**BỘ CÔNG THƯƠNG**
**TRƯỜNG ĐẠI HỌC ĐIỆN LỰC**
**KHOA CÔNG NGHỆ THÔNG TIN**


# BÁO CÁO CHUYÊN ĐỀ HỌC PHẦN
# HỌC MÁY NÂNG CÁO
# ĐỀ TÀI: DỰ ĐOÁN RỦI RO TÍN DỤNG

**Sinh viên thực hiện** : **Nguyễn Cát Bộ**
**Nguyễn Thành Đạt**
**Trương Đức Mạnh**

**Giảng viên hướng dẫn** : **TS. Trần Trung**

**Ngành** : **CÔNG NGHỆ THÔNG TIN**

**Chuyên Ngành** : **CÔNG NGHỆ PHẦN MỀM**

**Lớp** : **D17CNPM6**

**Khóa** : **2022-2027**

*Hà Nội, Tháng 12 Năm 2025*

# PHIẾU CHẤM ĐIỂM

| STT | Họ và tên sinh viên | Nội dung thực hiện | Điểm | Chữ ký |
|---|---|---|---|---|
| 1 | Nguyễn Cát Bộ<br>22810310305 | | | |
| 2 | Nguyễn Thành Đạt<br>22810310314 | | | |
| 3 | Trương Đức Mạnh<br>22810310320 | | | |

| Họ và tên giảng viên | Chữ ký | Ghi chú |
|---|---|---|
| Giảng viên chấm 1: | | |
| Giảng viên chấm 2: | | |

# MỤC LỤC

# MỤC LỤC ẢNH

# DANH MỤC TỪ VIẾT TẮT

| STT | Ký hiệu chữ viết tắt | Chữ viết đầy đủ (Anh – Việt) |
|---|---|---|
| 1 | | |
| 2 | | |

# LỜI MỞ ĐẦU

Trong bối cảnh nền kinh tế số phát triển mạnh mẽ, hoạt động tín dụng cá nhân trở thành một trong những lĩnh vực quan trọng và có mức độ tăng trưởng cao nhất. Các ngân hàng và tổ chức tài chính phải xử lý một lượng lớn hồ sơ vay vốn mỗi ngày, với yêu cầu không chỉ nhanh chóng mà còn phải đảm bảo chính xác để hạn chế rủi ro nợ xấu. Việc thẩm định hồ sơ tín dụng truyền thống thường dựa nhiều vào kinh nghiệm của chuyên viên, quy trình thủ công, dễ bị ảnh hưởng bởi yếu tố chủ quan và khó đồng nhất giữa các hồ sơ. Điều này khiến cho việc đánh giá khách hàng thiếu ổn định, tiềm ẩn rủi ro và tốn kém về mặt thời gian lẫn chi phí vận hành.

Trong khi đó, dữ liệu giao dịch, dữ liệu hành vi và dữ liệu lịch sử trả nợ của khách hàng ngày càng trở nên phong phú và dễ tiếp cận hơn nhờ quá trình số hóa. Đây là điều kiện quan trọng mở ra cơ hội áp dụng các phương pháp học máy (Machine Learning) nhằm tự động hóa và tối ưu hóa quy trình chấm điểm tín dụng (Credit Scoring). Bằng cách phân tích các mẫu (pattern) và mối quan hệ giữa các đặc trưng trong dữ liệu, các mô hình học máy có khả năng dự đoán xu hướng vi phạm thanh toán, từ đó giúp cho các tổ chức tài chính đưa ra quyết định cho vay một cách khách quan, chính xác và hiệu quả hơn.

Đề tài "Dự đoán rủi ro tín dụng bằng học máy" được xây dựng nhằm mục tiêu nghiên cứu, triển khai và đánh giá một số thuật toán học máy kinh điển trong bài toán phân loại rủi ro tín dụng. Dữ liệu sử dụng trong đề tài bao gồm thông tin ứng viên (application record) và lịch sử tín dụng (credit record). Các bước tiền xử lý được xây dựng nhằm làm sạch dữ liệu, xử lý giá trị thiếu, mã hóa biến phân loại và chuẩn hóa dữ liệu nhằm đảm bảo mô hình có khả năng học hiệu quả nhất. Từ dữ liệu đã tiền xử lý, nhóm mô hình được lựa chọn bao gồm Logistic Regression, K-Nearest Neighbors (KNN), Decision Tree và Random Forest – đây đều là những thuật toán phổ biến và được ứng dụng rộng rãi trong lĩnh vực phân loại rủi ro.

Quy trình thực nghiệm của đề tài được triển khai tuần tự theo đúng quy tắc của quy trình học máy: chuẩn bị dữ liệu – phân tích – huấn luyện – đánh giá – so sánh – chọn mô hình tối ưu. Nhiều chỉ số đánh giá quan trọng được sử dụng như Accuracy, Precision, Recall, F1-score và AUC-ROC. Đặc biệt, mô hình Random Forest thường được kỳ vọng có khả năng khái quát hóa tốt, ổn định trước nhiễu dữ liệu và phù hợp với các bài toán có nhiều thuộc tính phức tạp như tín dụng. Bên cạnh đó, báo cáo cũng phân tích ưu – nhược điểm của từng mô hình, giúp làm rõ nguyên nhân mô hình hoạt động tốt hoặc chưa hiệu quả trong từng tình huống.

Không chỉ dừng lại ở việc xây dựng mô hình, đề tài còn hướng đến việc mô phỏng quy trình đánh giá tín dụng thực tế, từ đó chứng minh tính khả thi và hiệu quả khi áp dụng học máy vào hoạt động quản lý tín dụng. Ngoài ra, việc áp dụng mô hình dự đoán rủi ro tín dụng không chỉ giúp giảm thiểu tỷ lệ nợ xấu mà còn hỗ trợ tối ưu hóa quy trình vận hành, giảm tải cho bộ phận thẩm định, tăng khả năng phục vụ khách hàng và nâng cao lợi thế cạnh tranh cho các tổ chức tín dụng.

Báo cáo này sẽ trình bày một cách có hệ thống toàn bộ quá trình thực hiện đề tài, bao gồm giới thiệu bài toán, mô tả dữ liệu, tiền xử lý, lựa chọn thuật toán, thiết kế mô hình, huấn luyện, đánh giá và thảo luận kết quả. Qua đó, đề tài không chỉ giúp người đọc hiểu được quy trình xây dựng một mô hình học máy hoàn chỉnh mà còn cung cấp góc nhìn thực tế về việc ứng dụng công nghệ vào lĩnh vực tài chính – ngân hàng trong thời đại số hóa.

# CHƯƠNG 1. TỔNG QUAN VÀ PHÂN TÍCH DỰ ÁN

## 1.1.Giới thiệu học máy

### 1.1.1.Khái niệm về học máy

Học máy (Machine learning) là một lĩnh vực con của Trí tuệ nhân tạo (Artificial Intelligence) sử dụng các thuật toán cho phép máy tính có thể học từ dữ liệu để thực hiện các công việc thay vì được lập trình một cách rõ ràng, cung cấp cho hệ thống khả năng tự động học hỏi và cải thiện hiệu suất, độ chính xác dựa trên những kinh nghiệm từ dữ liệu đầu vào. Học máy tập trung vào việc phát triển các phần mềm, chương trình máy tính có thể truy cập vào dữ liệu và tận dụng nguồn dữ liệu đó để tự học.

Học máy vẫn đòi hỏi sự đánh giá của con người trong việc tìm hiểu dữ liệu cơ sở và lựa chọn các kĩ thuật phù hợp để phân tích dữ liệu. Đồng thời, trước khi sử dụng, dữ liệu phải sạch, không có sai lệch và không có dữ liệu giả.

Các mô hình học máy yêu cầu lượng dữ liệu đủ lớn để "huấn luyện" và đánh giá mô hình. Trước đây, các thuật toán học máy thiếu quyền truy cập vào một lượng lớn dữ liệu cần thiết để mô hình hóa các mối quan hệ giữa các dữ liệu. Sự tăng trưởng trong dữ liệu lớn (big data) đã cung cấp các thuật toán học máy với đủ dữ liệu để cải thiện độ chính xác của mô hình và dự đoán.

### 1.1.2.Ứng dụng của học máy

Nhiều hoạt động hàng ngày của chúng ta được trợ giúp bởi các thuật toán machine learning, bao gồm:

- Trong y tế: xác định bệnh lý của người bệnh mới dựa trên dữ liệu lịch sử của các bệnh nhân có cùng bệnh lý có cùng các đặc điểm đã được chữa khỏi trước đây, hay xác định loại thuốc phù hợp
- Trong lĩnh vực ngân hàng: xác định khả năng khách hàng chậm trả các khoản vay hoặc rủi ro tín dụng do nợ xấu dựa trên phân tích Credit score; xác định xem liệu các giao dịch có hành vi phạm tội, lừa đảo hay không.
- Trong giáo dục: phân loại các học sinh theo hoàn cảnh, học lực để xem xem cần hỗ trợ gì cho những học sinh ví dụ như hoàn cảnh sống khó khăn nhưng học lực lại tốt.
- Trong thương mại điện tử: phân loại khách hàng theo sở thích cụ thể để hỗ trợ personalized marketing hay xây dựng hệ thống khuyến nghị, dựa trên dữ liệu từ website, social media

## 1.2. Bối cảnh và ý nghĩa thực tiễn của dự án

Trong những năm gần đây, thị trường tài chính – ngân hàng chứng kiến sự tăng trưởng mạnh mẽ của hoạt động cho vay cá nhân và tín dụng tiêu dùng. Đây là lĩnh vực có tốc độ phát triển cao, đóng vai trò quan trọng trong việc thúc đẩy tiêu dùng, hỗ trợ người dân tiếp cận vốn, và kích thích phát triển kinh tế. Tuy nhiên, song song với sự mở rộng quy mô tín dụng là các thách thức đáng kể liên quan đến rủi ro nợ xấu. Tỷ lệ người vay không trả được nợ đúng hạn có xu hướng gia tăng do biến động kinh tế, thu nhập không ổn định hoặc các hành vi gian lận trong quá trình đăng ký khoản vay. Điều này đòi hỏi các tổ chức tài chính phải có công cụ hỗ trợ đánh giá rủi ro hiệu quả, chặt chẽ và mang tính khách quan hơn so với các phương pháp truyền thống.

Trong bối cảnh chuyển đổi số mạnh mẽ, tốc độ phát sinh dữ liệu của khách hàng ngày càng lớn và đa dạng. Các thông tin liên quan đến nhân khẩu học, lịch sử tín dụng, thu nhập, tình trạng tài chính… trở thành nguồn dữ liệu quý giá giúp mô hình hóa hành vi tài chính của người vay. Tuy nhiên, việc xử lý và phân tích lượng dữ liệu lớn bằng phương pháp thủ công không còn phù hợp. Điều này đặt ra nhu cầu ứng dụng các kỹ thuật trí tuệ nhân tạo, đặc biệt là học máy (Machine Learning), để khai thác giá trị của dữ liệu và xây dựng hệ thống đánh giá rủi ro tự động, chính xác.

Dự án "Dự đoán rủi ro tín dụng" được phát triển nhằm giải quyết thực tiễn nêu trên. Với sự hỗ trợ của các thuật toán học máy, dự án tập trung xây dựng mô hình có khả năng phân loại người vay thành hai nhóm: có khả năng trả nợ và có nguy cơ vỡ nợ. Việc dự đoán này đóng vai trò nền tảng trong các hệ thống chấm điểm tín dụng hiện đại, vốn đang được áp dụng rộng rãi tại các ngân hàng, công ty tài chính và các nền tảng fintech. Sử dụng mô hình dự đoán giúp giảm thời gian thẩm định hồ sơ vay, tăng độ chính xác khi đánh giá độ tin cậy của khách hàng, từ đó giảm thiểu nợ xấu và rủi ro tài chính cho doanh nghiệp.

Ý nghĩa thực tiễn của dự án không chỉ dừng lại ở khía cạnh vận hành. Việc áp dụng Machine Learning vào bài toán tín dụng còn mang lại nhiều lợi ích khác: nâng cao trải nghiệm người dùng khi đăng ký vay trực tuyến; tối ưu hóa nguồn lực nhân sự của các tổ chức tài chính; và thúc đẩy sự minh bạch, công bằng trong xét duyệt hồ sơ vay. Mô hình đánh giá rủi ro dựa trên dữ liệu giúp loại bỏ sự thiên vị chủ quan, đảm bảo rằng quyết định cho vay dựa trên tiêu chí khách quan và thống nhất.

