·································································································

·································································································

·································································································

·································································································

Cần thơ, ngày...tháng...năm 2024

GIẢNG VIÊN PHẢN BIỆN

LỜI CAM ĐOAN

~~Chúng tôi bao gồm Vũ Huy, Phi Thuận, Văn Tài xin cam đoan rằng đề tài “Phương pháp nghiên cứu phân loại âm thanh bằng CNN” trong đồ án của chúng tôi đã được tiến hành một cách công khai và minh bạch, dựa trên sự cố gắng và nỗ lực của bản thân cũng như sự hướng dẫn tận tình từ giảng viên ThS. Hà Lê Ngọc Dung~~

~~Các số liệu nghiên cứu được trình bày trong đồ án đảm bảo tính trung thực, không có việc sao chép hay sử dụng kết quả từ bất kỳ công trình nào đã được công bố trước đó. Tôi cam đoan rằng nếu phát hiện có sự sao chép, chúng tôi sẽ chịu hoàn toàn trách nhiệm và sẵn sàng chấp nhận kỷ luật từ phía nhà trường.~~

Cần Thơ, ngày…tháng…năm 2024

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành đến giảng viên Lê Anh Nhã Uyên Khoa Công nghệ thông tin đã trang bị cho em những kiến thức, kỹ năng cơ bản cần có để hoàn thành đề tài nghiên cứu này.

Tuy nhiên, trong quá trình nghiên cứu đề tài, do kiến thức chuyên ngành còn hạn chế nên em vẫn còn nhiều thiếu sót khi tìm hiểu, đánh giá và trình bày về đề tài. Rất mông nhận được sự quan tâm, góp ý của thầy/ cô giảng viên trong khoa để đề tài của em được đầy đủ và hoàn chỉnh hơn.

Xin chân thành cảm ơn.

Cần Thơ, ngày…tháng…năm 2024

TÓM TẮT ĐỒ ÁN

Đề tài nghiên cứu phương pháp phân loại âm thanh bằng mạng nơ-ron tích chập (CNN) nhằm khai thác tiềm năng của CNN trong việc xử lý và nhận diện các mẫu âm thanh phức tạp. Với sự phát triển mạnh mẽ của công nghệ xử lý âm thanh và nhu cầu ngày càng tăng về các hệ thống phân loại âm thanh chính xác trong các ứng dụng như trợ lý ảo, hệ thống điều khiển bằng giọng nói và phân tích âm nhạc, nghiên cứu này tập trung vào việc cải tiến các phương pháp hiện có và khám phá các kỹ thuật mới để nâng cao hiệu quả và độ chính xác của quá trình phân loại âm thanh.

MỤC LỤC

[MỤC LỤC i](#_Toc205478530)

[DANH MỤC HÌNH iii](#_Toc205478531)

[DANH MỤC TỪ VIẾT TẮT iv](#_Toc205478532)

[CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI 1](#_Toc205478533)

[1.1. Lý Do Chọn Đề Tài 1](#_Toc205478534)

[1.2. Mục Tiêu Nghiên Cứu 1](#_Toc205478535)

[1.3. Phương Pháp Nghiên Cứu 2](#_Toc205478536)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 3](#_Toc205478537)

[2.1. Tìm Hiểu Về Máy Học 3](#_Toc205478538)

[2.1.1. Máy học là gì ? 3](#_Toc205478539)

[2.1.2. Lịch Sử Hình Thành Máy Học 4](#_Toc205478540)

[2.1.3. Phân Loại Máy Học 6](#_Toc205478541)

[2.1.4. Cách hoạt động của thuật toán trong máy học: 7](#_Toc205478542)

[2.2. Thuật toán CatBoost 10](#_Toc205478543)

[2.2.1. Các Lợi Ích Chính của CatBoost: 11](#_Toc205478544)

[2.2.2. Cơ Chế Hoạt Động của CatBoost 11](#_Toc205478545)

[2.2.3. Các Đặc Trưng Nổi Bật của CatBoost: 12](#_Toc205478546)

[2.2.4. Tham Số và Siêu Tham Số (Hyperparameters) của CatBoost 13](#_Toc205478547)

[2.2.5. Tinh chỉnh siêu tham số (Hyperparameter Tuning): 15](#_Toc205478548)

[2.2.6. Ứng dụng của CatBoost: 15](#_Toc205478549)

[2.2.7. Hạn chế của CatBoost: 15](#_Toc205478550)

[2.2.8. So sánh CatBoost với các thuật toán Boosting khác: 16](#_Toc205478551)

[2.2.9. Cải thiện mô hình: 17](#_Toc205478552)

[CHƯƠNG 3: TIỀN XỬ LÝ VÀ TRỰC QUANG HÓA DỮ LIỆU SINH VIÊN 17](#_Toc205478553)

[3.1. Dữ liệu sinh viên 17](#_Toc205478554)

[CHƯƠNG 4: XÂY DỰNG MÔ HÌNH MÁY HỌC DỰ ĐOÁN HIỆU XUẤT HỌC TẬP 21](#_Toc205478555)

[CHƯƠNG 5: KẾT LUẬN 21](#_Toc205478556)

[5.1. Hiểu Biết về Phân Loại Âm Thanh và CNN 21](#_Toc205478557)

[5.2. Xây dựng và Đánh Giá Mô Hình CNN 22](#_Toc205478558)

[5.3. Ứng Dụng và Tinh Chỉnh 23](#_Toc205478559)

[5.4. ĐÁNH GIÁ KẾT QUẢ 23](#_Toc205478560)

[5.4.1. Đánh giá triển khai tổng quan 23](#_Toc205478561)

[5.4.2. Đề Xuất Cải Tiến 24](#_Toc205478562)

[5.5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 25](#_Toc205478563)

[5.5.1. Kết Luận 25](#_Toc205478564)

[5.5.2. Hướng Phát Triển 25](#_Toc205478565)

[5.6. HẠN CHẾ 26](#_Toc205478566)

[TÀI LIỆU KHAM KHẢO 26](#_Toc205478567)

DANH MỤC HÌNH

[Hình 2.1: Học máy (Machine Learning) 3](#_Toc174565942)

[Hình 2.2: Quá trình phát triển của máy học 5](#_Toc174565943)

[Hình 2.3: Phân loại trong máy học 7](#_Toc174565944)

[Hình 2.4: Quá trình hoạt động của máy học 9](#_Toc174565945)

[Hình 2.5: Quá trình hoạt động của học sâu 11](#_Toc174565946)

[Hình 2.6: Lớp tích chập 13](#_Toc174565947)

[Hình 2.7: Lớp gộp 14](#_Toc174565948)

[Hình 2.8: Lớp kết nối đầy đủ 14](#_Toc174565949)

[Hình 2.9: Mô hình CNN hoạt động 15](#_Toc174565950)

[Hình 3.1: Minh họa về âm thanh 24](#_Toc174565951)

[Hình 3.2: minh họa quá trình số hóa âm thanh 26](#_Toc174565952)

[Hình 3.3: Phép toán DFT thuận 28](#_Toc174565953)

[Hình 3.4: Phép toán DFT nghịch 28](#_Toc174565954)

[Hình 3.5: Tín hiệu trong miền thời gian P(t) được biểu diễn trên miền tần số P(v) nhờ phép biến đổi Fourier rời rạc 29](#_Toc174565955)

[Hình 3.6: STFT thời gian liên tục 29](#_Toc174565956)

[Hình 3.7: STFT thời gian rời rạc 30](#_Toc174565957)

[Hình 3.8: STFT sử dụng phân tích tín hiệu âm thanh theo thời gian 30](#_Toc174565958)

[Hình 3.9: Biểu diễn trực quang bằng Spectrogram. 31](#_Toc174565959)

[Hình 3.10: Phát hiện âm thanh bất thường với 34](#_Toc174565960)

[Hình 4.1: Hàm Tạo Spectrogram Từ Tệp WAV 40](#_Toc174565961)

[Hình 4.2: Trực quan Spectrogram trên đồ thị 44](#_Toc174565962)

[Hình 4.3: giá trị Decibel Scale của một số loại âm thanh khác nhau 45](#_Toc174565963)

[Hình 4.4: quá trình học đặc trưng (feature learning) cho âm thanh 51](#_Toc174565964)

[Hình 4.5: Các thư viện cần thiết để xử lý dữ liệu âm thanh, tạo spectrograms, và huấn luyện mô hình CNN 53](#_Toc174565965)

[Hình 4.6: Quá trình train mô hình 63](#_Toc174565966)

[Hình 4.7: Test mô hình 64](#_Toc174565967)

DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **TT** | **Ký hiệu chữ viết tắt** | **Chữ viết tắt** |
| 1 | CNN | Convolutional Neural Network |
| 2 | AI | Artificial Intelligence |
| 3 | LSTM | Long Short-Term Memory |
| 4 | RNN | Recurrent Neural Network |
| 5 | GPU | Tensor Processing Unit |
| 6 | IDE | Integrated Development Environment |
| 7 | WAV | Waveform Audio File Format |
| 8 | FLAC | Free Lossless Audio Codec |
| 9 | TPU | Tensor Processing Unit |

# GIỚI THIỆU ĐỀ TÀI

## Lý Do Chọn Đề Tài

Hiệu suất học tập của sinh viên là một trong những yếu tố then chốt phản ánh chất lượng đào tạo của một cơ sở giáo dục. Trong bối cảnh giáo dục hiện đại, việc ứng dụng công nghệ và trí tuệ nhân tạo để hỗ trợ quản lý, giảng dạy và học tập ngày càng trở nên cần thiết. Tuy nhiên, nhiều sinh viên vẫn đang gặp khó khăn trong quá trình học tập mà nhà trường chưa thể phát hiện và hỗ trợ kịp thời.

Với sự phát triển mạnh mẽ của khoa học dữ liệu và học máy, việc khai thác dữ liệu học tập để xây dựng các mô hình dự đoán hiệu suất học tập có thể giúp nhà trường phát hiện sớm những sinh viên có nguy cơ tụt hậu hoặc bỏ học. Điều này giúp đưa ra các biện pháp hỗ trợ phù hợp, từ đó nâng cao chất lượng đào tạo và giảm tỷ lệ sinh viên bỏ học.

Đề tài này không chỉ có ý nghĩa thực tiễn cao trong quản lý giáo dục mà còn giúp em nâng cao kiến thức chuyên môn về xử lý dữ liệu, học máy và xây dựng ứng dụng thực tế. Vì vậy, em chọn thực hiện đề tài “Xây dựng mô hình dự đoán hiệu suất học tập của sinh viên” nhằm giải quyết một vấn đề thiết thực trong lĩnh vực giáo dục hiện nay.

## Mục Tiêu Nghiên Cứu

**Xây dựng hệ thống mô hình học máy** có khả năng dự đoán hiệu suất học tập của sinh viên dựa trên các đặc trưng như điểm số, mức độ chuyên cần, áp lực học tập, thời gian ngủ, và các yếu tố liên quan khác.

**Phân tích, xử lý và trực quan hóa dữ liệu** sinh viên nhằm hiểu rõ mối liên hệ giữa các đặc trưng đầu vào với kết quả học tập.

**So sánh và đánh giá hiệu quả giữa các mô hình** như Random Forest, CatBoost, XGBoost,... để chọn ra mô hình tối ưu nhất về độ chính xác và hiệu năng.

**Xây dựng giao diện ứng dụng dự đoán** thân thiện, cho phép người dùng nhập dữ liệu đầu vào và nhận kết quả dự đoán ngay lập tức.

**Hỗ trợ nhà trường và cố vấn học tập** trong việc phát hiện sớm sinh viên có nguy cơ gặp khó khăn học tập để có biện pháp can thiệp kịp thời.

## Phương Pháp Nghiên Cứu

**Phương pháp thu thập dữ liệu**: Thu thập bộ dữ liệu về sinh viên bao gồm điểm số, thời gian học, mức độ chuyên cần, áp lực học tập, thời gian ngủ, mức thu nhập của gia đình, mức độ căng thẳng v.v... từ các nguồn giả lập hoặc khảo sát thực tế.

**Phương pháp tiền xử lý dữ liệu (Data Preprocessing)**: Thực hiện làm sạch dữ liệu (loại bỏ dữ liệu thiếu, nhiễu), chuẩn hóa dữ liệu, mã hóa các biến phân loại và chia tập dữ liệu thành tập huấn luyện và tập kiểm tra.

**Phương pháp phân tích thống kê**: Áp dụng các kỹ thuật thống kê mô tả và trực quan hóa dữ liệu để khám phá mối quan hệ giữa các biến đầu vào và kết quả học tập.

**Phương pháp xây dựng mô hình học máy (Machine Learning)**: Sử dụng các thuật toán như Random Forest, CatBoost, và XGBoost để xây dựng mô hình dự đoán hiệu suất học tập của sinh viên.

