**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**

****

**ĐỒ ÁN 3**

Loan Status Prediction

NGÀNH: KHOA HỌC MÁY TÍNH

SINH VIÊN: **HOÀNG QUANG HUY**

MÃ LỚP: **12421TN**

HƯỚNG DẪN: **TS. NGUYỄN VĂN QUYẾT**

**HƯNG YÊN – 2024**

**NHẬN XÉT**

**Nhận xét của giáo viên hướng dẫn**

**GIÁO VIÊN HƯỚNG DẪN**

**NGUYỄN VĂN QUYẾT**

**LỜI CAM ĐOAN**

Em xin cam đoan đồ án “Loan Status Prediction” là sản phẩm của bản thân. Những phần sử dụng tài liệu tham khảo trong bài tập lớn đã được nêu rõ trong phần tài liệu tham khảo. Các số liệu, kết quả trình bày trong bài tập lớn là hoàn toàn trung thực, nếu sai em xin chịu hoàn toàn trách nhiệm và chịu mọi kỷ luật của bộ môn và nhà trường đề ra.

*Hưng yên, ngày … tháng … năm 2024*

Sinh viên

**LỜI CẢM ƠN**

Em xin bày tỏ lòng biết ơn chân thành tới Thầy Nguyễn Văn Quyết, người đã tận tình hướng dẫn và hỗ trợ em trong suốt quá trình thực hiện đồ án này. Sự nhiệt huyết và tận tâm của Thầy đã mang lại cho em những kiến thức quý báu và giúp em vượt qua những khó khăn, thách thức trong quá trình nghiên cứu và hoàn thiện đồ án.

Những lời khuyên, sự chỉ dẫn cụ thể và chi tiết của Thầy đã giúp em mở rộng tầm nhìn, hiểu sâu hơn về lĩnh vực mà em đang nghiên cứu. Sự kiên nhẫn và sẵn sàng dành thời gian để giải đáp mọi thắc mắc của Thầy đã tiếp thêm động lực cho em, giúp em không ngừng cố gắng và hoàn thiện bản thân.

Mặc dù em đã cố gắng hết sức để hoàn thành đồ án này, nhưng với kinh nghiệm và trình độ còn hạn chế, chắc chắn sẽ không tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đóng góp, nhận xét từ Thầy để em có thể học hỏi và cải thiện trong tương lai. Em sẵn sàng tiếp thu mọi ý kiến phê bình với tinh thần cầu tiến, vì em hiểu rằng chỉ qua đó, em mới có thể tiếp tục phát triển và hoàn thiện bản thân hơn.

Một lần nữa, em xin chân thành cảm ơn Thầy Nguyễn Văn Quyết vì sự giúp đỡ quý báu và những đóng góp quan trọng mà Thầy đã dành cho em trong suốt quá trình thực hiện đồ án. Em kính chúc Thầy luôn mạnh khỏe, hạnh phúc và tiếp tục gặt hái nhiều thành công trong sự nghiệp giảng dạy và nghiên cứu.

Em xin trân thành cảm ơn!

Mục Lục

[CHƯƠNG 1: TÌM HIỂU VỀ ĐỀ TÀI 7](#_Toc169314749)

[1.1 Lý do chọn đề tài 7](#_Toc169314750)

[1.2 Mục tiêu của đề tài 7](#_Toc169314751)

[1.3 Giới hạn và phạm vi của đề tài 8](#_Toc169314752)

[1.3.1 Đối tượng nghiên cứu 8](#_Toc169314753)

[1.3.2 Phạm vi nghiên cứu 8](#_Toc169314754)

[1.4 Nội dung thực hiện 8](#_Toc169314755)

[1.5 Phương pháp tiếp cận 8](#_Toc169314756)

[CHƯƠNG 2: TÌM HIỂU CÁC KỸ THUẬT HỌC MÁY 9](#_Toc169314757)

[2.1 Khái niệm khai phá dữ liệu và học máy 9](#_Toc169314758)

[2.1.1 Khái niệm khai phá dữ liệu 9](#_Toc169314759)

[2.1.2 Khái niệm học máy 10](#_Toc169314760)

[2.1.3 Liên hệ giữa khai phá dữ liệu và học máy 11](#_Toc169314761)

[2.2 Các kỹ thuật khai phá dữ liệu 11](#_Toc169314762)

[2.2.1 Kỹ thuật tiền xử lý dữ liệu 11](#_Toc169314763)

[2.2.2 Phân tích thống kê dữ liệu 12](#_Toc169314764)

[2.3 Nghiên cứu một số kỹ thuật học máy được áp dụng trong bài toán 13](#_Toc169314765)

[2.3.1 Mô hình Random Forest 13](#_Toc169314766)

[2.3.2 Mô hình Decision Tree 15](#_Toc169314767)

[2.3.3 Mô hình Logistic Regresstion 18](#_Toc169314768)

[CHƯƠNG 3: XÂY DỰNG MÔ HÌNH HỌC MÁY 20](#_Toc169314769)

[3.1 Thu thập dữ liệu 20](#_Toc169314770)

[3.1.1 Thu thập dữ liệu 20](#_Toc169314771)

[3.2 Phân tích dữ liệu thu thập 20](#_Toc169314772)

[3.2.1 Mô tả dữ liệu 20](#_Toc169314773)

[3.2.2 Tiền xử lý dữ liệu 24](#_Toc169314774)

[3.3 Xây dựng mô hình học máy 28](#_Toc169314775)

[3.3.1 Quan sát dữ liệu 28](#_Toc169314776)

[3.3.2 Xây dựng mô hình 31](#_Toc169314777)

[3.4 Phân tích kết quả thực nghiệm 31](#_Toc169314778)

[3.4.1 Thước đo đánh giá 31](#_Toc169314779)

[3.4.2 Chuẩn bị thực nghiệm 32](#_Toc169314780)

[3.4.3 Tiến hành thực nghiệm 32](#_Toc169314781)

[3.4.4 Kết quả thực nghiệm 33](#_Toc169314782)

[3.4.5 Thực nghiệm mô hình tinh chỉnh tham số cho từng mô hình 36](#_Toc169314783)

[3.5 Kết chương 40](#_Toc169314784)

[CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG KHAI PHÁ DỮ LIỆU 41](#_Toc169314785)

[4.1 Triển khai các chức năng nghiệp vụ 41](#_Toc169314786)

[4.2 Kiểm thử đơn vị và triển khai hệ thống 42](#_Toc169314787)

[4.3 Kết chương 45](#_Toc169314788)