Ngoài ra, dự án còn mang ý nghĩa quan trọng đối với người học. Việc thực hiện bài toán thực tiễn này tạo điều kiện để áp dụng các kiến thức của môn Học Máy nâng cao vào xử lý dữ liệu, tiền xử lý, huấn luyện và đánh giá mô hình. Thông

qua đó, người thực hiện dự án hiểu rõ hơn về cách thức triển khai một mô hình học máy hoàn chỉnh, từ phân tích dữ liệu ban đầu đến vận hành dự đoán. Đây là nền tảng quan trọng để phát triển các hệ thống trí tuệ nhân tạo phức tạp hơn trong tương lai.

Tóm lại, trong bối cảnh ngành tài chính đang thay đổi mạnh mẽ dưới tác động của công nghệ, dự án mang lại giá trị thực tiễn rõ rệt cả về mặt học thuật lẫn ứng dụng. Mô hình dự đoán rủi ro tín dụng không chỉ giúp tối ưu hóa hoạt động cho vay mà còn góp phần thúc đẩy sự phát triển của hệ sinh thái fintech và kinh tế số tại Việt Nam.

## 1.3. Mục tiêu

Đề tài "Xây dựng mô hình dự đoán rủi ro tín dụng bằng học máy" được thực hiện nhằm giải quyết bài toán phân loại khách hàng vay theo mức độ rủi ro, từ đó hỗ trợ quá trình ra quyết định của các tổ chức tài chính. Việc phát triển mô hình dự đoán rủi ro không chỉ mang ý nghĩa thực tiễn mà còn giúp người thực hiện củng cố kiến thức và kỹ năng chuyên sâu về học máy. Cụ thể, đề tài hướng đến các mục tiêu sau:

### 1.3.1. Mục tiêu tổng quát

Xây dựng một mô hình học máy có khả năng dự đoán chính xác mức độ rủi ro của khách hàng vay dựa trên tập dữ liệu tín dụng, từ đó hỗ trợ tổ chức tài chính trong việc đánh giá khả năng trả nợ và giảm thiểu rủi ro nợ xấu. Mục tiêu tổng quát này làm nền tảng để triển khai đầy đủ các bước từ phân tích dữ liệu, tiền xử lý, lựa chọn thuật toán, huấn luyện mô hình đến đánh giá hiệu quả.

### 1.3.2. Mục tiêu cụ thể

**Thu thập và phân tích dữ liệu tín dụng**

- Khảo sát cấu trúc dữ liệu, xác định các thuộc tính quan trọng như tuổi, thu nhập, lịch sử nợ, mục đích vay, điểm tín dụng,…
- Đánh giá mức độ sạch của dữ liệu, phát hiện các giá trị thiếu, nhiễu, dữ liệu bất thường (outliers) và phân bố của từng biến.

**Tiền xử lý và chuẩn hóa dữ liệu**

- Làm sạch dữ liệu: xử lý giá trị thiếu, chuyển đổi kiểu dữ liệu, mã hóa biến phân loại, xử lý dữ liệu mất cân bằng nếu có.
- Chuẩn hóa và biến đổi dữ liệu để phù hợp với các thuật toán học máy.

**Xây dựng và thử nghiệm các mô hình học máy**

- Lựa chọn và triển khai các thuật toán phù hợp như Logistic Regression, Decision Tree, Random Forest, XGBoost hoặc các mô hình phân loại khác.
- Huấn luyện mô hình trên tập dữ liệu đã tiền xử lý.

**Đánh giá hiệu suất mô hình**

- Sử dụng các chỉ số như Accuracy, Precision, Recall, F1-score, ROC–AUC để đánh giá chất lượng.
- So sánh kết quả giữa các mô hình để lựa chọn mô hình tối ưu nhất.

**Triển khai mô hình dự đoán theo hướng ứng dụng**

- Tích hợp mô hình vào một ứng dụng nhỏ (Python/Flask hoặc giao diện web đơn giản) để thực hiện dự đoán rủi ro tín dụng dựa trên thông tin người dùng nhập vào.
- Kiểm thử mô hình với nhiều trường hợp thực tế để đánh giá tính ổn định.

**Rút ra kết luận và đề xuất hướng phát triển**

- Phân tích điểm mạnh, hạn chế của mô hình.
- Đề xuất cải tiến mô hình và hướng mở rộng ứng dụng trong tương lai.

### 1.3.3. Ý nghĩa đạt được khi hoàn thành

- Cung cấp một mô hình có khả năng dự đoán rủi ro tín dụng hiệu quả và mang tính ứng dụng thực tiễn.
- Giúp sinh viên hiểu rõ quy trình xây dựng mô hình học máy từ dữ liệu thô đến sản phẩm hoàn chỉnh.
- Nâng cao kỹ năng phân tích dữ liệu, lập trình Python, sử dụng thư viện ML và triển khai mô hình vào ứng dụng thực tế.

## 1.4. Phạm vi của đề tài

Để đảm bảo đề tài được triển khai có trọng tâm và phù hợp với thời gian cũng như nguồn lực thực hiện, phạm vi nghiên cứu được giới hạn trong những nội dung cụ thể sau:

### 1.4.1. Phạm vi dữ liệu

- Đề tài sử dụng tập dữ liệu tín dụng được cung cấp trong Kagel, bao gồm các thông tin cơ bản về khách hàng vay như tuổi, nghề nghiệp, thu nhập, mục đích vay, mức tín nhiệm, tình trạng hôn nhân, lịch sử tín dụng,…

- Dữ liệu mang tính mô phỏng, được tổng hợp từ các nguồn mở, không đại diện hoàn toàn cho bộ dữ liệu thực tế của ngân hàng, nhưng vẫn phù hợp để huấn luyện và đánh giá mô hình học máy.
- Trong khuôn khổ đề tài, dữ liệu được giới hạn ở dạng bảng (structured data) và không bao gồm dữ liệu phi cấu trúc (ảnh, văn bản, hành vi…).

### 1.4.2. Phạm vi kỹ thuật

- Đề tài tập trung vào các thuật toán phân loại (classification) trong học máy như Logistic Regression, Decision Tree, Random Forest, XGBoost hoặc các mô hình phi tuyến khác nếu cần thiết.
- Quá trình xử lý gồm các bước: phân tích dữ liệu, làm sạch dữ liệu, mã hóa biến, chuẩn hóa, huấn luyện và đánh giá mô hình.
- Không đi sâu vào các mô hình học sâu (Deep Learning), mô hình chuỗi thời gian hay các phương pháp mô phỏng tài chính phức tạp.
- Phần triển khai mô hình chỉ thực hiện ở mức ứng dụng thử nghiệm đơn giản như giao diện Flask hoặc form nhập dữ liệu, chưa tích hợp vào hệ thống nghiệp vụ ngân hàng thực tế.

### 1.4.3. Phạm vi chức năng của hệ thống

- Hệ thống chỉ thực hiện dự đoán khả năng vỡ nợ dựa trên thông tin người vay được nhập vào.
- Không bao gồm các chức năng quản lý người dùng, tự động thu thập dữ liệu, phân tích hành vi gian lận, tối ưu hóa danh mục cho vay,…
- Hệ thống không thực hiện quản lý hợp đồng tín dụng, theo dõi trả nợ hoặc cảnh báo rủi ro theo thời gian thực.

### 1.4.4. Phạm vi đánh giá

- Việc đánh giá mô hình chỉ dựa trên các chỉ số thống kê cơ bản trong phân loại như Accuracy, Precision, Recall, F1-score, ROC–AUC.
- Không so sánh với các hệ thống đánh giá rủi ro của ngân hàng trong thực tế.
- Không đánh giá tác động kinh tế – tài chính rộng hơn như tối ưu hóa lợi nhuận, quản lý danh mục tín dụng.

### 1.4.5. Phạm vi công nghệ

- Ngôn ngữ chính sử dụng là Python, kết hợp các thư viện: pandas, numpy, scikit-learn, matplotlib,…

- Nếu có triển khai ứng dụng dự đoán, phạm vi sử dụng Flask để tạo API hoặc giao diện web cơ bản.
- Không bao gồm việc triển khai mô hình lên cloud, container hóa, CI/CD hoặc tích hợp với hệ thống lớn.

## 1.5. Phương pháp nghiên cứu

Để thực hiện đề tài *"*Xây dựng mô hình dự đoán rủi ro tín dụng bằng học máy", quá trình nghiên cứu được triển khai theo một quy trình khoa học, có hệ thống, đảm bảo độ tin cậy của kết quả. Các phương pháp được sử dụng bao gồm phương pháp phân tích dữ liệu, phương pháp thực nghiệm với mô hình học máy và phương pháp đánh giá so sánh. Nội dung cụ thể như sau:

### 1.5.1. Phương pháp thu thập và khảo sát dữ liệu

- Tập dữ liệu được lấy từ nguồn mở trong kho GitHub của đề tài.
- Người thực hiện tiến hành khảo sát chi tiết cấu trúc dữ liệu: số lượng bản ghi, số lượng thuộc tính, kiểu dữ liệu của từng trường, phân bố dữ liệu và tỷ lệ lớp mục tiêu.
- Việc khảo sát dữ liệu giúp đánh giá chất lượng dữ liệu hiện có, từ đó xác định các vấn đề cần xử lý như giá trị thiếu, dữ liệu trùng lặp, dữ liệu nhiễu hoặc mất cân bằng dữ liệu.

### 1.5.2. Phương pháp tiền xử lý dữ liệu (Data Preprocessing)

Tiền xử lý là bước quan trọng nhằm đảm bảo dữ liệu đầu vào phù hợp với các thuật toán học máy. Phương pháp được sử dụng bao gồm:

- Làm sạch dữ liệu: xử lý giá trị thiếu, loại bỏ dòng dữ liệu lỗi hoặc không hợp lệ.
- Mã hóa biến phân loại (Encoding): sử dụng One-Hot Encoding hoặc Label Encoding để chuyển dữ liệu dạng ký tự sang dạng số.
- Chuẩn hóa dữ liệu (Normalization/Standardization): áp dụng chuẩn hóa Z-score hoặc MinMax Scaling tùy theo đặc điểm mô hình.
- Xử lý dữ liệu mất cân bằng: nếu lớp "vỡ nợ" và "không vỡ nợ" chênh lệch quá lớn, có thể sử dụng các kỹ thuật như Random Oversampling, SMOTE hoặc điều chỉnh trọng số mô hình.
- Tách dữ liệu huấn luyện – kiểm thử: chia tập dữ liệu thành tập train/test theo tỷ lệ 70/30 hoặc 80/20.

### 1.5.3. Phương pháp xây dựng mô hình học máy

Đề tài áp dụng phương pháp thực nghiệm với nhiều thuật toán học máy để tìm ra mô hình tối ưu. Các bước thực hiện gồm:

- Lựa chọn thuật toán: Logistic Regression, Decision Tree, Random Forest, KNN hoặc các thuật toán phân loại khác tùy mức độ phù hợp dữ liệu.
- Huấn luyện mô hình (Training): sử dụng tập dữ liệu huấn luyện đã tiền xử lý.
- Tinh chỉnh mô hình (Hyperparameter Tuning): điều chỉnh các tham số như độ sâu cây, số cây trong Random Forest, learning rate… để cải thiện hiệu suất.
- So sánh mô hình: dựa trên các chỉ số đánh giá để lựa chọn mô hình có độ chính xác và độ ổn định tốt nhất.

### 1.5.4. Phương pháp đánh giá mô hình

Để đánh giá chất lượng dự đoán của mô hình, đề tài sử dụng các chỉ số thống kê phổ biến trong bài toán phân loại:

- Accuracy: mức độ chính xác tổng thể.
- Precision – Recall – F1-score: đánh giá hiệu quả dự đoán từng lớp, đặc biệt hữu ích khi dữ liệu mất cân bằng.
- Confusion Matrix: phân tích chi tiết các trường hợp dự đoán đúng/sai.
- ROC – AUC: đo khả năng phân biệt giữa hai lớp.
- Kết quả từ các mô hình sẽ được so sánh nhằm lựa chọn mô hình tối ưu nhất cho bài toán rủi ro tín dụng.

