
# BÀI TẬP LỚN MÔN HỌC: HỌC MÁY

Học kỳ I - Năm học 2025–2026

Bộ môn Khoa học Máy tính
Trường Đại học Bách Khoa, ĐHQG-HCM

**Giảng viên hướng dẫn:** TS. Lê Thành Sách

Ngày 1 tháng 9 năm 2025

**Phiên bản hiện tại: v1.1**

## Lịch sử phiên bản

| Version | Ngày | Nội dung cập nhật |
|---|---|---|
| 1.0 | 2025-31-08 | Khởi tạo tài liệu |
| 1.1 | 2025-09-01 | Cập nhật năm học và học kỳ. |

## 1 Giới thiệu

Bài tập lớn của môn học **Học máy** nhằm giúp sinh viên vận dụng các kiến thức lý thuyết đã học để giải quyết những bài toán thực tế trên nhiều loại dữ liệu khác nhau. Thông qua việc xây dựng quy trình xử lý dữ liệu, trích xuất đặc trưng, huấn luyện mô hình và đánh giá kết quả, sinh viên sẽ nắm vững các bước cơ bản trong việc triển khai một hệ thống học máy hoàn chỉnh.

Nội dung bài tập trải dài trên ba dạng dữ liệu phổ biến: dữ liệu dạng bảng, dữ liệu văn bản và dữ liệu ảnh. Mỗi dạng dữ liệu đều có những đặc thù riêng, yêu cầu sinh viên lựa chọn phương pháp xử lý và mô hình phù hợp. Qua đó, sinh viên không chỉ củng cố kiến thức lý thuyết mà còn rèn luyện kỹ năng thực hành, tư duy phân tích và khả năng viết báo cáo khoa học.

# 2 Tổng quan về pipeline học máy

Trong học máy, một hệ thống thường được triển khai theo chuỗi các bước gọi là *pipeline*. Pipeline mô tả toàn bộ quy trình từ dữ liệu thô ban đầu cho đến khi xây dựng được mô hình dự đoán và đánh giá kết quả. Hiện nay, có hai hướng tiếp cận phổ biến: **pipeline học máy truyền thống** và **pipeline học sâu (deep learning)**.

**Pipeline học máy truyền thống** thường bao gồm:

1. EDA (Exploratory Data Analysis): thống kê mô tả dữ liệu, trực quan hóa, phát hiện dữ liệu thiếu, giá trị ngoại lai và xu hướng phân phối.

2. Tiền xử lý dữ liệu: làm sạch dữ liệu, xử lý giá trị thiếu, chuẩn hóa hoặc chuẩn chỉnh dữ liệu, mã hóa dữ liệu phân loại, chia tập train/test.

3. Trích xuất và lựa chọn đặc trưng: sử dụng các kỹ thuật khác nhau tùy loại dữ liệu, ví dụ:

   - **Dữ liệu văn bản (Text):**
     - Truyền thống: Bag-of-Words, TF-IDF, n-gram.
     - Học sâu: RNN/LSTM, Transformer (BERT, GPT).
   - **Dữ liệu ảnh (Image):**
     - Truyền thống: HOG (Histogram of Oriented Gradients), SIFT, PCA để giảm chiều sau khi vector hóa ảnh.
     - Học sâu: CNN (Convolutional Neural Network), ResNet, VGG, EfficientNet, Transformer-based models (ViT, Swin Transformer).
   - **Dữ liệu bảng (Tabular):**
     - Truyền thống: đặc trưng thống kê (mean, std), chuẩn hóa (z-score, min-max), mã hóa dữ liệu phân loại (one-hot encoding, label encoding).
     - Học sâu: MLP (Multilayer Perceptron), TabNet, NODE (Neural Oblivious Decision Ensembles).

4. Huấn luyện mô hình: áp dụng các thuật toán học máy truyền thống như Logistic Regression, SVM, k-NN, Random Forest, Naive Bayes.

5. Đánh giá mô hình: sử dụng các chỉ số như accuracy, precision, recall, F1-score.

**Pipeline học sâu (deep learning)** hiện đại thường đơn giản hơn ở khâu trích xuất đặc trưng nhờ khả năng tự động học biểu diễn của mạng nơ-ron:

1. EDA (Exploratory Data Analysis): trực quan hóa dữ liệu, thống kê mô tả, kiểm tra phân phối, phát hiện dữ liệu thiếu hoặc bất thường.