[KẾT LUẬN 46](#_Toc169314789)

[TÀI LIỆU THAM KHẢO 47](#_Toc169314790)

**DANH MỤC TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| **Từ viết tắt** | **Viết đầy đủ** |
| P2P | Peer-to-Peer |
| RF | Random Forest |
| DC | Decision Tree |
| LR | Logistic Regresstion |
| ROC | Receiver Operating Characteristic |

**DANH MỤC HÌNH ẢNH**

[Hình 2. 1 Sơ đồ tổng quan về RandomForest 12](#_Toc169341390)

[Hình 2. 2: Công thức tính Entropy 14](#_Toc169341391)

[Hình 2. 3: Độ tinh khiết của Entropy 15](#_Toc169341392)

[Hình 2. 4: Công thức Entropy 15](#_Toc169341393)

[Hình 2. 5: bảng dữ liệu ví dụ decision tree 16](#_Toc169341394)

[Hình 2. 6: Cách tính Entropy trong decision tree 16](#_Toc169341395)

[Hình 2. 7: Hàm chi phí trong hồi quy logistic 17](#_Toc169341396)

[Hình 3. 1: Dữ liệu ban đầu 19](#_Toc169341274)

[Hình 3. 2: Tổng quan về dữ liệu 21](#_Toc169341275)

[Hình 3. 3:Thông tin tập dữ liệu 22](#_Toc169341276)

[Hình 3. 4: Bản tóm tắt thống kê 22](#_Toc169341277)

[Hình 3. 5: Các trạng thái đầy đủ của khoản vay 23](#_Toc169341278)

[Hình 3. 6: Kiểm tra kích thước của cột trạng thái khoản vay 24](#_Toc169341279)

[Hình 3. 7: Biểu đồ phân bố tỉ lệ 25](#_Toc169341280)

[Hình 3. 8 Kiểm tra có giá trị nào null hay không 25](#_Toc169341281)

[Hình 3. 9: Điền đầy giá trị còn thiếu 26](#_Toc169341282)

[Hình 3. 10: Chuẩn hóa cột term 26](#_Toc169341283)

[Hình 3. 11: Encoding 26](#_Toc169341284)

[Hình 3. 12: One hot encoding 27](#_Toc169341285)

[Hình 3. 13: Dữ liệu sao Encode 27](#_Toc169341286)

[Hình 3. 14: Tỉ lệ Fully Paid Và Charge Off trong cột term 27](#_Toc169341287)

[Hình 3. 15: Tỉ lệ Fully Paid và Charge Off trong cột verification\_status 28](#_Toc169341288)

[Hình 3. 16: Tỉ lệ Fully Paid Biểu diễn theo Employee 28](#_Toc169341289)

[Hình 3. 17: Tỉ lệ Fully Paid phân bố theo PurPose 29](#_Toc169341290)

[Hình 3. 18 29](#_Toc169341291)

[Hình 3. 19: Tỉ lệ Fully Paid Phân phối theo Home Ownership 29](#_Toc169341292)

[Hình 3. 20: Biểu đồ heatmap 30](#_Toc169341293)

[Hình 3. 21: Biều đồ Histogram cho từng biến số 30](#_Toc169341294)

[Hình 3. 22: Chương trình chia tập dữ liệu thành hai tập huấn luyện và kiểm tra 32](#_Toc169341295)

[Hình 3. 23: ROC của Random Forest trước tinh chỉnh 36](#_Toc169341296)

[Hình 3. 24: ROC Random Forest sau khi tinh chỉnh 37](#_Toc169341297)

[Hình 3. 25: ROC Decision Tree trước tinh chỉnh 37](#_Toc169341298)

[Hình 3. 26: ROC Decision Tree sau khi tinh chỉnh 38](#_Toc169341299)

[Hình 3. 27: ROC Linear trước tinh chỉnh 38](#_Toc169341300)

[Hình 3. 28: ROC mô hình Linear sau khi tinh chỉnh 39](#_Toc169341301)

[Hình 4. 1: Trang chủ 43](#_Toc169341355)

[Hình 4. 2: Trang chi tiết người đi vay 44](#_Toc169341356)

[Hình 4. 3: Trang chi tiết trạng tháy khoản vay 45](#_Toc169341357)

**DANH MỤC BẢNG BIỂU**

[Bảng 3. 1: Mô tả các trường dữ liệu 20](#_Toc169343042)

[Bảng 3. 2: Cột term 27](#_Toc169343043)

[Bảng 3. 3: Kết quả mô hình RF trước khi chuẩn hóa 33](#_Toc169343044)

[Bảng 3. 4: -Kết quả của mô hình sau khi đã được chuẩn hóa 33](#_Toc169343045)

[Bảng 3. 5: Kết quả thực nghiệm mô hình RF khi xử lý bằng kỹ thuật Undersampling: 33](#_Toc169343046)

[Bảng 3. 6: Kết quả thực nghiệm mô hình RF khi xử lý bằng kỹ thuật Oversampling 34](#_Toc169343047)

[Bảng 3. 7: Kết quả thực nghiệm mô hình RF khi xử lý bằng kỹ thuật SMOTETomek 34](#_Toc169343048)

[Bảng 3. 8: Ma trận nhầm lẫn 34](#_Toc169343049)

[Bảng 3. 9: Mô hình DC 35](#_Toc169343050)

[Bảng 3. 10: Mô hình LR 35](#_Toc169343051)

[Bảng 3. 11: Mô hình RF 35](#_Toc169343052)

[Bảng 3. 12: Ma trận nhầm lẫn 35](#_Toc169343053)

[Bảng 3. 13: Mô hình DC tinh chỉnh 36](#_Toc169343054)

[Bảng 3. 14: Mô hình LR tinh chỉnh 36](#_Toc169343055)

[Bảng 3. 15: Mô hình RF tinh chỉnh 36](#_Toc169343056)

[Bảng 3. 16: Ma trận nhầm lẫn 36](#_Toc169343057)

# TÌM HIỂU VỀ ĐỀ TÀI

## Lý do chọn đề tài

Đề tài "Dự đoán trạng thái khoản vay trên Lending Club được chọn vì nhiều lý do hấp dẫn cả về mặt học thuật lẫn ứng dụng thực tiễn. Trước hết, Lending Club là một trong những nền tảng cho vay ngang hàng lớn nhất tại Hoa Kỳ, nơi người vay và nhà đầu tư có thể kết nối trực tiếp với nhau mà không cần thông qua các tổ chức tài chính truyền thống. Khả năng dự đoán chính xác trạng thái khoản vay mang lại lợi ích to lớn cho các nhà đầu tư bằng cách giúp họ giảm rủi ro và tối ưu hóa lợi nhuận. Việc xác định liệu một khoản vay có được hoàn trả đầy đủ hay bị ghi giảm nợ là rất quan trọng trong việc ra quyết định đầu tư, đồng thời giúp các nhà quản lý quỹ cải thiện các mô hình quản lý rủi ro của mình.

