

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



BÁO CÁO BÀI TẬP LỚN
HỆ THỐNG THÔNG MINH CO3041
HỌC KÌ 251 NĂM HỌC 2025- 2026

LOAN DEFAULT PREDICTION

GVHD: PGS.TS.Quản Thành Thơ
GROUP: 18
HVTNs: Cao Nguyễn Minh Hiếu - 2470575
Trần Đăng Hùng - 2470750

Mục lục

1	Tổng quan	6
1.1	Giới thiệu bài toán và động cơ thực tiễn	6
1.2	Đặc trưng của bài toán	6
1.3	Các bên liên quan (Stakeholders)	6
1.4	Phương pháp tiếp cận	8
1.5	Kết quả mong đợi	9
1.6	Cấu trúc báo cáo	10
2	Thiết kế hệ thống	11
2.1	Model pipeline	11
2.2	UI/UX	12
2.2.1	Giao diện cho từng stakeholders	12
2.2.2	Phân quyền và kiểm soát truy cập.	15
2.2.3	Luồng vận hành end-to-end giữa các nhóm.	15
3	Giải thuật	17
3.1	Gradient Boosting Tree (GBT)	17
3.2	XGBoost: Phiên bản tăng cường của GBT	18
3.3	LightGBM: Tối ưu tốc độ và hiệu năng	19
3.4	So sánh GBT, XGBoost và LightGBM	20
3.5	Lý do lựa chọn mô hình cho bài toán Loan Default Prediction	20
4	Dữ liệu huấn luyện	21
4.1	Tổng quan	21
4.2	Phân tích khám phá dữ liệu (Exploratory Data Analysis – EDA)	22
4.2.1	Kiểm tra dữ liệu	23
4.2.2	Phân phối các biến	24
4.2.3	Phân tích tương quan	28
5	Huấn luyện, đánh giá, và triển khai mô hình	32
5.1	Huấn luyện mô hình	32
5.2	Dánh giá mô hình	35
5.3	Triển khai mô hình	38



5.3.1	Kiến trúc triển khai	38
5.3.2	Pipeline suy luận	39
5.3.3	Giao diện Gradio phân quyền theo vai trò (RBAC)	39
5.3.4	Các khía cạnh hệ thống	43
5.3.5	Quy trình triển khai tổng thể	43
5.3.6	Vai trò của triển khai trong hệ thống thông minh	44
5.4	Theo dõi mô hình	44
6	Kết luận và hướng phát triển	47
6.1	Kết luận	47
6.2	Hướng phát triển	48



Danh sách bảng

1	So sánh các thuật toán Boosting	20
2	Bảng mô tả dữ liệu (Data Dictionary) của bộ dữ liệu Loan Default Prediction . .	22
4	Thống kê số mẫu và tỷ lệ theo nhãn cho tập Train và Test.	32
5	Kết quả huấn luyện 5 mô hình theo 5-fold CV.	34
6	Kết quả tìm ngưỡng phân loại tối ưu theo từng fold.	34



Danh sách hình vẽ

1	Giao diện người dùng của Loan Officer.	12
2	Giao diện người dùng của Risk Manager.	13
3	Giao diện người dùng của Data Scientist.	14
4	Giao diện người dùng của End User.	15
5	Số giá trị thiếu, chuỗi rỗng, bằng 0, theo từng thuộc tính.	23
6	Phân phối các thuộc tính dạng số.	25
7	Phân phối các thuộc tính dạng phân loại.	27
8	Tương quan giữa <i>Default</i> và các biến kiểu số.	29
9	Tỷ lệ vỡ nợ theo từng giá trị của biến phân loại.	30
10	Dường cong ROC-AUC trên tập test.	35
11	Confusion matrix của mô hình trên tập test.	36
12	Biểu đồ SHAP summary thể hiện độ quan trọng của các đặc trưng.	37
13	Sơ đồ kiến trúc triển khai mô hình tổng thể	38
14	Gradio phân quyền theo vai trò của Loan Officer	40
15	Gradio phân quyền theo vai trò của Risk Manager	41
16	Gradio phân quyền theo vai trò của Data Scientist	42
17	Gradio phân quyền theo vai trò của User	43
18	Sơ đồ kiến trúc triển khai mô hình chi tiết	44



Các đường dẫn bài nộp

- Shared Drive: https://drive.google.com/drive/folders/11RgZe0mrABdSIz4LBqKIElIGi_1_0_ku?usp=sharing
- Dataset: <https://www.kaggle.com/datasets/nikhil1e9/loan-default/data>
- Repository: https://github.com/hieucnm/bku_intelligent_system
- Mockup Video: <https://youtu.be/fKLkMW883BY>



1 Tổng quan

1.1 Giới thiệu bài toán và động cơ thực tiễn

Trong lĩnh vực tài chính - ngân hàng, việc đánh giá khả năng khách hàng có hoàn trả khoản vay đúng hạn hay không là một vấn đề trọng yếu, ảnh hưởng trực tiếp đến hiệu quả quản trị rủi ro tín dụng. Khi số lượng khách hàng ngày càng tăng cùng với sự đa dạng về hành vi tài chính, việc dựa hoàn toàn vào các tiêu chí thủ công hoặc đánh giá định tính trở nên kém hiệu quả. Do đó, nhu cầu xây dựng các mô hình dự đoán tự động, chính xác và có khả năng mở rộng đã trở thành xu thế tất yếu trong các hệ thống tín dụng hiện đại.

Với sự phát triển của học máy (Machine Learning), các tổ chức tài chính ngày càng có khả năng tận dụng dữ liệu lịch sử để xây dựng các mô hình dự đoán xác suất vỡ nợ, giúp **ra quyết định cấp tín dụng một cách tự động, nhanh chóng và chính xác hơn**.

Dề tài “*Loan Default Prediction – Dự đoán rủi ro vỡ nợ*” được nhóm thực hiện với mục tiêu mô phỏng quy trình ứng dụng một mô hình học máy trong thực tiễn nghiệp vụ, từ việc xây dựng mô hình đến thiết kế giao diện phục vụ các bên liên quan trong tổ chức.

1.2 Đặc trưng của bài toán

Trong thực tiễn, bài toán dự đoán vỡ nợ là một **bài toán phân loại nhị phân (binary classification)**, trong đó:

- **Đầu vào:** tập các đặc trưng của khách hàng và khoản vay, bao gồm thông tin cá nhân (như tuổi, giới tính, thu nhập, ...), thông tin lịch sử tài chính (như tỷ lệ nợ, điểm tín dụng, số khoản vay trước, ...), và thông tin khoản vay hiện tại (số tiền vay, loại vay, thời gian làm việc, v.v.).
- **Đầu ra:** nhãn $y \in \{0, 1\}$, với $y = 1$ biểu thị khách hàng vỡ nợ (default), $y = 0$ biểu thị khách hàng trả đúng hạn (non-default).

Mục tiêu là dự đoán xác suất $P(y = 1 | X)$ cho mỗi khách hàng, từ đó hỗ trợ các quyết định duyệt vay hoặc quản lý rủi ro.

1.3 Các bên liên quan (Stakeholders)

Việc triển khai hệ thống dự đoán rủi ro vỡ nợ liên quan đến nhiều nhóm đối tượng khác nhau trong tổ chức tài chính cũng như khách hàng bên ngoài. Mỗi nhóm **stakeholder** đều có mục



tiêu và lợi ích riêng khi sử dụng kết quả từ mô hình. Phần này xác định các bên liên quan chính và mô tả lợi ích tiềm năng mà họ có thể nhận được.

- **Chuyên viên tín dụng (Loan Officers):** là người trực tiếp nhận và thẩm định hồ sơ vay. Hệ thống hỗ trợ họ bằng cách cung cấp xác suất vỡ nợ dự đoán cho từng khách hàng và hiển thị các yếu tố quan trọng ảnh hưởng đến quyết định mô hình. Điều này giúp nhân viên tín dụng ra quyết định cấp vay một cách nhanh chóng, khách quan và có cơ sở dữ liệu hỗ trợ thay vì chỉ dựa vào kinh nghiệm chủ quan.
- **Ban Giám đốc/Ban quản lý rủi ro (Managers):** chịu trách nhiệm giám sát rủi ro tín dụng ở cấp tổ chức. Nhóm này có thể theo dõi xu hướng rủi ro tổng thể, tỷ lệ vỡ nợ theo thời gian, và phân bổ khách hàng theo nhóm rủi ro (thấp – trung bình – cao). Dashboard cung cấp thông tin trực quan giúp họ đưa ra quyết định chiến lược, điều chỉnh chính sách tín dụng, và cân đối giữa mục tiêu tăng trưởng và kiểm soát rủi ro.
- **Chuyên viên khoa học dữ liệu (Data Scientists):** là nhóm kỹ thuật đảm nhiệm việc phân tích dữ liệu, huấn luyện và đánh giá mô hình. Các biểu đồ như ROC Curve, Confusion Matrix, cùng với các chỉ số hiệu năng (Accuracy, Precision, Recall, AUC) giúp họ giám sát chất lượng mô hình và phát hiện hiện tượng drift dữ liệu hoặc giảm hiệu năng theo thời gian. Lợi ích của nhóm này là khả năng cải tiến mô hình liên tục dựa trên phản hồi từ thực tế.
- **Khách hàng vay (End Users):** mặc dù không trực tiếp tương tác với toàn bộ hệ thống nội bộ, họ là đối tượng hưởng lợi gián tiếp từ mô hình. Giao diện người dùng có thể cho phép họ tra cứu điểm tín dụng tổng quát hoặc nhận phản hồi tại sao hồ sơ bị từ chối (ví dụ: tỷ lệ nợ cao, thu nhập chưa đủ, lịch sử tín dụng kém). Tuy nhiên, khác với nhóm Loan Officers, khi hiển thị lý do chấp nhận / từ chối cho vay cho người dùng cuối, chỉ một số yếu tố cụ thể của mô hình được hiển thị, nhằm đảm bảo bảo mật dữ liệu và tính toàn vẹn của hệ thống.

Các bên liên quan mở rộng: Ngoài các bên chính kể trên, hệ thống còn có sự tham gia của các bộ phận mở rộng khác. Tuy nhiên, nhóm đề xuất không dựng UI cho các bộ phận này vì họ hầu như tương tác trực tiếp với hệ thống:

- **Risk & Compliance:** chịu trách nhiệm giám sát các quyết định tín dụng, đảm bảo mô hình không vi phạm chính sách và quy định pháp lý.



- **IT/Operations:** phụ trách hạ tầng triển khai, bảo mật, sao lưu dữ liệu và đảm bảo uptime của hệ thống.
- **Business Analytics:** phân tích báo cáo từ dashboard, đề xuất điều chỉnh chính sách dựa trên hành vi thực tế.
- ... và các bộ phận đặc thù khác của doanh nghiệp.

Như vậy, hệ thống không chỉ giúp tự động hóa và tối ưu hóa quy trình phê duyệt tín dụng mà còn tạo ra giá trị cho nhiều cấp độ trong tổ chức, mang lại lợi ích cho tất cả các bên.

1.4 Phương pháp tiếp cận

Fang pháp tiếp cận của dự án được xây dựng nhằm mô phỏng toàn bộ quy trình triển khai một hệ thống dự đoán rủi ro vỡ nợ trong môi trường thực tế. Ở đây, nhóm trình bày quy trình này theo hai cấp độ: (1) phạm vi **dự án học thuật** hiện tại và (2) định hướng **triển khai thực tế trong doanh nghiệp**.

1. Phương pháp tiếp cận trong phạm vi dự án

Trong phạm vi báo cáo này, nhóm triển khai quy trình dự đoán vỡ nợ dựa trên dữ liệu công khai, với các bước cụ thể như sau:

1. **Thu thập dữ liệu:** sử dụng tập dữ liệu công khai (như đính kèm ở đầu báo cáo) bao gồm các đặc trưng được cung cấp sẵn. Đây là toàn bộ dữ liệu mà nhóm có, không phát sinh thêm theo thời gian. Nhóm chia tập dữ liệu này thành dữ liệu huấn luyện và dữ liệu kiểm thử.
2. **Tiền xử lý dữ liệu:** làm sạch dữ liệu, xử lý giá trị thiếu, loại bỏ nhiễu, mã hóa biến phân loại, và chuẩn hóa các biến định lượng để đảm bảo đầu vào cho mô hình nhất quán và chất lượng.
3. **Xây dựng mô hình:** huấn luyện mô hình học máy nhằm dự đoán xác suất vỡ nợ của khách hàng. Loại mô hình được chọn dựa trên khả năng xử lý dữ liệu dạng bảng và hiệu năng cao trong các bài toán tài chính.
4. **Đánh giá mô hình:** sử dụng tập kiểm thử để đánh giá độ chính xác của mô hình.
5. **Triển khai giao diện:** xây dựng một ứng dụng mô phỏng, gồm các giao diện tương ứng với các nhóm stakeholders, minh họa cách mô hình được sử dụng trong thực tế nghiệp vụ.



2. Phương pháp tiếp cận trong triển khai thực tế

Trong môi trường doanh nghiệp thực, quy trình triển khai một hệ thống dự đoán vỡ nợ có thể được mở rộng và tinh chỉnh như sau:

- Thu thập dữ liệu:** dữ liệu được lấy từ hệ thống quản lý tín dụng nội bộ của tổ chức (Loan Management System, CRM, Core Banking) và có thể được bổ sung bằng các nguồn dữ liệu bên ngoài như dữ liệu tín dụng quốc gia, hành vi tiêu dùng, dữ liệu viễn thông, hoặc dữ liệu định danh (KYC). Trong một số trường hợp, tổ chức có thể *mua dữ liệu bên thứ ba* để tăng độ bao phủ.
- Tiền xử lý và quản lý dữ liệu:** dữ liệu được làm sạch, mã hóa và chuẩn hóa thông qua pipeline tự động; có thể được lưu trữ và xử lý trong hệ thống Big Data (như Spark hoặc Data Warehouse) để đảm bảo khả năng mở rộng và bảo mật.
- Huấn luyện và tối ưu mô hình:** mô hình được huấn luyện định kỳ trên dữ liệu lịch sử mới, áp dụng các kỹ thuật tối ưu siêu tham số (hyperparameter tuning), cross-validation và giám sát độ trôi dữ liệu (data drift).
- Triển khai (Deployment):** mô hình được đóng gói dưới dạng API hoặc dịch vụ microservice và tích hợp vào quy trình phê duyệt tín dụng thực tế, giúp các bộ phận nghiệp vụ sử dụng trực tiếp qua hệ thống nội bộ.
- Giám sát và cập nhật mô hình:** theo dõi hiệu năng mô hình sau triển khai (Model Monitoring), đánh giá định kỳ độ chính xác và fairness, đồng thời tái huấn luyện khi phát hiện thay đổi trong hành vi người vay hoặc thị trường.