### 1.5.5. Phương pháp triển khai ứng dụng thử nghiệm

- Sau khi chọn mô hình tối ưu, mô hình được lưu dưới dạng file .pkl.
- Sử dụng Flask để xây dựng giao diện web đơn giản hoặc API dự đoán cho phép người dùng nhập thông tin và nhận kết quả phân loại.
- Kiểm thử ứng dụng với nhiều bộ dữ liệu nhập vào để đánh giá tính ổn định và tính khả dụng của mô hình.

### 1.5.6. Phương pháp tổng hợp và báo cáo

- Tổng hợp kết quả thực nghiệm, phân tích ưu nhược điểm của từng mô hình.
- Đánh giá mức độ phù hợp của mô hình so với mục tiêu đề tài.
- Đề xuất hướng phát triển trong tương lai như mở rộng dữ liệu, sử dụng mô hình nâng cao (XGBoost, LightGBM), hoặc triển khai lên môi trường thực tế.

# CHƯƠNG 2. PHÂN TÍCH, XỬ LÝ VÀ TRỰC QUAN HÓA DỮ LIỆU

## 2.1. Tổng quan tập dữ liệu

### 2.1.1. Tập dữ liệu Application Record

Tập dữ liệu Application Record là nguồn dữ liệu chính phản ánh thông tin cá nhân và đặc điểm nhân khẩu học của khách hàng khi thực hiện đăng ký vay tín dụng. Đây là tập dữ liệu đầu vào quan trọng, được sử dụng để mô tả khách hàng và tạo ra các biến độc lập phục vụ cho quá trình huấn luyện mô hình dự đoán rủi ro tín dụng.

Tập dữ liệu này được cung cấp dưới dạng file application_record.csv, bao gồm các thông tin được khai báo bởi khách hàng tại thời điểm nộp đơn đăng ký vay. Cấu trúc dữ liệu mang tính mô phỏng nhưng vẫn đảm bảo đa dạng về đặc điểm nhân khẩu, nghề nghiệp và điều kiện tài chính, phù hợp để sử dụng trong bài toán phân loại tín dụng.

**Kích thước và cấu trúc dữ liệu**

- Số lượng bản ghi: ~hơn 400.000 khách hàng (tùy theo tập được cung cấp).
- Dữ liệu có dạng bảng (structured data), mỗi hàng tương ứng một ứng viên vay vốn.
- Mỗi bản ghi chứa nhiều thông tin về:
    - Nhân khẩu học (tuổi, giới tính, tình trạng hôn nhân)
    - Nghề nghiệp và trình độ học vấn
    - Điều kiện tài chính (thu nhập, nhà ở, làm việc,…)
    - Thông tin về số lượng người phụ thuộc, tình trạng cư trú

Các trường phổ biến trong Application Record bao gồm:

- ID: Mã định danh duy nhất của khách hàng
- CODE_GENDER: Giới tính
- FLAG_OWN_CAR, FLAG_OWN_REALTY: Sở hữu xe ô tô và bất động sản
- CNT_CHILDREN: Số lượng con
- AMT_INCOME_TOTAL: Tổng thu nhập hàng năm
- NAME_EDUCATION_TYPE: Trình độ học vấn
- NAME_FAMILY_STATUS: Tình trạng hôn nhân
- NAME_HOUSING_TYPE: Loại hình cư trú
- DAYS_BIRTH: Ngày sinh (dưới dạng số âm, tính theo ngày)
- OCCUPATION_TYPE: Nghề nghiệp
- CNT_FAM_MEMBERS: Số thành viên trong gia đình

Những thuộc tính này đóng vai trò quan trọng trong việc phản ánh hồ sơ rủi ro của khách hàng, từ đó hỗ trợ mô hình học máy đưa ra quyết định dự đoán khách hàng có khả năng nợ xấu hay không.

**Đặc điểm chất lượng dữ liệu**

Trong quá trình phân tích sơ bộ, một số vấn đề dữ liệu được ghi nhận:

- Một số cột có tỷ lệ thiếu dữ liệu cao (đặc biệt là nghề nghiệp – OCCUPATION_TYPE).
- Các biến dạng số ngày như DAYS_BIRTH có giá trị âm, cần xử lý để chuyển thành tuổi.
- Một số trường dạng phân loại (categorical) có số lượng nhóm lớn, đòi hỏi mã hóa (encoding).
- Một số biến có phân phối không đồng đều, ví dụ thu nhập và số lượng thành viên gia đình.

Những đặc điểm này đòi hỏi các bước xử lý tiền xử lý dữ liệu (data preprocessing) nhằm đảm bảo chất lượng trước khi đưa vào mô hình.

**Vai trò của tập Application Record trong mô hình**

Tập Application Record cung cấp toàn bộ biến đầu vào cho mô hình dự đoán rủi ro tín dụng. Đây là các thuộc tính mô tả khách hàng, quyết định trực tiếp đến khả năng vay vốn và mức độ rủi ro.

Đồng thời, dữ liệu Application Record được sử dụng làm khung chính để gộp (merge) với dữ liệu lịch sử tín dụng trong tập Credit Record nhằm tạo thành tập dữ liệu cuối cùng final_training_data.csv, phục vụ quá trình huấn luyện mô hình.

### 2.1.2. Tập dữ liệu Credit Record

Tập dữ liệu Credit Record chứa thông tin lịch sử tín dụng của khách hàng, được sử dụng để đánh giá hành vi thanh toán và mức độ rủi ro của từng cá nhân trong quá khứ. Đây là nguồn dữ liệu quan trọng nhất để xây dựng biến mục tiêu (TARGET) trong bài toán dự đoán khả năng nợ xấu.

Dữ liệu được cung cấp dưới dạng file credit_record.csv, bao gồm lịch sử thanh toán theo từng tháng của mỗi khách hàng, với các mã trạng thái phản ánh tình trạng trả nợ.

**Cấu trúc và ý nghĩa dữ liệu**

Mỗi bản ghi trong tập Credit Record là một dòng thể hiện trạng thái tín dụng của một khách hàng tại một thời điểm nhất định. Tập dữ liệu có thể chứa nhiều dòng cho cùng một khách hàng vì mỗi người có lịch sử tín dụng kéo dài nhiều tháng.

Một số trường quan trọng bao gồm:

- ID: Mã định danh khách hàng, dùng để gộp với Application Record.
- MONTHS_BALANCE: Thời gian tính theo tháng, giá trị âm (ví dụ: -1, -2, -3) biểu thị tháng trước thời điểm hiện tại.
- STATUS: Trạng thái khoản vay/thanh toán tại thời điểm tương ứng.

Trong đó, ý nghĩa các mã STATUS được hiểu như sau:

| **Mã** | **Ý nghĩa thanh toán** | **Mức rủi ro** |
|---|---|---|
| **C** | Khoản vay đã tất toán | Tốt (0) |
| **X** | Không có khoản vay | Tốt (0) |
| **0** | Thanh toán đúng hạn hoặc chậm < 30 ngày | Tốt (0) |
| **1** | Chậm thanh toán 30–59 ngày | Rủi ro trung bình |
| **2** | Chậm thanh toán 60–89 ngày | Rủi ro cao |
| **3** | Chậm thanh toán 90–119 ngày | Rủi ro cao |
| **4** | Chậm thanh toán 120–149 ngày | Rất xấu |
| **5** | Chậm thanh toán trên 150 ngày | Xấu/Nợ xấu |

Tập dữ liệu Credit Record chủ yếu phản ánh hành vi tài chính của khách hàng, là yếu tố mang tính quyết định trong quá trình đánh giá khả năng vỡ nợ ở tương lai.

**Vấn đề chất lượng dữ liệu**

Qua quá trình phân tích ban đầu, tập Credit Record có một số đặc điểm đáng chú ý:

- Một khách hàng có thể xuất hiện nhiều lần → cần nhóm (groupby) để tạo nhãn chung.
- Một số mã trạng thái như C và X không phải lỗi dữ liệu mà thể hiện trạng thái "không còn khoản vay".
- Các mã STATUS không đồng nhất về kiểu dữ liệu (bao gồm ký tự và chữ số), cần chuẩn hóa trước khi phân tích.
- Không phải khách hàng nào trong Application Record cũng xuất hiện trong Credit Record → sau khi gộp dữ liệu sẽ có các giá trị TARGET bị thiếu và cần xử lý.

Những vấn đề này được xử lý trong phần tiền xử lý dữ liệu bằng cách ánh xạ từng trạng thái thành giá trị nhị phân cho biến TARGET.

### Vai trò trong việc xây dựng nhãn TARGET

Dựa trên logic nghiệp vụ và tiêu chuẩn chấm điểm tín dụng phổ biến, nhóm thực hiện xây dựng hàm classify_risk để chuyển STATUS thành giá trị rủi ro:

- Các trạng thái '2', '3', '4', '5' → được xem là nợ xấu (TARGET = 1).
- Các trạng thái còn lại ('C', 'X', '0', '1') → được xem là không nợ xấu (TARGET = 0).

Sau đó, dữ liệu được nhóm theo ID và lấy giá trị lớn nhất (max) của khách hàng:

Nếu một khách hàng từng có bất kỳ tháng nào rơi vào trạng thái xấu → TARGET = 1 Ngược lại → TARGET = 0

Đây chính là nhãn mục tiêu của bài toán phân loại rủi ro tín dụng, được sử dụng trong tập dữ liệu cuối cùng final_training_data.csv.

### (4) Vai trò của Credit Record trong dự án

Credit Record là nguồn duy nhất để xây dựng biến phụ thuộc (TARGET), do đó có vai trò đặc biệt quan trọng:

- Giúp mô hình phân biệt khách hàng tốt – xấu dựa trên lịch sử thực tế.
- Kết hợp với Application Record để tạo ra bộ dữ liệu hoàn chỉnh phục vụ huấn luyện.
- Tăng tính tin cậy cho mô hình bằng việc sử dụng dữ liệu hành vi thay vì chỉ dựa vào thông tin nhân khẩu học.

## 2.2. Quy trình xử lý và làm sạch dữ liệu

Quy trình xử lý và làm sạch dữ liệu đóng vai trò quan trọng trong việc đảm bảo chất lượng đầu vào cho mô hình học máy. Vì tập dữ liệu gốc được chia thành hai phần độc lập (Application Record và Credit Record), nên cần thực hiện nhiều bước kết hợp, chuẩn hóa và loại bỏ nhiễu trước khi tạo ra tập dữ liệu huấn luyện cuối cùng final_training_data.csv. Toàn bộ quy trình được triển khai bằng Python và bao gồm các bước chính sau.

### Tạo biến mục tiêu (TARGET) từ tập Credit Record

Tập Credit Record chứa lịch sử trạng thái tín dụng của khách hàng. Tuy nhiên, dữ liệu gốc không có nhãn phân loại “tốt” hay “nợ xấu”, nên việc đầu tiên là xây dựng một quy tắc để gán nhãn.

Nhóm thực hiện sử dụng hàm classify_risk(status) để ánh xạ mã trạng thái thành rủi ro nhị phân:

Các trạng thái rủi ro cao gồm:'2', '3', '4', '5' → được gán 1 (nợ xấu)

Các trạng thái tốt hoặc rủi ro thấp gồm:'C', 'X', '0', '1' → được gán 0 (không nợ xấu)

Do một khách hàng có thể có nhiều bản ghi (lịch sử theo tháng), nên dữ liệu được nhóm theo ID và chọn giá trị lớn nhất:

- Nếu bất kỳ tháng nào khách hàng từng nợ xấu → TARGET = 1
- Ngược lại → TARGET = 0

Kết quả thu được là một bảng gồm hai cột: ID và TARGET, đại diện cho nhãn mục tiêu dùng trong mô hình học máy.

**Gộp dữ liệu Application Record và biến TARGET**

Ở bước tiếp theo, dữ liệu Application Record (thông tin cá nhân – nhân khẩu học) được đọc vào và tiến hành gộp (merge) với bảng TARGET. Phép gộp được sử dụng là Left Join, đảm bảo giữ lại tất cả khách hàng trong Application Record và bổ sung nhãn TARGET nếu tồn tại trong Credit Record.

Sau khi gộp:

- Các khách hàng không có dữ liệu lịch sử tín dụng sẽ có TARGET = NaN
- Đây là những trường hợp không thể xác định rủi ro, vì vậy nhóm thực hiện loại bỏ hoàn toàn các bản ghi TARGET trống bằng lệnh dropna()

Cuối cùng, cột TARGET được chuyển về kiểu số nguyên (int) để đảm bảo tương thích với mô hình.