Dữ liệu từ Lending Club rất phong phú và có sẵn công khai, bao gồm thông tin chi tiết về người vay, khoản vay, lịch sử thanh toán và nhiều yếu tố khác. Điều này cung cấp một cơ sở dữ liệu phong phú và đa dạng để áp dụng các kỹ thuật học máy và phân tích dữ liệu, giúp các nhà nghiên cứu có thể khám phá và ứng dụng các phương pháp tiên tiến nhất để giải quyết vấn đề. Dự đoán trạng thái khoản vay là một bài toán phức tạp, yêu cầu phải hiểu rõ về dữ liệu và sử dụng thành thạo các kỹ thuật xử lý dữ liệu, từ tiền xử lý, chuyển đổi biến số, đến xây dựng và đánh giá mô hình. Đây là một cơ hội tuyệt vời để phát triển các kỹ năng quan trọng trong phân tích dữ liệu và học máy, những kỹ năng này có giá trị cao trong nhiều lĩnh vực khác nhau, từ tài chính đến công nghệ thông tin.

Bên cạnh đó, nghiên cứu này cũng có tiềm năng đóng góp lớn cho cộng đồng. Kết quả nghiên cứu không chỉ giúp các nhà đầu tư có cái nhìn rõ ràng hơn về rủi ro và lợi nhuận từ các khoản vay P2P mà còn cung cấp thông tin hữu ích cho các nhà quản lý nền tảng P2P để cải thiện dịch vụ và hệ thống đánh giá rủi ro của họ. Hơn nữa, việc nghiên cứu các yếu tố ảnh hưởng đến khả năng hoàn trả nợ của người vay cũng mở ra những hướng nghiên cứu mới, giúp hiểu rõ hơn về hành vi tài chính và cách quản lý rủi ro hiệu quả.

Cuối cùng, đề tài này còn mang lại thách thức và cơ hội học hỏi lớn cho người nghiên cứu. Việc áp dụng các mô hình học máy tiên tiến như decision trees, random forests để dự đoán trạng thái khoản vay không chỉ đòi hỏi sự hiểu biết sâu rộng về lý thuyết mà còn cần kỹ năng thực hành tốt trong việc triển khai và tối ưu hóa mô hình. Thêm vào đó, nghiên cứu này còn áp dụng các mô hình phức tạp hơn như học sâu (deep learning) và kết hợp các mô hình khác nhau để cải thiện độ chính xác.

Việc chọn đề tài "Dự đoán trạng thái khoản vay trên Lending Club " không chỉ mang lại giá trị thực tiễn cao mà còn giúp phát triển các kỹ năng quan trọng trong lĩnh vực phân tích dữ liệu và học máy. Đây là một đề tài thú vị và đầy thách thức, với tiềm năng đóng góp lớn cho cộng đồng và mở ra nhiều hướng nghiên cứu và phát triển cho các đồ án tiếp theo.

## Mục tiêu của đề tài

- Nghiên cứu và cài đặt một số mô hình phân lớp như Random Forest, Decision Tree, Logistic Regresstion.

- Thu thập và xử lý dữ liệu ở trên Kaggle

- Cài đặt một số mô hình Học Máy cho bài toán dự đoán trạng thái khoản vay và đánh giá hiệu quả của các mô hình này.

-Xây dựng được một chương trình cho phép dự đoán trạng thái khoản vay đối với người vay và người cho vay.

## Giới hạn và phạm vi của đề tài

### Đối tượng nghiên cứu

- Ứng dụng một số mô hình Học Máy hỗ trợ dự đoán trạng thái khoản vay.

### Phạm vi nghiên cứu

- 3 mô hình Học Máy: Random Forest, Decision Tree, Logistic Regresstion.

## Nội dung thực hiện

Đề tài gồm 4 chương:

- *Chương 1: Tìm hiểu về đề tài*

+ Lý do chọn đề tài

+ Mục tiêu của đề tài

+ Giới hạn và phạm vi của đề tài

+ Nội dung thực hiện

+ Phương pháp tiếp cận

- *Chương 2: Tìm hiểu các kỹ thuật học máy*

+ Nghiên cứu một số kỹ thuật học máy được áp dụng trong bài toán

- *Chương 3: Xây dựng mô hình học máy*

+ Thu thập dữ liệu

+ Phân tích dữ liệu

+ Xây dựng mô hình học máy

+ Phân tích kết quả thực nghiệm

- *Chương 4: Xây dựng ứng dụng khai phá dữ liệu*

*+* Xây dựng hệ thống để ứng dụng mô hình học máy đã xây dựng được vào thử

nghiệm và kiểm thử hệ thống.

## Phương pháp tiếp cận

- Tham khảo và lấy dữ liệu trên Kaggle, Tham khảo thông tin trên các trang internet.

# TÌM HIỂU CÁC KỸ THUẬT HỌC MÁY

## Khái niệm khai phá dữ liệu và học máy

### Khái niệm khai phá dữ liệu

- Khai phá dữ liệu là quá trình tìm kiếm thông tin cụ thể, tiềm ẩn và có giá trị từ một tập hợp dữ liệu lớn và phức tạp. Quá trình này bao gồm việc áp dụng các phương pháp và công cụ phân tích dữ liệu để tìm ra các mẫu, quy tắc, thông tin ẩn, hoặc kiến thức mới từ dữ liệu không được biết trước. Khai phá dữ liệu giúp hiểu rõ hơn về dữ liệu, tạo ra thông tin hữu ích và hỗ trợ quyết định.