**Phương pháp đánh giá mô hình**: Sử dụng các chỉ số như độ chính xác (accuracy), F1-score, confusion matrix... để đánh giá và so sánh hiệu quả các mô hình.

**Phương pháp triển khai ứng dụng**: Tích hợp mô hình dự đoán vào giao diện ứng dụng sử dụng thư viện Streamlit, giúp người dùng tương tác và sử dụng trực tiếp mô hình.

# CƠ SỞ LÝ THUYẾT

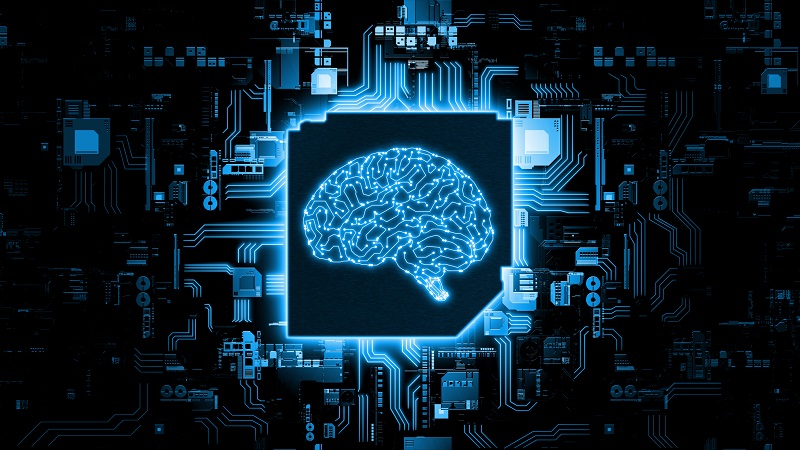
## Tìm Hiểu Về Máy Học

### Máy học là gì ?

Máy học (machine learning) là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kỹ thuật cho phép các hệ thống “học” tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Theo định nghĩa trong cuốn sách "Machine Learning" của Tom Mitchell (1997), máy học được mô tả như sau:

“A computer program is said to learn to perform a task T from experience E, if its performance at task T, as measured by a performance metric P, improves

with experience E over time.”



Học máy (Machine Learning)

Máy học là một thành phần quan trọng của lĩnh vực khoa học dữ liệu đang phát triển. Thông qua việc sử dụng phương pháp thống kê, các thuật toán được đào tạo để phân loại hoặc dự đoán và khám phá những thông tin chi tiết trong các dự án khai thác dữ liệu.

Những thông tin chi tiết này hỗ trợ, thúc đẩy việc đưa ra quyết định trong các ứng dụng, công cụ hỗ trợ doanh nghiệp, người dùng. Khi khối lượng dữ liệu tiếp tục mở rộng và phát triển, khả năng dự đoán, phân tích chính xác của máy học sẽ tăng lên.

### Lịch Sử Hình Thành Máy Học

Lịch sử hình thành và phát triển của máy học (machine learning) có thể được chia thành nhiều giai đoạn khác nhau:

1950s - 1960s: Khởi đầu

**1950**: Alan Turing đề xuất "Turing Test" trong bài viết "Computing Machinery and Intelligence," đặt nền móng cho việc nghiên cứu trí tuệ nhân tạo (AI).

**1952**: Arthur Samuel phát triển chương trình máy tính có khả năng chơi cờ vua và tự học từ kinh nghiệm.

**1957**: Frank Rosenblatt phát triển Perceptron, mô hình mạng neuron đơn giản, đánh dấu bước tiến đầu tiên trong nghiên cứu mạng neuron.

1970s - 1980s: Thời kỳ khó khan

**1970s**: AI gặp phải "Mùa đông AI" khi những kỳ vọng quá cao không được đáp ứng, dẫn đến giảm tài trợ và nghiên cứu.

**1980**: John Hopfield và David Rumelhart tái phát triển mạng neuron bằng cách giới thiệu mô hình Hopfield và thuật toán học ngược.

1990s: Thời kỳ phục hung

**1990s**: Máy học bắt đầu phát triển mạnh mẽ trở lại với sự gia tăng của dữ liệu và sức mạnh tính toán.

**1995**: Vladimir Vapnik và Alexey Chervonenkis phát triển Support Vector Machines (SVM), một phương pháp phân loại mạnh mẽ.

2000s: Phát triển nhanh chóng

**2006**: Geoffrey Hinton và các cộng sự giới thiệu thuật toán Deep Belief Networks, đặt nền móng cho sự phát triển của deep learning.

**2010**: Sự gia tăng của dữ liệu lớn (big data) và sự phát triển của GPU giúp deep learning và các phương pháp máy học khác phát triển mạnh mẽ.

2010s - Hiện tại: Kỷ nguyên AI, phát triển và ứng dụng

**2012**: AlexNet, một mô hình deep learning, chiến thắng cuộc thi ImageNet, chứng minh sức mạnh của mạng neuron sâu.

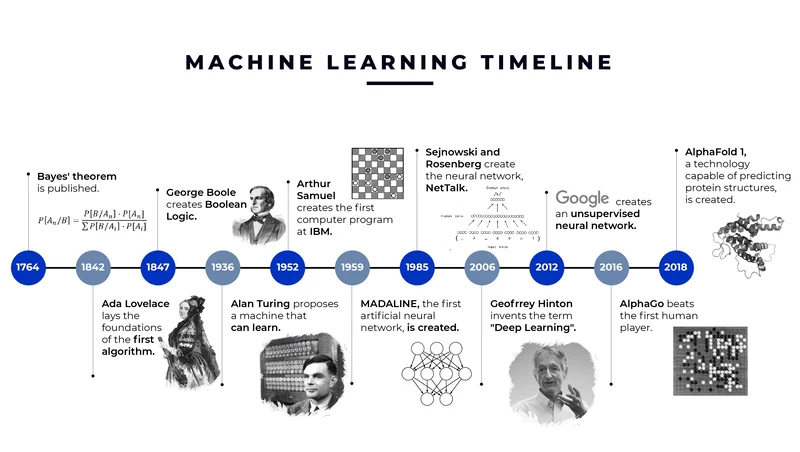
**2014**: Google DeepMind giới thiệu AlphaGo, một hệ thống AI có khả năng chơi cờ vây vượt qua người chơi chuyên nghiệp.

**2016**: AI được ứng dụng rộng rãi trong nhiều lĩnh vực như y tế, tài chính, và ô tô tự lái.

2017: Sự ra đời của Transformer (trong bài báo "Attention is All You Need" của Vaswani et al.), một mô hình mạng neuron mới đã cách mạng hóa nhiều ứng dụng trong xử lý ngôn ngữ tự nhiên (NLP).

2018: GPT-2 và sau đó là GPT-3 của OpenAI, mô hình ngôn ngữ lớn có khả năng tạo ra văn bản tự nhiên với độ phức tạp cao, đã gây ấn tượng mạnh.

2019: BERT (Bidirectional Encoder Representations from Transformers) của Google, một mô hình NLP mạnh mẽ, đã cải thiện nhiều nhiệm vụ xử lý ngôn ngữ tự nhiên.

2020: Sự phát triển của AI tự học không giám sát và học tăng cường, như AlphaFold của DeepMind, giải quyết vấn đề gấp cuộn protein, một trong những thách thức lớn nhất trong sinh học.

Quá trình phát triển của máy học

### Phân Loại Máy Học

**Học có giám sát (Supervised Learning):**

Học có giám sát là một trong những phương pháp phổ biến nhất trong máy học, nơi mà mô hình học từ một tập dữ liệu đã được gán nhãn. Tập dữ liệu huấn luyện gồm các cặp đầu vào (input) và đầu ra (output) mong muốn. Mục tiêu của mô hình là học cách ánh xạ từ đầu vào đến đầu ra để có thể dự đoán chính xác đầu ra cho các dữ liệu mới chưa thấy trước đó. Trong quá trình học, mô hình điều chỉnh các thông số của nó dựa trên sự khác biệt giữa dự đoán và thực tế, nhằm giảm thiểu lỗi dự đoán.

**Học không giám sát (Unsupervised Learning):**

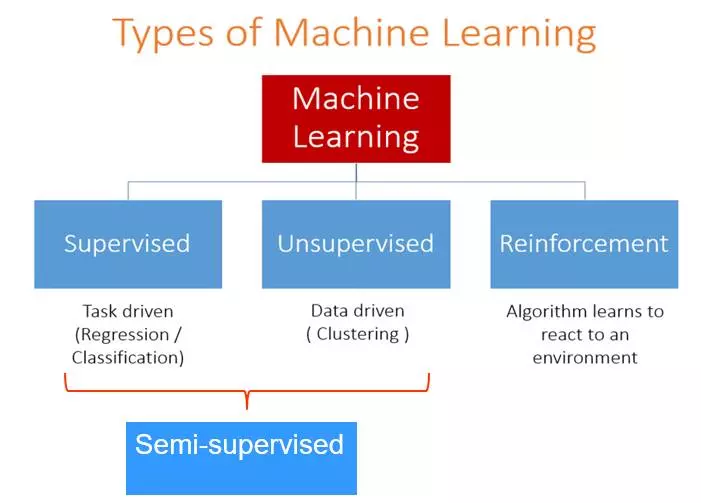
Học không giám sát là một kỹ thuật trong máy học mà không có nhãn đầu ra đi kèm với dữ liệu đầu vào. Thay vào đó, mục tiêu của mô hình là khám phá các cấu trúc, mẫu hoặc mối quan hệ ẩn trong dữ liệu. Điều này giúp hiểu rõ hơn về dữ liệu và có thể được sử dụng để phát hiện các nhóm (clusters), giảm chiều dữ liệu hoặc phát hiện các ngoại lệ (anomalies).

**Học tăng cường (Reinforcement Learning):**

Học tăng cường là một kỹ thuật trong máy học mà một tác nhân (agent) học cách tương tác với môi trường để tối đa hóa phần thưởng tích lũy theo thời gian. Môi trường đưa ra phản hồi dưới dạng phần thưởng hoặc hình phạt cho các hành động của tác nhân. Tác nhân sử dụng phản hồi này để điều chỉnh chính sách hành động của mình sao cho đạt được phần thưởng tối đa.

**Học bán giám sát (Semi-supervised Learning):**

Học bán giám sát là một phương pháp kết hợp giữa học có giám sát và học không giám sát. Mô hình được đào tạo trên một tập dữ liệu có cả nhãn và không nhãn. Điều này rất hữu ích khi việc gán nhãn dữ liệu tốn kém hoặc khó khăn, nhưng có nhiều dữ liệu không nhãn sẵn có. Học bán giám sát sử dụng dữ liệu không nhãn để cải thiện hiệu suất của mô hình hơn so với chỉ sử dụng dữ liệu có nhãn.



Phân loại trong máy học

### Cách hoạt động của thuật toán trong máy học:

#### Bước 1: Thu thập dữ liệu (Gathering Data / Data Collection)

Đây là bước đầu tiên và quan trọng nhất trong quy trình học máy. Trong bước này, cần thu thập dữ liệu từ các nguồn khác nhau mà mô hình của sẽ được huấn luyện. Dữ liệu có thể được thu thập từ nhiều nguồn như cơ sở dữ liệu, cảm biến, dịch vụ web, khảo sát, hoặc thậm chí từ các tệp tin đã có sẵn. Mục tiêu của bước này là đảm bảo có đủ dữ liệu chất lượng để huấn luyện mô hình.

#### Bước 2: Tiền xử lý dữ liệu (Data Preprocessing)

Tiền xử lý dữ liệu là một bước quan trọng để chuẩn bị dữ liệu cho mô hình học máy. Quá trình này bao gồm nhiều công việc khác nhau:

**Trích xuất dữ liệu (Data Extraction):** Lấy dữ liệu từ các nguồn thu thập và đưa vào định dạng dễ xử lý.

**Làm sạch dữ liệu (Data Cleaning):** Xóa bỏ hoặc sửa chữa các lỗi trong dữ liệu, chẳng hạn như dữ liệu bị thiếu, dữ liệu không chính xác, hoặc dữ liệu bị trùng lặp.

**Chuyển đổi dữ liệu (Data Transformation):** Chuyển đổi dữ liệu từ định dạng này sang định dạng khác để phù hợp với yêu cầu của mô hình, ví dụ như chuyển đổi dữ liệu văn bản thành số hoặc thay đổi định dạng thời gian.

**Chuẩn hóa dữ liệu (Data Normalization):** Đưa các thuộc tính dữ liệu về cùng một phạm vi giá trị để giúp mô hình học dễ dàng hơn. Ví dụ, chuẩn hóa các giá trị về khoảng từ 0 đến 1.