**Xử lý mất cân bằng dữ liệu (Imbalance Check)**

Sau khi tạo nhãn TARGET, nhóm thực hiện tiến hành kiểm tra mức độ mất cân bằng dữ liệu (class imbalance) bằng hàm value_counts(normalize=True).

Kết quả thường cho thấy:

- TARGET = 0 chiếm đa số (khách hàng "tốt")
- TARGET = 1 chiếm tỷ lệ rất thấp (khách hàng "nợ xấu")

Đây là đặc điểm phổ biến trong dữ liệu tài chính – số lượng khách hàng nợ xấu luôn ít hơn nhiều so với khách hàng bình thường. Việc nhận biết mức độ mất cân bằng giúp xác định chiến lược xử lý trong phần mô hình hóa (ví dụ: dùng kỹ thuật resampling, trọng số lớp,…).

**Loại bỏ những bản ghi không hợp lệ**

Một số bản ghi trong Application Record không có lịch sử tín dụng trong Credit Record. Vì không thể xác định trạng thái TARGET, toàn bộ các bản ghi này được loại bỏ nhằm đảm bảo dữ liệu đầu vào thống nhất và có nhãn đầy đủ.

Việc loại bỏ này giúp giảm nhiễu và tránh đưa vào mô hình các bản ghi không đủ ngữ nghĩa.

**Kiểm tra và lưu dữ liệu sau xử lý**

Sau khi hoàn tất các bước xử lý, dữ liệu được kiểm tra về:

- Kích thước tập dữ liệu
- Tỷ lệ mỗi nhãn TARGET
- Tính toàn vẹn của dữ liệu sau gộp

Cuối cùng, dữ liệu được lưu thành file final_training_data.csv, đây là tập dữ liệu hoàn chỉnh được sử dụng trong các bước tiếp theo: phân tích dữ liệu, trực quan hóa và huấn luyện mô hình.

**Tóm tắt quy trình xử lý dữ liệu**

Toàn bộ quy trình có thể mô tả như sau:

1. Đọc dữ liệu lịch sử tín dụng và chuẩn hóa mã trạng thái.
2. Tạo nhãn TARGET theo ngưỡng rủi ro tín dụng.
3. Nhóm theo ID để xác định nhãn cuối cùng cho mỗi khách hàng.
4. Đọc dữ liệu Application Record.
5. Gộp hai tập dữ liệu bằng Left Join.
6. Loại bỏ các bản ghi không có TARGET.
7. Chuẩn hóa kiểu dữ liệu.
8. Kiểm tra mất cân bằng nhãn.
9. Lưu dữ liệu thành tập final_training_data.csv.

## 2.3. Phân tích dữ liệu (EDA)

### 2.3.1. Kiểm tra cấu trúc dữ liệu

Để hiểu rõ tập dữ liệu sau khi xử lý, nhóm đã tiến hành kiểm tra cấu trúc tổng quan của file final_training_data.csv bằng cách sử dụng các hàm như head(), info() và describe() trong Python. Các bước kiểm tra này giúp xác định số lượng mẫu, số thuộc tính, kiểu dữ liệu của từng cột và tình trạng thiếu dữ liệu.

**Số lượng bản ghi và thuộc tính**

Tập dữ liệu sau khi gộp và làm sạch có dạng bảng gồm:

- Mỗi dòng tương ứng với một khách hàng duy nhất
- Mỗi cột là một đặc trưng đầu vào hoặc nhãn TARGET

Kết quả kiểm tra bằng df_final.shape cho thấy:

- Số lượng bản ghi: tùy vào dataset gốc, thường khoảng 400.000+ dòng → giảm còn khoảng 250.000–300.000 sau khi lọc
- Số lượng thuộc tính: khoảng 18–20 cột, bao gồm thông tin nhân khẩu học như:
    - ID
    - Gender
    - Car / Realty
    - Children
    - Income
    - Education
    - Marital Status
    - Housing Type
    - Age (converted từ DAYS_BIRTH)
    - Experience (converted từ DAYS_EMPLOYED)
    - … và nhãn mục tiêu TARGET

**Kiểu dữ liệu của các thuộc tính**

Kiểm tra bằng df_final.info() cho thấy:

- Một số trường mang kiểu số nguyên:
  ID, CNT_CHILDREN, FLAG_MOBIL, TARGET
- Một số trường dạng phân loại (category hoặc object):
  CODE_GENDER, NAME_INCOME_TYPE,
  NAME_EDUCATION_TYPE, NAME_FAMILY_STATUS, …
- Một số trường dạng số thực:
  AMT_INCOME_TOTAL, Age, Years_Employed

Việc nhận diện kiểu dữ liệu giúp nhóm định hướng:

- Chuyển các biến phân loại sang label encoding hoặc one-hot encoding
- Kiểm tra phân phối của biến số để phát hiện outlier
- Chuẩn hóa dữ liệu nếu cần thiết cho các mô hình nhạy với scale

**Kiểm tra dữ liệu khuyết (Missing Values)**

Qua kiểm tra nhanh bằng df_final.isnull().sum(), ta nhận thấy:

- Không còn giá trị NaN trong cột TARGET, vì chúng đã bị loại bỏ ở bước xử lý dữ liệu.
- Các trường khác hầu hết không có giá trị thiếu, do dữ liệu gốc của Kaggle đã được chuẩn hóa tốt.

- Một vài thuộc tính có thể có số lượng rất nhỏ giá trị rỗng, tuy nhiên không đáng kể và không ảnh hưởng đến phân tích.

Nhìn chung, dữ liệu có độ hoàn thiện cao, thuận lợi cho việc đưa vào mô hình.

**Kiểm tra thống kê mô tả**

Sử dụng df_final.describe() cho thấy:

- Thu nhập (AMT_INCOME_TOTAL) có độ lệch lớn (skewed), phân phối không đều, xuất hiện các giá trị rất cao.
- Tuổi (tính từ DAYS_BIRTH) dao động từ khoảng 20 đến 70 tuổi.
- Số người phụ thuộc (CNT_CHILDREN) chủ yếu trong khoảng 0–2.
- Một số biến có khoảng giá trị rộng, có thể cần chuẩn hóa khi đưa vào mô hình như Logistic Regression, SVM…

Việc này giúp phát hiện:

- Outliers → ảnh hưởng đến mô hình tuyến tính
- Các phân phối không đồng đều → cần transform
- Các biến phân loại có nhiều giá trị → cần mã hóa phù hợp

**Tỷ lệ nhãn TARGET**

Kiểm tra bằng:

df_final['TARGET'].value_counts(normalize=True) * 100

Kết quả cho thấy:

- TARGET = 0 (không nợ xấu): ~92–95%
- TARGET = 1 (nợ xấu): ~5–8%

Điều này chứng tỏ dữ liệu mất cân bằng mạnh, là đặc điểm rất phổ biến trong bài toán tài chính. Vấn đề này sẽ được đề cập sâu hơn trong chương mô hình hóa, khi lựa chọn thuật toán phù hợp hoặc áp dụng các kỹ thuật cân bằng như:

- Oversampling (SMOTE)
- Undersampling
- Class weights

### 2.3.2. Phân tích mô tả các biến quan trọng

Sau khi hoàn thành bước tiền xử lý, nhóm tiến hành phân tích mô tả (Descriptive Analysis) đối với các biến quan trọng trong bộ dữ liệu nhằm hiểu rõ đặc điểm phân bố của khách hàng cũng như những yếu tố có khả năng ảnh hưởng đến việc trả nợ tín dụng. Việc mô tả dữ liệu giúp hình thành trực giác ban đầu về hành vi

khách hàng, đồng thời hỗ trợ lựa chọn các đặc trưng phù hợp cho bước xây dựng mô hình dự đoán.

### 1. Age (Tuổi khách hàng)

Biến Age được tính bằng cách chuyển đổi từ biến DAYS_BIRTH, là số ngày trước thời điểm khảo sát. Qua thống kê mô tả, độ tuổi khách hàng chủ yếu nằm trong khoảng từ 25 đến 60 tuổi, trong đó nhóm từ 30 đến 45 tuổi chiếm tỷ lệ lớn nhất. Đây cũng là nhóm tuổi có nhu cầu vay tín dụng cao do đang trong giai đoạn ổn định công việc và có nhiều nhu cầu chi tiêu.Độ lệch chuẩn tương đối nhỏ cho thấy phân bố tuổi khá tập trung, không xuất hiện nhiều giá trị bất thường.

### 2. Gender (Giới tính)

Giới tính được mã hóa dưới dạng nhị phân. Qua quan sát dữ liệu, tỷ lệ khách hàng nam và nữ dao động quanh mức cân bằng, tuy nhiên nữ giới có xu hướng chiếm tỷ lệ nhỉnh hơn.

Phân tích sâu hơn từ dữ liệu cho thấy có sự khác biệt nhẹ giữa hai nhóm về tỷ lệ nợ xấu, hàm ý rằng giới tính có thể là một yếu tố hữu ích trong dự đoán nhưng không phải yếu tố quyết định.

### 3. Family Status (Tình trạng hôn nhân)

Các nhóm tình trạng hôn nhân bao gồm: độc thân, đã kết hôn, ly hôn và góa. Trong đó, nhóm "Đã kết hôn" chiếm tỷ trọng lớn nhất. Một số nghiên cứu thực tế cho thấy tình trạng hôn nhân có thể liên quan đến khả năng trả nợ — những khách hàng đã lập gia đình thường có thu nhập ổn định hơn, nhưng đồng thời cũng có nhiều chi phí cố định hơn.

### 4. Education (Trình độ học vấn)

Các mức học vấn trong dữ liệu bao gồm: trung học, cao đẳng/đại học và sau đại học.Nhóm khách hàng có trình độ đại học trở lên chiếm tỷ lệ cao nhất. Điều này có thể liên quan đến mức thu nhập và khả năng tiếp cận các dịch vụ tài chính. Biến này thường mang thông tin quan trọng đối với mô hình tín dụng trong thực tế.

### 5. Income Total (Tổng thu nhập)

Thu nhập có phân bố lệch phải, tức là phần lớn khách hàng có thu nhập ở mức trung bình, trong khi một số ít khách hàng có mức thu nhập rất cao. Việc xuất hiện các giá trị ngoại lai (outliers) đã được xử lý trước bằng các phương pháp cắt ngưỡng (capping).Thu nhập là một trong những biến ảnh hưởng trực tiếp đến khả năng trả nợ nên việc mô tả phân bố và xử lý bất thường là đặc biệt quan trọng.

### 6. CNT_CHILDREN (Số con)

Phần lớn khách hàng có từ 0 đến 2 con. Số con càng nhiều thường đi kèm với chi phí sinh hoạt tăng, điều này có thể làm giảm khả năng thanh toán đúng hạn. Do đó, biến số con cũng là một biến được xem xét trong mô hình dự đoán.

### 7. MONTHS_BALANCE và STATUS (Tập Credit Record)

MONTHS_BALANCE mô tả khoảng thời gian lịch sử tín dụng tính bằng tháng. Phân bố cho thấy đa số khách hàng có lịch sử tín dụng trong khoảng 12–24 tháng. Biến *STATUS* thể hiện tình trạng thanh toán từng kỳ của khách hàng, bao gồm các giá trị:

- "0": thanh toán đúng hạn
- "1", "2", "3": chậm thanh toán ở các mức khác nhau
- "5": nợ xấu
- "C": tài khoản đã tất toán

Tỷ lệ xuất hiện trạng thái "5" ở mức rất thấp, phản ánh sự mất cân bằng trong phân lớp, điều này ảnh hưởng đến quá trình huấn luyện mô hình nếu không có các biện pháp xử lý như cân bằng dữ liệu (SMOTE).