Cách tiếp cận hai tầng này giúp nhóm vừa đảm bảo **khả năng thực hiện trong phạm vi học thuật**, vừa thể hiện được **tư duy hệ thống và quy trình vận hành thực tế** – điều cần thiết nếu hệ thống được triển khai trong một tổ chức tài chính thực thụ.

1.5 Kết quả mong đợi

Kết quả của dự án gồm:

- Một mô hình học máy có khả năng dự đoán xác suất vỡ nợ với độ chính xác cao ($AUC > 0.85$).
- Hiểu rõ mối quan hệ giữa các đặc trưng tài chính và rủi ro vỡ nợ của khách hàng.



-
- Một giao diện tương tác minh họa rõ ràng lợi ích của mô hình trong thực tế nghiệp vụ tín dụng, cho từng bên liên quan.

1.6 Cấu trúc báo cáo

Báo cáo được tổ chức theo các chương như sau:

- **Chương 1 – Tổng quan:** Giới thiệu tổng thể về bài toán dự đoán rủi ro vỡ nợ, bối cảnh thực tiễn, cùng các nhóm *stakeholders* chính và vai trò của họ trong hệ thống.
- **Chương 2 – Thiết kế hệ thống:** Trình bày cấu trúc tổng thể của hệ thống dự đoán rủi ro, bao gồm:
 - **Backend:** mô tả cách thu thập và lưu trữ dữ liệu, cấu trúc file/dataset, và cơ chế cập nhật dữ liệu mới.
 - **Model Pipeline:** giới thiệu luồng xử lý dữ liệu và huấn luyện mô hình ở mức khái quát; chi tiết sẽ được trình bày trong Chương 5.
 - **UI/UX:** mô tả thiết kế giao diện người dùng, phục vụ các nhóm *stakeholders* đã nêu.
- **Chương 3 – Giải thuật Gradient Boosting Tree:** Giới thiệu cơ sở lý thuyết của phương pháp *Gradient Boosting*, các biến thể như *XGBoost* và *LightGBM*, cùng với lý do lựa chọn mô hình này cho bài toán dự đoán vỡ nợ.
- **Chương 4 – Dữ liệu huấn luyện:** Mô tả chi tiết về tập dữ liệu sử dụng trong dự án:
 - **Tổng quan:** nguồn gốc và đặc trưng của dữ liệu.
 - **Phân tích thống kê mô tả (EDA):** khám phá phân bố, tương quan giữa các biến và đặc điểm của nhãn mục tiêu.
 - **Tiền xử lý (Preprocessing):** các bước làm sạch dữ liệu, xử lý giá trị thiếu, mã hóa biến phân loại, chuẩn hóa thang đo, v.v.
- **Chương 5 – Huấn luyện và triển khai mô hình:** Trình bày quy trình phát triển, huấn luyện, đánh giá, và triển khai mô hình dự đoán
- **Chương 6 – Kết luận và hướng phát triển:** Tổng hợp kết quả báo cáo, các kết luận chính, và hướng phát triển của dự án trong tương lai.



2 Thiết kế hệ thống

2.1 Model pipeline

Chuyển đổi điểm rủi ro thành quyết định nghiệp vụ. Đầu ra của mô hình là một *default score / risk score* nằm trong khoảng [0, 1], biểu diễn xác suất default. Dựa trên kết quả này, hệ thống áp dụng quy tắc ngưỡng để chuyển đổi thành các nhóm quyết định (các giá trị ngưỡng dưới đây đang là giá trị đề xuất, sẽ được cập nhật sau khi train xong model):

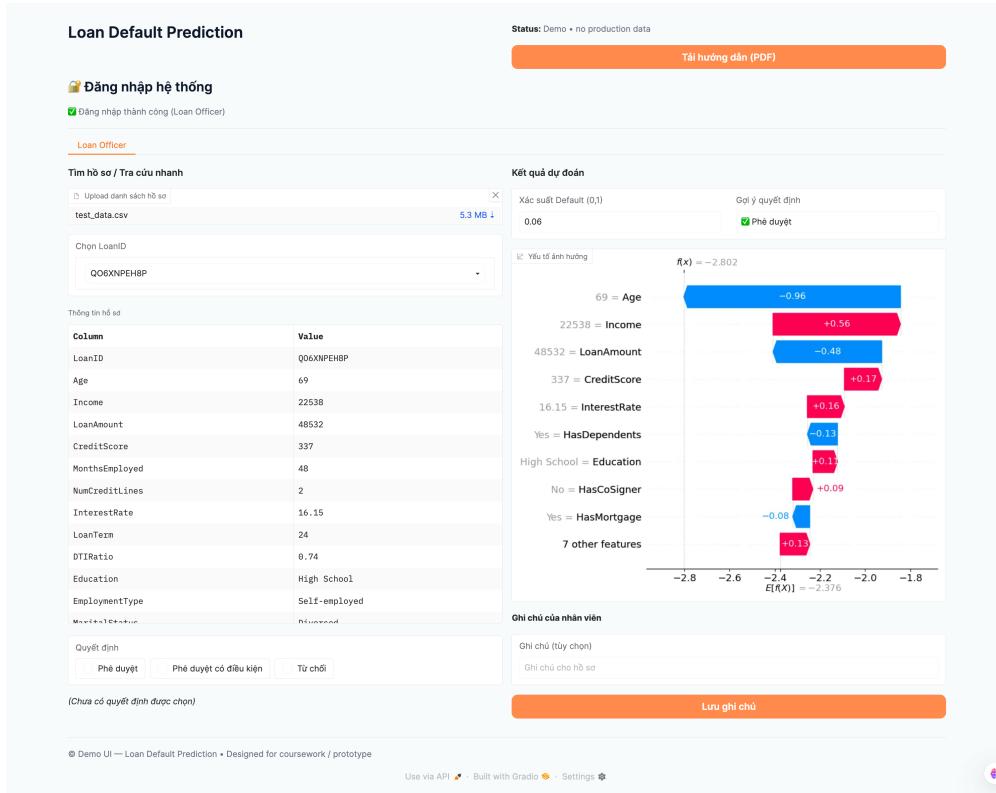
- **Approved:** nếu default score < 0.176
- **Rejected:** nếu default score ≥ 0.176

Các ngưỡng trên được xác định thông qua quá trình phân tích nhằm cân bằng giữa 2 tiêu chí: độ chính xác và độ phủ (Precision và Recall). Các giá trị này hoàn toàn có thể được điều chỉnh định kỳ theo hiệu năng thực tế của mô hình hoặc yêu cầu từ bộ phận quản trị rủi ro.

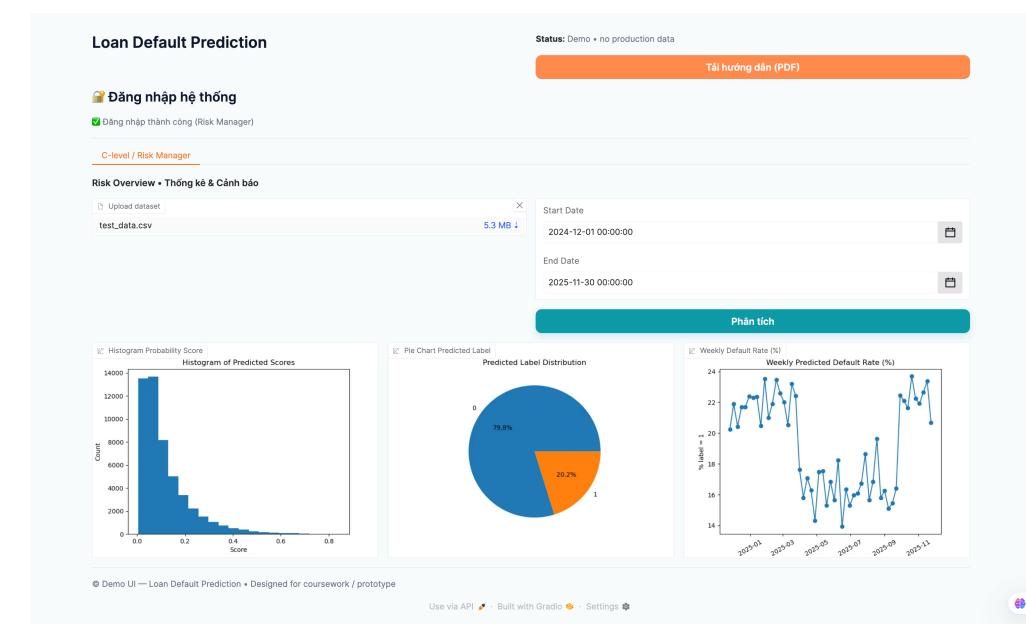


2.2 UI/UX

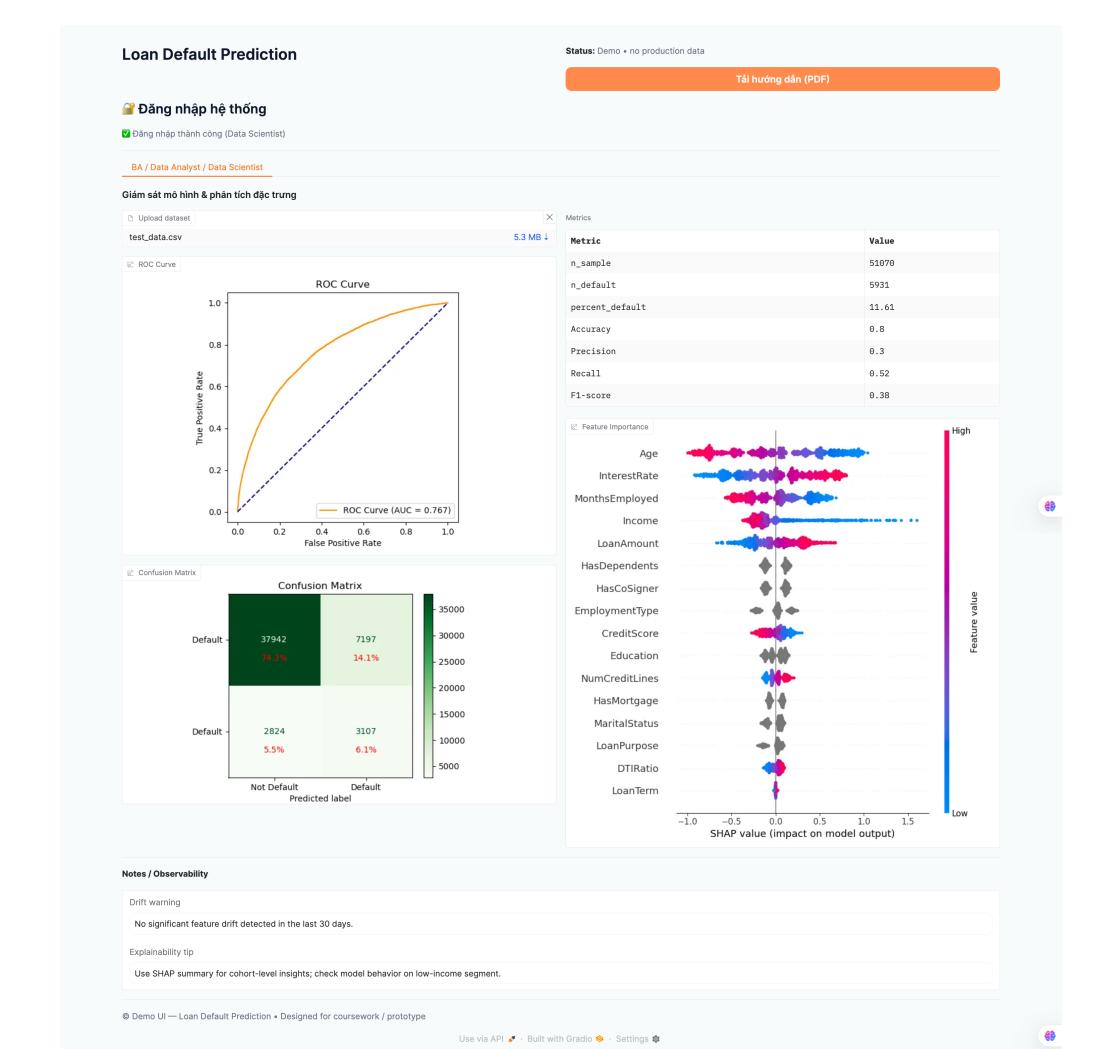
2.2.1 Giao diện cho từng stakeholders



Hình 1: Giao diện người dùng của Loan Officer.



Hình 2: Giao diện người dùng của Risk Manager.



Hình 3: Giao diện người dùng của Data Scientist.



Loan Default Prediction

Status: Demo • no production data

[Tải hướng dẫn \(PDF\)](#)

Đăng nhập hệ thống

Đăng nhập thành công (End User)

Người dùng cuối — Tra cứu kết quả

Tra cứu kết quả vay vốn

Số điện thoại
0909123456

Gửi mã OTP

Nhập OTP
847991

Xác thực & Tra cứu

Trạng thái tra cứu
 Xác thực thành công!

Kết quả tra cứu

Họ tên	Duyệt lúc	Kết quả	Lý do
Nguyen Van A	2025-11-29 15:24:00	<input checked="" type="checkbox"/> Được phê duyệt	Điểm tín dụng cao (CreditScore 789), thể hiện lịch sử tín dụng tốt và thói quen trả nợ ổn định. Thời gian làm việc dài (102 tháng) trong hình thức Full-time, cho thấy nguồn thu nhập tương đối ổn định. Số lượng dòng tín dụng hiện tại không quá cao (4 credit lines), giảm áp lực trả nợ. Mục đích vay là Education, thường gần với mức rủi ro thấp hơn so với các mục đích kinh doanh hoặc tiêu dùng quy mô lớn.