2. Tiền xử lý dữ liệu cơ bản: chuẩn hóa (normalization), resize ảnh, tokenization văn bản, padding chuỗi.

3. Mạng nơ-ron học đặc trưng: CNN cho ảnh, RNN/LSTM cho dữ liệu chuỗi, Transformer cho văn bản và ảnh (ViT, BERT).

4. Huấn luyện đầu-cuối (end-to-end training): toàn bộ mô hình được tối ưu cùng một lúc bằng gradient descent.

5. Đánh giá và triển khai: sử dụng các chỉ số phù hợp (accuracy, F1-score, BLEU, IoU), và triển khai mô hình vào ứng dụng thực tế.

Trong phạm vi môn học này, sinh viên **bắt buộc** phải triển khai pipeline theo hướng **truyền thống**. Nếu có nhóm thực hiện thêm pipeline học sâu và tiến hành so sánh, báo cáo sẽ được **cộng điểm thưởng** để khuyến khích mở rộng và sáng tạo.

# 3 Mục tiêu

Bài tập lớn được thiết kế với các mục tiêu cụ thể sau:

- Hiểu và áp dụng được quy trình **pipeline học máy truyền thống**, bao gồm: tiền xử lý dữ liệu, trích xuất đặc trưng, huấn luyện và đánh giá mô hình.
- Rèn luyện kỹ năng triển khai mô hình học máy trên các loại dữ liệu khác nhau: bảng, văn bản, và ảnh.
- Phát triển khả năng phân tích, so sánh, và đánh giá hiệu quả của các mô hình học máy thông qua các chỉ số đo lường.
- Rèn luyện kỹ năng lập trình, thử nghiệm, và tổ chức báo cáo khoa học.
- **Bắt buộc:** triển khai pipeline học máy truyền thống trên ít nhất một loại dữ liệu.
- **Khuyến khích:** nếu có thể, thực hiện thêm pipeline học sâu (deep learning) để so sánh với pipeline truyền thống. Các nhóm có phần mở rộng này sẽ được **cộng điểm thưởng**.

# 4 Nội dung

Bài tập lớn của môn học bao gồm ba chủ đề tương ứng với ba loại dữ liệu phổ biến: dữ liệu dạng bảng, dữ liệu văn bản và dữ liệu ảnh. Mỗi nhóm hoặc cá nhân sẽ lựa chọn ít nhất một chủ đề để triển khai. Đối với mỗi loại dữ liệu, sinh viên cần thực hiện đầy đủ các bước trong **pipeline học máy truyền thống**, từ EDA, tiền xử lý, trích xuất đặc trưng, huấn luyện mô hình đến đánh giá kết quả.

Việc thực hiện **EDA (Exploratory Data Analysis)** là yêu cầu bắt buộc, nhằm giúp sinh viên hiểu rõ đặc điểm của dữ liệu trước khi đưa vào phân tích và xây dựng mô hình. Ngoài ra, nếu sinh viên triển khai thêm **pipeline học sâu (deep learning)** để so sánh với pipeline truyền thống, báo cáo sẽ được cộng thêm điểm thưởng. Điều này khuyến khích sinh viên mở rộng kiến thức và có cái nhìn toàn diện hơn về hai hướng tiếp cận trong học máy.

## 4.1 Bài 1: Học máy với dữ liệu dạng bảng (Tabular Data)

Sinh viên được chia thành các nhóm (xem chi tiết ở mục Hình thức làm việc) và mỗi nhóm phải lựa chọn một tập dữ liệu phù hợp để triển khai bài tập, với các ràng buộc sau:

- Các nhóm không được chọn trùng tập dữ liệu.

- Tập dữ liệu phải có **missing value** để sinh viên thực hành kỹ thuật *imputation*.
- Tập dữ liệu phải có **categorical value** để sinh viên thực hành các kỹ thuật *encoding*.
- Số lượng mẫu (*sample size*) đủ lớn để pipeline có ý nghĩa; việc lựa chọn cụ thể nên được thảo luận trực tiếp với giảng viên trên lớp.