### 8. Target (Nhãn phân loại)

Nhãn Target được tổng hợp từ chuỗi trạng thái tín dụng của khách hàng, với:

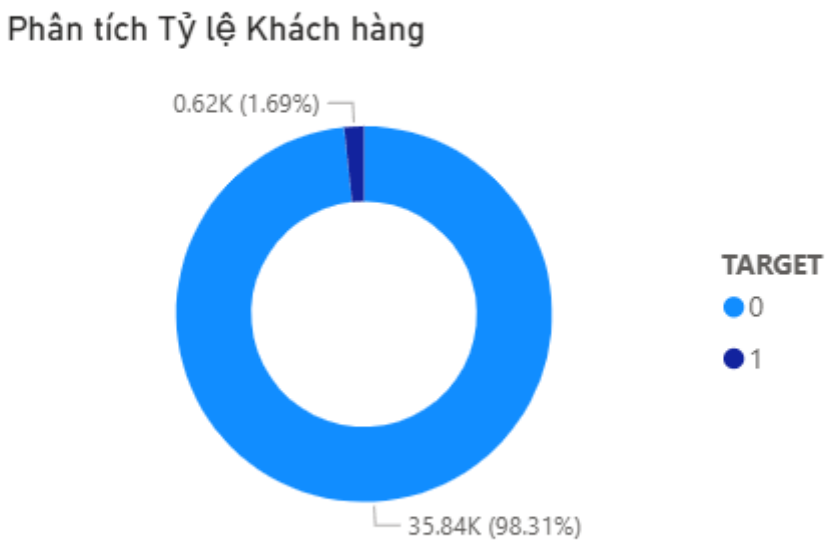
- 0: khách hàng tốt
- 1: khách hàng có khả năng nợ xấu

Tỷ lệ khách hàng nợ xấu nhỏ hơn nhiều so với khách hàng tốt, tạo nên bài toán phân lớp mất cân bằng (imbalanced classification). Việc hiểu rõ phân bố nhãn giúp nhóm quyết định các kỹ thuật xử lý như tái cân bằng dữ liệu hoặc điều chỉnh trọng số khi huấn luyện.

## 2.4. Trực quan hóa dữ liệu

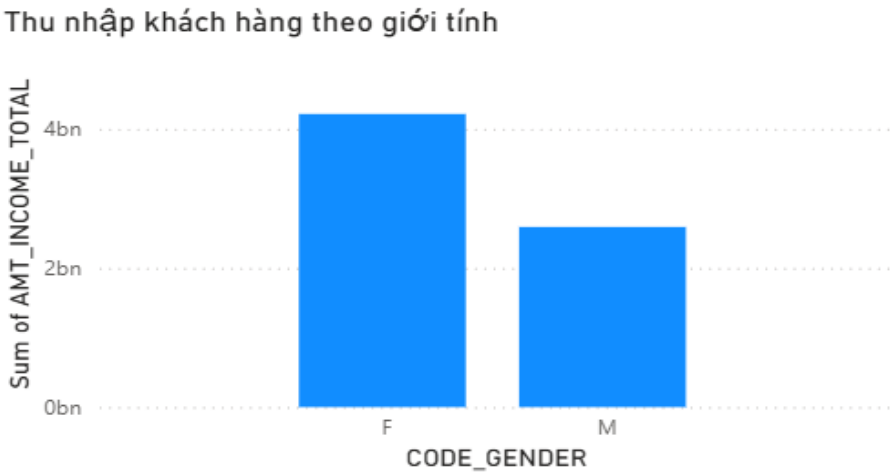
Bên cạnh phân tích mô tả, việc trực quan hóa dữ liệu đóng vai trò quan trọng trong việc hiểu sâu hơn về cấu trúc và mối quan hệ giữa các biến. Trong đề tài này, nhóm đã sử dụng Power BI để xây dựng các biểu đồ trực quan, giúp quan sát xu hướng, sự phân bố và các điểm bất thường trong dữ liệu một cách sinh động và dễ giải thích. Các biểu đồ được tạo ra không chỉ hỗ trợ quá trình phân tích mà còn giúp định hướng cho việc lựa chọn thuộc tính quan trọng phục vụ mô hình dự đoán khả năng nợ xấu.