© Demo UI — Loan Default Prediction • Designed for coursework / prototype

Use via API · Built with Gradio · Settings

Hình 4: Giao diện người dùng của End User.

2.2.2 Phân quyền và kiểm soát truy cập.

Hệ thống áp dụng cơ chế *role-based access control (RBAC)* nhằm đảm bảo mỗi nhóm người dùng chỉ truy cập được thông tin phù hợp:

- **Loan Officer:** xem kết quả dự đoán của model cho từng hồ sơ và ra quyết định *Approved* / *Rejected* hoặc *Conditional*.
- **Risk Officer:** xem báo cáo tổng thể điểm Default.
- **Data Scientist:** xem hiệu suất của mô hình trên tập các hồ sơ.
- **End User:** xem kết quả duyệt hồ sơ của chính mình.

Tất cả hành động truy cập đều được ghi lại trong nhật ký bảo mật để phục vụ kiểm toán.

2.2.3 Luồng vận hành end-to-end giữa các nhóm.

Quy trình tổng thể được thiết kế như một dòng chảy khép kín:

1. Bộ phận **Sales** nhập hồ sơ khách hàng vào hệ thống.
2. Dữ liệu được chuyển tới **Model API**, nơi mô hình dự đoán xác suất rủi ro và trả về kết quả.



HCMUT - Khoa Khoa học và Kỹ thuật máy tính

GVHD: PGS.TS. Quản Thành Thơ. HVTHs: Cao Nguyễn Minh Hiếu - 2470575 & Trần Đăng Hùng - 2470750

-
3. Bộ phận **Risk/Compliance** rà soát các hồ sơ “Conditional Approval”.
 4. **Data Science team** định kỳ tổng hợp thống kê hiệu năng, nhận phản hồi từ nghiệp vụ và cập nhật mô hình nếu cần.

Luồng tương tác này được thể hiện trực quan trên giao diện dashboard, đảm bảo mọi bên liên quan đều nắm được vị trí và vai trò của mình trong chu trình ra quyết định.



3 Giải thuật

Trong phần này, chúng tôi trình bày tổng quan về các giải thuật được sử dụng trong hệ thống, bao gồm: Gradient Boosting Tree (GBT), XGBoost và LightGBM. Đây đều là các thuật toán thuộc họ mô hình Gradient Boosting, nổi tiếng với khả năng xử lý dữ liệu dạng bảng và mang lại hiệu năng cao trong nhiều bài toán phân loại.

3.1 Gradient Boosting Tree (GBT)

Gradient Boosting Tree (GBT) là thuật toán học máy mạnh mẽ được đề xuất ban đầu bởi Friedman [5]. Ý tưởng cốt lõi của Gradient Boosting là xây dựng mô hình theo hướng tăng cường dần dần: mỗi cây quyết định được huấn luyện để sửa lỗi của mô hình trước đó, bằng cách tối thiểu hóa hàm mất mát theo hướng gradient.

Gradient Boosting Tree (GBT) xây dựng mô hình dự đoán dưới dạng tổng của nhiều cây quyết định nhỏ (shallow decision trees):

$$\hat{y}(x) = \sum_{m=1}^M f_m(x),$$

trong đó $f_m(x)$ là cây quyết định thứ m , và mỗi cây đóng vai trò hiệu chỉnh (corrective model) cho các sai lệch còn tồn đọng tại các bước trước đó.

Quy trình huấn luyện GBT bao gồm:

1. Khởi tạo mô hình ban đầu:

$$F_0(x) = \arg \min_c \sum_i L(y_i, c).$$

2. Lặp qua từng vòng boosting (với $m = 1..M$):

(a) Tính gradient của hàm mất mát theo từng mẫu:

$$r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right].$$

(b) Huấn luyện một cây quyết định nhỏ dựa trên gradient r_{im} .

(c) Cập nhật mô hình:

$$F_m(x) = F_{m-1}(x) + \eta f_m(x),$$



trong đó η là hệ số học (learning rate).

3. Dự đoán cuối cùng: tổng hợp toàn bộ cây đã học.

Các ưu điểm nổi bật của Gradient Boosting gồm:

- khả năng mô hình hóa quan hệ phi tuyến,
- hiệu quả cao trên dữ liệu thực tế,
- giải thích được thông qua độ quan trọng đặc trưng.

3.2 XGBoost: Phiên bản tăng cường của GBT

XGBoost (Extreme Gradient Boosting) được phát triển bởi Chen và Guestrin [6], là một trong những thuật toán Boosting được sử dụng rộng rãi nhất trong các cuộc thi Kaggle và các ứng dụng thực tế. XGBoost (Extreme Gradient Boosting) là một mở rộng mạnh mẽ của GBT với các cải tiến đáng chú ý:

(1) Regularization

XGBoost bổ sung hệ số điều chỉnh nhằm giảm overfitting:

$$\Omega(f) = \gamma T + \frac{\lambda}{2} \sum_{j=1}^T w_j^2,$$

trong đó T là số lá (leaves) của cây, và w_j là trọng số của lá thứ j .

(2) Tối ưu hóa quá trình xây dựng cây

XGBoost sử dụng:

- parallelized split finding,
- pruning để loại bỏ nhánh kém hiệu quả,
- cơ chế xử lý giá trị thiếu tự động.



(3) Hỗ trợ dữ liệu mất cân bằng

Đặc biệt quan trọng đối với bài toán dự đoán vỡ nợ, XGBoost hỗ trợ tham số:

$$\text{scale_pos_weight} = \frac{\text{negative samples}}{\text{positive samples}},$$

giúp cải thiện độ phủ (recall) của lớp thiểu số.

3.3 LightGBM: Tối ưu tốc độ và hiệu năng

LightGBM, được giới thiệu bởi Ke et al. [7], là một biến thể tối ưu hoá của Gradient Boosting Tree do Microsoft phát triển, tập trung vào việc tăng tốc và giảm bộ nhớ. LightGBM sử dụng hai kỹ thuật chính:

- **Leaf-wise growth** với giới hạn độ sâu, cho phép tăng tốc hội tụ và giảm lỗi huấn luyện nhanh hơn so với XGBoost;
- **Histogram-based splitting**, giúp giảm đáng kể chi phí tính toán và bộ nhớ khi tìm điểm chia.

Ngoài ra, LightGBM còn có hai kỹ thuật quan trọng:

- *Gradient-based One-Side Sampling (GOSS)*: ưu tiên giữ lại những mẫu có gradient lớn;
- *Exclusive Feature Bundling (EFB)*: gom các đặc trưng hiếm xảy ra cùng lúc (mutually exclusive) để giảm số chiều.

Nhờ đó, LightGBM phù hợp với dữ liệu lớn và có nhiều đặc trưng dạng phân loại hoặc rời rạc. Trong bài toán phân loại rủi ro vỡ nợ, LightGBM được ưu tiên sử dụng do tốc độ nhanh, hỗ trợ tốt dữ liệu mất cân bằng và khả năng mở rộng khi triển khai thực tế.



3.4 So sánh GBT, XGBoost và LightGBM

Tiêu chí	GBT	XGBoost	LightGBM
Tốc độ xử lý	Chậm	Nhanh	Rất nhanh
Regularization	Không	Có	Có
Xử lý missing values	Không	Có	Có
Memory efficiency	Thấp	Trung bình	Cao
Hiệu năng trên dữ liệu lớn	Trung bình	Cao	Rất cao
Nguy cơ overfitting	Thấp	Trung bình	Cao

Bảng 1: So sánh các thuật toán Boosting

3.5 Lý do lựa chọn mô hình cho bài toán Loan Default Prediction

Mặc dù cả XGBoost và LightGBM đều là các thuật toán Boosting mạnh mẽ, LightGBM thường có hiệu năng cao hơn trong các bài toán tín dụng nhờ:

- Thời gian huấn luyện nhanh hơn;
- Hỗ trợ tốt dữ liệu nhiều đặc trưng phân loại;
- Khả năng xử lý bộ dữ liệu lớn nhờ Histogram-based splitting và EFB;
- Leaf-wise growth giúp mô hình học tốt các tương tác phức tạp giữa các thuộc tính.

Do đó, LightGBM được lựa chọn làm mô hình chính cho hệ thống, trong khi XGBoost và CatBoost [8] cũng được xem xét như những lựa chọn thay thế khả thi trong tương lai.



4 Dữ liệu huấn luyện

Ở phạm vi học thuật, nhóm chọn một bộ dữ liệu công khai (đường dẫn đặt tên đầu báo cáo) để huấn luyện và đánh giá mô hình.

4.1 Tổng quan

Bộ dữ liệu này được lấy từ Cuộc thi Dự đoán Vỡ nợ Khoản vay (Loan Default Prediction Challenge) trên Coursera. Với bộ dữ liệu này, người tham gia được mời xây dựng một mô hình phân loại nhị phân nhằm dự đoán liệu người vay có khả năng vỡ nợ hay không, dựa trên các đặc điểm nhân khẩu học, thông tin tín dụng và đặc trưng của khoản vay.

Về quy mô và cấu trúc, bộ dữ liệu bao gồm 255,347 bản ghi (dòng) và 18 thuộc tính (cột). Các cột này thể hiện sự kết hợp giữa đặc điểm của người vay (như tuổi, điểm tín dụng, thu nhập, hoặc lịch sử việc làm), thông tin của khoản vay (số tiền vay, lãi suất, thời hạn), cùng với biến mục tiêu cho biết người vay có vỡ nợ hay không. Sự kết hợp này giúp người xây dựng mô hình có một tập đặc trưng phong phú để khám phá các mối quan hệ, mô hình hóa hành vi, và cuối cùng là phát triển bộ phân loại dự đoán.

Về tỷ lệ nhãn, bộ dữ liệu này có 29,653 nhãn 1 - ứng với trường hợp vỡ nợ, và 225,694 nhãn 0 - ứng với trường hợp không vỡ nợ. Tỷ lệ nhãn 1 là khoảng 11%. Nhóm đánh giá tỷ lệ này là tương đối mất cân bằng, ta sẽ cần các metrics và cách đánh giá phù hợp đối với dạng dữ liệu mất cân bằng như vậy.

Ý nghĩa các cột được mô tả như sau:



Bảng 2: Bảng mô tả dữ liệu (Data Dictionary) của bộ dữ liệu Loan Default Prediction

Tên cột	Loại cột	Kiểu dữ liệu	Mô tả
LoanID	Identifier	string	Mã định danh duy nhất cho mỗi khoản vay.
Age	Feature	integer	Tuổi của người vay.
Income	Feature	integer	Thu nhập hàng năm của người vay.
LoanAmount	Feature	integer	Số tiền vay.
CreditScore	Feature	integer	Điểm tín dụng của người vay, phản ánh khả năng trả nợ.
MonthsEmployed	Feature	integer	Số tháng người vay đã đi làm.
NumCreditLines	Feature	integer	Số lượng tài khoản tín dụng mà người vay đang mở.
InterestRate	Feature	float	Lãi suất của khoản vay.
LoanTerm	Feature	integer	Thời hạn khoản vay tính theo tháng.
DTIRatio	Feature	float	Tỷ lệ nợ trên thu nhập (Debt-to-Income), phản ánh mức nợ so với thu nhập của người vay.
Education	Feature	string	Trình độ học vấn cao nhất của người vay (PhD, Thạc sĩ, Cử nhân, Trung học).
EmploymentType	Feature	string	Loại hình công việc của người vay (Toàn thời gian, Bán thời gian, Tự doanh, Thất nghiệp).
MaritalStatus	Feature	string	Tình trạng hôn nhân của người vay (Độc thân, Đã kết hôn, Ly dị).
HasMortgage	Feature	string	Người vay có thể chấp nhận hay không (Có/Không).
HasDependents	Feature	string	Người vay có người phụ thuộc hay không (Có/Không).
LoanPurpose	Feature	string	Mục đích khoản vay (Nhà, Ô tô, Giáo dục, Kinh doanh, Khác).
HasCoSigner	Feature	string	Khoản vay có người đồng ký hay không (Có/Không).
Default	Target	integer	Biến mục tiêu nhị phân, cho biết khoản vay có bị vỡ nợ (1) hay không (0).

4.2 Phân tích khám phá dữ liệu (Exploratory Data Analysis – EDA)

Phân tích khám phá dữ liệu (EDA) là bước quan trọng nhằm hiểu rõ cấu trúc, phân phối và các đặc điểm quan trọng của dữ liệu trước khi xây dựng mô hình dự đoán. Các nội dung chính



bao gồm: kiểm tra tính toàn vẹn của dữ liệu, khám phá phân phối của biến, và phân tích mối quan hệ giữa các thuộc tính với nhãn *Default*.

4.2.1 Kiểm tra dữ liệu

(a) **Kiểm tra dữ liệu thiếu (Missing Values)** Đầu tiên, nhóm tiến hành thống kê số lượng giá trị bị thiếu theo từng cột. Kết quả được minh họa trong Hình 5.

	null_count	empty_str_count	zero_count
LoanID	0	0	0
Age	0	0	0
Income	0	0	0
LoanAmount	0	0	0
CreditScore	0	0	0
MonthsEmployed	0	0	2122
NumCreditLines	0	0	0
InterestRate	0	0	0
LoanTerm	0	0	0
DTIRatio	0	0	0
Education	0	0	0
EmploymentType	0	0	0
MaritalStatus	0	0	0
HasMortgage	0	0	0
HasDependents	0	0	0
LoanPurpose	0	0	0
HasCoSigner	0	0	0
Default	0	0	225694

Hình 5: Số giá trị thiếu, chuỗi rỗng, bằng 0, theo từng thuộc tính.

Nhìn chung, tập dữ liệu không có giá trị thiếu (missing values) nào, với các cột categorical thì không có giá trị chuỗi rỗng (empty string) nào, và với các cột numerical thì không có giá trị bằng 0 bất thường nào (các cột *MonthsEmployed* và *Default* có giá trị bằng 0 là hoàn toàn bình



thường). Do đó, ta không cần thực hiện các bước xử lý giá trị thiếu và tất cả các thuộc tính có thể được sử dụng trực tiếp cho các bước phân tích và huấn luyện mô hình.