**Trích xuất đặc trưng (Feature Extraction):** Xác định và chọn lựa các đặc trưng (features) quan trọng từ dữ liệu gốc mà mô hình sẽ sử dụng. Đây là bước quan trọng để cải thiện hiệu suất của mô hình bằng cách loại bỏ các đặc trưng không cần thiết hoặc không liên quan.

#### Bước 3: Phân tích dữ liệu (Data Analysis)

Trong bước này, thực hiện các phân tích sơ bộ để hiểu rõ hơn về dữ liệu của mình. Điều này có thể bao gồm việc khám phá các mối quan hệ giữa các biến, phân phối của dữ liệu, và các mẫu hoặc xu hướng có thể có trong dữ liệu. Phân tích dữ liệu giúp xác định các vấn đề tiềm ẩn và hướng dẫn việc chọn lựa các thuật toán và kỹ thuật phù hợp cho việc xây dựng mô hình.

#### Bước 4: Xây dựng mô hình máy học (Model Building)

Sau khi dữ liệu đã được chuẩn bị và phân tích, tiến hành xây dựng mô hình học máy. Trong bước này, chọn lựa và thiết kế các thuật toán học máy phù hợp với bài toán của mình. Có nhiều loại mô hình khác nhau như hồi quy tuyến tính, cây quyết định, mạng nơ-ron, và máy vector hỗ trợ (SVM). Mô hình được xây dựng dựa trên các đặc trưng đã chọn và cấu hình các tham số của mô hình.

#### Bước 5: Huấn luyện mô hình (Model Training)

Huấn luyện mô hình là bước quan trọng để mô hình học máy học từ dữ liệu. Trong bước này, dữ liệu huấn luyện được sử dụng để điều chỉnh các tham số của mô hình sao cho mô hình có thể dự đoán chính xác các kết quả. sẽ chia dữ liệu thành hai tập: tập huấn luyện và tập kiểm tra. Tập huấn luyện được dùng để cập nhật các tham số của mô hình, trong khi tập kiểm tra dùng để đánh giá hiệu suất của mô hình trong quá trình huấn luyện.

#### Bước 6: Đánh giá mô hình (Model Evaluation)

Sau khi mô hình đã được huấn luyện, cần đánh giá hiệu suất của nó để đảm bảo rằng nó hoạt động tốt và có thể dự đoán chính xác trên dữ liệu mới. Đánh giá mô hình thường được thực hiện bằng cách sử dụng các chỉ số hiệu suất như độ chính xác, độ nhạy, độ đặc hiệu. cũng có thể sử dụng các kỹ thuật như kiểm tra chéo để đảm bảo rằng mô hình không bị overfit với dữ liệu huấn luyện và có thể hoạt động tốt với dữ liệu chưa thấy.

**Các chỉ số đánh giá chính**

Accuracy (Độ chính xác): Tỷ lệ dự đoán đúng trên tổng số mẫu. Đây là chỉ số phổ biến nhưng không phản ánh toàn diện nếu dữ liệu mất cân bằng (ví dụ: đa số sinh viên đều có kết quả tốt).

Precision (Độ chính xác theo lớp): Tỷ lệ dự đoán đúng trong số những mẫu mà mô hình dự đoán là "sinh viên kém".

Recall (Độ nhạy/Sensitivity): Tỷ lệ phát hiện đúng những sinh viên thực sự kém trong tất cả những sinh viên kém thực tế. Chỉ số này đặc biệt quan trọng nếu mục tiêu là phát hiện sớm để can thiệp.

F1-score:Trung bình điều hòa giữa Precision và Recall, giúp cân bằng hai yếu tố trên.

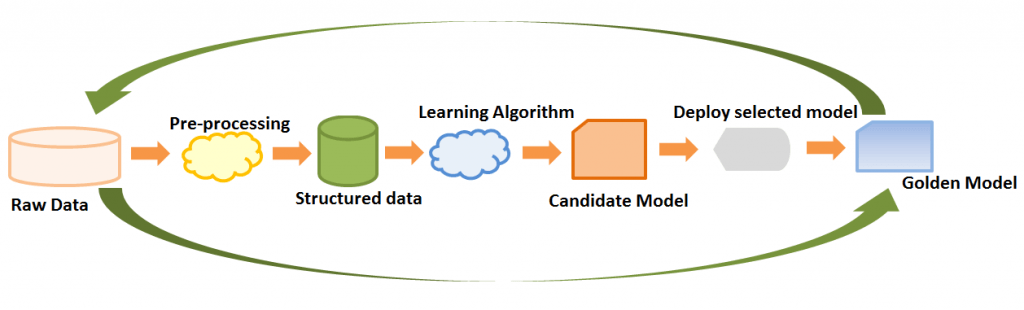
**Kỹ thuật kiểm tra chéo (Cross-validation)**

Để đảm bảo rằng mô hình không chỉ hoạt động tốt trên dữ liệu huấn luyện mà còn tổng quát hóa tốt trên dữ liệu mới, nhóm thực hiện k-fold cross-validation.  
Kỹ thuật này giúp chia tập dữ liệu thành nhiều phần và hoán đổi luân phiên giữa các phần để huấn luyện và kiểm tra mô hình, từ đó giảm thiểu rủi ro của kết quả đánh giá bị lệch do chia tập dữ liệu không đồng đều.

**So sánh mô hình và lựa chọn mô hình tối ưu**

Sau khi đánh giá nhiều mô hình như Random Forest, XGBoost, CatBoost,... trên cùng một bộ tiêu chí, nhóm tiến hành so sánh hiệu suất giữa các mô hình. Việc lựa chọn mô hình cuối cùng không chỉ dựa trên chỉ số chính xác cao nhất, mà còn cân nhắc đến tốc độ xử lý, khả năng diễn giải (interpretability), và khả năng mở rộng cho các ứng dụng thực tế như tích hợp vào giao diện web.

Trực quan hóa kết quả: Cuối cùng, nhóm sử dụng các biểu đồ như biểu đồ cột (bar chart), biểu đồ đường (line chart), và confusion matrix dạng heatmap để trình bày trực quan hiệu suất mô hình. Điều này giúp dễ dàng thuyết minh kết quả với người dùng không chuyên và nâng cao tính minh bạch trong mô hình dự đoán.



Quá trình hoạt động của máy học

## Thuật toán CatBoost

CatBoost là một bộ công cụ mã nguồn mở được phát triển bởi Yandex, rất phổ biến cho kỹ thuật **gradient boosting trên cây quyết định**. Nó có thể được áp dụng để giải quyết nhiều vấn đề học máy khác nhau, bao gồm phân loại (classification), hồi quy (regression), xếp hạng (ranking), và nhiều hơn nữa.

### Các Lợi Ích Chính của CatBoost:

* **Xử lý tự động các đặc trưng phân loại (categorical features)**: CatBoost có khả năng xử lý các đặc trưng phân loại một cách tự động mà không cần mã hóa hay tiền xử lý. Điều này giúp tránh các vấn đề như ma trận thưa thớt (sparse matrix) hoặc hiện tượng quá khớp (overfitting) thường gặp khi sử dụng các kỹ thuật mã hóa truyền thống như One-Hot Encoding.
* **Giảm thiểu quá khớp**: Nó sử dụng một lược đồ gradient boosting mới lạ và các kỹ thuật điều chuẩn (regularization) để giảm quá khớp. CatBoost sử dụng các kỹ thuật thông minh như boosting có thứ tự (ordered boosting), kết hợp đặc trưng ngẫu nhiên (random feature combinations), và các phương pháp boosting mạnh mẽ để giúp mô hình hoạt động tốt ngay cả trên dữ liệu mới, chưa từng thấy.
* **Hiệu suất cao và khả năng mở rộng**: CatBoost đạt được hiệu suất cao và khả năng mở rộng nhờ các triển khai hiệu quả cho CPU và GPU. Nó hỗ trợ huấn luyện tăng tốc bằng GPU, giúp đẩy nhanh quá trình xây dựng mô hình, đặc biệt với các bộ dữ liệu lớn. Thư viện này cũng tận dụng các kỹ thuật xử lý song song để sử dụng nhiều lõi CPU trong quá trình huấn luyện, làm cho quá trình này hiệu quả và có khả năng mở rộng.

### Cơ Chế Hoạt Động của CatBoost

CatBoost hoạt động dựa trên kỹ thuật **gradient boosting**. Quy trình này xây dựng các cây quyết định một cách tuần tự để giảm thiểu sai số và cải thiện dự đoán.

1. **Xây dựng cây quyết định đầu tiên**: Một cây quyết định được xây dựng và đánh giá mức độ sai số trong các dự đoán của nó.
2. **Sửa lỗi từ cây trước**: Cây tiếp theo được tạo ra để sửa các lỗi mà cây trước đó đã mắc phải.
3. **Quy trình lặp lại**: Quá trình này tiếp tục lặp đi lặp lại, với mỗi cây mới tập trung vào việc cải thiện dự đoán của mô hình bằng cách giảm các sai số trước đó, cho đến khi đạt đến số lần lặp đã định trước. Kết quả là một tập hợp các cây quyết định hoạt động cùng nhau để đưa ra các dự đoán chính xác.

CatBoost đặc biệt phù hợp với các bộ dữ liệu lớn có nhiều đặc trưng độc lập. Không giống như các thuật toán gradient boosting khác, CatBoost được thiết kế đặc biệt để xử lý cả các đặc trưng phân loại và số một cách liền mạch mà không yêu cầu mã hóa đặc trưng thủ công. Nó cũng sử dụng thuật toán **Symmetric Weighted Quantile Sketch (SWQS)** để xử lý các giá trị thiếu (missing values), giảm quá khớp và cải thiện hiệu suất mô hình.

### Các Đặc Trưng Nổi Bật của CatBoost:

* **Xử lý đặc trưng phân loại**: CatBoost tự động chuyển đổi các đặc trưng phân loại thành đặc trưng số một cách hiệu quả, sử dụng các chiến lược mã hóa mục tiêu (target encoding) và mã hóa một nóng (one-hot encoding) nội bộ.
* **Xử lý giá trị thiếu**: CatBoost có thể xử lý các giá trị thiếu trong dữ liệu đầu vào mà không cần phương pháp điền khuyết (imputation). Thuật toán SWQS giúp xử lý dữ liệu thiếu hiệu quả, giảm quá khớp và cải thiện hiệu suất.
* **Huấn luyện và phân tích mô hình**:
  + Hỗ trợ huấn luyện tăng tốc GPU, giúp tăng tốc quá trình xây dựng mô hình, đặc biệt với bộ dữ liệu lớn.
  + Sử dụng các kỹ thuật xử lý song song để tận dụng nhiều lõi CPU trong quá trình huấn luyện, làm cho quá trình này hiệu quả và có khả năng mở rộng.
* **Các chỉ số CatBoost (Metrics)**: CatBoost sử dụng các chỉ số để kiểm tra hiệu suất của mô hình, bao gồm độ chính xác (accuracy), độ chuẩn xác (precision), độ thu hồi (recall), F1-score, ROC-AUC cho phân loại, và RMSE cho hồi quy.

### Tham Số và Siêu Tham Số (Hyperparameters) của CatBoost

Việc hiểu và tinh chỉnh các tham số và siêu tham số là rất quan trọng để tối ưu hóa hiệu suất của mô hình CatBoost.

* **Tham số CatBoost:**

Đây là các cài đặt nội bộ của mô hình mà nó học được trong quá trình huấn luyện. Một số tham số quan trọng:

* **iterations**: Chỉ định số lần lặp (số cây) được sử dụng trong quá trình huấn luyện.
* **learning\_rate**: Kiểm soát kích thước bước ở mỗi lần lặp khi di chuyển về phía cực tiểu của hàm mất mát.
* **depth**: Xác định độ sâu tối đa của các cây quyết định riêng lẻ trong tập hợp.
* **l2\_leaf\_reg**: Thuật ngữ điều chuẩn ngăn chặn quá khớp bằng cách phạt các giá trị tham số lớn.
* **cat\_features**: Một mảng các chỉ số cho biết đặc trưng nào là phân loại.
* **loss\_function**: Chỉ định hàm mất mát cần tối ưu hóa trong quá trình huấn luyện. Ví dụ: 'RMSE' cho hồi quy, 'Logloss' hoặc 'MultiClass' cho phân loại.
* **Siêu tham số CatBoost (Hyperparameters):**

Đây là các tham số mà phải cung cấp trước khi huấn luyện mô hình. Chúng kiểm soát các biến liên quan đến huấn luyện. Việc lựa chọn siêu tham số phù hợp là rất cần thiết để mô hình hoạt động tốt. CatBoost cung cấp một giao diện linh hoạt để cấu hình và tinh chỉnh siêu tham số, được chia thành nhiều loại:

* **Siêu tham số chung (Common hyperparameters)**: Áp dụng cho bất kỳ vấn đề học máy nào, như hàm mất mát, tốc độ học, hoặc hạt giống ngẫu nhiên.
* **Siêu tham số Bootstrap (Bootstrap hyperparameters)**: Kiểm soát việc lấy mẫu dữ liệu cho mỗi cây, như loại bootstrap hoặc tỷ lệ lấy mẫu con (subsample rate).
* **Siêu tham số cấu trúc cây (Tree structure hyperparameters)**: Kiểm soát hình dạng và kích thước của mỗi cây, như độ sâu, số lá, hoặc số mẫu tối thiểu trong một lá.
* **Siêu tham số tầm quan trọng của đặc trưng (Feature importance hyperparameters)**: Kiểm soát cách các đặc trưng được chọn và phân tách cho mỗi cây.
* **Siêu tham số điều chuẩn (Regularization hyperparameters)**: Kiểm soát mức độ phức tạp của mô hình bị phạt, như điều chuẩn L2 hoặc phương pháp ước tính lá.
* **Siêu tham số phát hiện quá khớp (Overfitting detector hyperparameters)**: Kiểm soát cách dừng huấn luyện khi xảy ra quá khớp, như chỉ số đánh giá (eval metric) hoặc tùy chọn sử dụng mô hình tốt nhất.