Dưới đây là một số ứng dụng thường thấy của khai phá dữ liệu:

1. **Dự đoán và phân loại**: Sử dụng khai phá dữ liệu để dự đoán các sự kiện tương lai hoặc phân loại dữ liệu vào các nhóm khác nhau. Ví dụ, dự đoán nhu cầu thị trường, phát hiện gian lận tín dụng, hoặc phân loại email thành thư rác và thư không mong muốn.
2. **Khai phá chuỗi thời gian**: Dùng khai phá dữ liệu để xác định mô hình và xu hướng trong dữ liệu theo thời gian, ví dụ như dự đoán giá cổ phiếu, dự báo lưu lượng truy cập trang web, hoặc phát hiện các biểu đồ chuỗi trong dữ liệu thời gian.
3. **Khai phá tập hợp dữ liệu lớn (Big Data)**: Với sự gia tăng đáng kể của dữ liệu số, khai phá dữ liệu được sử dụng để trích xuất thông tin và mô hình từ các nguồn dữ liệu lớn, chẳng hạn như dữ liệu xã hội, dữ liệu IoT (Internet of Things), hoặc dữ liệu từ các nguồn trực tuyến.
4. **Khám phá tri thức ẩn**: Xác định các mẫu và tri thức ẩn trong dữ liệu, như mối quan hệ giữa các yếu tố khác nhau. Ví dụ, khám phá quy luật mua sắm của khách hàng hoặc mô hình hành vi của người dùng trên mạng xã hội.
5. **Phân tích dữ liệu y tế**: Khai phá dữ liệu y tế để dự đoán bệnh tật, xác định yếu tố nguy cơ, và cải thiện chẩn đoán bệnh.
6. **Quản lý tài chính và ngân hàng**: Sử dụng khai phá dữ liệu để theo dõi giao dịch tài chính, phát hiện gian lận, và tối ưu hóa quản lý rủi ro.
7. **Tối ưu hóa chuỗi cung ứng**: Áp dụng khai phá dữ liệu để tối ưu hóa việc quản lý hàng tồn kho, dự đoán nhu cầu nguồn cung ứng, và cải thiện hiệu suất chuỗi cung ứng.
8. **Phát triển sản phẩm và tiếp thị**: Sử dụng dữ liệu khách hàng để tạo sản phẩm mới, dự đoán hành vi mua sắm, và tối ưu hóa chiến dịch tiếp thị.

Khai phá dữ liệu đóng vai trò quan trọng trong nhiều ngành, giúp các tổ chức hiểu sâu hơn về dữ liệu của họ và tận dụng giá trị tiềm năng từ dữ liệu đó.

### Khái niệm học máy

**Học máy (Machine Learning)** là một phần của trí tuệ nhân tạo (AI) và là lĩnh vực nghiên cứu và ứng dụng trong đó máy tính và hệ thống thông minh được lập trình để tự động học và cải thiện từ dữ liệu mà chúng tiếp xúc. Thay vì phải lập trình cụ thể để thực hiện một nhiệm vụ, học máy cho phép máy tính tự học thông qua sự phân tích và hiểu dữ liệu.

Trong quá trình học máy, các mô hình và thuật toán được sử dụng để tìm hiểu mối quan hệ, mẫu, và cấu trúc trong dữ liệu, sau đó sử dụng kiến thức này để đưa ra dự đoán hoặc quyết định trong các tình huống mới. Học máy có thể được áp dụng rộng rãi trong nhiều lĩnh vực, bao gồm nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên, dự đoán tài chính, dự báo thời tiết, tối ưu hóa chuỗi cung ứng, và nhiều ứng dụng khác.

Phân loại học máy thường được thực hiện dựa trên cách mô hình học từ dữ liệu:

1. **Học máy có giám sát (Supervised Learning)**: Trong học máy có giám sát, mô hình được đào tạo bằng cách sử dụng tập dữ liệu có nhãn, trong đó kết quả mong muốn đã được biết trước. Mô hình sẽ học cách dự đoán hoặc phân loại dữ liệu mới dựa trên mối quan hệ giữa các đặc trưng đầu vào và đầu ra đã biết. Ví dụ bao gồm phân loại email thành thư rác hoặc không phải thư rác, và dự đoán giá cổ phiếu trong tương lai.
2. **Học máy không giám sát (Unsupervised Learning)**: Trong học máy không giám sát, mô hình không được cung cấp dữ liệu đào tạo có nhãn. Thay vào đó, mô hình cố gắng tìm hiểu cấu trúc ẩn trong dữ liệu, thường bằng cách phân loại dữ liệu thành các cụm (phân loại) hoặc tìm kiếm các mẫu dữ liệu phổ biến. Ví dụ bao gồm phân loại tự động các nhóm người dựa trên sở thích hoặc phát hiện nguyên nhân tiềm ẩn trong dữ liệu y tế.
3. **Học máy bán giám sát (Semi-supervised Learning)**: Học máy bán giám sát kết hợp cả hai loại học máy có giám sát và không giám sát. Một phần của dữ liệu có nhãn được sử dụng để đào tạo mô hình, và sau đó mô hình cố gắng tự động phân loại dữ liệu không có nhãn. Điều này hữu ích khi dữ liệu có nhãn khá đắt đỏ hoặc khó khăn để thu thập.
4. **Học máy tăng cường (Reinforcement Learning)**: Trong học máy tăng cường, mô hình được đào tạo để tương tác với môi trường và đưa ra các hành động để đạt được mục tiêu. Mô hình nhận phản hồi từ môi trường về mức độ thành công của các hành động và cố gắng tối ưu hóa chiến lược để đạt được kết quả tốt nhất. Ví dụ bao gồm việc đào tạo máy tính chơi cờ, tự động lái xe, hoặc quản lý cổng thông tin tự động.
5. **Học máy suy luận (Inductive Learning)**: Học máy suy luận là một hình thức của học máy có giám sát, trong đó mô hình được đào tạo để tìm ra một quy tắc tổng quát từ dữ liệu đào tạo. Mô hình sau đó có khả năng áp dụng quy tắc này để dự đoán hoặc phân loại dữ liệu mới mà nó chưa từng thấy.

### Liên hệ giữa khai phá dữ liệu và học máy

- Giống nhau:

+ Cần sử dụng dữ liệu; thường là rất nhiều dữ liệu

+ Phát hiện tri thức từ dữ liệu

- Khác nhau:

Bảng 2. 1: Phân biệt Học máy và Khai phá

|  |  |  |
| --- | --- | --- |
|  | Học máy | Khai phá dữ liệu |
| Trọng tâm: | Tập trung vào việc học(learning) của hệ thống máy tính. | Tập trung vào việc hiểu(understanding) dữ liệu |
| Mục đích sử dụng: | Nhằm dự đoán các kết quả trong tương lai | Nhằm phân tích các dữ liệu hiện có (quá khứ) |

## Các kỹ thuật khai phá dữ liệu

### Kỹ thuật tiền xử lý dữ liệu

#### Xử lý Dữ liệu Thiếu (Handling Missing Data)

Loại bỏ dữ liệu thiếu: Xóa các hàng hoặc cột chứa dữ liệu thiếu.