(b) Kiểm tra dữ liệu trùng lặp (Duplicated Rows).

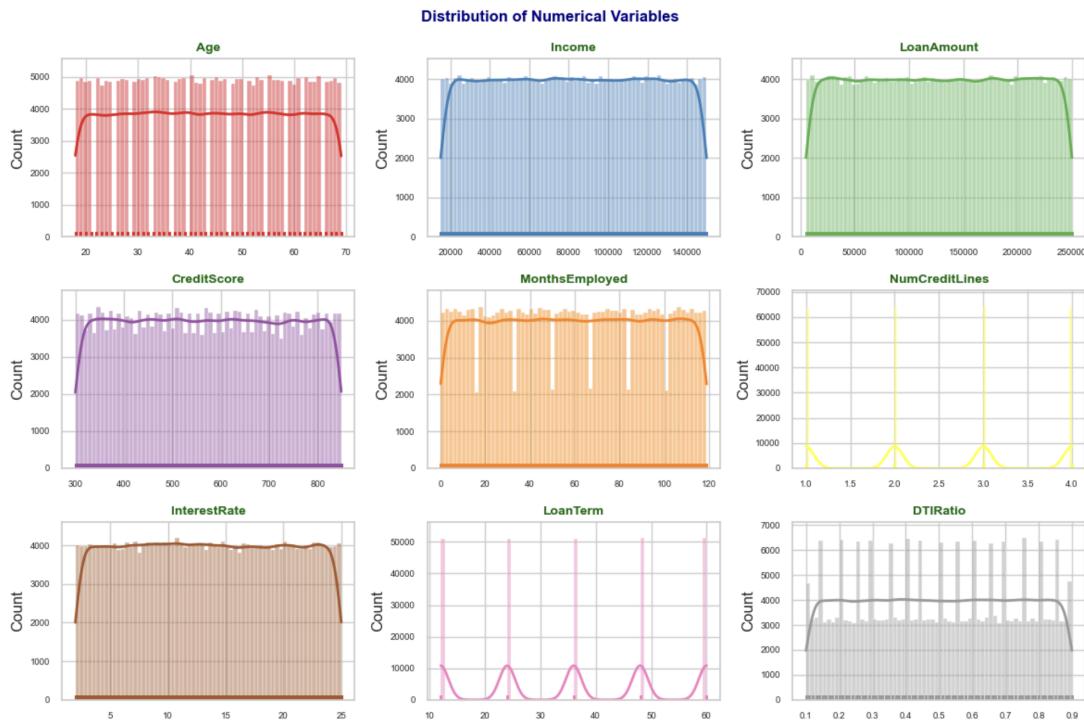
```
>>> df.drop(columns=["LoanID"]).duplicated().sum()  
0  
>>> df.drop(columns=["LoanID", "Default"]).duplicated().sum()  
0
```

Nhóm tiến hành kiểm tra các dòng dữ liệu trùng lặp theo toàn bộ các thuộc tính và nhận thấy tập dữ liệu không có bản ghi nào bị trùng lặp. Do đó, tất cả các dòng dữ liệu đều có thể sử dụng trực tiếp cho các bước phân tích và huấn luyện mô hình.

4.2.2 Phân phối các biến

4.2.2.1 Phân phối các biến liên tục (Numerical Features)

Nhóm phân tích phân phối của các thuộc tính dạng số gồm: *Age*, *Income*, *LoanAmount*, *CreditScore*, *MonthsEmployed*, *NumCreditLines*, *InterestRate*, *LoanTerm*, *DTIRatio*. Hình 6 mô tả các histogram phân phối của các biến này.



Hình 6: Phân phối các thuộc tính dạng số.

Nhận xét.

- **Age:** phân phối gần như đồng đều trong khoảng 18–70 tuổi, không có độ lệch rõ rệt, cho thấy tập khách hàng trải dài ở nhiều nhóm tuổi.
- **Income:** phân phối khá đồng đều trên toàn bộ khoảng 15,000–150,000, không xuất hiện outlier rõ rệt, phù hợp cho việc huấn luyện mô hình.
- **LoanAmount:** cũng phân phối gần như đồng đều từ mức thấp tới cao, cho thấy đa dạng khoản vay trong dữ liệu.
- **CreditScore:** phân phối đều từ 300–850, cho thấy dữ liệu bao quát toàn bộ thang điểm tín dụng tiêu chuẩn.
- **MonthsEmployed:** phân phối đều từ 0–120 tháng, cho thấy dữ liệu có khách hàng mới đi làm và khách hàng đã đi làm lâu năm.
- **NumCreditLines:** thuộc tính này là các số đếm, nên nhìn phân phối ta thấy có các spike ở các giá trị nguyên, điều này là bình thường. Ta cũng thấy số lượng tại các giá trị nguyên



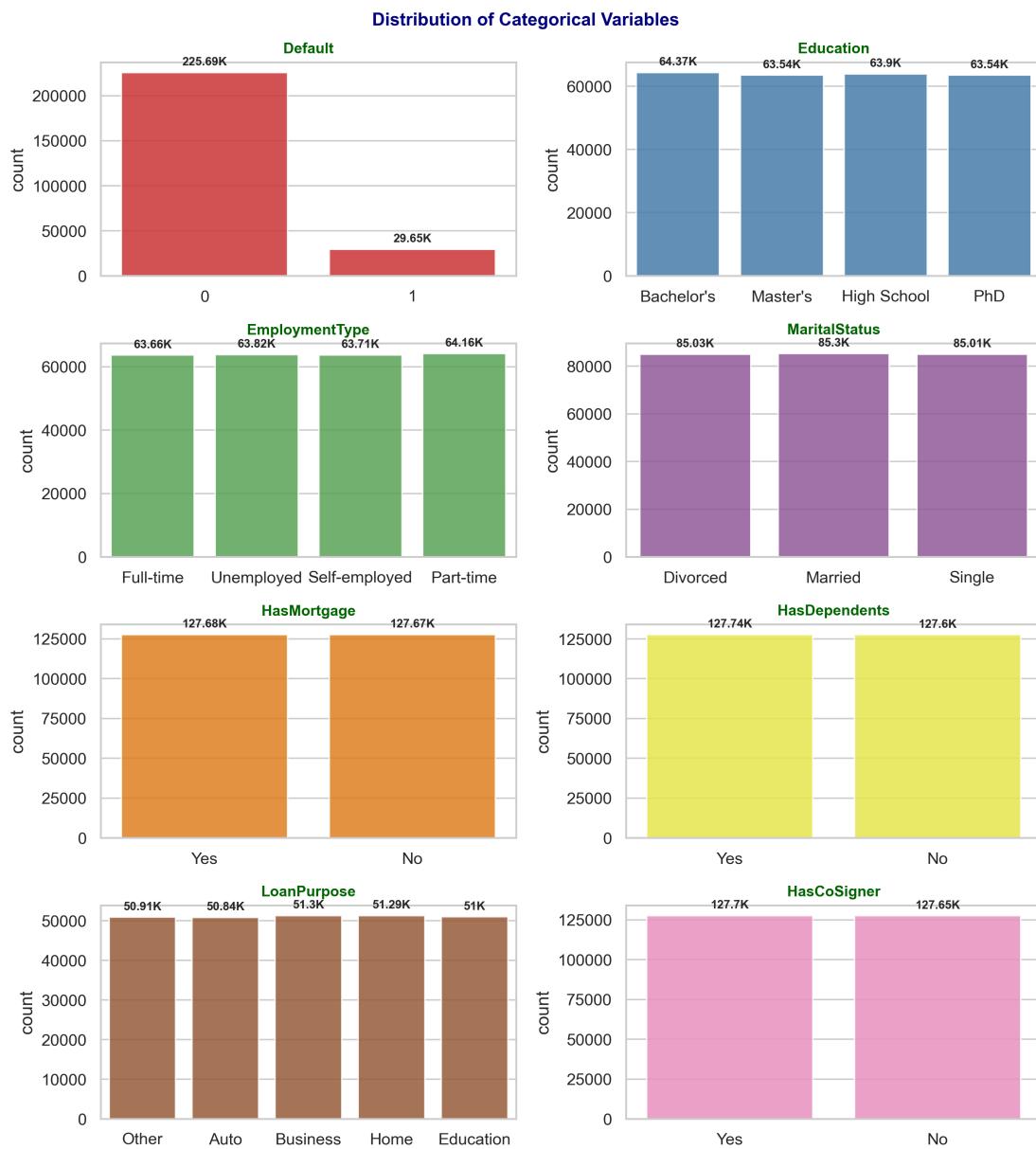
đó cũng khá đồng đều.

- NumCreditLines = 1: 63,554 profiles
- NumCreditLines = 2: 64,130 profiles
- NumCreditLines = 3: 63,834 profiles
- NumCreditLines = 4: 63,829 profiles

- **InterestRate:** phân phối khá đồng đều trong khoảng 2–25%, không có đuôi lệch đặc biệt.
- **LoanTerm:** thuộc tính này xét theo các kỳ hạn chuẩn (12, 24, 36, 48, 60 tháng), và cũng rất đồng đều, phản ánh các lựa chọn vay phổ biến.
 - LoanTerm = 12: 50,957 profiles
 - LoanTerm = 24: 51,009 profiles
 - LoanTerm = 36: 51,061 profiles
 - LoanTerm = 48: 51,166 profiles
 - LoanTerm = 60: 51,154 profiles
- **DTIRatio:** phân phối đều, tuy nhiên xuất hiện các spike tại các giá trị đặc trưng, phản ánh các giới hạn DTI định sẵn trong hệ thống cho vay.

4.2.2.2 Phân phối các biến rác (Categorical Features)

Các thuộc tính rác gồm: *Default, Education, EmploymentType, MaritalStatus, HasMortgage, HasDependents, LoanPurpose, HasCoSigner*. Hình 7 biểu diễn tỷ lệ phân loại của các biến này dưới dạng bar chart.



Hình 7: Phân phối các thuộc tính dạng phân loại.

Nhận xét.

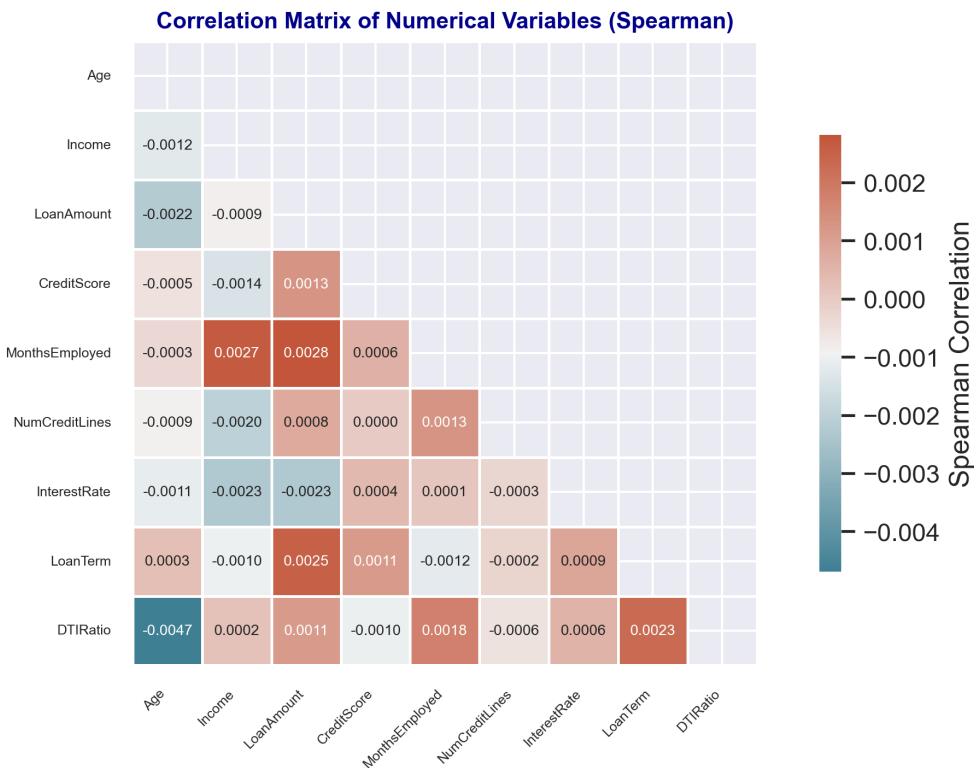
- **Default (nhân):** dữ liệu có sự mất cân bằng rõ rệt với phần lớn khách hàng không vỡ nợ (khoảng 225.7K) so với số ít vỡ nợ (khoảng 29.6K), tỷ lệ vỡ nợ (cũng là tỷ lệ nhãn positive của mô hình) chỉ khoảng 11.6%.



-
- **Education:** các mức học vấn phân phối tương đồng đều, với số lượng mỗi nhóm dao động quanh 63–64K; không có nhóm nào chiếm ưu thế rõ rệt.
 - **EmploymentType:** phân phối gần như đồng đều giữa các loại hình lao động (Full-time, Unemployed, Self-employed, Part-time), dao động khoảng 63–64K mỗi nhóm.
 - **MaritalStatus:** số lượng mỗi nhóm (Divorced, Married, Single) xấp xỉ nhau, khoảng 85K, cho thấy dữ liệu cân bằng về trạng thái hôn nhân.
 - **HasMortgage:** số lượng khách hàng có và không có thẻ chấp gần như bằng nhau, khoảng 127.7K mỗi nhóm.
 - **HasDependents:** phân phối gần như đồng đều giữa có và không có người phụ thuộc, dao động khoảng 127.6–127.7K.
 - **LoanPurpose:** các mục đích vay (Other, Auto, Business, Home, Education) đều có số lượng tương đương, khoảng 50–51K, không có mục đích nào chiếm ưu thế.
 - **HasCoSigner:** gần như cân bằng giữa có và không có người bảo lãnh, khoảng 127.65–127.7K mỗi nhóm.