Một số siêu tham số chung được sử dụng để tinh chỉnh:

* **Learning rate (tốc độ học)**: Giảm bước gradient. Giá trị càng nhỏ thì quá trình huấn luyện càng lâu, nhưng cần ít lần lặp hơn.
* **Tree Depth (độ sâu của cây)**: Xác định độ sâu tối đa của mỗi cây quyết định trong tập hợp. Cây sâu hơn có thể nắm bắt các mẫu phức tạp hơn, nhưng có thể quá khớp nếu giá trị quá cao.
* **Bagging temperature**: Điều chỉnh mức độ ngẫu nhiên của các mẫu được chọn để huấn luyện. Giá trị cao hơn (> 1) làm mẫu trở nên xác định hơn, trong khi giá trị nhỏ hơn (ví dụ: 1) làm mẫu trở nên ngẫu nhiên, giúp cải thiện khả năng tổng quát hóa.
* **Border count**: Kiểm soát số lượng phân tách tối đa cho các đặc trưng số, ảnh hưởng đến độ phức tạp của mô hình và hiệu quả huấn luyện.
* **L2 regularization (Điều chuẩn L2)**: Thêm một thuật ngữ phạt vào hàm mất mát trong quá trình huấn luyện để ngăn chặn các giá trị trọng số cao và khuyến khích một mô hình đơn giản hơn, giúp ngăn ngừa quá khớp. Giá trị cao hơn áp đặt điều chuẩn mạnh hơn, được kiểm soát bởi siêu tham số "reg\_lambda".

### Tinh chỉnh siêu tham số (Hyperparameter Tuning):

Là quá trình chọn bộ siêu tham số tối ưu cho một vấn đề và bộ dữ liệu cụ thể. Các bước chung để tinh chỉnh siêu tham số bao gồm:

1. **Xác định không gian tìm kiếm**: Một phạm vi hoặc danh sách các giá trị có thể có cho mỗi siêu tham số muốn tinh chỉnh.
2. **Xác định hàm mục tiêu**: Một hàm đánh giá mức độ hoạt động của mô hình trên tập xác thực với một bộ siêu tham số nhất định.
3. **Xác định chiến lược tìm kiếm**: Một phương pháp quyết định cách khám phá không gian tìm kiếm và tìm ra bộ siêu tham số tối ưu (ví dụ: grid search, random search, Bayesian optimization, Optuna).
4. **Chạy tìm kiếm**: Thực hiện chiến lược tìm kiếm và thu thập kết quả.
5. **Phân tích kết quả**: So sánh và trực quan hóa hiệu suất của các bộ siêu tham số khác nhau và chọn bộ tốt nhất.

### Ứng dụng của CatBoost:

CatBoost có thể được sử dụng trong nhiều nhiệm vụ học máy:

* **Phân loại (Classification Tasks)**:
  + Phân tích cảm xúc (Sentiment analysis)
  + Phát hiện thư rác (Email spam detection)
  + Dự đoán ung thư vú (Breast cancer prediction)
* **Hồi quy (Regression Tasks)**:
  + Dự đoán giá nhà (House price prediction)
  + Dự đoán tiêu thụ nhiên liệu (Fuel consumption prediction)
  + Dự đoán thị trường chứng khoán (Stock market prediction)

### Hạn chế của CatBoost:

Mặc dù CatBoost là một trong những thuật toán học máy tiên tiến và được đánh giá cao trong các bài toán về phân loại và hồi quy, nhưng giống như bất kỳ công cụ nào khác, nó cũng tồn tại một số hạn chế nhất định trong quá trình sử dụng:

**Tiêu thụ bộ nhớ**: Có thể yêu cầu tài nguyên bộ nhớ đáng kể, đặc biệt đối với các bộ dữ liệu lớn.

**Tiêu thụ bộ nhớ:** Mô hình có thể sử dụng lượng lớn bộ nhớ, đặc biệt khi xử lý tập dữ liệu lớn với nhiều đặc trưng phân loại.

**Thời gian huấn luyện:** Quá trình huấn luyện thường chậm hơn các thuật toán boosting khác như LightGBM do sử dụng cơ chế boosting có thứ tự (ordered boosting).

**Tinh chỉnh siêu tham số:** Việc lựa chọn và tinh chỉnh các siêu tham số hiệu quả đòi hỏi nhiều kinh nghiệm, kiến thức chuyên môn và thời gian thử nghiệm với nhiều tổ hợp khác nhau để đạt được hiệu suất tối ưu.

**Hạn chế phân tán:** Mặc dù có hỗ trợ GPU, CatBoost vẫn còn hạn chế trong việc huấn luyện phân tán trên nhiều máy, gây khó khăn khi áp dụng ở quy mô rất lớn.

**Cộng đồng và tài liệu:** So với XGBoost và LightGBM, CatBoost có cộng đồng người dùng nhỏ hơn và tài liệu hỗ trợ vẫn chưa thực sự phong phú, gây trở ngại cho người mới bắt đầu.

### So sánh CatBoost với các thuật toán Boosting khác:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tính năng** | **CatBoost** | **LightGBM** | **XGBoost** |
| **Xử lý đặc trưng phân loại** | Tự động, không cần tiền xử lý | Hỗ trợ mã hóa one-hot, đặc trưng phân loại trực tiếp | Yêu cầu tiền xử lý |
| **Chiến lược phân tách cây** | Đối xứng (Symmetric) | Dựa trên lá (Leaf-wise) | Dựa trên độ sâu (Depth-wise) |
| **Khả năng giải thích** | Tầm quan trọng của đặc trưng, SHAP | Tầm quan trọng của đặc trưng, biểu đồ giá trị phân tách | Tầm quan trọng của đặc trưng, biểu đồ cây |
| **Tốc độ và hiệu quả** | Tối ưu hóa cho tốc độ và bộ nhớ | Hiệu quả cho bộ dữ liệu lớn | Có khả năng mở rộng và nhanh |

Bảng so sánh cho thấy CatBoost, đặc biệt là phiên bản đã tinh chỉnh, thường có hiệu suất cạnh tranh hoặc tốt hơn so với LightGBM và XGBoost trên một số bộ dữ liệu, với lỗi thấp hơn.

### Cải thiện mô hình:

Để nâng cao hơn nữa mô hình của , hãy cân nhắc:

* **Kỹ thuật đặc trưng (Feature engineering)**: Tạo ra các đặc trưng thông tin từ dữ liệu thô.
* **Khám phá các chỉ số đánh giá khác**: Đặc biệt quan trọng nếu làm việc với dữ liệu không cân bằng.
* **Các kỹ thuật điều chuẩn**: Để ngăn chặn quá khớp.

# TIỀN XỬ LÝ VÀ TRỰC QUANG HÓA DỮ LIỆU SINH VIÊN

## Dữ liệu sinh viên

Dữ liệu được sử dụng trong đồ án đến từ nguồn mở trên nền tảng Kaggle, tại địa chỉ:  
🔗 <https://www.kaggle.com/datasets/mahmoudelhemaly/students-grading-dataset>

Tập dữ liệu chứa thông tin chi tiết của từng sinh viên bao gồm yếu tố học tập, xã hội và cá nhân – có ảnh hưởng trực tiếp hoặc gián tiếp đến kết quả học tập. Tổng cộng có **23 thuộc tính** (columns), trong đó thuộc tính mục tiêu (target) chính là **“Grade”** hoặc **“Total\_Score”**. Dưới đây là mô tả từng thuộc tính và nhận định ảnh hưởng của nó trong quá trình học máy:

|  |  |
| --- | --- |
| **Thuộc tính** | Giải thích chi tiết |
| Student\_ID | Mã định danh duy nhất cho mỗi sinh viên. Được dùng để theo dõi và phân biệt các bản ghi trong tập dữ liệu, không mang thông tin học thuật. |
| First\_Name, Last\_Name | Họ và tên sinh viên. Dữ liệu nhận diện cá nhân, không phản ánh yếu tố học tập. Có thể được ẩn danh nếu cần bảo mật. |
| Email | Địa chỉ liên lạc của sinh viên. Tương tự như tên, chỉ mang tính định danh hoặc quản lý, không ảnh hưởng trực tiếp đến kết quả học tập. |
| Gender | Giới tính của sinh viên (Nam, Nữ hoặc Khác). Có thể liên quan đến các khác biệt về phương pháp học tập, môi trường xã hội hoặc các yếu tố tâm lý. |
| Age | Tuổi của sinh viên. Tuổi tác có thể ảnh hưởng đến sự trưởng thành, khả năng tiếp thu kiến thức và phương pháp học tập. |
| Department | Ngành học của sinh viên (ví dụ: Khoa học máy tính, Kinh doanh,...). Ngành học khác nhau có mức độ khó, yêu cầu kỹ năng và cách đánh giá khác nhau. |
| Attendance (%) | Tỷ lệ chuyên cần trên lớp. Chuyên cần cao thường phản ánh mức độ cam kết và khả năng nắm bắt bài giảng tốt hơn, từ đó ảnh hưởng đến điểm số. |
| Midterm\_Score | Điểm thi giữa kỳ (trên thang 100). Là chỉ số quan trọng thể hiện kết quả học tập trong giai đoạn đầu học kỳ. |
| Final\_Score | Điểm thi cuối kỳ (trên 100). Đại diện cho thành tích học tập toàn bộ học kỳ, thường có trọng số cao trong tổng điểm. |
| Assignments\_Avg | Trung bình điểm bài tập. Cho thấy mức độ chăm chỉ, đều đặn và khả năng áp dụng kiến thức trong quá trình học. |
| Quizzes\_Avg | Trung bình điểm các bài kiểm tra ngắn. Giúp phản ánh sự hiểu bài ngay tại lớp và khả năng duy trì kiến thức trong thời gian ngắn. |
| Participation\_Score | Mức độ tham gia hoạt động lớp học (trên thang 10). Thể hiện sự chủ động học tập và thái độ tích cực trong môi trường lớp học. |
| Projects\_Score | Điểm đánh giá bài tập lớn/đồ án. Là chỉ số quan trọng thể hiện kỹ năng áp dụng thực tế, làm việc nhóm, giải quyết vấn đề. |
| Total\_Score | Tổng điểm tích lũy – thường là trung bình có trọng số từ các yếu tố như bài tập, kiểm tra, thi cuối kỳ... Là chỉ số tổng hợp để xác định học lực. |
| Grade | Xếp loại học lực bằng chữ (A, B, C, D, F). Là biến mục tiêu cuối cùng trong các mô hình dự đoán hiệu suất học tập. |
| Study\_Hours\_per\_Week | Số giờ tự học trung bình mỗi tuần. Là một trong những yếu tố then chốt thể hiện tính kỷ luật và mức độ đầu tư học tập. |
| Extracurricular\_Activities | Tham gia các hoạt động ngoại khóa (Có/Không). Ảnh hưởng gián tiếp đến sự phát triển kỹ năng mềm và khả năng quản lý thời gian. |
| Internet\_Access\_at\_Home | Có Internet tại nhà (Có/Không). Là yếu tố hỗ trợ học tập, đặc biệt quan trọng trong học online hoặc tìm kiếm tài liệu. |
| Parent\_Education\_Level | Trình độ học vấn cao nhất của phụ huynh (Không học, THPT, Cử nhân, Thạc sĩ, Tiến sĩ). Có thể ảnh hưởng đến môi trường học tập, định hướng và hỗ trợ từ gia đình. |
| Family\_Income\_Level | Mức thu nhập gia đình (Thấp, Trung bình, Cao). Phản ánh điều kiện kinh tế và khả năng tiếp cận nguồn tài nguyên học tập (sách vở, Internet, lớp thêm,...). |
| Stress\_Level (1-10) | Mức độ căng thẳng tự đánh giá (1 là thấp, 10 là cao). Mức stress cao thường ảnh hưởng tiêu cực đến khả năng tiếp thu và kết quả học tập. |
| Sleep\_Hours\_per\_Night | Số giờ ngủ trung bình mỗi đêm. Giấc ngủ đủ giúp cải thiện trí nhớ, khả năng tập trung và sức khỏe học tập tổng thể. |