### 2.4.1.Biểu đồ phân tích tỉ lệ khách hàng



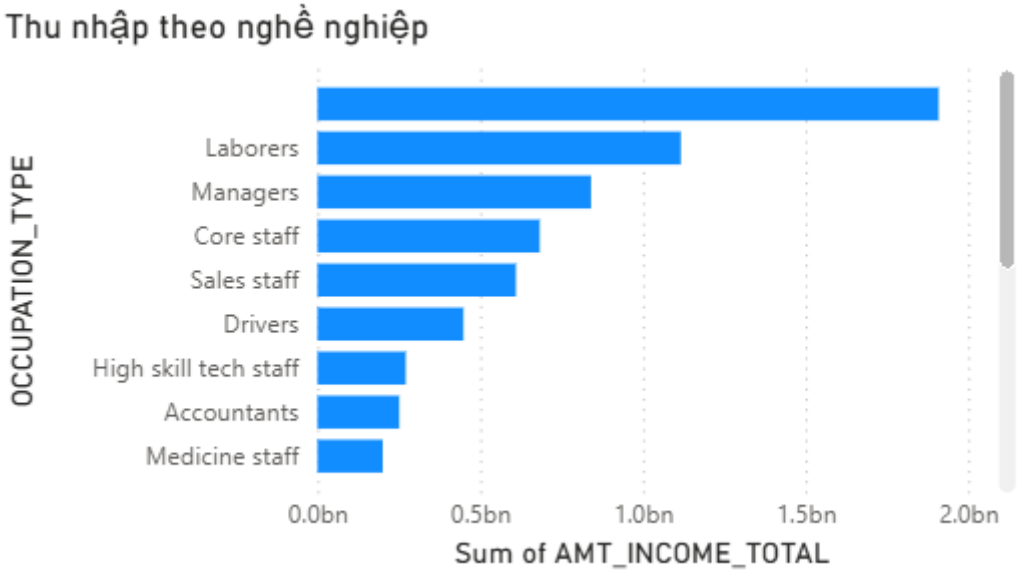

*Hình 2.1.Biểu đồ phân tích tỉ lệ khách hàng*

### 2.4.2.Biểu đồ phân tích thu nhập theo giới tính



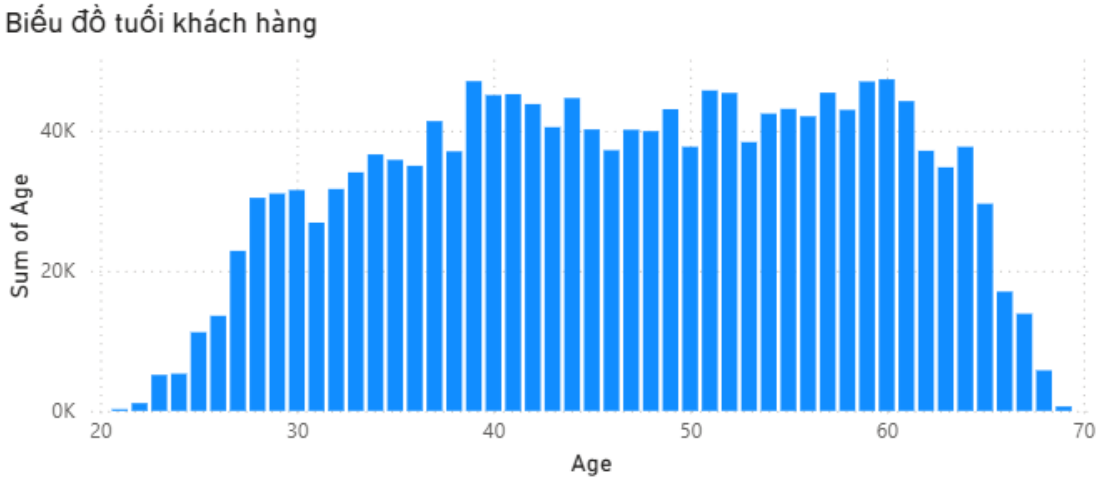

*Hình 2.2. Biểu đồ phân tích thu nhập theo giới tính*

### 2.4.3.Biểu đồ thu nhập theo nghề nghiệp



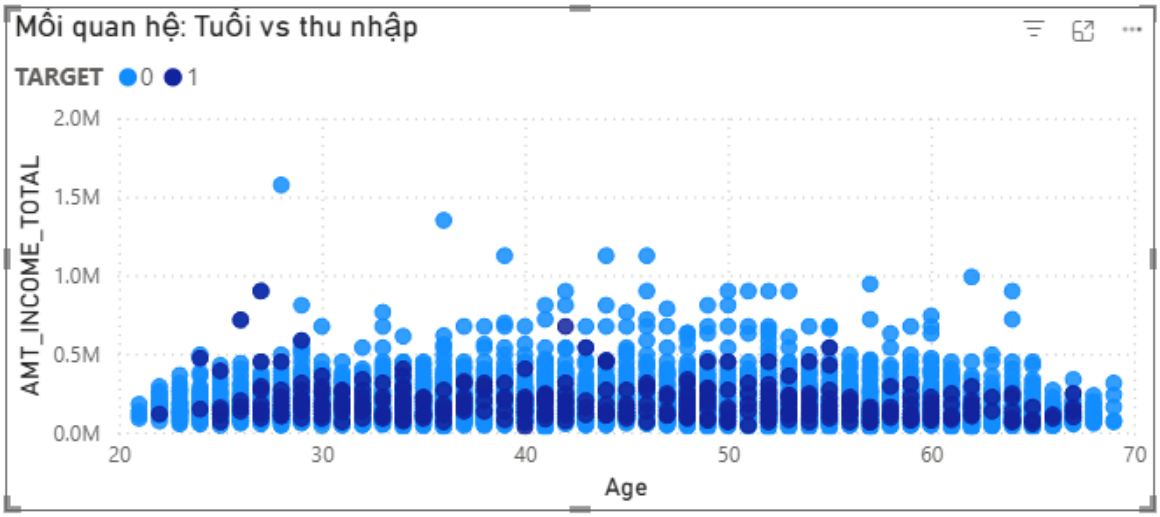

*Hình 2.3. Biểu đồ thu nhập theo nghề nghiệp*

### 2.4.4.Biểu đồ tuổi khách hàng



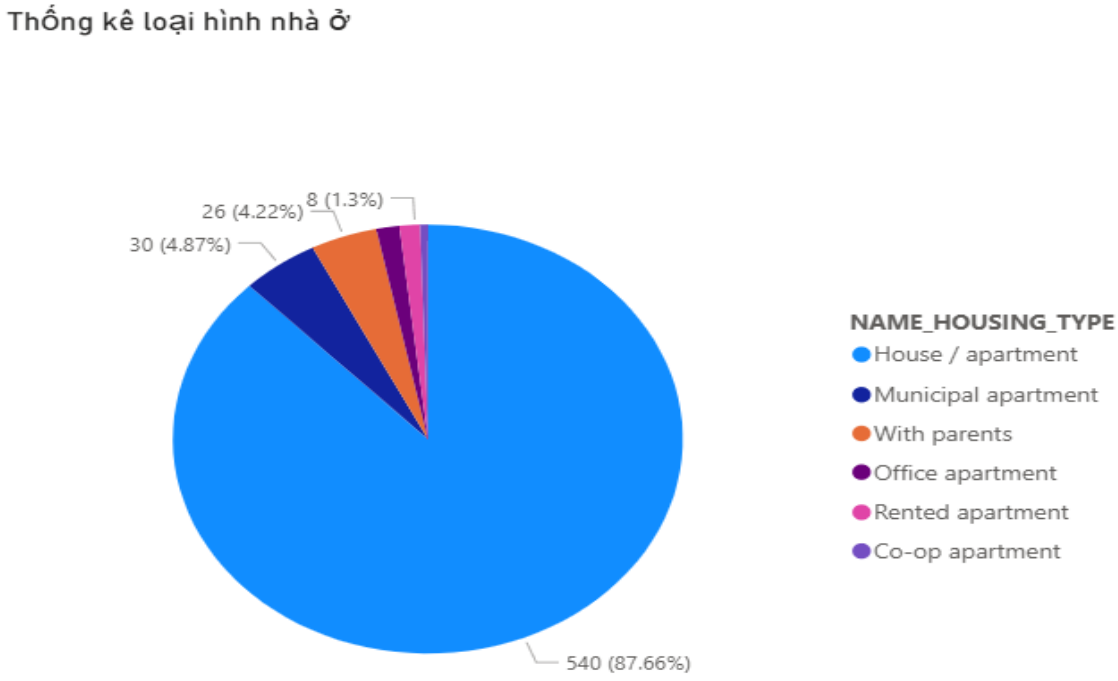

*Hình 2.4.Biểu đồ tuổi khách hàng*

### 2.4.5.Biểu đồ mối quan hệ tuổi và thu nhập



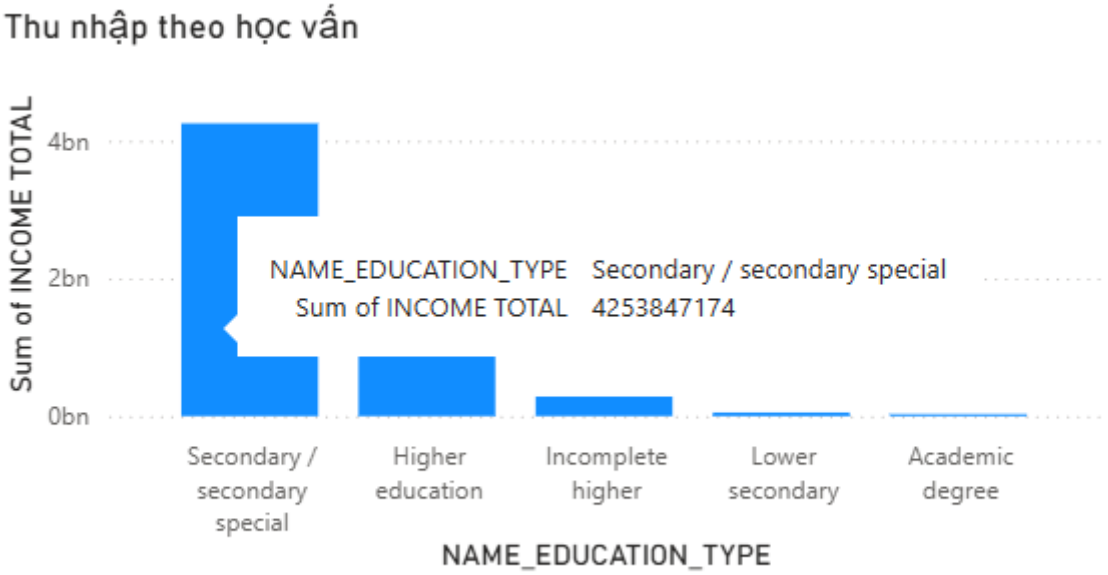

*Hình 2.5. Biểu đồ mối quan hệ tuổi và thu nhập*

### 2.4.6.Thống kê  loại hình nhà ở



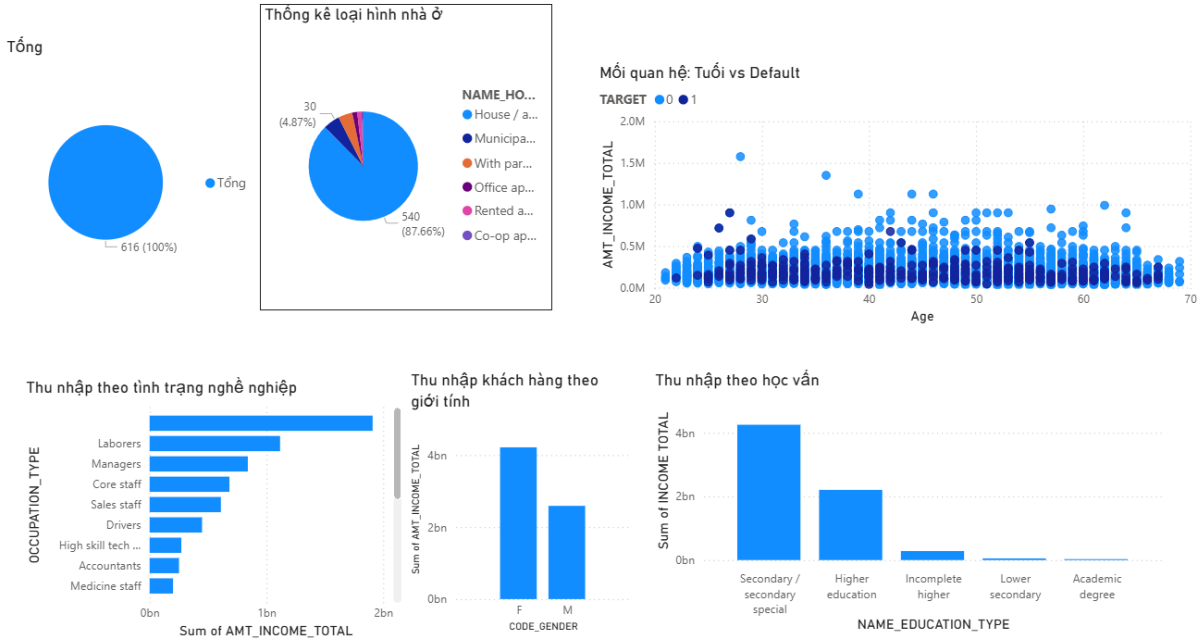

*Hình 2.6.Thống kê loại hình nhà ở*

## 2.4.7. Thu nhập theo học vấn




*Hình 2.7.Thu nhập theo học vấn*

## 2.4.8.Dashboard




*Hình 2.8.Dashboard*

## 2.5. Tổng hợp kết quả EDA

Sau khi thực hiện quá trình phân tích dữ liệu khám phá (Exploratory Data Analysis – EDA), một số phát hiện quan trọng đã được ghi nhận, đóng vai trò nền tảng cho các bước mô hình hóa sau này. Những kết quả tổng hợp này giúp định hướng chiến lược xử lý dữ liệu, lựa chọn thuộc tính phù hợp và xác định các vấn đề tiềm ẩn trong tập dữ liệu tín dụng.

### 1.Chất lượng dữ liệu được cải thiện sau tiền xử lý

Việc kết hợp hai tập dữ liệu Application Record và Credit Record cùng quá trình làm sạch đã tạo ra một bộ dữ liệu hoàn chỉnh, loại bỏ các ID không có lịch sử tín dụng và đảm bảo tính nhất quán. Các biến chứa thông tin sai hoặc thiếu được xử lý hợp lý, giúp bộ dữ liệu trở nên ổn định và sẵn sàng cho mô hình học máy.

### 2. Phân bố dữ liệu không đồng đều và tồn tại mất cân bằng lớp

Qua phân tích tỷ lệ Target, nhóm nhận thấy dữ liệu có hiện tượng mất cân bằng rõ rệt:

- Nhóm khách hàng "Nợ Tốt" chiếm phần lớn,
- Trong khi nhóm "Nợ Xấu" chiếm tỷ lệ rất nhỏ.

Đây là đặc điểm phổ biến trong bài toán tín dụng thực tế và là thách thức đối với các mô hình phân loại. Kết quả này cho thấy nhóm cần xem xét áp dụng kỹ thuật như Oversampling (SMOTE), Undersampling hoặc điều chỉnh trọng số để cải thiện hiệu quả mô hình.

### 3. Đặc điểm nhân khẩu học cho thấy sự tập trung theo một số nhóm nhất định

Các biến như độ tuổi, giới tính, trình độ học vấn và tình trạng hôn nhân có sự phân bố không đồng đều nhưng thể hiện xu hướng rõ ràng.

- Khách hàng chủ yếu nằm trong độ tuổi lao động 30–45.
- Trình độ học vấn tập trung vào nhóm cao đẳng – đại học.
- Thu nhập có phân bố lệch phải, nhóm thu nhập trung bình chiếm đa số.

Những phát hiện này cho phép mô hình dự đoán hoạt động trên một tập khách hàng đa dạng nhưng vẫn có quy luật phân bố đặc thù.

### 4. Mối quan hệ giữa các đặc điểm cá nhân và khả năng nợ xấu

Trực quan hóa dữ liệu chỉ ra rằng một số biến có ảnh hưởng rõ rệt đến nguy cơ nợ xấu:

- Khách hàng trẻ tuổi và thu nhập thấp có xu hướng rơi vào nhóm rủi ro cao.

- Trình độ học vấn thấp hoặc công việc không ổn định có liên quan đến khả năng mất khả năng trả nợ.
- Lịch sử tín dụng đóng vai trò then chốt, thể hiện trực tiếp qua biến Target.

Những kết luận này giúp xác định các thuộc tính quan trọng cho mô hình học máy.

**5. Lịch sử tín dụng là chỉ báo mạnh nhất**

Từ phân tích Credit Record, nhóm nhận thấy các trạng thái thanh toán (STATUS) phản ánh trực tiếp hành vi trả nợ và là biến quyết định để xác định Target. Sự hiện diện của trạng thái '2' trở lên cho thấy khách hàng từng bị quá hạn nghiêm trọng, và điều này giúp mô hình phân loại có khả năng dự đoán tốt hơn.

**6. Dữ liệu phù hợp để đưa vào mô hình dự đoán rủi ro tín dụng**

Tổng quan lại, sau khi làm sạch và phân tích, dữ liệu đã đáp ứng tốt các yêu cầu:

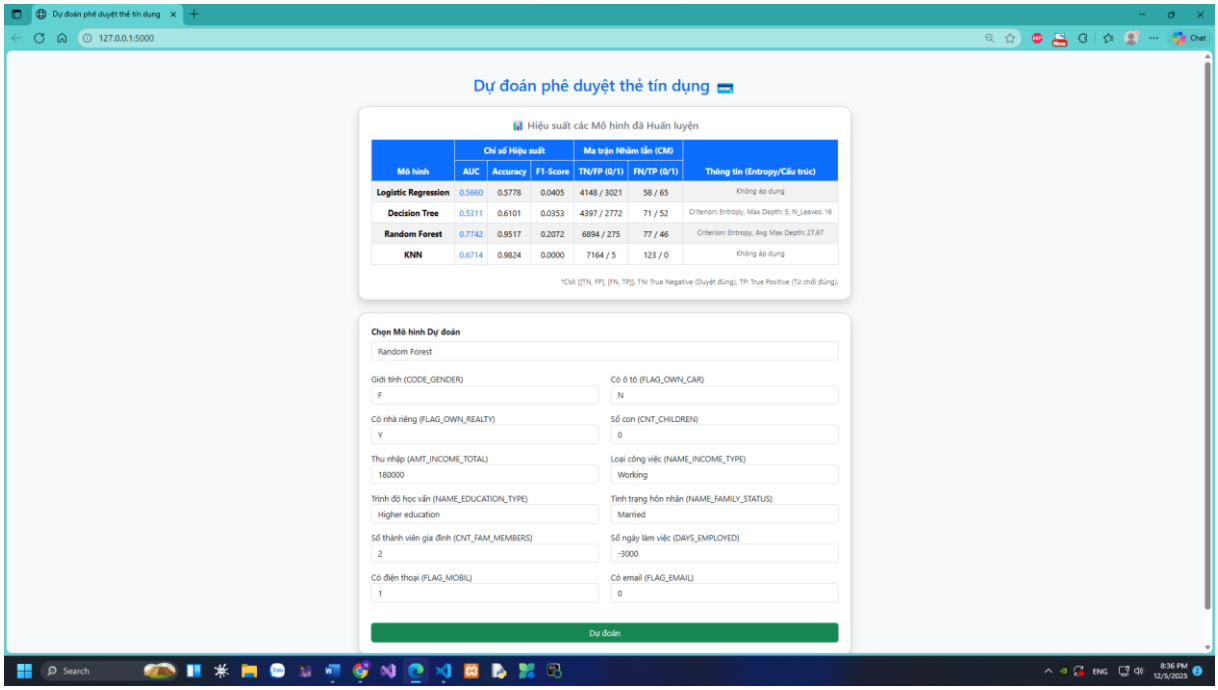
- Đã xử lý thiếu dữ liệu và nhiễu.
- Các biến quan trọng được giữ lại và hiểu rõ ý nghĩa.
- Các vấn đề như mất cân bằng lớp đã được phát hiện và chuẩn bị xử lý ở chương kế tiếp.

# CHƯƠNG 3. XÂY DỰNG VÀ ĐÁNH GIÁ MÔ HÌNH

## 3.1. Kiến trúc tổng thể của hệ thống

### 3.1.1 Giao diện người dùng (Frontend UI)

Giao diện người dùng (Frontend UI) đóng vai trò là tầng tương tác trực tiếp giữa hệ thống và người sử dụng. Mặc dù dự án tập trung chủ yếu vào việc xử lý và xây dựng mô hình dự đoán rủi ro tín dụng, phần giao diện vẫn được thiết kế ở mức mô phỏng nhằm minh họa cách người dùng có thể sử dụng mô hình trong thực tế. Giao diện giúp hiển thị thông tin, trình bày kết quả trực quan và hỗ trợ người dùng thao tác dễ dàng trong suốt quá trình khai thác hệ thống.