4.2.3 Phân tích tương quan

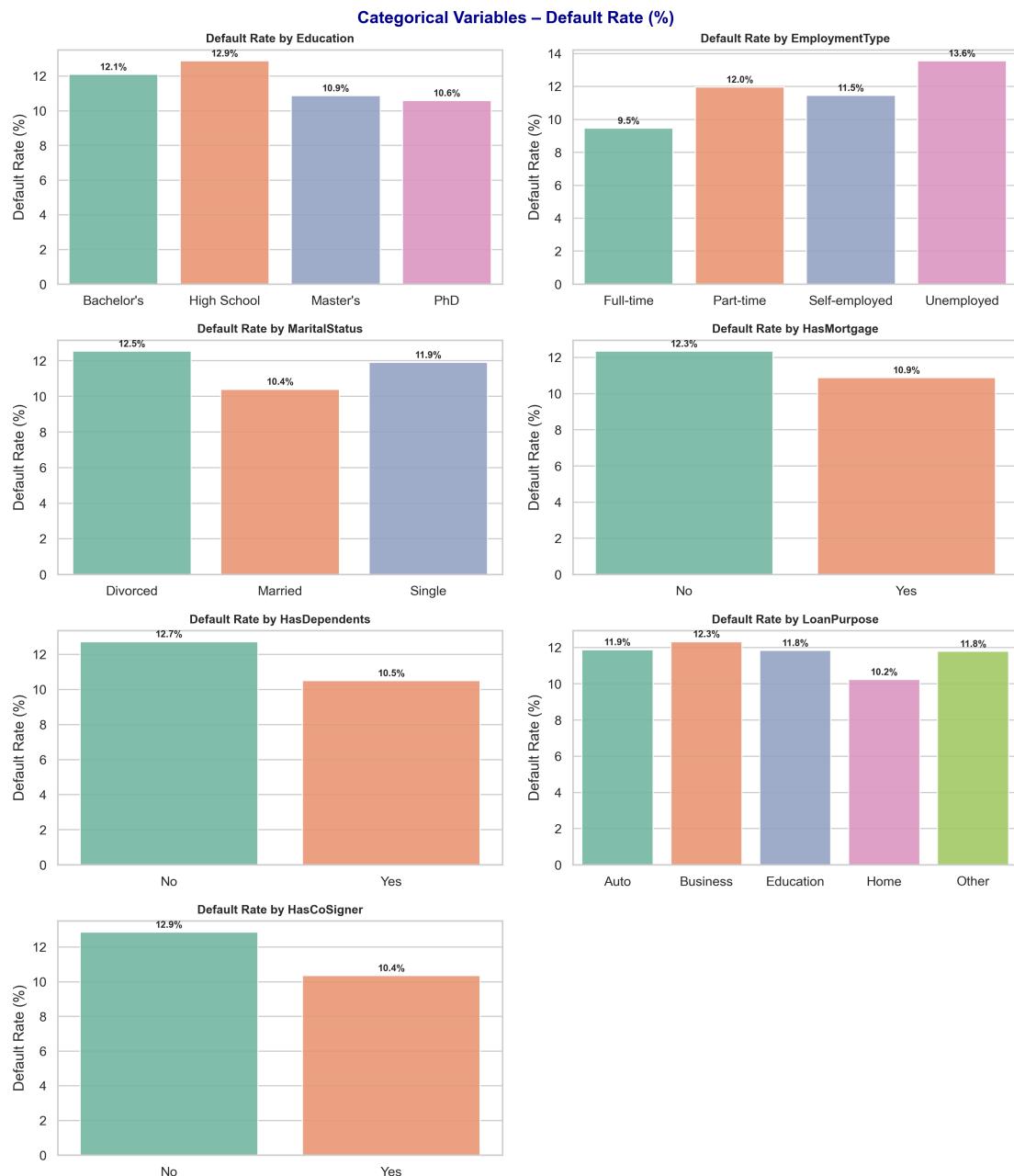
(a) **Tương quan giữa nhãn Default và các biến số.** Nhóm tính hệ số tương quan Spearman giữa nhãn *Default* và các thuộc tính dạng số. Kết quả được biểu diễn bằng heatmap trong Hình 8.



Hình 8: Tương quan giữa *Default* và các biến kiểu số.

Nhận xét. Trong thực tế, thường thì các biến về tài chính cá nhân như *CreditScore*, *InterestRate*, *DTIRatio* sẽ tương quan cao với khả năng vỡ nợ. Tuy nhiên, ta thấy tất cả các biến đều không có tương quan mạnh, dù thuận hay nghịch, với biến target (*Default*). Điều này cho thấy với bộ dữ liệu này, model sẽ cần học sự kết hợp giữa nhiều biến với nhau thì mới dự đoán được khả năng vỡ nợ.

(b) Tương quan giữa nhãn *Default* và các biến phân loại. Nhóm tính tỷ lệ phần trăm vỡ nợ theo từng giá trị của biến phân loại và biểu diễn dưới dạng biểu đồ cột như Hình 9.



Hình 9: Tỷ lệ vỡ nợ theo từng giá trị của biến phân loại.

Nhận xét.

- **EmploymentType (Loại Hình Việc làm)** là yếu tố dự đoán mạnh mẽ: Nhóm **Unemployed** (Thất nghiệp) có Tỷ lệ Vỡ nợ cao nhất (**13.6%**), trong khi **Full-time** (Toàn thời gian) là



thấp nhất (**9.5%**).

- **Education (Trình độ Học vấn)** và **MaritalStatus (Hôn nhân)**: **High School (12.9%)** và **Divorced (12.5%)** là các nhóm có rủi ro cao. **Married** có rủi ro thấp nhất (**10.4%**).
- **Các yếu tố Giảm thiểu Rủi ro (Mitigating Factors)**: Việc có **CoSigner** (Người đồng ký tên) và **Dependents** (Người phụ thuộc) liên quan đến Tỷ lệ Vỡ nợ thấp hơn đáng kể (lần lượt là **10.4%** và **10.5%**), so với việc không có (**12.9%** và **12.7%**).
- **LoanPurpose (Mục đích Khoản vay)**: Vay cho mục đích **Business** (Kinh doanh) có rủi ro cao nhất (**12.3%**), còn vay cho **Home** (Nhà ở) có rủi ro thấp nhất (**10.2%**).

Tổng hợp lại, các phân tích EDA cung cấp cái nhìn sâu về đặc tính dữ liệu và giúp xác định các biến tiềm năng có ảnh hưởng lớn đến khả năng vỡ nợ. Điều này hỗ trợ trực tiếp cho quá trình lựa chọn đặc trưng và tối ưu mô hình ở các bước sau.



5 Huấn luyện, đánh giá, và triển khai mô hình

5.1 Huấn luyện mô hình

Định nghĩa bài toán và metric tối ưu. Mục tiêu của mô hình là dự đoán xác suất khách hàng rời vào trạng thái *default* trong vòng 12 tháng kể từ ngày cấp khoản vay. Do đây là bài toán phân loại nhị phân, metric chính được tối ưu là **AUC (Area Under the ROC Curve)**, nhằm phản ánh năng lực phân tách giữa hai lớp tốt hơn so với accuracy trong trường hợp dữ liệu mêt cân bằng.

Chiến lược chia dữ liệu. Tập dữ liệu được chia theo tỷ lệ **80% huấn luyện, 20% test**, đảm bảo tính ngẫu nhiên có kiểm soát bằng cách *stratified sampling* trên nhãn Default. Việc hiệu chỉnh siêu tham số được thực hiện bằng **cross-validation 5-fold** trên tập dữ liệu huấn luyện (80% nói trên).

Dataset	Default	Số mẫu	Phần trăm (%)
Train			
	0	180555	88.39
	1	23722	11.61
Test			
	0	45139	88.39
	1	5931	11.61

Bảng 4: Thống kê số mẫu và tỷ lệ theo nhãn cho tập Train và Test.

Quy trình huấn luyện mô hình. Quá trình huấn luyện được tiến hành qua ba bước chính nhằm bảo đảm mô hình đạt được hiệu năng tối ưu và ổn định trên tập dữ liệu. Chi tiết các bước được trình bày như sau [7].

Bước 1: Huấn luyện mô hình baseline và xác định số cây tối ưu. Đầu tiên, một mô hình baseline được huấn luyện nhằm mục tiêu xác định số cây (*num_boosting_rounds* hay *n_estimators* hay *num_iterations*) phù hợp. Giá trị này được chọn dựa trên quá trình *early stopping* bằng cách theo dõi metric tối ưu (AUC) trên tập validation. Số cây thu được từ bước này sẽ làm mốc tham khảo cho quá trình tìm kiếm tham số ở bước tiếp theo.

**Bước 2: Tìm kiếm bộ siêu tham số tối ưu bằng Grid Search kết hợp Cross Validation 5-fold.**

Tiếp theo, một lưới tham số (param grid) được thiết lập và tiến hành tìm kiếm bằng kỹ thuật *Grid Search* kết hợp *5-fold Cross Validation* trên tập dữ liệu huấn luyện. Bộ tham số tối ưu được chọn dựa trên hiệu năng trung bình của mô hình trên năm fold, với metric đánh giá chính là AUC.

Listing 1: Param grid và kết quả best_params_

```
param_grid = {  
    'num_leaves': [31, 50],  
    'max_depth': [3, 5],  
    'learning_rate': [0.01, 0.05],  
    'n_estimators': [90, 100, 500],  
    'subsample': [0.5],  
    'colsample_bytree': [0.5],  
    'reg_alpha': [0, 5],  
    'reg_lambda': [0, 5],  
    'verbose': [-1]  
}  
  
# Found best params:  
best_params_ = {'colsample_bytree': 0.5,  
    'learning_rate': 0.05,  
    'max_depth': 3,  
    'n_estimators': 500,  
    'num_leaves': 31,  
    'reg_alpha': 5,  
    'reg_lambda': 0,  
    'subsample': 0.5  
}
```

Bước 3: Huấn luyện 5 mô hình theo 5-fold Cross Validation và lựa chọn ngưỡng dự đoán. Sau khi thu được bộ siêu tham số tối ưu, mô hình được huấn luyện lại theo 5-fold Cross Validation trên toàn bộ dữ liệu huấn luyện. Mỗi fold sinh ra một mô hình riêng biệt, và số cây tối ưu (với early stopping), cùng với AUC trên tập validation, được ghi nhận như bảng sau:

Dựa trên các mô hình đã huấn luyện, xác suất dự đoán (y_score) trên tập validation của



Fold	Num Iteration	Validation AUC
1	392	0.76
2	479	0.76
3	500	0.76
4	336	0.76
5	481	0.76

Bảng 5: Kết quả huấn luyện 5 mô hình theo 5-fold CV.

mỗi fold được dùng để tìm ngưỡng phân loại tối ưu. Nguồn được chọn là ngưỡng tối đa hóa chỉ số F1-score. Các kết quả chi tiết cho từng fold được biểu diễn trong bảng dưới:

Fold	Best Threshold	Best F1	Precision	Recall	Accuracy
1	0.17	0.37	0.29	0.52	0.79
2	0.18	0.38	0.30	0.49	0.81
3	0.18	0.36	0.29	0.48	0.81
4	0.17	0.37	0.29	0.52	0.80
5	0.18	0.37	0.29	0.50	0.80

Bảng 6: Kết quả tìm ngưỡng phân loại tối ưu theo từng fold.

Nhận xét về ngưỡng tối ưu. Qua kết quả thu được từ năm fold, ngưỡng phân loại tối ưu (theo tiêu chí tối đa hóa F1-score) được xác định là **0.176** - chính là trung bình ngưỡng tối ưu của 5 fold. Nguồn này giúp cân bằng tốt giữa Precision và Recall, đặc biệt quan trọng trong bài toán có phân bố dữ liệu mất cân bằng, và do đó được chọn làm nguồn dự đoán chính thức cho mô hình cuối cùng.

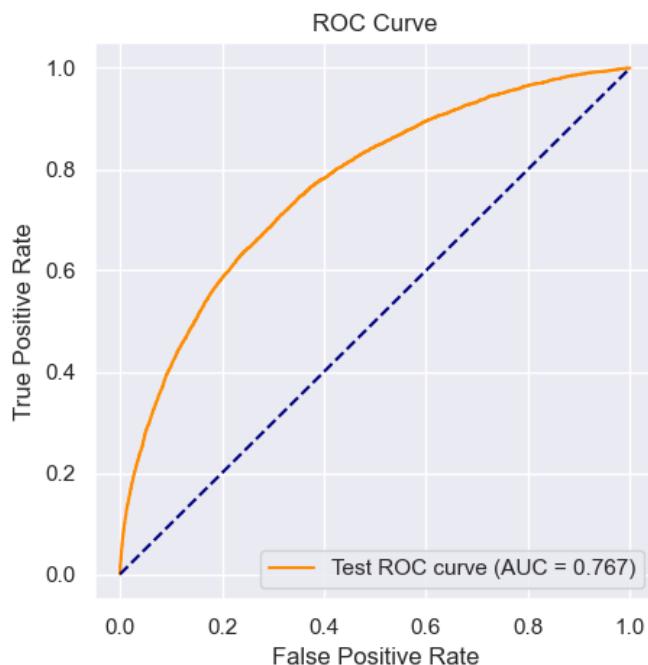
Ghi chú về phạm vi và chiến lược xây dựng mô hình. Trong thực tế, việc phân tích dữ liệu chuyên sâu và xây dựng thêm các đặc trưng mới (feature engineering) thường đóng vai trò quan trọng trong việc cải thiện hiệu quả của mô hình phân lớp, thậm chí mang lại mức tăng hiệu năng lớn hơn so với việc chỉ tinh chỉnh siêu tham số. Tuy nhiên, trong bối cảnh của bài tập nhóm này, mục tiêu chính là xây dựng một quy trình huấn luyện và triển khai mô hình hoàn chỉnh theo hướng hệ thống, hơn là tập trung tối đa vào tối ưu hóa từng điểm số AUC. Do đó, nhóm lựa chọn chỉ tập trung vào việc tinh chỉnh hyperparameters và không đầu tư nhiều thời gian vào việc thiết kế đặc trưng mới. Việc này giúp bảo đảm tiến độ, đồng thời vẫn minh họa được đầy đủ các bước quan trọng trong quá trình xây dựng một mô hình học máy hoàn chỉnh.