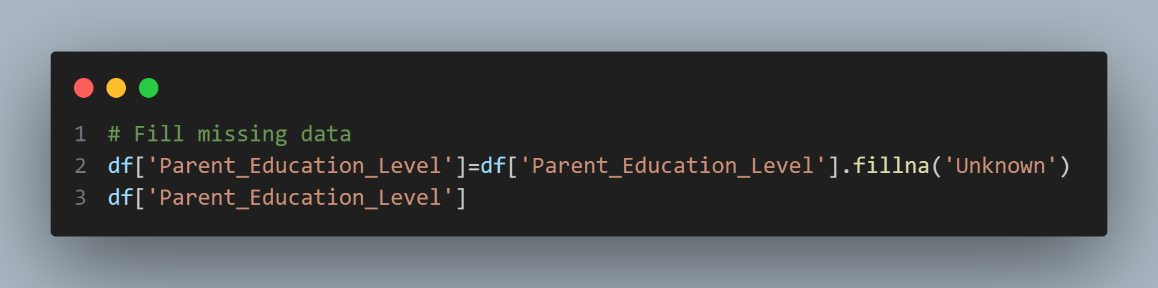
## Tiền xử lý dữ liệu

Tiền xử lý dữ liệu là bước quan trọng trước khi xây dựng mô hình học máy, nhằm đảm bảo dữ liệu đầu vào có chất lượng tốt, giảm nhiễu và phù hợp với yêu cầu của thuật toán. Trong nghiên cứu này, quá trình tiền xử lý dữ liệu được thực hiện theo các bước sau:

### Kiểm tra và làm sạch dữ liệu

Trong quá trình khảo sát tập dữ liệu, nhóm tiến hành kiểm tra số lượng giá trị không khuyết (non-null) của từng cột bằng phương thức df.info(). Kết quả cho thấy đa số các cột đều có **5.000 bản ghi đầy đủ**, ngoại trừ cột **Parent\_Education\_Level** chỉ có **3.975 giá trị**, tương ứng với **1.025 giá trị bị thiếu**.

Để đảm bảo dữ liệu đầu vào không gây ảnh hưởng đến quá trình huấn luyện mô hình, nhóm tiến hành các bước làm sạch dữ liệu như sau:

* **Xử lý giá trị khuyết:**
  + Với cột **Parent\_Education\_Level**, các giá trị bị thiếu được thay thế bằng giá trị "Unknown". Điều này giúp duy trì số lượng bản ghi ban đầu, đồng thời vẫn phân biệt được những trường hợp không có thông tin về trình độ học vấn của phụ huynh.

điền giá trị bị thi

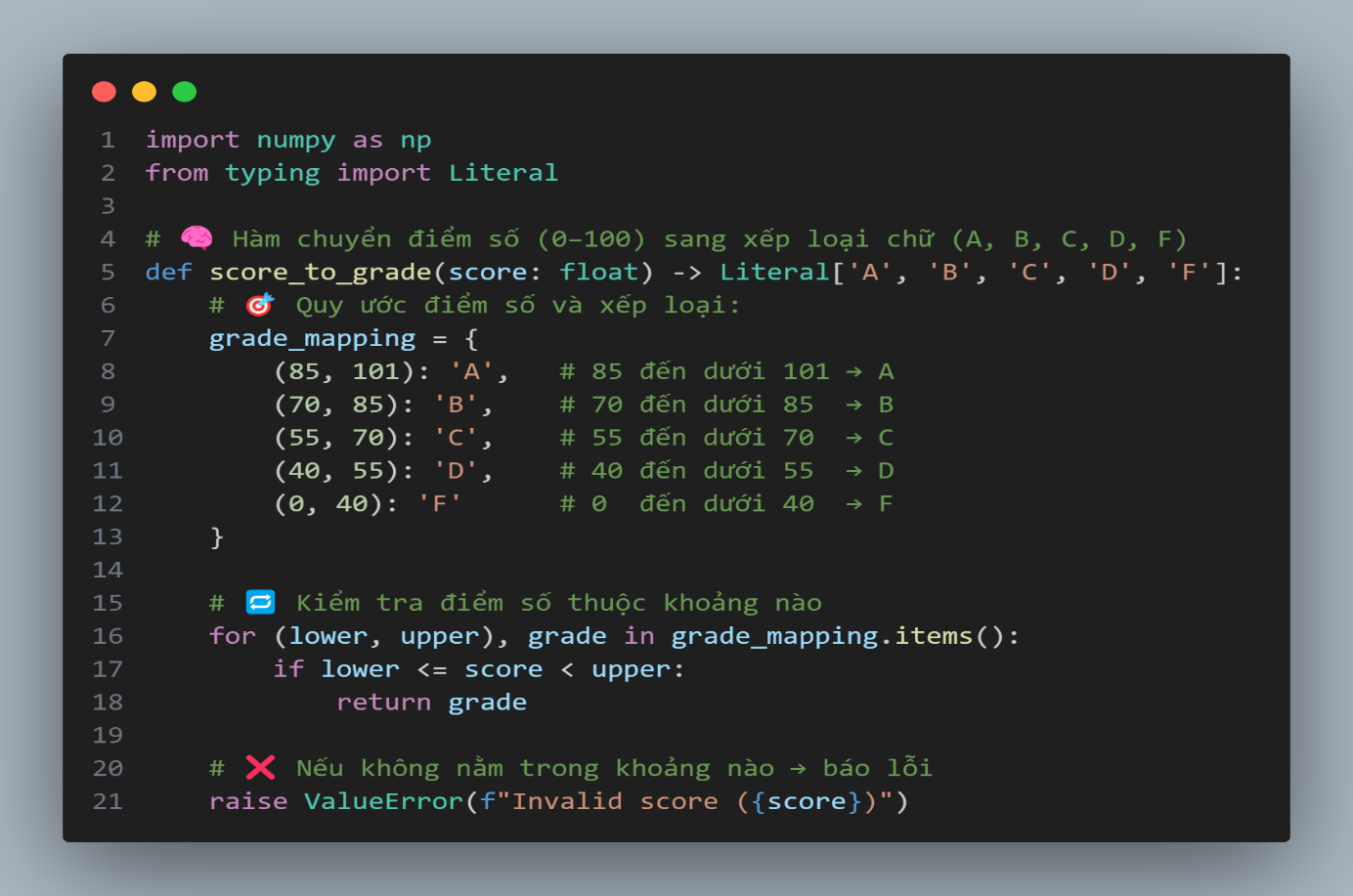
* + Các cột định lượng khác không xuất hiện giá trị khuyết nên không cần xử lý bổ sung.

### **Chuẩn hóa dữ liệu điểm và xếp loại**

Trong tập dữ liệu gốc, biến GRADE (xếp loại học lực của sinh viên) được lưu trữ dưới dạng ký tự (A, B, C, D, F). Tuy nhiên, trong quá trình kiểm tra chéo giữa biến **Total\_Score** (tổng điểm) vàGRADE, phát hiện nhiều trường hợp không nhất quán.

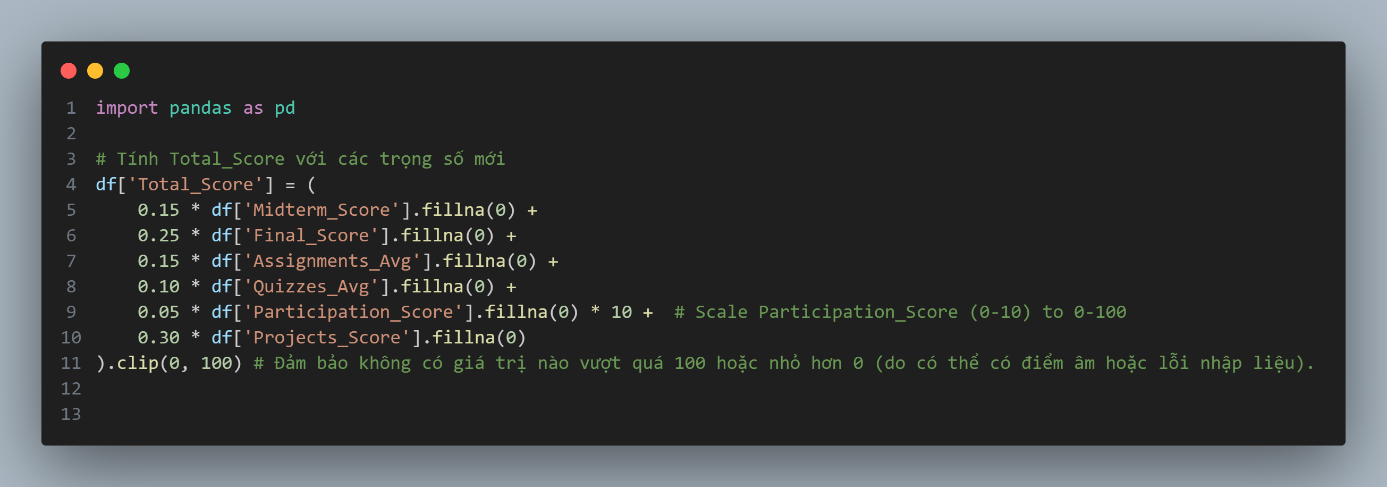
***ví dụ:*** sinh viên có **Total\_Score = 51.03** nhưng lại được gán **Grade = 'A'**. Điều này cho thấy biến GRADE không đảm bảo độ tin cậy nếu sử dụng trực tiếp làm biến mục tiêu trong mô hình dự đoán.

Sinh viên đạt từ 85 đến 100 điểm được xếp loại **A**; từ 70 đến dưới 85 điểm được xếp loại **B**; từ 55 đến dưới 70 điểm được xếp loại **C**; từ 40 đến dưới 55 điểm được xếp loại **D**; và dưới 40 điểm sẽ được xếp loại **F**.

****Việc chuẩn hóa này giúp loại bỏ sự sai lệch và đảm bảo rằng Grade phản ánh chính xác thành tích học tập của sinh viên, đồng thời tăng độ tin cậy khi sử dụng làm biến mục tiêu (target) trong huấn luyện mô hình CatBoost.

Hàm chuyển điểm số sang chữ

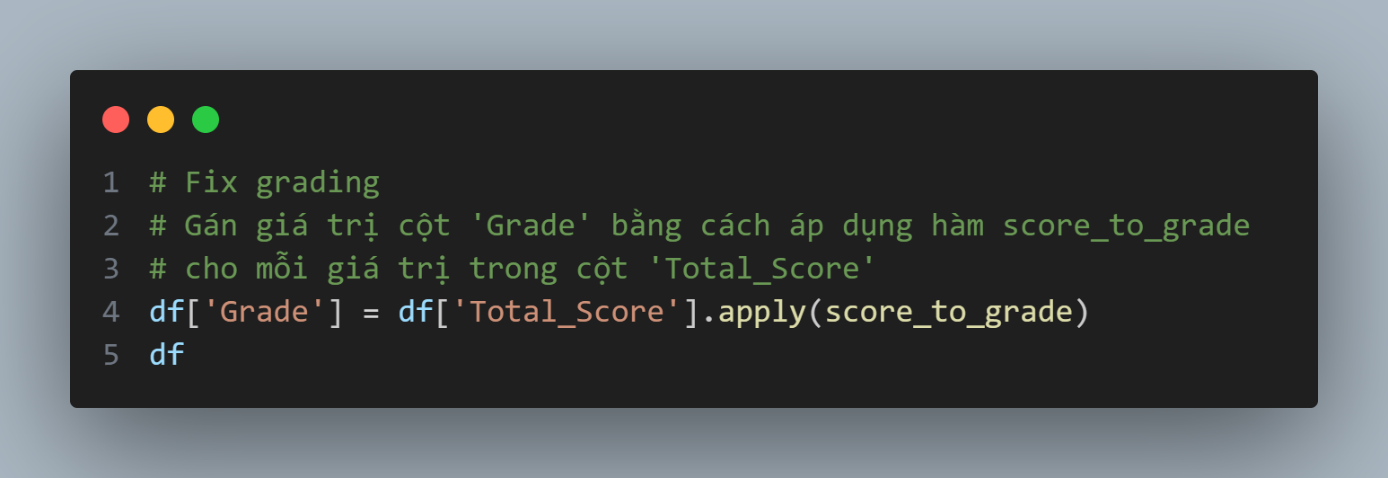
Phần tính toán **điểm tổng (Total\_Score)** được thực hiện dựa trên việc kết hợp các thành phần đánh giá khác nhau của sinh viên với các trọng số đã được xác định trước. Cụ thể, công thức tính được xây dựng như sau:

* **Điểm giữa kỳ (Midterm\_Score)** chiếm 15% tổng điểm.
* **Điểm cuối kỳ (Final\_Score)** chiếm 25% tổng điểm.
* **Điểm trung bình bài tập (Assignments\_Avg)** chiếm 15% tổng điểm.
* **Điểm trung bình bài kiểm tra ngắn (Quizzes\_Avg)** chiếm 10% tổng điểm.
* **Điểm tham gia lớp học (Participation\_Score)** chiếm 5% tổng điểm, tuy nhiên do được đánh giá theo thang điểm 0–10 nên cần được quy đổi sang thang 0–100 trước khi nhân trọng số.
* **Điểm dự án (Projects\_Score)** chiếm 30% tổng điểm.