Điền giá trị thiếu: Sử dụng các phương pháp như điền giá trị trung bình, trung vị hoặc giá trị xuất hiện nhiều nhất để thay thế giá trị thiếu.

#### Xử lý Dữ liệu Nhiễu (Noise Handling)

Loại bỏ nhiễu: Sử dụng các thuật toán lọc nhiễu để loại bỏ dữ liệu nhiễu không mong muốn.

Sử dụng kỹ thuật smoothing: Áp dụng kỹ thuật như trung bình động để làm mịn dữ liệu và giảm ảnh hưởng của nhiễu.

#### Chuẩn hóa Dữ liệu (Data Normalization)

Min-Max Scaling: Chuyển đổi dữ liệu về khoảng giá trị cụ thể, chẳng hạn [0, 1].

Z-score normalization: Chuyển đổi dữ liệu để có giá trị trung bình bằng 0 và độ lệch chuẩn bằng 1.

#### Chuẩn hóa Phân phối (Data Standardization)

Power Transformations: Sử dụng biến đổi mũ để làm cho phân phối dữ liệu gần với phân phối chuẩn.

#### Xử lý Dữ liệu Categorical

One-Hot Encoding: Chuyển đổi biến phân loại thành các biến giả mà mô hình có thể hiểu được.

Label Encoding: Gán nhãn số cho các giá trị phân loại.

#### Loại bỏ Đặc trưng (Feature Selection)

Univariate Feature Selection: Chọn các đặc trưng dựa trên các thống kê như ANOVA.

Recursive Feature Elimination (RFE): Lặp lại việc loại bỏ đặc trưng ít quan trọng nhất từ mô hình.

#### Xử lý Dữ liệu không Cân bằng (Handling Imbalanced Data)

Over-sampling: Tăng số lượng mẫu trong lớp thiếu số lượng.

Under-sampling: Giảm số lượng mẫu trong lớp có số lượng lớn hơn.

Combining Oversampling and Undersampling: kết hợp cả Over-sampling và Under-sampling

### Phân tích thống kê dữ liệu

- Các phương phức thống kê dữ liệu:

+ [**Describe()**: Phương thức này tạo ra các thống kê mô tả, bao gồm những thống kê tóm tắt về xu hướng trung tâm, sự phân tán và hình dạng của phân phối của tập dữ liệu, loại trừ giá trị NaN](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.describe.html). [Nó phân tích cả chuỗi số và đối tượng, cũng như các cột DataFrame có các loại dữ liệu hỗn hợp](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.describe.html" \t "_blank). [Kết quả sẽ khác nhau tùy thuộc vào những gì được cung cấp](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.describe.html" \t "_blank).

+ [**Info()**: Phương thức này cho phép chúng ta tìm hiểu về hình dạng và kiểu đối tượng của dữ liệu](https://campus.datacamp.com/courses/python-for-spreadsheet-users/diving-in?ex=7). [Nó cung cấp thông tin tổng quan về DataFrame bao gồm số lượng hàng và cột, kiểu dữ liệu của mỗi cột, số lượng giá trị không null và sử dụng bộ nhớ](https://campus.datacamp.com/courses/python-for-spreadsheet-users/diving-in?ex=7" \t "_blank).

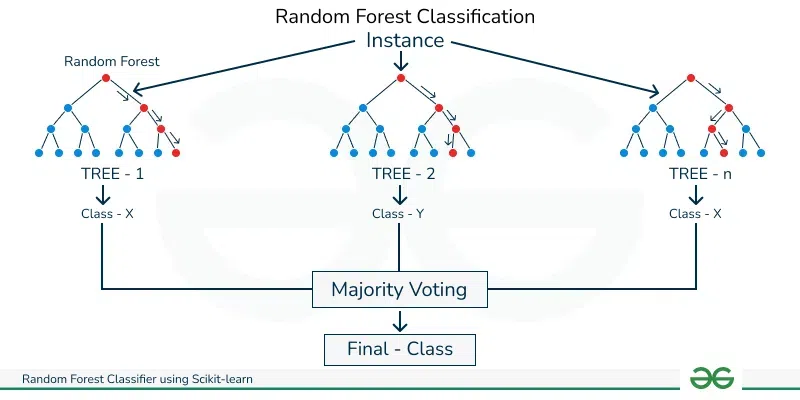
## Nghiên cứu một số kỹ thuật học máy được áp dụng trong bài toán

### Mô hình Random Forest

Rừng ngẫu nhiên hoặc Rừng quyết định ngẫu nhiên là một thuật toán Machine learning được giám sát được sử dụng để phân loại, hồi quy và các tác vụ khác sử dụng cây quyết định. Rừng ngẫu nhiên đặc biệt phù hợp để xử lý các tập dữ liệu lớn và phức tạp, xử lý các không gian đối tượng nhiều chiều và cung cấp thông tin chi tiết về tầm quan trọng của đối tượng. Khả năng duy trì độ chính xác dự đoán cao của thuật toán này trong khi giảm thiểu việc trang bị quá mức khiến nó trở thành lựa chọn phổ biến trên nhiều lĩnh vực khác nhau, bao gồm tài chính, chăm sóc sức khỏe và phân tích hình ảnh, cùng nhiều lĩnh vực khác.

* + **Phân Loại**

Trình phân loại rừng ngẫu nhiên tạo một [tập hợp](https://www.geeksforgeeks.org/set-in-cpp-stl/" \t "_blank) cây [quyết định](https://www.geeksforgeeks.org/decision-tree/" \t "_blank) từ tập hợp con được chọn ngẫu nhiên của tập huấn luyện. Đó là một tập hợp các cây quyết định (DT) từ một tập hợp con được chọn ngẫu nhiên của tập huấn luyện và sau đó Nó thu thập phiếu bầu từ các cây quyết định khác nhau để quyết định dự đoán cuối cùng.



Hình 2. 1Sơ đồ tổng quan về RandomForest

Ngoài ra, trình phân loại rừng ngẫu nhiên có thể xử lý cả nhiệm vụ phân loại và hồi quy, đồng thời khả năng cung cấp điểm quan trọng của tính năng khiến nó trở thành một công cụ có giá trị để hiểu tầm quan trọng của các biến khác nhau trong tập dữ liệu.