5.2 Dánh giá mô hình

Kết quả đánh giá

Sau khi hoàn thiện quá trình huấn luyện, mô hình được đánh giá trên tập *test* (20% dữ liệu ban đầu). Năm mô hình thu được từ quá trình Cross Validation 5-fold được sử dụng để dự đoán trên tập *test*; điểm dự đoán cuối cùng (*y_score*) được tính bằng cách lấy trung bình điểm dự đoán của cả năm mô hình, tương đương kỹ thuật *voting ensemble* ở dạng trung bình xác suất.

Đường cong ROC-AUC. Đầu tiên, điểm dự đoán *y_score* của mô hình trên tập *test* được sử dụng để vẽ đường cong ROC (Receiver Operating Characteristic) và tính diện tích dưới đường cong (AUC). Đây là metric tổng quát đánh giá chất lượng phân tách lớp của mô hình mà không phụ thuộc vào ngưỡng phân loại.



Hình 10: Đường cong ROC-AUC trên tập test.

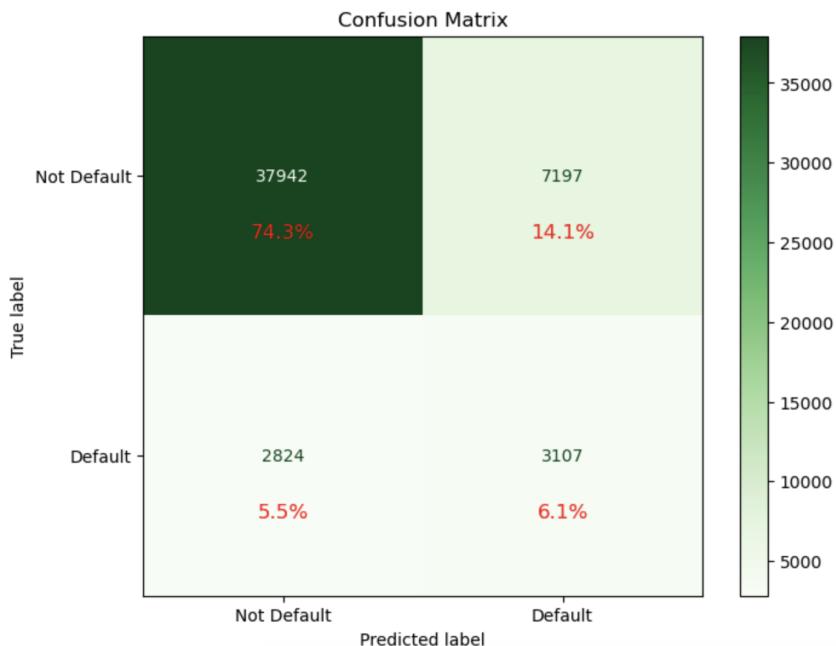
Tính các độ đo phân lớp. Ngưỡng tối ưu đã tìm được trong phần huấn luyện (**threshold = 0.176**) được áp dụng để chuyển đổi từ xác suất dự đoán sang nhãn cứng 0, 1. Trên cơ sở đó, các độ đo phân lớp tiêu chuẩn, bao gồm Precision, Recall, F1-score và Accuracy, được tính bằng

hàm `classification_report` trong thư viện `sklearn.metrics`.

Listing 2: Classification report

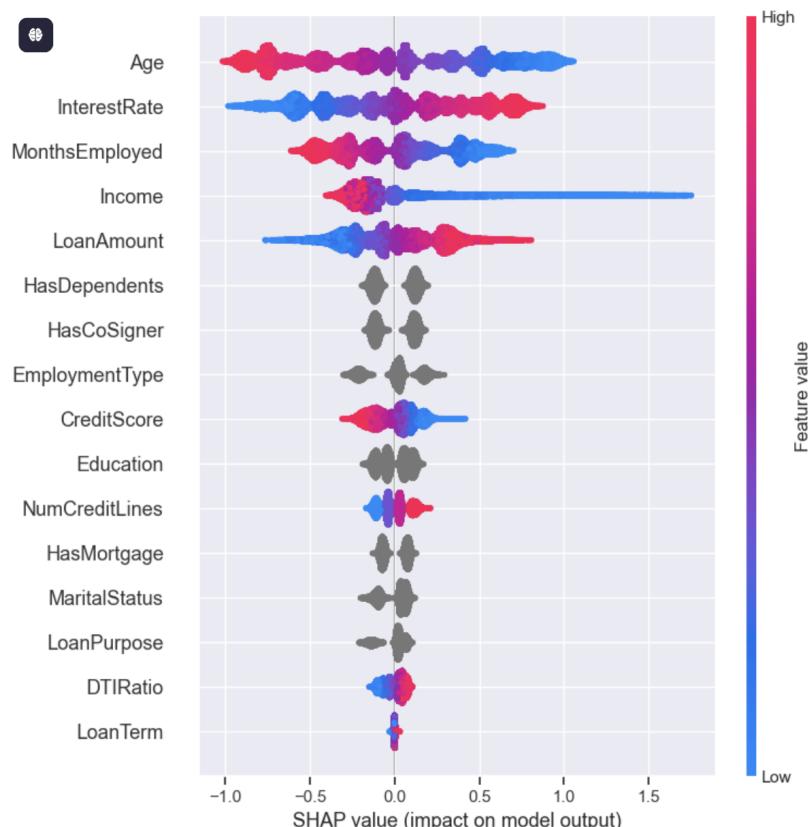
	precision	recall	f1-score	support
0	0.93	0.84	0.88	45139
1	0.30	0.52	0.38	5931
accuracy			0.80	51070
macro avg	0.62	0.68	0.63	51070
weighted avg	0.86	0.80	0.83	51070

Confusion Matrix (Ma trận nhầm lẩn) Để quan sát trực quan hơn về phân bố dự đoán đúng/sai của mô hình, ma trận nhầm lẩn (confusion matrix) được vẽ dưới dạng heatmap. Điều này đặc biệt hữu ích trong bối cảnh dữ liệu mất cân bằng, giúp đánh giá chi tiết số lượng mẫu dương tính và âm tính được dự đoán đúng hoặc sai.



Hình 11: Confusion matrix của mô hình trên tập test.

Phân tích độ quan trọng của đặc trưng bằng SHAP. Cuối cùng, độ quan trọng của các đặc trưng được phân tích bằng thư viện SHAP, một kỹ thuật giải thích mô hình phổ biến đối với mô hình dạng cây như LightGBM. Biểu đồ SHAP summary plot góp phần cung cấp cái nhìn sâu hơn về mức độ đóng góp của từng đặc trưng vào quyết định của mô hình.



Hình 12: Biểu đồ SHAP summary thể hiện độ quan trọng của các đặc trưng.

Giả định và hạn chế của hệ thống

- **Giả định dữ liệu:** mô hình giả định rằng các biến đầu vào phản ánh đầy đủ hành vi tín dụng của khách hàng, và không có sai lệch đáng kể trong quá trình thu thập.
- **Rủi ro bias:** do dữ liệu lịch sử chịu ảnh hưởng từ chính sách phê duyệt trong quá khứ, mô hình có thể kế thừa *selection bias*.
- **Overfitting:** được kiểm soát bằng cơ chế early-stopping và cross-validation, tuy nhiên vẫn cần theo dõi định kỳ sau khi triển khai.

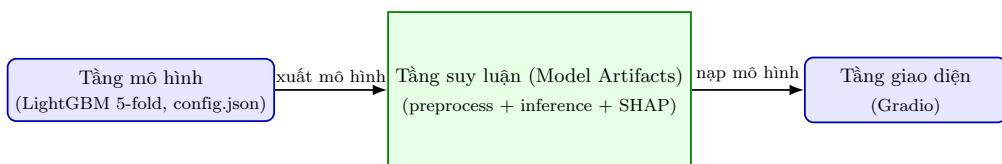
- **Explainability:** hệ thống cung cấp khả năng giải thích thông qua *SHAP values*, giúp người dùng hiểu rõ đóng góp của từng feature đối với quyết định cuối cùng.

5.3 Triển khai mô hình

Phần này trình bày quy trình triển khai mô hình dự đoán rủi ro vỡ nợ dựa trên LightGBM theo kiến trúc *train-serve-explain*. Kiến trúc triển khai mô hình (Model Deployment Architecture) được xây dựng theo hướng *train-serve-explain*, một mô hình tiêu chuẩn trong triển khai hệ thống học máy được mô tả chi tiết trong [1], [14]. Hệ thống được triển khai thông qua các notebook huấn luyện và triển khai, trong đó tầng suy luận được đóng gói thành lớp **ModelPredictor**, còn tầng giao diện được xây dựng bằng Gradio kèm cơ chế phân quyền nhiều vai trò.

5.3.1 Kiến trúc triển khai

Hệ thống được triển khai theo mô hình tổng quát gồm ba tầng: (i) Tầng mô hình (Model Layer) – lưu trữ các mô hình LightGBM và cấu hình; (ii) Tầng suy luận (Inference Layer) – sử dụng các Model Artifacts (các file .booster và model_config.json) để thực hiện tiền xử lý, suy luận và giải thích; (iii) Tầng giao diện (Application Layer) – giao diện dựa trên Gradio, phân quyền theo vai trò. Kiến trúc tổng quát được mô tả trong Hình 13.



Hình 13: Sơ đồ kiến trúc triển khai mô hình tổng thể

Thành phần chính của kiến trúc:

- Model Artifacts: gồm 5 mô hình LightGBM dạng .booster (tương ứng 5 fold) và tệp model_config.json chứa: danh sách đặc trưng đầu vào (features), các cột phân loại (cat_features), tên nhãn Default, ngưỡng phân loại tối ưu threshold = 0.176.
- ModelPredictor Class: nạp 5 mô hình và cấu hình, tiền xử lý dữ liệu (ép kiểu, chọn feature), suy luận bằng cách trung bình xác suất 5 mô hình, sinh nhãn dự đoán, sinh biểu đồ giải thích SHAP, đánh giá mô hình (ROC curve, confusion matrix, F1-score, ...).
- Gradio Multi-Role App: mô phỏng cơ chế đăng nhập và phân quyền (RBAC), cung cấp 4 nhóm chức năng theo từng vai trò nghiệp vụ.



5.3.2 Pipeline suy luận

Pipeline suy luận được đóng gói trong lớp `ModelPredictor`, bao gồm các bước tiền xử lý, chạy dự đoán ensemble và sinh giải thích SHAP. Đây là cấu trúc module hoá theo hướng phục vụ triển khai ML trong môi trường vận hành thực, phù hợp với khuyến nghị về xây dựng pipeline trong sách Introducing MLOps: *How to Scale Machine Learning in the Enterprise* [1]. Pipeline/Quy trình suy luận của hệ thống bao gồm bốn bước liên tiếp:

1. **Tiền xử lý dữ liệu:** Chuẩn hóa các biến phân loại (category dtype) để nhất quán với tập huấn luyện; Chọn đúng danh sách đặc trưng đã được lưu trong cấu hình; Loại bỏ các giá trị thiếu hoặc sai định dạng.
2. **Dự đoán:** mỗi hồ sơ được mô phỏng bằng vector đặc trưng; năm mô hình LightGBM chạy song song và trung bình xác suất để tạo ra kết quả ensemble.
3. **Phân loại nhị phân:** sử dụng ngưỡng tối ưu hoá theo F1-score (0.176) để tạo nhãn dự đoán. Trả về nhãn dự đoán:
 - 0 → Hồ sơ an toàn (Approved)
 - 1 → Nguy cơ vỡ nợ cao (Rejected)
4. **Giải thích mô hình:** dùng TreeSHAP để sinh biểu đồ SHAP Summary và SHAP Waterfall cho từng hồ sơ.

Pipeline này đảm bảo tính nhất quán giữa dữ liệu huấn luyện và dữ liệu triển khai, đồng thời duy trì tính minh bạch qua mô-đun giải thích.