Nhiệm vụ của mỗi nhóm là xây dựng một **pipeline học máy truyền thống** cho dữ liệu dạng bảng. Pipeline này phải được thiết kế sao cho cho phép cấu hình các kỹ thuật và tham số ở từng bước. Ví dụ:

- *Scaling*: có thể lựa chọn `MinMaxScaler` hoặc `StandardScaler`; nếu chọn `MinMaxScaler`, sinh viên cần cấu hình `feature_range`.
- *Giảm số chiều*: có thể lựa chọn PCA với các mức giữ lại phương sai khác nhau (90%, 95%, ...).
- *Mô hình*: có thể lựa chọn Logistic Regression, SVM, Random Forest và so sánh kết quả.

Kết quả cuối cùng cần bao gồm: báo cáo phân tích EDA, mô tả pipeline, các tham số đã thử nghiệm, và so sánh hiệu quả giữa các cấu hình.

## 4.2 Bài 2: Học máy với dữ liệu dạng văn bản (Text Data)

Sinh viên được chia thành các nhóm (xem chi tiết ở mục Hình thức làm việc) và mỗi nhóm lựa chọn một tập dữ liệu văn bản để triển khai bài tập, với các ràng buộc sau:

- Các nhóm không được chọn trùng tập dữ liệu.
- Tập dữ liệu phải có độ đa dạng về nội dung (ví dụ: nhiều chủ đề hoặc nhiều nhãn phân loại khác nhau).
- Số lượng mẫu (*sample size*) phải đủ lớn để mô hình có ý nghĩa; việc lựa chọn cụ thể nên được thảo luận trực tiếp với giảng viên trên lớp.

Trong bài tập này, sinh viên cần hiện thực ít nhất **hai phương pháp trích xuất đặc trưng** cho dữ liệu văn bản:

- **Truyền thống:** Bag-of-Words (BoW), TF-IDF, hoặc n-gram.
- **Hiện đại (deep learning embeddings):** sử dụng các mô hình học sâu để biểu diễn văn bản. Gợi ý:
  - Word2Vec hoặc GloVe (embedding từ mô hình huấn luyện trước).
  - BERT, RoBERTa, hoặc DistilBERT (Transformer-based contextual embeddings).

  Sinh viên cần **chạy trích xuất embedding và lưu kết quả** thành file định dạng `.npy` hoặc `.h5`, sau đó sử dụng các file embedding này làm đầu vào cho các mô hình phân loại truyền thống hoặc mạng nơ-ron đơn giản.

Pipeline cần được xây dựng linh hoạt, cho phép cấu hình các bước tiền xử lý (loại bỏ stopwords, tokenization, padding), lựa chọn kỹ thuật embedding, và mô hình phân loại (Naive Bayes, Logistic Regression, SVM, hoặc fine-tuning mô hình BERT).

Kết quả cuối cùng cần bao gồm: báo cáo phân tích EDA (thống kê số lượng từ, phân phối độ dài văn bản, tần suất từ), mô tả pipeline, các tham số đã thử nghiệm, và so sánh hiệu quả giữa cách tiếp cận truyền thống và hiện đại.

### 4.3 Bài 3: Học máy với dữ liệu dạng ảnh (Image Data)

Sinh viên được chia thành các nhóm (xem chi tiết ở mục Hình thức làm việc) và mỗi nhóm lựa chọn một tập dữ liệu ảnh để triển khai bài tập, với các ràng buộc sau:

- Các nhóm không được chọn trùng tập dữ liệu.
- Tập dữ liệu phải có số lượng ảnh đủ lớn và đa dạng để việc huấn luyện có ý nghĩa; cần thảo luận với giảng viên để thống nhất trước khi thực hiện.
- Kích thước và chất lượng ảnh nên phù hợp để có thể xử lý trên môi trường thực hành (Colab hoặc máy tính cá nhân).

Trong bài tập này, sinh viên cần thực hiện **EDA** cơ bản cho dữ liệu ảnh (kích thước ảnh, số kênh màu, phân phối nhãn, số lượng mẫu cho mỗi lớp), sau đó tiến hành xây dựng pipeline theo các bước:

- **Trích xuất đặc trưng hiện đại (deep learning features):** sử dụng các mô hình học sâu đã được huấn luyện trước (pretrained models) để rút trích đặc trưng ảnh. Gợi ý:
  - CNN: VGG16, ResNet, EfficientNet.
  - Transformer-based: Vision Transformer (ViT), Swin Transformer.