Tính toán điểm tổng (Total\_Score)

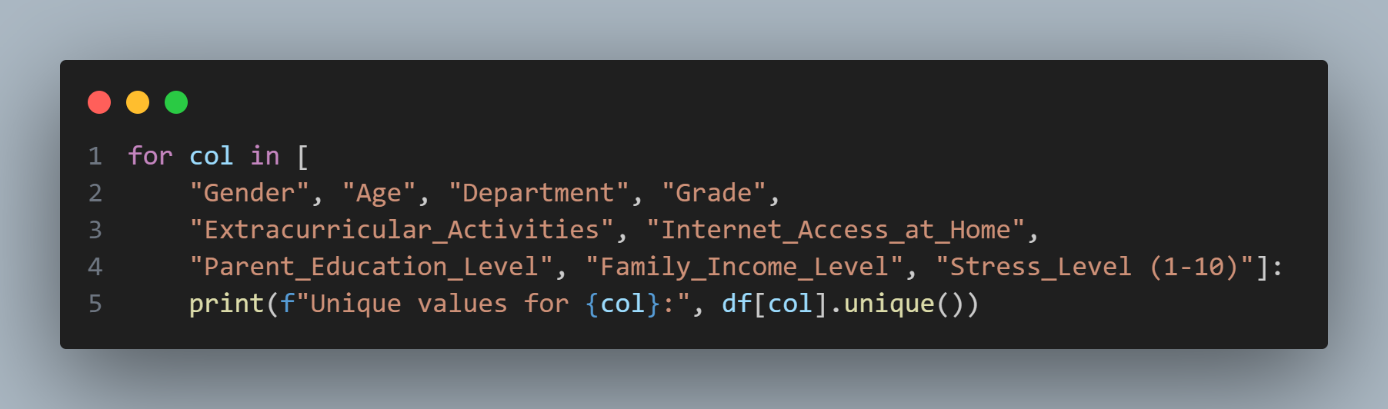
Các giá trị bị thiếu (**NaN**) trong từng thành phần sẽ được thay thế bằng 0 nhằm tránh gây sai lệch trong quá trình tính toán. Bên cạnh đó, sau khi tính toán, điểm tổng được giới hạn trong khoảng từ 0 đến 100 bằng cách sử dụng phương pháp **clipping**, nhằm loại bỏ các trường hợp bất thường như điểm âm hoặc vượt quá thang điểm tối đa.

Cách tính này giúp đảm bảo rằng **Total\_Score** phản ánh tương đối chính xác kết quả học tập của sinh viên, đồng thời thuận lợi cho việc phân loại học lực và đưa vào mô hình học máy như CatBoost để dự đoán hoặc phân tích.

Sau khi áp dụng lại hàm **score\_to\_grade** để sửa xếp loại dựa trên **Total\_Score**, dữ liệu cho thấy các giá trị trong cột **Grade** đã được chuẩn hóa và không còn các trường hợp bất hợp lý như trước.

Áp dụng hàm score\_to\_grade dựa trên total\_sore

### Kiểm tra giá trị duy nhất của các thuộc tính

Sau khi xử lý dữ liệu thiếu, nhóm tiến hành kiểm tra các giá trị duy nhất trong từng cột phân loại để đảm bảo tính hợp lệ và thống nhất. Kết quả cho thấy:

Kiểm tra các thuộc tính là duy nhất

* + **Gender**: Chỉ có 2 giá trị hợp lệ "Male" và "Female", không có dữ liệu nhiễu.
  + **Age**: Các giá trị trong khoảng từ 18 đến 24 tuổi, phù hợp với độ tuổi sinh viên đại học.
  + **Department**: Bao gồm 4 khoa "Mathematics", "Business", "Engineering", "CS", không xuất hiện khoa ngoài danh sách.
  + **Grade**: Có 5 mức xếp loại "A", "B", "C", "D", "F", đúng theo thang điểm chữ chuẩn.
  + **Extracurricular\_Activities**: Chỉ gồm "Yes" và "No".
  + **Internet\_Access\_at\_Home**: Chỉ gồm "Yes" và "No".
  + **Parent\_Education\_Level**: Có các mức "Master's", "High School", "Bachelor's", "PhD" và giá trị "Unknown" (được gán cho dữ liệu thiếu).
  + **Family\_Income\_Level**: Gồm "Low", "Medium", "High".
  + **Stress\_Level (1-10)**: Giá trị nguyên từ 1 đến 10, phản ánh mức độ căng thẳng.

Kết quả trên cho thấy dữ liệu phân loại đã được chuẩn hóa tốt, không cần thêm bước làm sạch nâng cao ngoài xử lý giá trị thiếu.

## Trực quang hóa dữ liệu

Phân tích dữ liệu sinh viên là quá trình thu thập, xử lý và khai thác thông tin từ các dữ liệu liên quan đến học tập, hành vi, và đặc điểm cá nhân của sinh viên. Ý nghĩa của việc này bao gồm:

* Giúp nhà trường, giáo viên hiểu rõ hơn về tình hình học tập, các yếu tố ảnh hưởng đến kết quả học tập của sinh viên.
* Phát hiện sớm các sinh viên có nguy cơ học lực yếu để có biện pháp hỗ trợ kịp thời.
* Đánh giá hiệu quả của các chương trình đào tạo, hoạt động ngoại khóa, và các chính sách giáo dục.
* Hỗ trợ cá nhân hóa việc học, xây dựng lộ trình phù hợp cho từng sinh viên.
* Là cơ sở để dự báo, ra quyết định quản lý, cải tiến chất lượng đào tạo.

Tóm lại, phân tích dữ liệu sinh viên giúp tối ưu hóa quá trình giáo dục, nâng cao chất lượng và hiệu quả đào tạo.

### Phân tích phân bố điểm chữ của sinh viên

Biểu đồ cột phân bố điểm

* **Grade B** chiếm tỷ lệ lớn nhất: **2,808 sinh viên (≈ 56,16%)**.
* **Grade C** đứng thứ hai: **1,997 sinh viên (≈ 39,94%)**.
* **Grade A** rất ít: **153 sinh viên (≈ 3,06%)**.
* **Grade D** gần như hiếm: **42 sinh viên (≈ 0,84%)**.

#### a. Nhận xét

* Phân bố này cho thấy phần lớn sinh viên đạt **mức trung bình – khá** (B và C), trong khi số sinh viên xuất sắc (A) hoặc yếu (D) rất nhỏ.
* Khoảng chênh lệch lớn giữa nhóm **B** và các nhóm còn lại có thể phản ánh:
  + **Đề thi, bài tập hoặc tiêu chí chấm điểm** được thiết kế để đa số sinh viên đạt mức khá.
  + **Hiệu suất học tập** của phần lớn sinh viên tương đối đồng đều, không có nhiều cá nhân nổi trội hoặc yếu kém.
* Nhóm **A** và **D** đều là thiểu số, cho thấy độ phân tán kết quả thấp, dữ liệu tương đối “tập trung” quanh trung bình.

#### Ý nghĩa đối với mô hình học máy

* **Vấn đề mất cân bằng lớp (class imbalance)**:
  + Lớp B và C chiếm gần 96% dữ liệu → mô hình dễ **thiên lệch** dự đoán vào hai nhóm này, khó nhận diện chính xác A và D.
  + Các lớp A và D cần **kỹ thuật xử lý imbalance** (oversampling, SMOTE, hoặc trọng số lớp trong CatBoost).
* **Khó phân loại nhóm hiếm**:
  + Vì dữ liệu nhóm A và D quá ít, mô hình cần được huấn luyện với cơ chế penalize lỗi nhiều hơn ở nhóm hiếm.
  + Có thể kết hợp **feature engineering** để tìm đặc trưng giúp tách bạch A/D khỏi B/C.

### Phân tích các yếu tố ảnh hưởng đến học tập

Mối quan hệ giữa các yếu tố học tập

#### a. Grade vs. Stress Level

**Điểm A, B, C**: Trung vị stress dao động từ ~5 đến 6, phân bố khá rộng từ 1 đến 10, Không có chênh lệch lớn về mức độ stress giữa các nhóm điểm này.

**Điểm D**: Trung vị stress cao hơn (~6), cho thấy nhóm học sinh điểm thấp có xu hướng chịu áp lực nhiều hơn. Có thể stress là yếu tố góp phần làm giảm hiệu suất học tập.

**Ý nghĩa**: Stress không hoàn toàn tỷ lệ nghịch với điểm số, nhưng nhóm điểm D đáng chú ý vì áp lực cao hơn.

#### Grade vs. Study Hours per Week

**Điểm A**: Trung vị số giờ học cao nhất (~18 giờ/tuần). Khoảng phân bố lệch về phía nhiều giờ học, cho thấy nhóm này thường đầu tư thời gian học đáng kể.

**Điểm B và C**: Trung vị thấp hơn (khoảng 17 giờ cho C, 17–18 giờ cho B).

Có sự phân tán cao, tức là trong nhóm này vẫn có học sinh học rất nhiều hoặc rất ít.

**Điểm D**: Đáng ngạc nhiên là trung vị giờ học (~21–22 giờ/tuần) cao hơn cả nhóm B và C. Điều này có thể cho thấy học sinh điểm D dù học nhiều nhưng phương pháp học chưa hiệu quả.

**Ý nghĩa**: Thời gian học không phải yếu tố duy nhất quyết định điểm số chất lượng và phương pháp học cũng quan trọng.

#### Grade vs. Attendance (%)

**Điểm A**: Trung vị điểm danh khoảng 76–77%, không quá cao.

**Điểm B và C**: Trung vị tương tự nhau (~75%), phân bố rộng từ 50% đến 100%.

**Điểm D**: Trung vị cao nhất (~83–85%), điều này khá bất thường — nhóm điểm thấp lại đi học đều hơn. Có thể nguyên nhân là phương pháp học chưa phù hợp, hoặc chỉ tham gia lớp học không đủ để đạt điểm cao.

**Ý nghĩa**: Tỉ lệ điểm danh cao không đồng nghĩa với kết quả học tập cao, cho thấy cần kết hợp nhiều yếu tố khác như phương pháp học và kỹ năng làm bài.

### Phân tích hoạt động ảnh hưởng đến tổng điểm

#### Extracurricular Activities vs. Total Score (Hoạt động ngoại khóa)

* **Yes** và **No** có trung vị điểm gần như tương đương (~71–72).
* Phân bố điểm ở cả hai nhóm khá rộng (từ ~50 đến >90).
* Nhóm “Yes” có một số điểm cao nổi bật, nhưng khác biệt tổng thể không lớn.

**Nhận xét:** Tham gia hoạt động ngoại khóa không tạo ra sự khác biệt rõ rệt về điểm số trung bình; có thể ảnh hưởng gián tiếp qua kỹ năng mềm hoặc tinh thần học tập.