5.3.3 Giao diện Gradio phân quyền theo vai trò (RBAC)

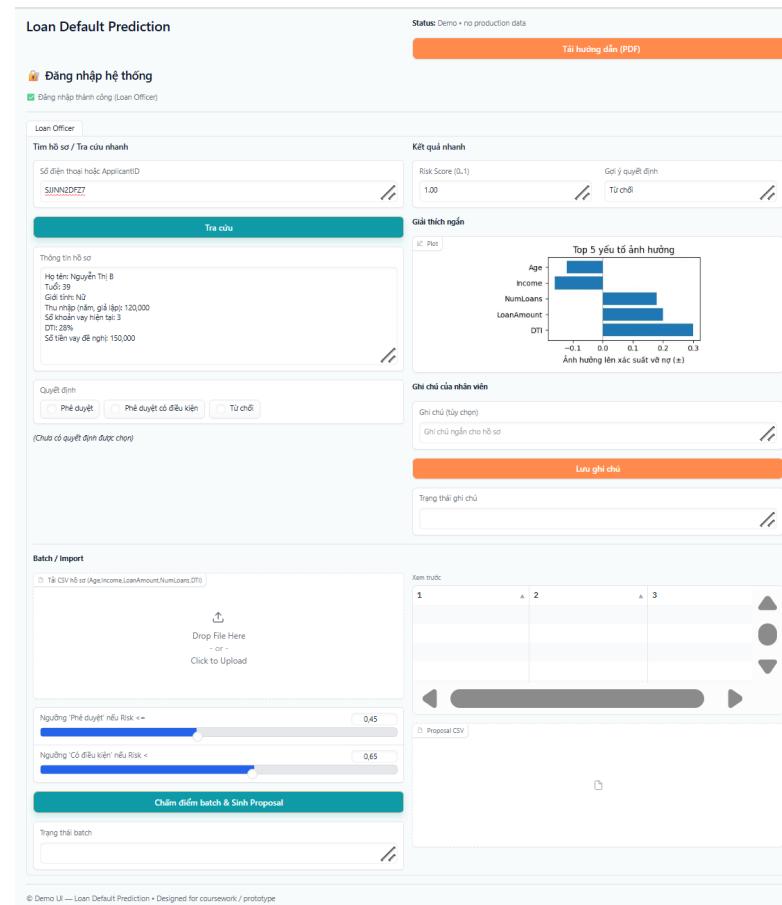
Giao diện triển khai hỗ trợ bốn vai trò:

1. **Loan Officer – Ra quyết định cấp tín dụng:**

- Upload file dữ liệu hồ sơ khách hàng.
- Chọn một LoanID → hệ thống chạy inference.
- Hiển thị:
 - xác suất vỡ nợ;
 - Gợi ý quyết định (“Phê duyệt” / “Từ chối”);



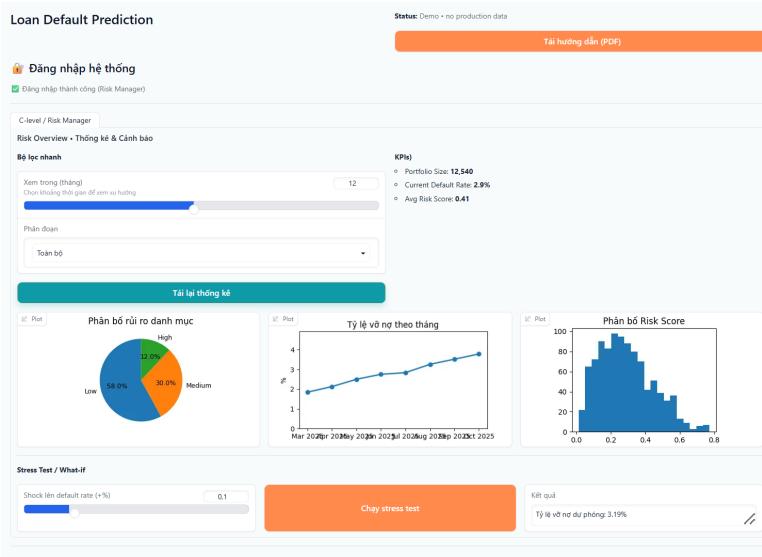
- Biểu đồ SHAP Waterfall cho từng khách hàng.
- Nhân viên có thể ghi chú hoặc thay đổi quyết định → mô phỏng quá trình ra quyết định có hỗ trợ bởi AI.



Hình 14: Gradio phân quyền theo vai trò của Loan Officer

2. Risk Manager – Giám sát danh mục tín dụng:

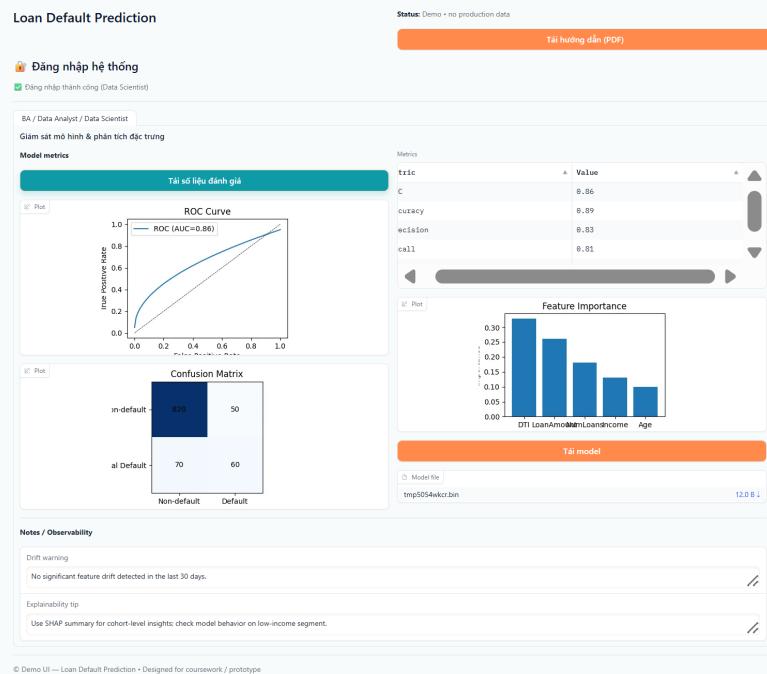
- Tải dữ liệu danh mục theo thời gian.
- Hệ thống sinh:
 - Histogram phân phối xác suất;
 - Biểu đồ pie theo tỷ lệ Rejected/Approved;
 - timeline theo tuần (weekly default risk).
- Dùng để quan sát xu hướng rủi ro của toàn bộ portfolio.



Hình 15: Gradio phân quyền theo vai trò của Risk Manager

3. Data Scientist – Giám sát hiệu năng mô hình:

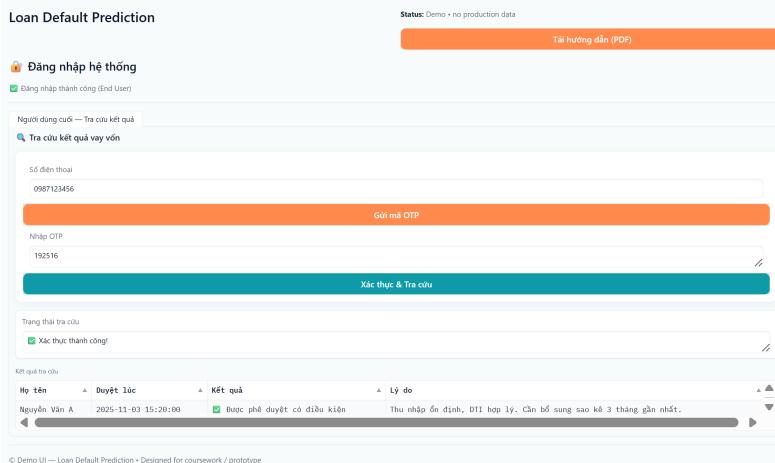
- Tải tập dữ liệu có nhãn để đánh giá mô hình. Hệ thống chạy evaluation:
 - ROC curve,
 - confusion matrix,
 - classification report,
 - SHAP feature importance.
- Form "Observability Notes" để ghi chú tình trạng drift.



Hình 16: Gradio phân quyền theo vai trò của Data Scientist

4. **User -Người dùng cuối:** tra cứu trạng thái khoản vay qua OTP, xem kết quả xét duyệt và giải thích rút gọn.

- Nhập số điện thoại → hệ thống sinh mã OTP ngẫu nhiên.
- Khi xác thực đúng, hệ thống truy vấn dữ liệu mẫu và trả về:
 - Trạng thái hồ sơ,
 - Kết quả xét duyệt,
 - Thông báo, giải thích ngắn gọn.



Hình 17: Gradio phân quyền theo vai trò của User

Giao diện người dùng được xây dựng bằng Gradio theo cơ chế phân quyền nhiều vai trò (RBAC), mô phỏng một hệ thống tín dụng thực tế với 4 nhóm người dùng: Loan Officer, Risk Manager, Data Scientist và End User. Cách tiếp cận này bám sát mô hình "multi-stakeholder intelligent system" được trình bày trong [2], [14].

5.3.4 Các khía cạnh hệ thống

Hệ thống triển khai hỗ trợ khả năng mở rộng (triển khai qua Docker hoặc chuyển sang API với FastAPI), đảm bảo bảo mật (RBAC, OTP, local-first), và cung cấp công cụ quan sát mô hình (quan sát drift, biến động phân phối đầu vào, theo dõi F1-score theo thời gian).

5.3.5 Quy trình triển khai tổng thể

Quy trình triển khai đầy đủ của hệ thống bao gồm:

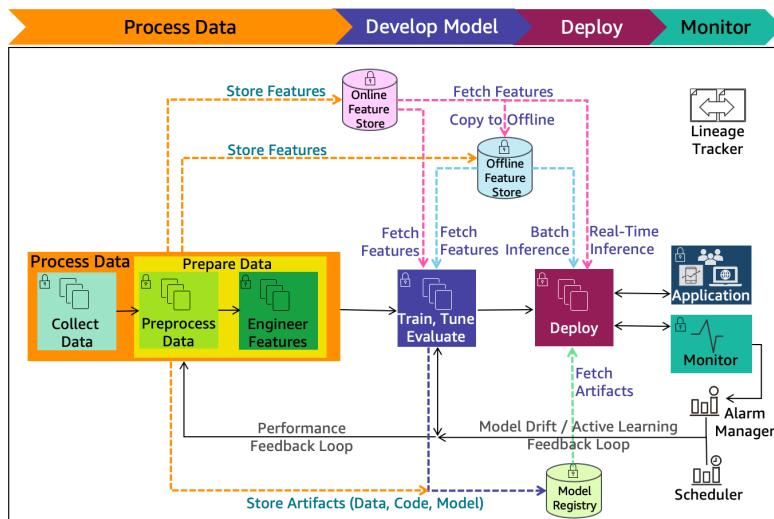
1. Huấn luyện mô hình.
2. Đóng gói mô hình thành .booster + model_config.json.
3. Khởi tạo lớp ModelPredictor để chuẩn hóa pipeline suy luận.
4. Tạo ứng dụng Gradio phục vụ nhiều nhóm người dùng.
5. Kiểm thử triển khai bằng tập dữ liệu thật và dữ liệu mô phỏng.
6. Tích hợp Explainability qua SHAP để tăng tính minh bạch.

5.3.6 Vai trò của triển khai trong hệ thống thông minh

- Hỗ trợ ra quyết định tự động (autonomous decision support).
- Giảm rủi ro nhờ phân tích và cảnh báo dựa trên mô hình dự đoán.
- Minh bạch hóa thuật toán nhờ Explainability (SHAP).
- Kết nối giữa kỹ thuật và nghiệp vụ qua giao diện phân vai.

Những vai trò trên của việc triển khai mô hình đã xây dựng hệ thống đáp ứng đúng tinh thần của một Intelligent System, gồm: Kết hợp mô hình học máy, trực quan hóa, phân quyền, minh bạch và kiểm soát rủi ro để hỗ trợ ra quyết định trong môi trường phức tạp.

Phản triển khai đã hoàn thiện từ tầng mô hình, pipeline suy luận, giao diện, phân quyền, cho đến các khía cạnh mở rộng, bảo mật và giám sát mô hình. Kiến trúc train-serve-explain đảm bảo mô hình không chỉ được xây dựng đúng kỹ thuật mà còn có thể sử dụng thực tế bởi nhiều nhóm đối tượng khác nhau trong tổ chức. Kiến trúc chi tiết được mô tả trong Hình 18.



Hình 18: Sơ đồ kiến trúc triển khai mô hình chi tiết

5.4 Theo dõi mô hình

Việc theo dõi mô hình là cực kỳ quan trọng. Mô hình được huấn luyện và đánh giá dựa trên dữ liệu tại một thời điểm trong quá khứ. Không có gì đảm bảo mô hình vẫn hoạt động tốt với dữ liệu tương lai. Hơn nữa, việc hiệu suất mô hình kém đi dần theo thời gian là hiện tượng phổ biến. Do đó, hệ thống cần có module theo dõi hiệu suất mô hình, và tái huấn luyện với dữ liệu



mới nhất có thể. Trong thực tế, module này thường được thiết kế bởi Data Scientist, và được triển khai vận hành bởi chính Data Scientists hoặc Machine Learning Operators (MLOPs).

Vòng đời mô hình và vai trò của Data Scientist. Data Scientist chịu trách nhiệm toàn bộ vòng đời của mô hình bao gồm:

1. **Data Cleaning & Feature Engineering:** làm sạch, chọn lọc, chuẩn hoá dữ liệu đầu vào, và tổng hợp features
2. **Model Development:** thử nghiệm nhiều thuật toán, tinh chỉnh siêu tham số, đánh giá bằng cross-validation.
3. **Deployment:** chuyển đổi codebase từ môi trường phát triển (Jupyter notebooks) sang môi trường triển khai (Python scripts)
4. **Monitoring:** theo dõi các chỉ số drift (như PSI), hiệu năng (gồm model metrics như AUC, ..., và business metrics như lợi nhuận từ cho vay đúng khách hàng, ...), và cảnh báo khi có sai lệch (có thể dựng dashboard và/hoặc cài đặt các alert nếu cần thiết).
5. **Versioning:** Làm việc với các repositories liên quan, bao gồm nhiều loại repo: code versioning (ví dụ: git), model versioning (ví dụ: MLFlow), data versioning (ví dụ: DVC), ... Bằng việc này, Data Scientists luôn commit và theo dõi tất cả mọi thứ liên quan: models, hyper-parameters, features, processing pipelines, thresholds, global variables, APIs, secret keys, data paths, ..., nhằm giúp Data Scientists nắm rõ toàn bộ hệ thống. Lưu ý rằng chi tiết của những khái niệm này nằm ngoài phạm vi project nên nhóm không viết cụ thể vào báo cáo, tuy nhiên nhóm đã trang bị đủ kiến thức và có thể trả lời trong buổi present nếu được hỏi.
6. **Retraining:** huấn luyện lại mô hình theo chu kỳ (tuỳ theo tính chất bài toán, ở bài này nhãn chậm có nên thường là tái huấn luyện sau mỗi 1 tháng trở lên) hoặc khi hiệu năng suy giảm đáng kể.

Cơ chế phản hồi từ nghiệp vụ: Sau khi mô hình được triển khai, các phản hồi thực tế từ Ban Giám đốc và bộ phận tín dụng được ghi nhận để cập nhật pipeline:

- Các hồ sơ có quyết định khác với dự đoán mô hình được đánh dấu và đưa vào tập “manual review”.



-
- Sau 6–12 tháng (thời gian trung bình để có kết quả khách hàng có vỡ nợ hay không), kết quả thực tế (*default hoặc không*) được dùng để đánh giá lại độ chính xác và hiệu chỉnh những phân loại.
 - Báo cáo phản hồi được tổng hợp định kỳ và gửi cho nhóm Data Science để cải thiện mô hình. Vì dạng phản hồi này không thể tự động hoá nên chúng sẽ được phân tích xử lý thủ công, không thể đưa vào pipeline triển khai tự động của hệ thống.