* + **Tầm quan trọng của tính năng rừng ngẫu nhiên**

Một ưu điểm tuyệt vời khác của thuật toán rừng ngẫu nhiên là rất dễ dàng đo lường tầm quan trọng tương đối của từng đặc điểm đối với dự đoán. Sklearn cung cấp một công cụ tuyệt vời để đo lường tầm quan trọng của một tính năng bằng cách xem xét mức độ giảm tạp chất của các nút cây sử dụng tính năng đó trên tất cả các cây trong rừng. Nó tự động tính toán điểm này cho từng tính năng sau khi đào tạo và chia tỷ lệ kết quả sao cho tổng của tất cả tầm quan trọng bằng một.

Nếu bạn không biết cây quyết định hoạt động như thế nào hoặc lá hoặc nút là gì, thì đây là một mô tả hay từ Wikipedia: “Trong cây quyết định, mỗi nút bên trong đại diện cho một 'kiểm tra' trên một thuộc tính (ví dụ: liệu một đồng xu có lật xuất hiện mặt ngửa hoặc mặt ngửa), mỗi nhánh biểu thị kết quả của bài kiểm tra và mỗi nút lá biểu thị một nhãn lớp (quyết định được đưa ra sau khi tính toán tất cả các thuộc tính). Một nút không có nút con là một chiếc lá.”

Bằng cách xem xét tầm quan trọng của tính năng, bạn có thể quyết định những tính năng nào có thể bị loại bỏ vì chúng không đóng góp đủ (hoặc đôi khi không đóng góp gì cả) cho quá trình dự đoán. Điều này rất quan trọng vì nguyên tắc chung trong học máy là bạn càng có nhiều tính năng thì mô hình của bạn càng có nhiều khả năng bị trang bị quá mức và ngược lại.

* + **Cách rừng ngẫu nhiên hoạt động**

**Tập hợp các Cây quyết định:** Rừng ngẫu nhiên tận dụng sức mạnh của [việc học tập tổng hợp](https://www.geeksforgeeks.org/ensemble-methods-in-python/) bằng cách xây dựng một đội quân [Cây quyết định](https://www.geeksforgeeks.org/decision-tree/) . Những cây này giống như các chuyên gia riêng lẻ, mỗi chuyên gia chuyên về một khía cạnh cụ thể của dữ liệu. Điều quan trọng là chúng hoạt động độc lập, giảm thiểu rủi ro mô hình bị ảnh hưởng quá mức bởi các sắc thái của một cây duy nhất.

**Lựa chọn tính năng ngẫu nhiên:** Để đảm bảo rằng mỗi cây quyết định trong tập hợp mang lại một góc nhìn độc đáo, Rừng ngẫu nhiên sử dụng [lựa chọn tính năng ngẫu nhiên](https://www.geeksforgeeks.org/feature-selection-techniques-in-machine-learning/). Trong quá trình huấn luyện mỗi cây, một tập hợp con các đặc tính ngẫu nhiên sẽ được chọn. Tính ngẫu nhiên này đảm bảo rằng mỗi cây tập trung vào các khía cạnh khác nhau của dữ liệu, thúc đẩy một tập hợp các yếu tố dự đoán đa dạng trong quần thể.

**Tổng hợp hoặc đóng bao Bootstrap:** Kỹ thuật đóng bao là nền tảng trong chiến lược đào tạo của Random Forest, bao gồm việc tạo nhiều mẫu bootstrap từ tập dữ liệu gốc, cho phép lấy mẫu các phiên bản thay thế. Điều này dẫn đến các tập hợp dữ liệu khác nhau cho mỗi cây quyết định, tạo ra tính biến đổi trong quá trình đào tạo và làm cho mô hình trở nên mạnh mẽ hơn.

**Ra quyết định và biểu quyết:** Khi đưa ra dự đoán, mỗi cây quyết định trong Rừng ngẫu nhiên sẽ bỏ phiếu. Đối với [các nhiệm vụ phân loại](https://www.geeksforgeeks.org/ml-classification-vs-regression/), dự đoán cuối cùng được xác định theo [chế độ](https://www.geeksforgeeks.org/mode/) (dự đoán thường xuyên nhất) trên tất cả các cây. Trong [các tác vụ hồi quy](https://www.geeksforgeeks.org/regression-classification-supervised-machine-learning/), giá trị trung bình của các dự đoán cây riêng lẻ được lấy. Cơ chế bỏ phiếu nội bộ này đảm bảo quá trình ra quyết định tập thể và cân bằng.

### ****Mô hình Decision Tree****

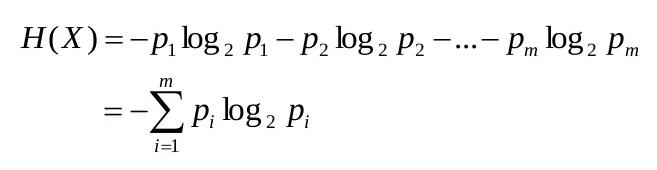
Decision tree là một mô hình supervised learning, có thể được áp dụng vào cả hai bài toán classification và regression. Mỗi một nút trong (internal node) tương ứng với một biến; đường nối giữa nó với nút con của nó thể hiện một giá trị cụ thể cho biến đó. Mỗi nút lá đại diện cho giá trị dự đoán của biến mục tiêu, cho trước các giá trị của các biến được biểu diễn bởi đường đi từ nút gốc tới nút lá đó. Kỹ thuật học máy dùng trong cây quyết định được gọi là học bằng cây quyết định, hay chỉ gọi với cái tên ngắn gọn là cây quyết định.

**Entropy**

Giả sử ta có biến ngẫu nhiên rời rạc X:

Có không gian mẫu là {x1, x2, ..., xm} với xác suất

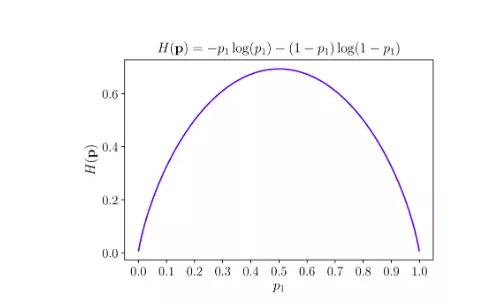
P(X = x1) = p1, P(X = x2) = p2, ..., P(X = xm) = pm Số bit trung bình nhỏ nhất để truyền một đơn vị dữ liệu theo phân phối P(X):



Hình 2. 2: Công thức tính Entropy

H(X) là entropy của X (0 <= H(X) <= log2m)

Xem xét m = 2, trong trường hợp X là tinh khiết nhất, tức là một trong hai pi bằng 0 và giá trị kia bằng 1, khi đó H(X) = 0. Khi X là vẩn đục nhất tức cả hai xác suất mang giá trị pi = 0.5, hàm entropy đạt giá trị cao nhất.