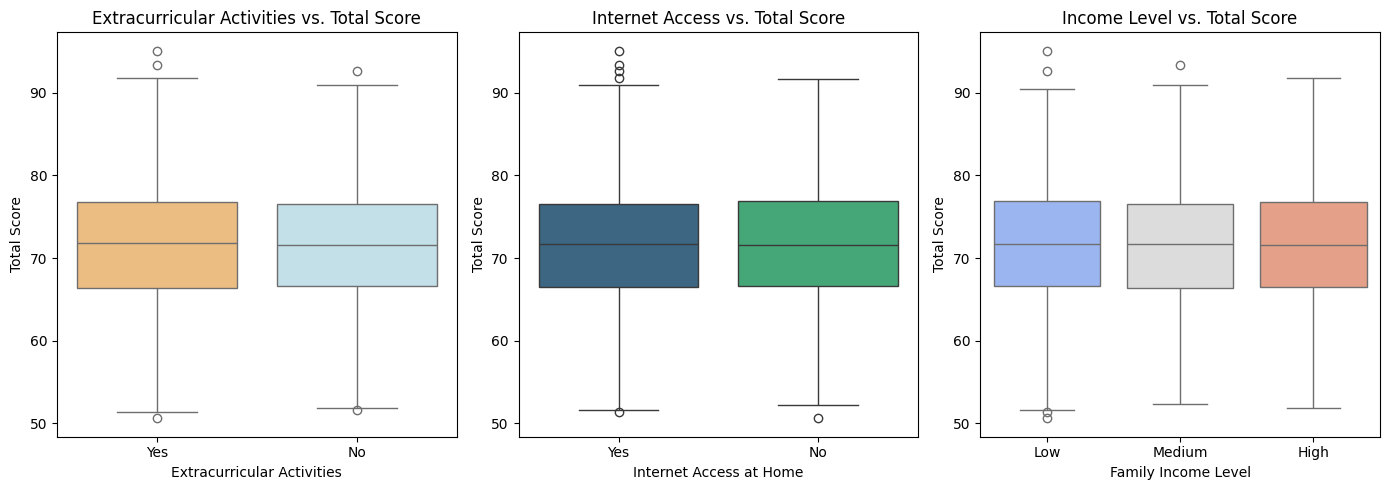
#### Internet Access vs. Total Score (Truy cập Internet tại nhà)

* **Yes** và **No** có trung vị điểm tương tự (~71–72).
* Nhóm “Yes” có nhiều giá trị ngoại lai ở mức điểm cao hơn.
* Khoảng phân bố điểm rộng ở cả hai nhóm.

**Nhận xét:** Có Internet ở nhà không đảm bảo điểm số cao hơn, nhưng có thể giúp một số học sinh đạt thành tích vượt trội.

#### Income Level vs. Total Score (Mức thu nhập gia đình)

* **Low, Medium, High** đều có trung vị gần nhau (~71–72).
* Phân bố điểm rất tương tự, không có nhóm nào vượt trội rõ rệt.
* Cả ba nhóm đều có điểm cao và thấp.

**Nhận xét:** Mức thu nhập gia đình không cho thấy tác động rõ rệt đến điểm số tổng thể; yếu tố cá nhân và môi trường học tập có thể quan trọng hơn.

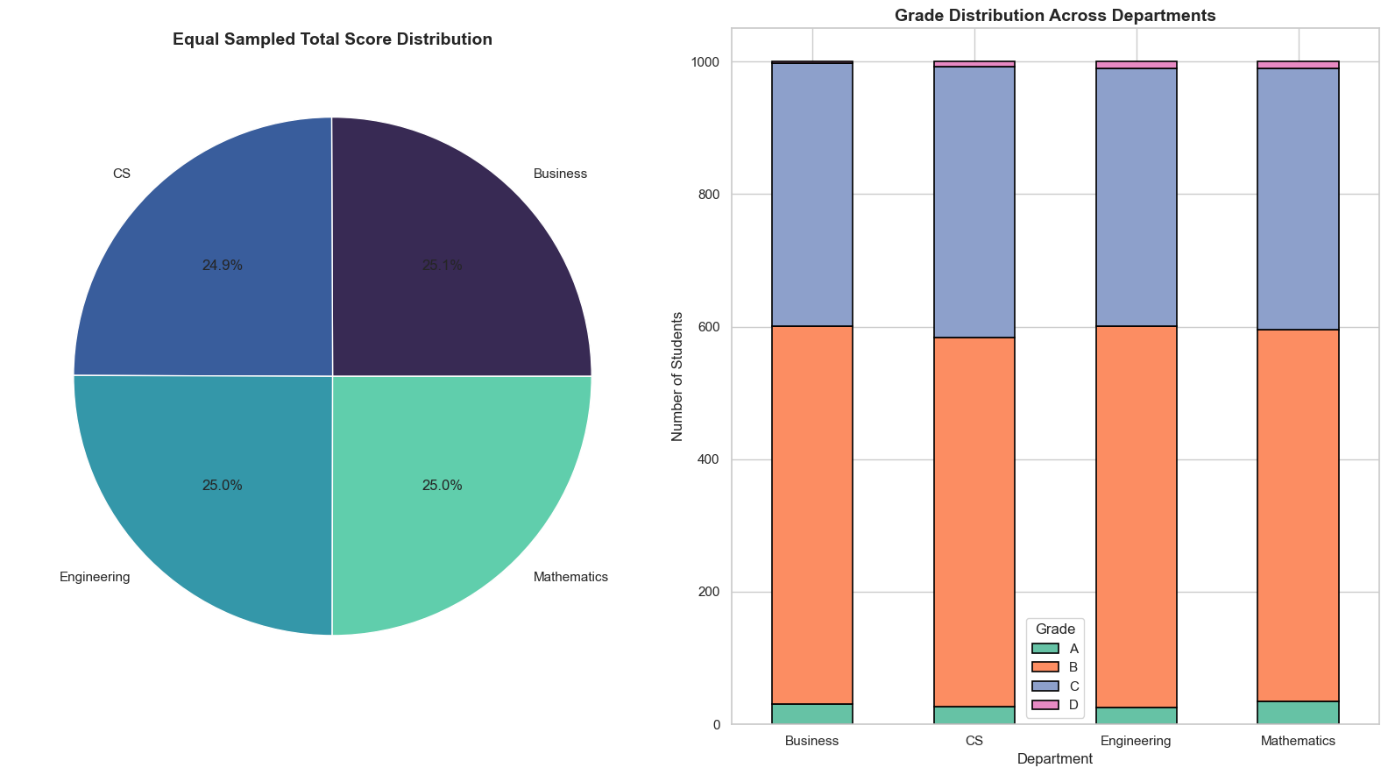
Hoạt động ảnh hưởng đến tổng điểm

### Phân tích phân bố điểm số theo khoa

#### Equal Sampled Total Score Distribution (Phân bố mẫu đều theo khoa)

* Tỉ lệ sinh viên giữa các khoa **Business**, **CS**, **Engineering**, và **Mathematics** gần như bằng nhau (~25%).
* Điều này cho thấy dữ liệu đã được **lấy mẫu cân bằng** nhằm tránh thiên lệch khi so sánh thành tích học tập giữa các khoa.

#### Grade Distribution Across Departments (Phân bố loại điểm theo khoa)

* Ở cả 4 khoa, điểm **B** chiếm tỉ lệ áp đảo (~58–60%), tiếp theo là điểm **C** (~38–40%).
* Điểm **A** chiếm tỉ lệ rất thấp (~2–3%) ở tất cả các khoa.
* Điểm **D** xuất hiện với tỉ lệ cực nhỏ (<1%), gần như không đáng kể.
* Không có sự khác biệt rõ rệt về phân bố điểm giữa các khoa — cấu trúc điểm gần như giống hệt nhau.

Phân tích phân bố điểm theo khoa

### Phân tích tổng quan với biểu đồ nhiệt

Biểu đồ nhiệt

Biểu đồ nhiệt này thể hiện **hệ số tương quan Pearson** giữa các yếu tố trong bộ dữ liệu và cho thấy mức độ liên hệ tuyến tính giữa chúng.  
Dựa vào số liệu trong biểu đồ, có thể rút ra các ý chính:

#### a. Các yếu tố liên quan chặt chẽ với nhau

* + Total\_Score có tương quan mạnh với:
    - Grade (0.86) → Tổng điểm ảnh hưởng trực tiếp tới xếp loại.
    - Projects\_Score (0.60) → Điểm dự án góp phần lớn vào tổng điểm.
    - Final\_Score (0.59) → Điểm cuối kỳ là thành phần quan trọng của tổng điểm.
  + Final\_Score tương quan vừa với Grade (0.51).
  + Projects\_Score tương quan vừa với Grade (0.50).

#### Các yếu tố gần như không liên quan

* + Study\_Hours\_per\_Week,Internet\_Access\_at\_Home, Parent\_Education\_Level, Stress\_Level, Sleep\_Hours\_per\_Night, Age, và Attendance (%) có hệ số tương quan rất thấp (gần 0) với hầu hết các yếu tố điểm số → nghĩa là trong dữ liệu này, các yếu tố này không cho thấy mối liên hệ tuyến tính rõ ràng với hiệu suất học tập.
  + Midterm\_Score và Total\_Score (0.33) → Điểm giữa kỳ có ảnh hưởng nhưng không mạnh như điểm cuối kỳ hoặc điểm dự án.
  + Assignments\_Avg và Total\_Score (0.32) → Trung bình bài tập cũng góp phần nhưng không lớn.

#### Ý nghĩa tổng quan

* + Các yếu tố liên quan trực tiếp đến **điểm đánh giá (Final, Midterm, Projects, Assignments)** mới có ảnh hưởng đáng kể đến tổng điểm và xếp loại.
  + Các yếu tố về **điều kiện học tập, thời gian học, hoặc yếu tố cá nhân** trong bộ dữ liệu này không thể hiện tương quan mạnh với kết quả học tập.

#### Điểm bất cập

Trong phần tới sẽ huấn luyện mô hình dựa trên các thuộc tính như Study\_Hours\_per\_Week,Internet\_Access\_at\_Home, Parent\_Education\_Level, Stress\_Level, Sleep\_Hours\_per\_Night, Age, và Attendance (%)… Tuy nhiên, theo biểu đồ nhiệt về hệ số tương quan, các yếu tố này đều có **độ tương quan rất thấp** với các chỉ số kết quả học tập (Total\_Score, Grade, Final\_Score, ...). Điều này đồng nghĩa rằng về mặt thống kê, chúng không thể hiện mối liên hệ tuyến tính mạnh mẽ với hiệu suất học tập.

Nguyên nhân không sử dụng các thuộc tính có độ tương quan cao như Final\_Score, Midterm\_Score, Projects\_Score, hay Assignments\_Avg là vì:

* Các thuộc tính này **trực tiếp phản ánh kết quả học tập** (ví dụ: điểm giữa kỳ, cuối kỳ, điểm dự án).
* Nếu dùng chúng để huấn luyện mô hình, kết quả dự đoán sẽ gần như **chỉ là phép tính cộng/trung bình** từ các điểm thành phần, chứ không còn là quá trình “dự đoán” thực sự.
* Mục tiêu ở đây là **dự đoán hiệu suất học tập trước khi có điểm thi hoặc điểm dự án**, nên cần dựa vào các yếu tố gián tiếp như thói quen học tập, điều kiện học tập, yếu tố cá nhân, v.v.

Do đó, việc huấn luyện mô hình trên các thuộc tính có tương quan thấp tuy khó khăn hơn, nhưng sẽ giúp mô hình phản ánh đúng năng lực dự đoán từ các dữ liệu đầu vào chưa chứa sẵn thông tin kết quả.

### Kết luận chung qua quá trình phân tích

#### Kết luận chi tiết

Kết quả phân tích cho thấy yếu tố quyết định mạnh mẽ nhất đến thành tích học tập của sinh viên là **điểm các bài kiểm tra đánh giá**, đặc biệt:

* **Final Exam** có mức ảnh hưởng lớn nhất, thể hiện qua hệ số tương quan cao.
* **Midterm**, **dự án** và **bài tập** cũng góp phần quan trọng nhưng mức độ thấp hơn một chút so với Final Exam.

Ngược lại, các yếu tố như **thời gian học mỗi tuần**, **mức độ căng thẳng**, **điểm danh**, hay **tham gia hoạt động ngoại khóa** cho thấy mối liên hệ yếu hoặc không đáng kể với điểm số. Điều này gợi ý rằng, **chỉ số điểm số được hình thành chủ yếu từ hiệu quả làm bài thi, không chỉ từ thời gian hoặc tần suất tham gia lớp học**.

#### Một số phát hiện đáng chú ý khác:

* **Trình độ học vấn của cha mẹ** có xu hướng tác động tích cực: Cha mẹ có trình độ học vấn cao (Thạc sĩ, Tiến sĩ) thường có con đạt điểm cao hơn.
* **Thời gian học hợp lý** (tối đa khoảng 30 giờ/tuần) thường đi kèm kết quả tốt; tuy nhiên, học quá nhiều không đảm bảo điểm cao hơn.
* **Mức độ căng thẳng vừa phải** (4–6/10) có thể mang tính thúc đẩy, nhưng căng thẳng quá cao hoặc quá thấp đều không tối ưu cho kết quả học tập.
* **Truy cập internet tại nhà** mang lại lợi thế nhỏ, nhưng không tạo khác biệt quá lớn.
* **Hoạt động ngoại khóa** có ảnh hưởng gián tiếp: Tham gia vừa phải giúp cải thiện kỹ năng mềm và điểm số, nhưng tham gia quá nhiều dễ gây quá tải và giảm thành tích học tập.