  Các đặc trưng này cần được **lưu lại thành file `.npy` hoặc `.h5`** để sử dụng trong các bước huấn luyện mô hình phân loại phía sau.

Pipeline phải cho phép linh hoạt trong việc cấu hình các bước (ví dụ: resize ảnh về kích thước nào, chọn mô hình pretrained nào để trích xuất đặc trưng, lựa chọn mô hình phân loại Logistic Regression, SVM hoặc Random Forest).

Kết quả cuối cùng cần bao gồm: báo cáo EDA, mô tả pipeline, các đặc trưng đã lưu, và so sánh hiệu quả giữa các lựa chọn mô hình học sâu khác nhau.

## 5 Yêu cầu sản phẩm

Toàn bộ bài làm phải có **front-end là Google Colab Notebook**. Sinh viên có thể (và được khuyến khích) viết thêm các **module hỗ trợ** bằng Python, sau đó import và sử dụng trực tiếp trong Colab.

File Colab nộp lên phải đảm bảo yêu cầu:

- Khi chọn chức năng `Runtime` → `Run all`, toàn bộ notebook phải chạy thành công mà không phát sinh lỗi.
- Không được mount vào các dịch vụ cloud cá nhân (Google Drive, Dropbox, ...).

- Dữ liệu cần được download từ nguồn công khai (có link cụ thể trong notebook) và giải nén trực tiếp vào môi trường máy chủ Colab.
- Các bước chuẩn bị dữ liệu, cài đặt thư viện bổ sung (nếu có) phải được ghi rõ và chạy tự động trong notebook.

Ngoài notebook, sản phẩm nộp phải bao gồm:

- **Báo cáo PDF**: trình bày EDA, pipeline, các thí nghiệm đã thực hiện, so sánh kết quả và phân tích.
- **Các file đặc trưng đã trích xuất** (embedding, vector đặc trưng) được lưu ở định dạng `.npy` hoặc `.h5`, nộp kèm theo hoặc ghi rõ đường link tải.
- **Cấu trúc thư mục** trong file nộp (`.zip`) phải rõ ràng: gồm thư mục `notebooks/`, `modules/`, `reports/`, và `features/`.
- Sinh viên được **khuyến khích** tải toàn bộ mã nguồn và tài liệu lên GitHub, sau đó chia sẻ đường link trong Colab và báo cáo. Việc này sẽ được cộng điểm khuyến khích vì thể hiện khả năng quản lý và công bố mã nguồn khoa học. README của repository GitHub cần có đầy đủ thông tin:
  - Tên môn học, mã môn học, học kỳ, năm học.
  - Thông tin giảng viên hướng dẫn (GVHD).
  - Thông tin các thành viên nhóm: họ tên, mã số sinh viên, email.
  - Mục tiêu của bài tập lớn.
  - Hướng dẫn chạy notebook (yêu cầu thư viện, cách tải dữ liệu).
  - Cấu trúc thư mục của dự án.
  - Link tới báo cáo PDF và link Colab notebook (nếu có).

# 6 Hình thức làm việc

Sinh viên cần tự chia thành các nhóm từ 2–3 thành viên. Mỗi nhóm sẽ thực hiện toàn bộ các bài tập lớn (bao gồm cả phần mở rộng nếu có) theo nhóm đã đăng ký.

- **Đăng ký nhóm:** Các nhóm cần đăng ký tên nhóm và danh sách thành viên vào trang Google Sheet được chia sẻ trên LMS của môn học.
- **Tài liệu minh chứng:** Trong quá trình làm việc, các nhóm được khuyến khích xây dựng kho tư liệu minh chứng (hình ảnh, video họp nhóm, biên bản thảo luận, ...) để chứng minh sự hợp tác thực tế.
- **Phân công công việc:** Trong báo cáo nộp cuối kỳ, mỗi nhóm bắt buộc phải có bảng phân công công việc chi tiết cho từng thành viên, kèm theo tỷ lệ phần trăm hoàn thành. Điểm của mỗi thành viên sẽ được tính theo công thức:

  **Điểm cá nhân = Điểm nhóm × Tỷ lệ đóng góp** (%).

# 7 Tiêu chí đánh giá

Bài tập lớn sẽ được chấm điểm dựa trên các tiêu chí sau:

- **Hoàn thiện pipeline truyền thống (40%):** Bao gồm đầy đủ các bước EDA, tiền xử lý, trích xuất đặc trưng, huấn luyện mô hình, đánh giá kết quả.
- **Chất lượng phân tích và thí nghiệm (25%):** So sánh nhiều cấu hình, giải thích kết quả, phân tích ưu điểm và hạn chế.
- **Báo cáo (20%):** Trình bày rõ ràng, có bảng biểu, hình ảnh minh họa, bảng phân công công việc và tỷ lệ đóng góp cá nhân.
- **Tính hoàn chỉnh của sản phẩm nộp (10%):** File Colab chạy thành công với `Run all`, cấu trúc thư mục rõ ràng, file đặc trưng được lưu lại đúng định dạng.
- **Tinh thần làm việc nhóm (5%):** Có minh chứng hoạt động nhóm (ảnh/video họp nhóm, biên bản).

Ngoài ra, sinh viên có thể nhận **điểm cộng**:

- **Pipeline học sâu (deep learning):** +5% nếu triển khai thêm để so sánh với pipeline truyền thống.
- **Công bố trên GitHub:** +5% nếu tổ chức dự án đầy đủ với README chứa thông tin môn học, giảng viên hướng dẫn, thành viên nhóm, hướng dẫn chạy, và liên kết đến báo cáo/Colab.

**Lưu ý:** Trong suốt thời gian học, các nhóm được yêu cầu nộp *báo cáo tiến độ định kỳ* (progress report) theo lịch mà giảng viên công bố. Đây là căn cứ để đánh giá sự tham gia, tiến độ, và sự phân công công việc trong nhóm.

# 8 Thời hạn & cách nộp

- **Hình thức nộp:** Bài tập lớn được nộp thông qua **thư mục Google Drive** đã được giảng viên tạo và chia sẻ cho toàn lớp.
- **Thời hạn:** Thời gian nộp cụ thể xem trên LMS của môn học.

# Phần mở rộng cho lớp Kỹ sư Tài năng

Ngoài bài tập lớn chung, các nhóm thuộc lớp Kỹ sư Tài năng sẽ thực hiện thêm một phần mở rộng. Mỗi nhóm (vẫn giữ nguyên nhóm đã làm bài tập lớn) cần chọn **một trong ba chủ đề sau** để triển khai:

1. **Bayesian Network (BN)**
    - Hiện thực mô hình Bayesian Network, cài đặt các thuật toán suy luận (inference) gồm: **suy luận chính xác** và **suy luận xấp xỉ** cơ bản.
    - Ứng dụng: giải một bài toán tiêu biểu, ví dụ phân loại hoặc dự đoán dựa trên quan hệ nhân quả trong dữ liệu.

- Giảng viên sẽ cung cấp sẵn **thư viện Python** hỗ trợ thao tác trên đồ thị DAG và thực hiện *topological sort*, nhằm giúp sinh viên tập trung vào phần hiện thực thuật toán suy luận.

2. **Hidden Markov Model (HMM)**
   - Hiện thực các thuật toán chính: Forward, Viterbi, và (tùy chọn) Baum-Welch để huấn luyện tham số.
   - Ứng dụng: giải một bài toán chuỗi thời gian tiêu biểu, ví dụ nhận dạng chuỗi ký tự, gán nhãn chuỗi hoặc phân tích tín hiệu.
3. **Conditional Random Field (CRF)**
   - Hiện thực CRF và cài đặt thuật toán suy luận cơ bản.
   - Ứng dụng: giải một bài toán gán nhãn chuỗi (sequence labeling), ví dụ nhận dạng thực thể (NER) trong văn bản hoặc phân đoạn chuỗi.

Mỗi nhóm cần:

- Xây dựng mô hình, cài đặt hoặc sử dụng thư viện thích hợp.
- Đánh giá mô hình bằng thí nghiệm cụ thể, phân tích ưu điểm và hạn chế.
- Trình bày ứng dụng tiêu biểu của kỹ thuật trong báo cáo và minh họa bằng ví dụ chạy thử.