Hình 2. 3: Độ tinh khiết của Entropy

Vậy với m > 2,hàm entropy đạt giá trị nhỏ nhất nếu có một giá trị pi = 1 (tức các giá trị pi còn lại mang giá trị 0), đạt giá trị lớn nhất nếu tất cả pi bằng nhau.

Giá trị entropy:

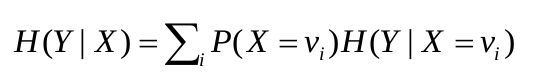
Lớn: Phân phối P(X) gần với dạng phân phối đồng nhất (uniform distribution)

Nhỏ: Phân phối P(X) xa dạng phân phối đồng nhất

Những giá trị này khiến nó được sử dụng trong việc đo độ vẩn đục của một pháp phân chia trong ID3.

**Entropy điều kiện H(Y|X)**

H(Y|X) = trung bình các giá trọ entropy điều kiện cụ thể H(Y|X = v)



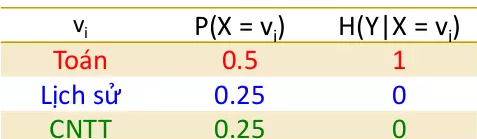
Hình 2. 4: Công thức Entropy

**Ví dụ:**



Hình 2. 5: bảng dữ liệu ví dụ decision tree

Ta có:



Hình 2. 6: Cách tính Entropy trong decision tree

Vậy nên: H(Y|X) = 0.5 x 1 + 0.25 x 0 + 0.25 x 0 = 0.5

Information Gain IG(Y|X)

IG(Y|X) là số lượng bit trung bình có thể tiết kiệm khi truyền Y mà hai đầu gửi và nhận đã biết X.

Tức là: IG(Y|X) = H(Y) - H(Y|X)

Ví dụ:

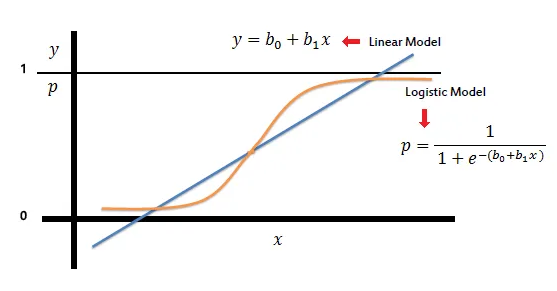
H(Y) = 1

H(Y|X) = 0.5

IG(Y|X) = 1 - 0.5 = 0.5

### ****Mô hình Logistic Regresstion****

Hồi quy Logistic là một mô hình thống kê được sử dụng để phân loại nhị phân, tức dự đoán một đối tượng thuộc vào một trong hai nhóm. Hồi quy Logistic làm việc dựa trên nguyên tắc của hàm sigmoid – một hàm phi tuyến tự chuyển đầu vào của nó thành xác suất thuộc về một trong hai lớp nhị phân.



Hình 2. 7: Hàm chi phí trong hồi quy logistic

Hồi quy Logistic hoạt động dựa trên hàm Sigmoid, được biểu diễn như sau:

𝑆(𝑧)=1/(1+𝑒−𝑧)S(z)=1/(1+e−z)

Hàm Sigmoid nhận đầu vào là một giá trị z bất kỳ, và trả về đầu ra là một giá trị xác suất nằm trong khoảng [0,1][0,1]. Khi áp dụng vào mô hình Hồi quy Logistic với đầu vào là ma trận dữ liệu 𝑋*X* và trọng số 𝑤*w*, ta có 𝑧=𝑋𝑤*z*=*Xw*.

Việc huấn luyện của mô hình là tìm ra bộ trọng số 𝑤*w* sao cho đầu ra dự đoán của hàm Sigmoid gần với kết quả thực tế nhất. Để làm được điều này, ta sử dụng hàm mất mát ([Loss](https://trituenhantao.io/tu-dien-thuat-ngu/loss/) Function) để đánh giá hiệu năng của mô hình. Mô hình càng tốt khi hàm mất mát càng nhỏ.

Hàm mất mát ([Loss](https://trituenhantao.io/tu-dien-thuat-ngu/loss/) Function) là một hàm số được sử dụng để đo lường mức độ lỗi mà mô hình của chúng ta tạo ra khi dự đoán các kết quả từ dữ liệu đầu vào. Trong bài toán Hồi quy Logistic, chúng ta sử dụng hàm mất mát [Cross-Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/cross-entropy/) (còn gọi là Log [Loss](https://trituenhantao.io/tu-dien-thuat-ngu/loss/)) để đánh giá hiệu năng của mô hình.

Hàm mất mát [Cross-Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/cross-entropy/) được định nghĩa như sau:  
𝐿(𝑤)=−1𝑛∑𝑖=1𝑛[𝑦𝑖log⁡𝑝𝑖+(1−𝑦𝑖)log⁡(1−𝑝𝑖)]*L*(*w*)=−*n*1​∑*i*=1*n*​[*yi*​log*pi*​+(1−*yi*​)log(1−*pi*​)],

Trong đó:

* 𝑛*n*: số lượng mẫu dữ liệu trong tập huấn luyện.
* 𝑦𝑖*yi*​: giá trị thực tế của đầu ra thứ i.
* 𝑝𝑖*pi*​: xác suất dự đoán thuộc lớp 1 của mô hình cho đầu vào thứ i.

Hàm [Cross-Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/cross-entropy/) đo lường khoảng cách giữa hai phân phối xác suất 𝑦𝑖*yi*​ và 𝑝𝑖*pi*​. Khi mô hình dự đoán chính xác, tức là nếu 𝑦𝑖=1*yi*​=1 thì 𝑝𝑖*pi*​ càng gần 1, và nếu 𝑦𝑖=0*yi*​=0 thì 𝑝𝑖*pi*​ càng gần 0, sau đó hàm mất mát sẽ tiến gần về 0.

Trong quá trình huấn luyện, chúng ta tìm cách cập nhật bộ trọng số 𝑤*w* sao cho giá trị hàm mất mát [Cross-Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/cross-entropy/) đạt giá trị nhỏ nhất, dẫn đến một mô hình dự đoán tốt nhất.

Để tìm giá trị tối ưu cho bộ trọng số 𝑤*w*, chúng ta có thể sử dụng kỹ thuật [Gradient](https://trituenhantao.io/tu-dien-thuat-ngu/gradient/) Descent. Tại mỗi bước lặp, chúng ta cập nhật 𝑤*w* theo phương từm ứng với đạo hàm của hàm mất mát 𝐿(𝑤)*L*(*w*) theo 𝑤*w*.