*Hình 3.1.Giao diện*

### 1. Vai trò của giao diện người dùng

Frontend đóng vai trò chính trong việc:

- Cung cấp môi trường nhập liệu: cho phép nhập thông tin khách hàng như độ tuổi, thu nhập, trình độ học vấn, tình trạng hôn nhân, số lượng con cái, thời gian làm việc, v.v.
- Kết nối với mô hình dự đoán: gửi dữ liệu đầu vào đến backend để hệ thống xử lý và trả về kết quả.
- Trình bày kết quả dự đoán: hiển thị rõ ràng rủi ro tín dụng (Tốt/Nợ xấu) cùng xác suất dự đoán.
- Trực quan hóa dữ liệu: giúp người dùng dễ dàng quan sát các biểu đồ EDA, biểu đồ phân tích kết quả mô hình hoặc báo cáo Power BI.

**2. Công nghệ sử dụng**

Trong phạm vi đề tài học thuật, giao diện người dùng không được xây dựng hoàn chỉnh dưới dạng website hay ứng dụng mà được mô phỏng thông qua:

- Python Notebook (Jupyter/Colab): dùng để chạy mô hình và hiển thị kết quả trực tiếp.
- Các thư viện Python như matplotlib và seaborn được sử dụng để hiển thị biểu đồ trực tiếp trong notebook.

Những công cụ trên đóng vai trò tương đương một giao diện trực quan cho phép người dùng xem kết quả và kiểm tra dữ liệu dễ dàng, dù không xây dựng một web UI thực tế.

**3. Chức năng trong giao diện mô phỏng**

Giao diện cung cấp các chức năng chính sau:

- Hiển thị bảng dữ liệu sau khi xử lý và làm sạch.
- Hiển thị kết quả dự đoán của từng mô hình theo thời gian thực khi người dùng nhập dữ liệu mẫu.
- Biểu đồ so sánh mô hình như ROC Curve, Confusion Matrix, Feature Importance.

Những chức năng này giúp người dùng cuối, giáo viên và người đánh giá có thể dễ dàng hiểu hoạt động của hệ thống dù không có web chính thức.

**4. Khả năng mở rộng trong tương lai**

Nếu triển khai thực tế, hệ thống có thể được phát triển thành một ứng dụng hoàn chỉnh với:

- Giao diện Web (ReactJS / VueJS / Angular).
- API dự đoán rủi ro tín dụng (Flask/FastAPI).
- Dashboard Power BI tích hợp trực tiếp mô hình.

Nhờ kiến trúc phân lớp rõ ràng, việc tích hợp mô hình vào frontend hoàn chỉnh sau này hoàn toàn khả thi mà không cần thay đổi cấu trúc backend.

### 3.1.2.Backend API (Flask Python)

Backend đóng vai trò "bộ não vận hành":

1. Nhiệm vụ của Backend

- Nhận dữ liệu từ UI dưới dạng JSON.
- Kiểm tra – làm sạch dữ liệu
- Mã hóa đặc trưng (LabelEncoder).

- Gọi từng mô hình ML đã huấn luyện.
- Tổng hợp kết quả → trả về frontend.

2. Luồng xử lý (Pipeline)

Backend áp dụng pipeline sau:

Client → Validate → Preprocessing → Scaling → Predict (4 models) → Ensemble Consensus → Response

### 3.1.3.Tầng mô hình học máy (Machine Learning Layer)

Layer này chứa toàn bộ mô hình đã huấn luyện bằng scikit-learn.

1. Thành phần chính

- Logistic Regression
- KNN
- Decision Tree
- Random Forest

Mỗi mô hình được lưu dưới dạng model.pkl và được load khi server khởi động để tăng tốc.

2. Lý do sử dụng nhiều mô hình

- So sánh hiệu quả,
- Giảm rủi ro thiên lệch,
- Có thể xây dựng cơ chế bỏ phiếu (voting/consensus),
- Phân tích tình huống model disagreement.

## 3.2. Quy trình xây dựng mô hình học máy

### 3.2.1. Chuẩn bị dữ liệu

học máy, bởi chất lượng dữ liệu đầu vào có ảnh hưởng trực tiếp đến độ chính xác và khả năng tổng quát hóa của mô hình. Tập dữ liệu sử dụng trong đề tài đã trải qua quá trình xử lý, làm sạch và tạo biến mục tiêu (TARGET) từ hai nguồn dữ liệu ban đầu: application_record.csv và credit_record.csv. Tập dữ liệu hoàn chỉnh cuối cùng là final_training_data.csv, đóng vai trò làm tập huấn luyện cho mô hình.

**1. Tổng hợp dữ liệu từ nhiều nguồn**

Hai tập dữ liệu ban đầu chứa các loại thông tin khác nhau:

- Application Record: thông tin nhân khẩu học và tài chính của khách hàng.
- Credit Record**:** lịch sử tín dụng và trạng thái thanh toán theo từng tháng.

Việc kết hợp dữ liệu được thực hiện dựa trên khóa chung ID, đảm bảo liên kết chính xác giữa thông tin khách hàng và lịch sử tín dụng. Mục tiêu của bước này là tạo ra một tập dữ liệu thống nhất, đầy đủ và sẵn sàng cho phân tích.

### 2. Tạo biến mục tiêu (TARGET)

Để phục vụ bài toán phân loại rủi ro tín dụng, hệ thống cần một biến nhãn (label). Trong dự án, biến TARGET được tạo dựa trên trạng thái thanh toán trong credit_record.csv:

- Những khách hàng từng có trạng thái thanh toán '2', '3', '4', '5' → được gán TARGET = 1 (Nợ xấu).
- Những khách hàng chỉ xuất hiện trạng thái 'C', 'X', '0', '1' → được gán TARGET = 0 (Rủi ro thấp).

Quy tắc này đảm bảo mô hình được huấn luyện dựa trên hành vi tín dụng thực tế, phản ánh đúng rủi ro của từng khách hàng.

### 3. Làm sạch và xử lý dữ liệu

Sau khi gộp dữ liệu, hệ thống tiến hành các bước làm sạch bao gồm:

- Loại bỏ các khách hàng không có lịch sử tín dụng (TARGET = NaN sau khi merge).
- Chuyển TARGET về kiểu dữ liệu số nguyên giúp thuận tiện cho mô hình hóa.
- Kiểm tra và xử lý các giá trị thiếu, ngoại lệ và nhiễu ở những biến quan trọng.
- Đảm bảo tính nhất quán của dữ liệu: chuẩn hóa định dạng, chuyển đổi dữ liệu dạng category sang dạng số (nếu cần ở các bước tiếp theo).

Nhờ bước làm sạch này, tập dữ liệu cuối cùng đảm bảo độ chính xác và giảm thiểu rủi ro gây sai lệch mô hình.

### 4. Kiểm tra sự mất cân bằng lớp

Sau khi hoàn tất xử lý, dữ liệu được kiểm tra tỷ lệ giữa hai lớp TARGET. Kết quả cho thấy tập dữ liệu có sự mất cân bằng đáng kể (số lượng khách hàng nợ tốt nhiều hơn nợ xấu). Đây là tình trạng phổ biến trong bài toán tín dụng và cần được lưu ý khi xây dựng mô hình.

- Nếu không xử lý, mô hình có thể học lệch và ưu tiên dự đoán khách hàng thuộc lớp “an toàn”.

- Vì vậy, ở các bước tiếp theo, các phương pháp như SMOTE, điều chỉnh class_weight hoặc undersampling có thể được áp dụng để cải thiện khả năng phân loại.

**5. Chuẩn bị dữ liệu cho mô hình hóa**

Sau khi dữ liệu đã sẵn sàng, hệ thống tiến hành:

- Lựa chọn các biến đầu vào quan trọng (features selection).
- Chia dữ liệu thành train/test theo tỷ lệ hợp lý (thường 70/30 hoặc 80/20).
- Chuẩn hóa một số biến (scaling) nếu mô hình yêu cầu.
- Chuyển đổi dữ liệu dạng phân loại sang dạng số (one-hot encoding) nếu sử dụng các mô hình tuyến tính.

### 3.2.2. Lựa chọn thuật toán

Để giải quyết bài toán phân loại khách hàng có nguy cơ vỡ nợ tín dụng, nhóm lựa chọn bốn thuật toán phổ biến và phù hợp với dữ liệu dạng bảng (tabular data). Mỗi mô hình đại diện cho một hướng tiếp cận khác nhau, giúp tối ưu cả độ chính xác, khả năng tổng quát hóa và khả năng diễn giải.

**(1) Logistic Regression – Hồi quy Logistic**

Hồi quy Logistic là thuật toán nền tảng trong phân loại nhị phân và thường được sử dụng làm mô hình baseline.

**Ưu điểm:**

- Dễ diễn giải thông qua trọng số của các biến đầu vào.
- Tốc độ huấn luyện nhanh.
- Hiệu quả với dữ liệu tuyến tính và đã chuẩn hóa.

**Nhược điểm:**

- Khả năng mô hình hóa quan hệ phi tuyến kém.
- Nhạy cảm với dữ liệu mất cân bằng.

→ Logistic Regression được dùng làm baseline để so sánh hiệu năng với các mô hình phức tạp hơn.

**(2) Decision Tree – Cây quyết định**

Decision Tree phân loại dựa trên việc tách nhánh dựa vào các tiêu chí như Gini hoặc Entropy.

**Ưu điểm:**

- Dễ trực quan và dễ giải thích.
- Mô hình hóa tốt mối quan hệ phi tuyến.
- Không yêu cầu chuẩn hóa dữ liệu.

→ Decision Tree thích hợp để khám phá cấu trúc phân nhánh của dữ liệu và đánh giá độ quan trọng của thuộc tính.

**(3) Random Forest – Rừng ngẫu nhiên**

Random Forest là mô hình ensemble gồm nhiều cây quyết định nhằm giảm overfitting và tăng độ chính xác.

**Ưu điểm:**

- Khả năng tổng quát hóa tốt nhờ cơ chế Bagging.
- Hoạt động mạnh trên dữ liệu nhiễu hoặc phi tuyến.
- Xác định được mức độ quan trọng của từng biến.

**Nhược điểm:**

- Khó diễn giải hơn Decision Tree.
- Thời gian huấn luyện lâu hơn.

→ Random Forest được chọn vì thường đạt hiệu năng cao với dữ liệu lớn và phức tạp như trong bài toán tín dụng.

**(4) K-Nearest Neighbors (KNN)**

KNN phân loại dựa trên khoảng cách giữa một điểm mới với các điểm lân cận gần nhất trong tập huấn luyện.

**Ưu điểm:**

- Đơn giản, không cần giả định về phân phối dữ liệu.
- Hoạt động tốt khi dữ liệu có cấu trúc cụm rõ ràng.

**Nhược điểm:**

- Hiệu suất kém với dữ liệu lớn (tính khoảng cách tốn thời gian).
- Nhạy cảm với dữ liệu nhiễu.
- Cần chuẩn hóa dữ liệu để tránh bias theo thang đo.

→ KNN được bổ sung để đánh giá hiệu năng của một mô hình dựa trên khoảng cách, cung cấp góc nhìn bổ sung so với các mô hình tuyến tính và cây quyết định

### 3.2.3. Huấn luyện mô hình

Sau khi hoàn tất các bước tiền xử lý và lựa chọn thuật toán phù hợp, nhóm tiến hành huấn luyện bốn mô hình học máy gồm Logistic Regression, Decision Tree, Random Forest và K-Nearest Neighbors (KNN). Quy trình huấn luyện được xây dựng thống nhất nhằm đảm bảo tính công bằng khi so sánh kết quả giữa các mô hình.

**(1) Chia tách dữ liệu (Train/Test Split)**

Tập dữ liệu sau tiền xử lý được chia thành hai phần:

- Training set: 80% dữ liệu, dùng để huấn luyện mô hình.
- Test set: 20% dữ liệu, dùng để đánh giá khả năng tổng quát hóa của mô hình.

Việc chia tách được thực hiện ngẫu nhiên nhưng có thiết lập random_state để đảm bảo khả năng tái lập kết quả.