6 Kết luận và hướng phát triển

6.1 Kết luận

Trong báo cáo này, nhóm đã xây dựng một hệ thống dự đoán rủi ro vỡ nợ (*Loan Default Prediction*) theo hướng một hệ thống thông minh tương đối hoàn chỉnh, bao gồm: hiểu bài toán và các bên liên quan, phân tích dữ liệu, lựa chọn giải thuật, huấn luyện và đánh giá mô hình, thiết kế giao diện cho từng nhóm người dùng, triển khai pipeline suy luận và đề xuất cơ chế theo dõi mô hình sau triển khai.

Về mặt dữ liệu, nhóm sử dụng bộ dữ liệu công khai với 255.347 bản ghi và 18 thuộc tính, trong đó nhãn **Default** có tỷ lệ khoảng 11% (dữ liệu mất cân bằng). Nhóm đã tiến hành kiểm tra dữ liệu thiếu, dữ liệu trùng lặp, phân tích phân phối các biến số và biến phân loại, cũng như khảo sát tương quan giữa các thuộc tính với nhãn mục tiêu. Kết quả EDA cho thấy không có giá trị thiếu, không có bản ghi trùng lặp, phân phối của các biến tương đối “sạch”, đồng thời chỉ ra một số thuộc tính có tín hiệu mạnh hơn về rủi ro vỡ nợ như **EmploymentType**, **MaritalStatus**, **HasCoSigner**, **LoanPurpose**, v.v. Về mặt mô hình, nhóm lựa chọn họ mô hình Gradient Boosting trên cây quyết định và sử dụng LightGBM làm mô hình chính, dựa trên các ưu điểm: hiệu năng tốt trên dữ liệu dạng bảng, hỗ trợ dữ liệu mất cân bằng, khả năng mở rộng và tốc độ suy luận nhanh. Mô hình được huấn luyện với chiến lược chia dữ liệu 80% train – 20% test (stratified theo nhãn), kết hợp Grid Search và 5-fold Cross Validation để tìm bộ siêu tham số tối ưu. Năm mô hình tương ứng với 5 fold được sử dụng như một tổ hợp (*ensemble*), điểm dự đoán cuối cùng trên tập test được tính bằng trung bình xác suất của cả năm mô hình.

Kết quả đánh giá cho thấy mô hình đạt AUC khoảng 0,76 trên tập validation và test, với Accuracy khoảng 80% và F1-score của lớp **Default** khoảng 0,38 khi sử dụng ngưỡng phân loại tối ưu 0,176 – được xác định thông qua việc tối đa hóa F1 trên các fold. Trong bối cảnh dữ liệu mất cân bằng và không có bước thiết kế đặc trưng chuyên sâu, mức hiệu năng này là chấp nhận được và có thể sử dụng để hỗ trợ ra quyết định nghiệp vụ. Ngoài ra, trong phạm vi project thì nhóm chấp nhận mức $AUC \approx 0.76$ như một baseline hợp lý. Tuy nhiên, so với mục tiêu kỳ vọng ban đầu ($AUC > 0.85$), mô hình hiện tại vẫn còn dư địa để cải thiện nếu đầu tư nhiều hơn cho feature engineering, thử nghiệm thêm các biến thể mô hình và tinh chỉnh siêu tham số trên không gian rộng hơn.

Một điểm nhấn quan trọng của hệ thống là việc tích hợp mô-đun giải thích mô hình (Explainability) thông qua SHAP. Các biểu đồ SHAP summary và SHAP waterfall giúp giải thích đóng góp của từng đặc trưng đối với quyết định của mô hình, vừa hỗ trợ Data Scientist



trong việc phân tích mô hình, vừa cung cấp “lý do” ở mức phù hợp cho Loan Officer và cho người dùng cuối. Bên cạnh đó, nhóm cũng xây dựng giao diện tương tác bằng Gradio với cơ chế phân quyền theo vai trò (RBAC) cho bốn nhóm chính: Loan Officer, Risk Manager, Data Scientist và End User; qua đó mô phỏng tương đối sát cách một hệ thống dự đoán rủi ro có thể được sử dụng trong môi trường nghiệp vụ thực tế.

Cuối cùng, báo cáo không chỉ dừng lại ở việc xây dựng một mô hình đơn lẻ, mà còn trình bày kiến trúc triển khai theo hướng *train-serve-explain* và đề xuất cơ chế theo dõi mô hình (monitoring), vòng đời mô hình (model lifecycle) cũng như vai trò của Data Scientist trong việc vận hành, giám sát và tái huấn luyện mô hình. Điều này thể hiện đúng tinh thần của môn Hệ thống thông minh: không chỉ tập trung vào thuật toán, mà còn chú trọng đến việc tích hợp mô hình vào một hệ thống có thể vận hành, giám sát và cải tiến liên tục trong môi trường phức tạp.

6.2 Hướng phát triển

Dù đã đạt được một hệ thống tương đối hoàn chỉnh về mặt chức năng, dự án vẫn còn nhiều hướng mở rộng và cải tiến có thể thực hiện trong tương lai:

1. Mở rộng dữ liệu và gắn với bối cảnh doanh nghiệp thực

Một trong những hạn chế lớn nhất của hệ thống hiện tại là sử dụng bộ dữ liệu công khai, có cấu trúc đơn giản và không phản ánh đầy đủ các đặc trưng nghiệp vụ của một tổ chức tín dụng thực. Trong triển khai tương lai, việc mở rộng và tích hợp thêm dữ liệu từ các hệ thống nghiệp vụ là hướng phát triển quan trọng. Cụ thể:

- (i) *Tích hợp nhiều nguồn dữ liệu nội bộ*: kết hợp dữ liệu từ Core Banking, hệ thống thẩm định, CRM, dữ liệu thu hồi nợ, lịch sử giao dịch, điểm tín dụng nội bộ, ... giúp mô hình học được hành vi khách hàng toàn diện hơn.
- (ii) *Bổ sung dữ liệu bên ngoài*: như dữ liệu CIC, dữ liệu viễn thông (telco scoring), dữ liệu hành vi số (digital footprint), hoặc dữ liệu từ các nền tảng thanh toán. Đây là hướng cực kỳ quan trọng để tăng độ chính xác của mô hình.
- (iii) *Mở rộng dữ liệu theo thời gian (temporal data)*: cho phép mô hình khai thác chuỗi thời gian (time-series patterns), ví dụ: biến động thu nhập, tần suất giao dịch, sự ổn định trong thanh toán, giúp dự báo rủi ro sớm hơn.
- (iv) *Làm sạch và chuẩn hóa dữ liệu ở cấp độ doanh nghiệp*: áp dụng data governance như định danh dữ liệu, dictionary — giúp dữ liệu đầu vào ổn định, hạn chế drift do



thay đổi quy trình nghiệp vụ.

Việc mở rộng dữ liệu không chỉ nâng cao hiệu năng mô hình, mà còn giúp hệ thống tiêm cận cách thức vận hành trong các tổ chức tài chính thực, vốn dựa mạnh vào dữ liệu lịch sử phong phú và đa dạng.

2. Hoàn thiện kiến trúc hệ thống và MLOps

Một hệ thống dự đoán rủi ro chỉ thật sự hữu ích khi có khả năng vận hành ổn định, cập nhật mô hình tự động và đảm bảo khả năng mở rộng. Do đó, hướng phát triển tự nhiên là chuyển từ prototype (Gradio + Jupyter) sang một kiến trúc sản xuất chuyên nghiệp với MLOps. Các hướng phát triển gồm:

- (i) *Dóng gói mô hình thành dịch vụ (Model Serving)*: sử dụng FastAPI/Flask để tạo REST API phục vụ suy luận, kết hợp Docker để đảm bảo khả năng triển khai đa môi trường.
- (ii) *Tự động hóa pipeline huấn luyện-triển khai*: sử dụng Airflow, Kubeflow, MLflow hoặc DVC để theo dõi thí nghiệm, quản lý version mô hình, tự động re-train theo điều kiện, ví dụ: drift vượt ngưỡng.
- (iii) *Xây dựng cơ chế giám sát thời gian thực (Model Monitoring)*: theo dõi AUC, F1, tỷ lệ default theo ngày/tuần, độ lệch phân phối đầu vào, heatmap drift... để phát hiện kịp thời khi mô hình xuống cấp.
- (iv) *Tối ưu kiến trúc hạ tầng*: triển khai mô hình trên Kubernetes, sử dụng autoscaling để phục vụ khối lượng yêu cầu lớn trong giờ cao điểm; thiết kế chiến lược cache để tăng tốc suy luận.

Hướng này giúp hệ thống chuyển từ phiên bản demo sang mức độ **sẵn sàng triển khai thực tế** trong doanh nghiệp.

3. Tăng cường fairness, explainability và governance

Trong bối cảnh ra quyết định tín dụng, các mô hình AI không chỉ cần chính xác, mà còn phải **minh bạch, có thể giải thích**, và **không có bias** đối với các nhóm khách hàng. Dù hệ thống hiện tại đã tích hợp SHAP, vẫn còn nhiều hướng nâng cấp quan trọng:



-
- (i) *Mở rộng khả năng giải thích*: ngoài SHAP, có thể bổ sung mô hình surrogate (LIME), partial dependence plots (PDP), accumulated local effects (ALE) để cung cấp góc nhìn đa chiều.
 - (ii) *Dánh giá fairness giữa các nhóm dân số*: ví dụ theo giới tính, nghề nghiệp, độ tuổi, thu nhập; đo bằng các chỉ số Equal Opportunity, Demographic Parity, Predictive Parity...
 - (iii) *Thiết lập audit trail cho quyết định mô hình*: lưu lại từng dự đoán kèm lý do SHAP để phục vụ kiểm toán nội bộ hoặc yêu cầu của cơ quan quản lý.
 - (iv) *Áp dụng Counterfactual Explanations*: giúp khách hàng biết “cần thay đổi điều gì”, chẳng hạn: tăng thu nhập, lịch sử thanh toán ổn định hơn, ... để tăng khả năng được duyệt vay.
 - (v) *Governance cho toàn bộ vòng đời mô hình*: thiết lập chính sách kiểm soát version, quy trình phê duyệt mô hình, đánh giá rủi ro mô hình, kiểm thử định kỳ.

Đây là xu hướng tất yếu trong các hệ thống AI tại ngân hàng — đảm bảo sự minh bạch, công bằng và tuân thủ quy định.

4. Nâng cấp giao diện và trải nghiệm người dùng

Giao diện Gradio trong bài tập giúp minh họa chức năng cơ bản, nhưng để vận hành trong doanh nghiệp, cần một UI/UX chuyên nghiệp hơn, đa nền tảng hơn. Các hướng phát triển gồm:

- (i) *Xây dựng Frontend thực thụ*: sử dụng React, Angular hoặc Vue; tối ưu hoá layout cho từng vai trò người dùng Loan Officer, Risk Manager, Data Scientist.
- (ii) *Thiết kế dashboard trực quan hơn*: - Heatmap rủi ro theo thời gian - Biểu đồ funnel cấp tín dụng - Bộ lọc theo tỉnh/thành/phân khúc - Xu hướng thay đổi điểm rủi ro cá nhân
- (iii) *Tối ưu hóa trải nghiệm người dùng cuối*: hỗ trợ mobile-first, thêm chatbot giải thích lý do từ chối, mở rộng chức năng giải thích theo dạng “What-if scenarios”.
- (iv) *Cải thiện tương tác batch*: cho phép Loan Officer xử lý hàng loạt hồ sơ (batch approval), Risk Manager tải và phân tích portfolio lớn với hàng trăm nghìn bản ghi.
- (v) *Tăng cường bảo mật giao diện*: đăng nhập đa lớp (MFA), phân quyền chi tiết hơn, mã hóa thông tin khách hàng khi hiển thị.



Nâng cấp UI/UX giúp hệ thống trở nên thân thiện, trực quan và phù hợp với môi trường doanh nghiệp quy mô lớn.

Tổng kết lại, Nhóm đã đạt được mục tiêu xây dựng một hệ thống thông minh mô phỏng khá sát quy trình dự đoán rủi ro vỡ nợ trong thực tế, từ dữ liệu tới mô hình, triển khai và theo dõi. Các hướng phát triển nêu trên không chỉ giúp nâng cao hiệu quả mô hình, mà còn phát triển prototype hiện tại thành một hệ thống có thể triển khai trong môi trường doanh nghiệp thực thụ.



Tài liệu

- [1] Mark Treveil et al. Introducing MLOps: How to Scale Machine Learning in the Enterprise. O'Reilly Media, 1st Ed., 2021.
- [2] Hulten, G. Building Intelligent Systems: A Guide to Machine Learning Engineering. MIT Press, 2018.
- [3] Russell, S; Norvig, P. Artificial Intelligence: A Modern Approach, 4th Edition, Pearson, 2021.
- [4] Haykin, S. Neural Networks and Learning Machines, Pearson, 3rd Edition.
- [5] Friedman, J. H. "Greedy Function Approximation: A Gradient Boosting Machine." *Annals of Statistics*, 2001.
- [6] Chen, T., & Guestrin, C. "XGBoost: A Scalable Tree Boosting System." *KDD*, 2016.
- [7] Ke, G. et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." *NeurIPS*, 2017.
- [8] Prokhorenkova, L. et al. "CatBoost: Unbiased Boosting with Categorical Features." *NeurIPS*, 2018.
- [9] Anderson, R. *An Introduction to Credit Scoring*. Oxford University Press.
- [10] Chollet, F. *Deep Learning with Python*, 2nd Edition, Manning, 2021.
- [11] Hand, D.; Henley, W. "Statistical Classification Methods in Consumer Credit Scoring: A Review." *Journal of the Royal Statistical Society*, 1997.
- [12] Louzada, F. et al. "Classification Methods Applied to Credit Scoring: Systematic Review." *Expert Systems with Applications*, 2016.
- [13] Lessmann, S. et al. "Benchmarking State-of-the-Art Classification Algorithms for Credit Scoring." *European Journal of Operational Research*, 2015.
- [14] Huyen, C. *Designing Machine Learning Systems: An Iterative Process for Production-Ready Applications*. O'Reilly, 2022.
- [15] Quản Thành Thơ, Mạng Nơ-ron nhân tạo: Từ hồi quy đến học sâu, NXB ĐHQG TPHC, 2022.