#### Gợi ý các phương pháp cải hiện hiệu xuất học tập

1. **Tập trung cải thiện kỹ năng làm bài thi và dự án**, vì đây là yếu tố then chốt quyết định điểm số.
2. **Ưu tiên chất lượng học** hơn là thời lượng; học quá nhiều giờ chưa chắc hiệu quả.
3. **Đầu tư hạ tầng công nghệ và tài nguyên học tập** để đảm bảo sinh viên có đủ điều kiện học tập.
4. **Quản lý căng thẳng hợp lý**, tránh để áp lực học tập vượt quá khả năng chịu đựng.
5. **Khuyến khích hoạt động ngoại khóa** ở mức cân bằng để vừa phát triển kỹ năng mềm vừa duy trì kết quả học tập.

# XÂY DỰNG MÔ HÌNH MÁY HỌC DỰ ĐOÁN HIỆU XUẤT HỌC TẬP

# KẾT LUẬN

## Hiểu Biết về Phân Loại Âm Thanh và CNN

- **Khái niệm cơ bản:** Nhóm đã nắm vững các khái niệm cơ bản về phân loại âm thanh, bao gồm cách dữ liệu âm thanh được chuyển đổi thành các đặc trưng như hình ảnh phổ, tần số cơ bản (F0), và các formant (F1, F2). Các kỹ thuật như Biến đổi Fourier nhanh (FFT) giúp mô hình học máy sử dụng các đặc trưng này để phân loại âm thanh hiệu quả.

- **Mô hình CNN:** Nhóm đã hiểu rõ vai trò của mạng nơ-ron tích chập (CNN) trong phân loại âm thanh. CNN có khả năng trích xuất các đặc trưng quan trọng từ hình ảnh phổ và các đặc trưng âm thanh khác, giúp nâng cao độ chính xác của việc phân loại thông qua các lớp tích chập, gộp, và kết nối đầy đủ.

## Xây dựng và Đánh Giá Mô Hình CNN

- **Xây Dựng Mô Hình:** Thay vì sử dụng các kiến trúc RNN như LSTM vốn phổ biến trong xử lý dữ liệu âm thanh do tính chất chuỗi thời gian của nó, nhóm đã chọn sử dụng CNN để xây dựng mô hình phân loại âm thanh. Điều này được thực hiện bằng cách chuyển đổi âm thanh thành hình ảnh phổ (spectrogram), cho phép CNN xử lý dữ liệu âm thanh như một vấn đề thị giác. Mô hình CNN sau đó trích xuất các đặc trưng quan trọng từ hình ảnh phổ này, từ đó học cách phân loại các loại âm thanh khác nhau một cách hiệu quả.

- **Đánh Giá Hiệu Suất:** Sau khi huấn luyện, mô hình CNN đã được kiểm tra trên tập dữ liệu thử nghiệm, trong đó các tệp âm thanh được chuyển đổi thành hình ảnh phổ (spectrogram). Mô hình được tải từ tệp model.h5 và tiến hành dự đoán các lớp cho từng hình ảnh phổ. Quá trình dự đoán bao gồm việc tính toán xác suất cho mỗi lớp và chọn lớp có xác suất cao nhất làm kết quả cuối cùng.Kết quả dự đoán của mô hình được so sánh với nhãn thực tế trong tập kiểm thử. Những chỉ số như độ chính xác (accuracy) và độ nhạy (sensitivity) đều đạt mức cao, cho thấy mô hình có khả năng phân loại âm thanh hiệu quả. Nhóm đã hiển thị một số kết quả dự đoán và nhận thấy rằng chúng tương đối khớp với nhãn thực tế, khẳng định rằng mô hình CNN hoạt động tốt trên dữ liệu kiểm thử, và có thể ứng dụng trong phân loại âm thanh một cách tin cậy.

## Ứng Dụng và Tinh Chỉnh

- **Ứng dụng thực tế:** Mô hình đã chứng minh khả năng ứng dụng trong các tình huống thực tế như nhận diện âm thanh từ các tệp WAV, ví dụ như phân loại “tiếng chó sủa” hay “tiếng còi xe”. Điều này mang lại hiệu quả rõ rệt trong việc tự động hóa và cải thiện quy trình làm việc trong các hệ thống giám sát âm thanh và nhận diện môi trường.

- **Tinh chỉnh:** Nhóm đã thực hiện các tinh chỉnh cần thiết để cải thiện hiệu suất của mô hình, bao gồm việc tối ưu hóa các siêu tham số của CNN và nâng cao chất lượng dữ liệu đầu vào thông qua việc chuyển đổi âm thanh thành hình ảnh phổ (spectrogram). Những điều chỉnh này đã giúp nâng cao độ chính xác và khả năng phân loại âm thanh của mô hình.

## ĐÁNH GIÁ KẾT QUẢ

### Đánh giá triển khai tổng quan

Hiệu quả triển khai:

Việc triển khai mô hình CNN đã chứng tỏ khả năng phân loại âm thanh hiệu quả và mạnh mẽ. Mô hình đã đạt được độ chính xác cao trong việc nhận diện các loại âm thanh khác nhau, ví dụ như “tiếng chó sủa”, “tiếng còi xe”, và nhiều âm thanh khác. Sự linh hoạt của mô hình trong việc xử lý các dạng âm thanh khác nhau cho thấy sự phù hợp của nó với các ứng dụng thực tế. Mô hình đã thực hiện tốt trong các tình huống khác nhau, từ môi trường kiểm thử đến các điều kiện thực tế, cho thấy khả năng ứng dụng rộng rãi và hiệu quả trong việc tự động hóa quy trình phân loại âm thanh.

Tuy nhiên, trong quá trình triển khai, nhóm đã gặp phải một số khó khăn. Việc điều chỉnh mô hình để nhận diện các loại âm thanh mới hoặc chưa thấy trước đó là một thách thức đáng kể. Điều này thường yêu cầu phải có kiến thức sâu rộng về cấu trúc và hoạt động của mô hình CNN cũng như các kỹ thuật tiền xử lý dữ liệu. Bên cạnh đó, việc tối ưu hóa mô hình cho các loại âm thanh chưa được định danh cũng đòi hỏi nhiều thử nghiệm và tinh chỉnh, điều này có thể làm tăng thời gian và công sức cần thiết cho việc phát triển mô hình.

### Đề Xuất Cải Tiến

#### Tối ưu hóa mô hình:

Để nâng cao hiệu suất của mô hình CNN, nên xem xét việc áp dụng các kiến trúc mạng nơ-ron tiên tiến hơn, chẳng hạn như các mạng nơ-ron tích chập sâu (Deep CNNs) hoặc các mô hình mạng nơ-ron tích chập biến thể (CNN Variants) như ResNet hoặc DenseNet. Các phương pháp học sâu tiên tiến, như học chuyển giao (Transfer Learning) hoặc tăng cường học sâu (Deep Learning Enhancement), cũng có thể giúp cải thiện khả năng phân loại của mô hình bằng cách tận dụng các mô hình đã được huấn luyện trên các tập dữ liệu lớn và đa dạng.

#### Cải thiện chất lượng dữ liệu:

Để nâng cao chất lượng của mô hình, việc thu thập và làm phong phú thêm tập dữ liệu âm thanh là cần thiết. Điều này có thể bao gồm việc mở rộng quy mô dữ liệu bằng cách thu thập thêm các loại âm thanh từ nhiều nguồn khác nhau, cũng như sử dụng các kỹ thuật tăng cường dữ liệu (Data Augmentation) như thay đổi tốc độ, tăng cường âm lượng, hoặc thêm nhiễu để làm phong phú tập dữ liệu. Các kỹ thuật này giúp mô hình học được nhiều đặc điểm và biến thể của âm thanh hơn, từ đó cải thiện độ chính xác và khả năng tổng quát**.**

#### Tinh chỉnh siêu tham số:

Để đạt được kết quả tốt nhất, cần tiếp tục tinh chỉnh các siêu tham số của mô hình CNN, bao gồm số lượng lớp tích chập, kích thước kernel, và tỷ lệ dropout. Việc thử nghiệm với các cấu hình khác nhau của các tham số như tốc độ học (learning rate), kích thước batch, và các phương pháp tối ưu hóa (như Adam, RMSprop) cũng có thể giúp tối ưu hóa hiệu suất của mô hình. Quá trình tinh chỉnh này yêu cầu thử nghiệm và đánh giá thường xuyên để xác định cấu hình tốt nhất cho các bài toán phân loại âm thanh cụ thể.

## KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### Kết Luận

**Tổng kết:**

Nghiên cứu đã chứng minh rằng mô hình CNN là một công cụ mạnh mẽ và hiệu quả trong việc phân loại âm thanh. Mô hình đã đạt được kết quả tốt trong việc nhận diện chính xác các loại âm thanh khác nhau, từ các âm thanh phổ biến như “tiếng chó sủa” đến các loại âm thanh khác. Khả năng của mô hình trong việc phân loại chính xác và linh hoạt cho thấy tiềm năng ứng dụng rộng rãi trong nhiều lĩnh vực thực tế, chẳng hạn như hệ thống giám sát âm thanh và tự động hóa các quy trình phân loại âm thanh.

**Ý nghĩa:**

Việc áp dụng CNN trong phân loại âm thanh không chỉ giúp tự động hóa quy trình phân loại mà còn cải thiện đáng kể hiệu suất làm việc trong các ứng dụng liên quan đến âm thanh. Mô hình CNN cho phép xử lý và phân tích âm thanh một cách hiệu quả, giảm thiểu sự can thiệp của con người và nâng cao độ chính xác trong việc nhận diện âm thanh.

### Hướng Phát Triển

Hướng nghiên cứu tiếp theo:

Để nâng cao khả năng tổng quát và hiệu suất của mô hình, nghiên cứu nên mở rộng bằng cách khám phá các kiến trúc mạng nơ-ron sâu hơn hoặc tích hợp các phương pháp học sâu tiên tiến như học không giám sát (unsupervised learning) hoặc học tăng cường (reinforcement learning). Việc này có thể giúp mô hình học được các đặc trưng âm thanh phức tạp và cải thiện khả năng phân loại các loại âm thanh chưa thấy trước đó.

Ứng dụng thực tiễn:

Khuyến khích triển khai mô hình vào các ứng dụng thực tiễn, chẳng hạn như hệ thống nhận diện âm thanh trong môi trường công nghiệp hoặc giám sát an ninh. Đồng thời, cần theo dõi và áp dụng các cải tiến công nghệ mới để nâng cao hiệu suất và khả năng của mô hình, giúp đáp ứng các yêu cầu ngày càng cao của thực tiễn.

## HẠN CHẾ

-**Hiểu biết chưa đầy đủ về các kỹ thuật xử lý tín hiệu:** Nhóm nhận thức rằng vẫn còn một số kỹ thuật xử lý tín hiệu âm thanh, như STFT, wavelet, chưa được khai thác hết tiềm năng.

-**Kiến thức về các kiến trúc mạng CNN còn hạn chế:** Nhóm chưa khám phá hết các biến thể của CNN như ResNet, Inception, và chưa tối ưu hóa được siêu tham số của mô hình.

-**Chưa hiểu sâu về lý thuyết học sâu**: Một số khái niệm như overfitting, underfitting, gradient vanishing, exploding gradient vẫn còn chưa được nắm vững hoàn toàn.

-**Chất lượng dữ liệu:** Dữ liệu âm thanh sử dụng để huấn luyện mô hình còn hạn chế về số lượng và đa dạng, ảnh hưởng đến khả năng tổng quát của mô hình.

-**Tính toán:** Việc huấn luyện mô hình đòi hỏi nhiều tài nguyên tính toán, nhóm chưa khai thác tối đa các công cụ và thư viện hỗ trợ.

TÀI LIỆU KHAM KHẢO

* 1. Myad S. E. O. (2024, July 18). Thuật toán CNN là gì? Thông tin cấu trúc của mạng CNN. FPT Aptech
  2. Học máy (Machine Learning) là gì? Cách hoạt động và ứng dụng. (n.d.). FPT IS - Công ty TNHH Hệ thống thông tin FPT.
  3. 4. T.-Q. lý N. (2023, February 27). Deep Learning là gì? Hiểu tổng quan về Deep Learning và ứng dụng. Tanca.io; Tanca là Sự kết hợp giữa trải nghiệm, hiểu biết sâu sắc, chiến lược con người với sự hỗ trợ của trí tuệ nhân tạo AI.
  4. PHÉP BIẾN ĐỔI FOURIER NHANH VÀ ỨNG DỤNG CỦA CHÚNG TRÊN NỀN CÔNG NGHỆ NHÚNG ARM TRONG NGHIÊN CỨU THIẾT KẾ, CHẾ TẠO THIẾT BỊ GIÁM SÁT CÁC GIẢI TẦN SỐ VÔ TUYẾN ĐIỆN. (n.d.). Org.vn.
  5. (N.d.). Wikipedia.org.
  6. SuNT. (n.d.). SuNT’s blog. Github.Io.