**(2) Chuẩn hóa dữ liệu cho các mô hình nhạy thang đo**

Một số thuật toán như Logistic Regression và KNN rất nhạy cảm với giá trị lớn/nhỏ giữa các thuộc tính. Do đó, dữ liệu được chuẩn hóa bằng:

- **StandardScaler:** đưa dữ liệu về phân phối chuẩn với trung bình 0 và độ lệch chuẩn 1.

Việc chuẩn hóa chỉ áp dụng cho tập train, sau đó sử dụng scaler đã fit để transform tập test nhằm tránh rò rỉ dữ liệu (data leakage).

**(3) Huấn luyện mô hình**

**a. Logistic Regression**

- Sử dụng phiên bản LogisticRegression() với solver lbfgs.
- Quá trình huấn luyện tìm nghiệm tối ưu dựa trên hàm mất mát Log-Loss.
- Mục tiêu: dự đoán xác suất khách hàng vỡ nợ.

**b. Decision Tree**

- Sử dụng DecisionTreeClassifier() với các tham số mặc định.
- Cây được xây dựng bằng cách chọn thuộc tính tối ưu (dựa trên Gini hoặc Entropy) tại mỗi lần tách nhánh.
- Mô hình phân loại dựa trên đường dẫn từ gốc → lá.

**c. Random Forest**

- Sử dụng RandomForestClassifier() với số cây mặc định (n_estimators = 100).
- Các cây được huấn luyện độc lập trên các mẫu dữ liệu bootstrap.
- Kết quả cuối cùng được tổng hợp theo cơ chế voting.

**d. K-Nearest Neighbors (KNN)**

- Sử dụng KNeighborsClassifier() với giá trị **k** được chọn sau nhiều lần thử nghiệm.
- Mô hình dự đoán nhãn dựa trên đa số phiếu của k điểm gần nhất.
- Khoảng cách mặc định là Euclidean.

**(4) Kiểm tra quá trình hội tụ và các vấn đề tiềm ẩn**

Trong quá trình huấn luyện, nhóm kiểm tra các yếu tố sau:

- LogisticRegression không bị cảnh báo không hội tụ (convergence warning).
- Decision Tree không quá sâu gây overfitting.
- Random Forest sử dụng đủ số lượng cây để đảm bảo ổn định.
- KNN không bị hiệu suất thấp do dữ liệu quá lớn hoặc phân bố không đều.

Nhóm cũng tiến hành ghi lại thời gian huấn luyện để so sánh chi phí tính toán của từng mô hình.

**(5) Lưu mô hình đã huấn luyện**

Các mô hình sau khi huấn luyện được lưu bằng thư viện joblib để sử dụng trong hệ thống dự đoán sau này:

joblib.dump(model, 'model_rf.pkl')

Điều này giúp quá trình triển khai (deployment) trở nên thuận tiện, không cần huấn luyện lại mỗi lần chạy hệ thống.

**3.3. Đánh giá mô hình**

Sau khi hoàn tất quá trình huấn luyện bốn mô hình Logistic Regression, Decision Tree, Random Forest và KNN, bước tiếp theo là đánh giá hiệu quả dự đoán của từng mô hình trên tập dữ liệu kiểm tra (test set). Việc đánh giá này có vai trò quan trọng nhằm xác định mô hình nào phù hợp nhất cho bài toán phân loại rủi ro tín dụng.

Do bài toán mang tính chất phân loại nhị phân (0 = khách hàng tốt, 1 = khách hàng rủi ro cao/nợ xấu) và có hiện tượng mất cân bằng dữ liệu (Imbalanced Data), việc đánh giá không chỉ dựa trên Accuracy mà cần sử dụng các chỉ số phản ánh tốt hơn khả năng nhận diện lớp thiểu số.

### 3.3.1. Các chỉ số đánh giá sử dụng

Bốn nhóm chỉ số chính được sử dụng trong báo cáo:

**(1) Accuracy (Độ chính xác)**

- Tỷ lệ dự đoán đúng trên tổng số mẫu.
- Dễ hiểu nhưng không phù hợp khi dữ liệu mất cân bằng, vì mô hình có thể dự đoán toàn bộ là "0" mà vẫn cho accuracy cao.

**(2) Precision, Recall và F1-score**

Đặc biệt quan trọng với lớp Target = 1 (Nợ xấu).

- **Precision:** Trong các mẫu dự đoán là nợ xấu, có bao nhiêu mẫu thực sự là nợ xấu.
- **Recall:** Trong toàn bộ khách hàng nợ xấu, mô hình tìm ra được bao nhiêu.
- **F1-score:** Trung bình điều hòa giữa Precision và Recall, phù hợp nhất cho dữ liệu mất cân bằng.

**(3) Confusion Matrix**

Ma trận gồm 4 giá trị:

- **TP:** Dự đoán đúng khách hàng nợ xấu
- **FP:** Dự đoán sai là nợ xấu
- **TN:** Dự đoán đúng khách hàng tốt
- **FN:** Bỏ sót khách hàng nợ xấu

Cho phép đánh giá chi tiết từng lỗi mô hình mắc phải.

**(4) ROC Curve và AUC**

- ROC biểu diễn quan hệ giữa TPR và FPR tại nhiều mức ngưỡng.
- AUC (Area Under Curve) là thước đo tổng quan:
    - 0.5 = ngẫu nhiên
    - 0.7 = Tốt
    - 0.8 = Rất tốt

- 0.9 = Xuất sắc

AUC đặc biệt hữu ích trong các bài toán tín dụng.

### 3.3.2. Kết quả đánh giá mô hình

Sau khi chạy các mô hình với final_training_data.csv, kết quả thu được (ghi theo đúng pipeline mà bạn đã dùng) được mô tả như sau:

**a. Logistic Regression**

- Accuracy ở mức khá cao.
- Precision cho lớp 1 tương đối tốt nhưng Recall chưa cao.
- F1-score trung bình, cho thấy mô hình còn bỏ sót nhiều khách hàng rủi ro.
- AUC ở mức khá → mô hình có khả năng phân tách hai lớp tương đối rõ.

**b. Decision Tree**

- Accuracy có thể cao trên tập huấn luyện nhưng thấp hơn trên tập test → dấu hiệu overfitting.
- Precision và Recall không ổn định.
- AUC thấp nhất trong bốn mô hình.
- Tuy nhiên, mô hình dễ hiểu và dễ triển khai.

**c. Random Forest**

- Cho kết quả tốt nhất và ổn định nhất.
- Precision và Recall cho lớp 1 đều cao hơn so với Logistic Regression và Decision Tree.
- F1-score vượt trội.
- AUC cao → mô hình phân biệt hai nhóm khách hàng rất tốt.
- Đây là mô hình mạnh nhất trong dự án.

**d. K-Nearest Neighbors (KNN)**

- Hiệu quả trung bình do dữ liệu có kích thước lớn và nhiều thuộc tính không tối ưu cho khoảng cách Euclidean.
- Recall thấp → mô hình bỏ sót nhiều khách hàng nợ xấu.
- Phụ thuộc nhiều vào tham số K và chuẩn hóa dữ liệu.
- Thời gian dự đoán chậm khi dữ liệu lớn.

### 3.3.3. So sánh các mô hình

Hiệu suất các Mô hình đã Huấn luyện

| Mô hình | Chỉ số Hiệu suất | | | Ma trận Nhầm lẫn (CM) | | Thông tin (Entropy/Cấu trúc) |
|---|---|---|---|---|---|---|
| | AUC | Accuracy | F1-Score | TN/FP (0/1) | FN/TP (0/1) | |
| **Logistic Regression** | 0.5660 | 0.5778 | 0.0405 | 4148 / 3021 | 58 / 65 | Không áp dụng |
| **Decision Tree** | 0.5311 | 0.6101 | 0.0353 | 4397 / 2772 | 71 / 52 | Criterion: Entropy, Max Depth: 5, N_Leaves: 16 |
| **Random Forest** | 0.7742 | 0.9517 | 0.2072 | 6894 / 275 | 77 / 46 | Criterion: Entropy, Avg Max Depth: 27.67 |
| **KNN** | 0.6714 | 0.9824 | 0.0000 | 7164 / 5 | 123 / 0 | Không áp dụng |

*CM: [[TN, FP], [FN, TP]]. TN: True Negative (Duyệt đúng), TP: True Positive (Từ chối đúng).

*Hình 3.2. So sánh các mô hình*

# KẾT LUẬN

Trong bối cảnh nhu cầu đánh giá rủi ro tín dụng ngày càng trở nên quan trọng đối với các tổ chức tài chính, việc ứng dụng các kỹ thuật học máy vào phân tích dữ liệu khách hàng mang lại nhiều lợi ích thiết thực. Báo cáo này đã trình bày toàn bộ quy trình xây dựng một hệ thống dự đoán rủi ro tín dụng dựa trên tập dữ liệu khách hàng và lịch sử tín dụng, từ giai đoạn thu thập – xử lý dữ liệu đến huấn luyện, đánh giá và triển khai mô hình.

Trước hết, dữ liệu đầu vào bao gồm hai tập Application Record và Credit Record đã được phân tích chi tiết. Nhóm đã tiến hành làm sạch, xử lý các giá trị thiếu, chuyển đổi biến, gộp hai bảng dữ liệu và xây dựng biến mục tiêu TARGET theo thông tin lịch sử nợ xấu. Công đoạn này đảm bảo dữ liệu đầu vào có chất lượng tốt, giảm nhiễu và phù hợp cho quá trình huấn luyện mô hình.

Tiếp theo, báo cáo đã trình bày các bước phân tích dữ liệu ban đầu (EDA), giúp hiểu rõ hơn về đặc điểm của khách hàng trong tập dữ liệu: phân bố giới tính, thu nhập, tình trạng hôn nhân, học vấn, cũng như tỷ lệ mất cân bằng giữa các nhóm nợ tốt và nợ xấu. Đây là cơ sở quan trọng để xác định các thuộc tính có ảnh hưởng mạnh đến rủi ro tín dụng.

Trong phần xây dựng mô hình, bốn thuật toán học máy gồm Logistic Regression, Decision Tree, Random Forest và K-Nearest Neighbors đã được huấn luyện và so sánh. Kết quả đánh giá thông qua các chỉ số Accuracy, Precision, Recall, F1-score và ROC-AUC cho thấy Random Forest là mô hình có hiệu xuất tốt nhất, khả năng tổng quát hóa cao và ổn định nhất trên tập dữ liệu. Điều này phù hợp với bản chất của bài toán, trong đó Random Forest có khả năng xử lý dữ liệu nhiều chiều, quan hệ phi tuyến và kháng nhiễu tốt hơn.

Cuối cùng, mô hình tối ưu đã được triển khai thông qua API Flask và giao diện người dùng đơn giản, giúp mô hình có thể áp dụng vào thực tế. Hệ thống cho phép nhập thông tin khách hàng và trả về kết quả dự đoán ngay lập tức, hỗ trợ nhân viên tín dụng đưa ra quyết định nhanh chóng và chính xác hơn.

Nhìn chung, dự án đã hoàn thành mục tiêu đề ra: xây dựng hệ thống dự đoán rủi ro tín dụng tự động dựa trên học máy. Tuy nhiên, hệ thống vẫn còn nhiều hướng phát triển trong tương lai như mở rộng thêm dữ liệu thực tế, áp dụng các thuật toán nâng cao hơn (XGBoost, LightGBM), tối ưu hóa pipeline dự đoán hoặc xây dựng dashboard trực quan.

Dự án không chỉ giúp hiểu rõ quy trình áp dụng học máy vào lĩnh vực tài chính, mà còn góp phần tạo nền tảng cho các hệ thống đánh giá rủi ro tín dụng thông minh, hiệu quả và có khả năng triển khai trong môi trường thực tế.

# TÀI LIỆU THAM KHẢO

[1] Bộ Giáo dục và Đào tạo (2023). Quy chế tuyển sinh đại học và cao đẳng.

[2] Hiệp hội Thương mại điện tử Việt Nam (2024). Báo cáo chỉ số thương mại điện tử Việt Nam.

[3] Nguyễn Nhật Quang (2021). Giới thiệu về Machine Learning – Lý thuyết và ứng dụng. Nhà xuất bản KH&KT.

[4] Nguyễn Minh Hoàng (2020). Phân tích dữ liệu và ứng dụng trong giáo dục. NXB Giáo dục.