**Công thức toán**

Trong quá trình học và làm việc với Hồi quy Logistic, chúng ta cần nắm vững một số công thức toán quan trọng như sau:

Hàm sigmoid:

𝑆(𝑧)=1/(1+𝑒−𝑧)*S*(*z*)=1/(1+*e*−*z*)

Xác suất thuộc lớp 1 cho mỗi ví dụ:

𝑃(𝑦=1∣𝑋)=𝑆(𝑋𝑤)*P*(*y*=1∣*X*)=*S*(*Xw*)

Hàm mất mát [cross-entropy](https://trituenhantao.io/tu-dien-thuat-ngu/cross-entropy/) (sử dụng cho Logistic Regression):

𝐿(𝑤)=−1𝑛∑𝑖=1𝑛[𝑦𝑖log⁡𝑝𝑖+(1−𝑦𝑖)log⁡(1−𝑝𝑖)]*L*(*w*)=−*n*1​∑*i*=1*n*​[*yi*​log*pi*​+(1−*yi*​)log(1−*pi*​)]

[Gradient](https://trituenhantao.io/tu-dien-thuat-ngu/gradient/) của hàm mất mát (đạo hàm theo 𝑤*w* của hàm mất mát):

∇𝐿(𝑤)=1𝑛𝑋𝑇(𝑆(𝑋𝑤)−𝑌)∇*L*(*w*)=*n*1​*XT*(*S*(*Xw*)−*Y*)

Sử dụng các công thức trên, chúng ta có thể huấn luyện mô hình Hồi quy Logistic và dự đoán kết quả cho dữ liệu mới.

# XÂY DỰNG MÔ HÌNH HỌC MÁY

## Thu thập dữ liệu

### Thu thập dữ liệu

Dữ liệu được thu thập trên trang Kaggle: [Cerebral Stroke Prediction-Imbalanced Dataset (kaggle.com)](https://www.kaggle.com/datasets/shashwatwork/cerebral-stroke-predictionimbalaced-dataset/data)

Thông tin về dữ liệu cần để phục vụ cho việc hỗ trợ chẩn đoán trạng thái khoản vay:

Dữ liệu ban đầu của 887379 trạng thái khoản vay với 21 thông tin chi tiết về khoản vay.



Hình 3. 1: Dữ liệu ban đầu

## Phân tích dữ liệu thu thập

### Mô tả dữ liệu

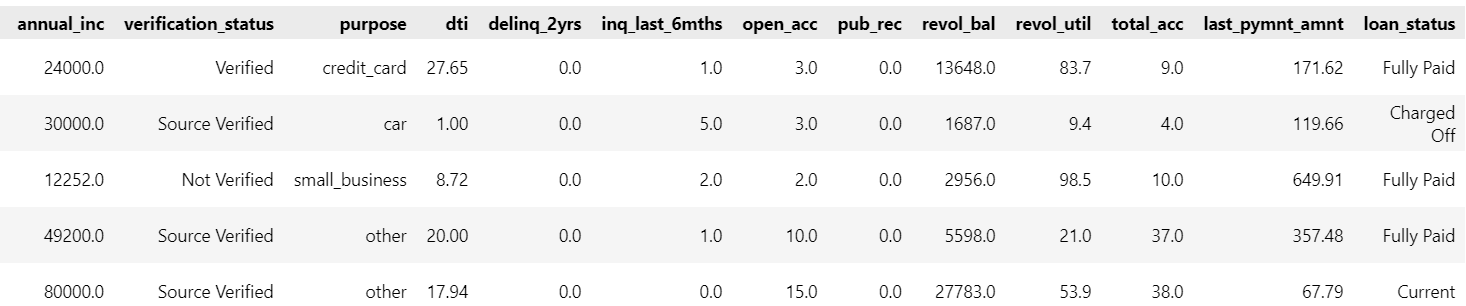
* Mô tả các trường dữ liệu ban đầu

Bảng 3. 1: Mô tả các trường dữ liệu

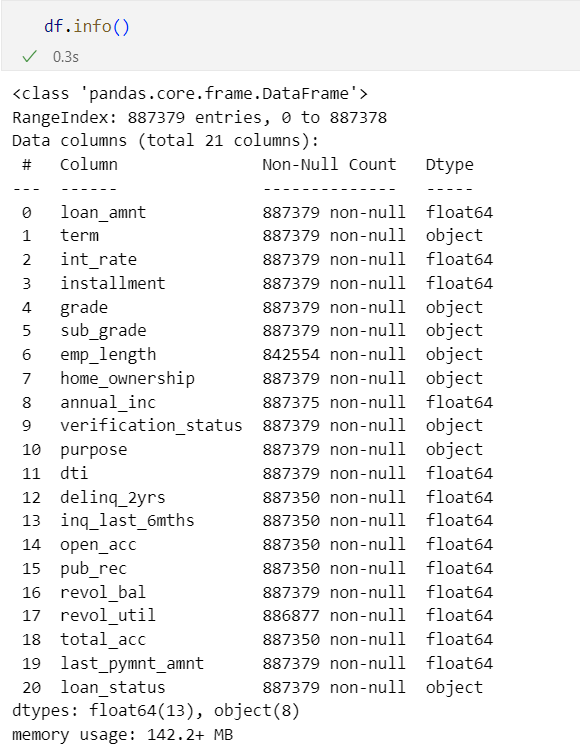
| **STT** | **Thông tin** | **Tên cột** | **Mô tả thông tin** |
| --- | --- | --- | --- |
|  | loan\_amnt | 1.0 | Số tiền của khoản vay |
|  | term | 2.0 | Thời hạn của khoản vay (36 hoặc 60 tháng) |
|  | int\_rate | 3.0 | Lãi suất của khoản vay |
|  | installment | 4.0 | Số tiền trả hàng tháng của khoản vay |
|  | grade | 5.0 | Hạng được gán bởi LendingClub dựa trên khả năng tín dụng của người vay, được phân ra theo các hạng A, B, C, D, E, F, G. |
|  | sub\_grade | 6.0 | Một hạng chi tiết hơn dựa trên khả năng tín dụng của người vay |
|  | emp\_length | 7.0 | Thời gian làm việc của người vay được tính theo năm '10+ years', '< 1 year', '3 years', '9 years', '4 years', '5 years', '1 year' |
|  | home\_ownership | 8.0 | Tình trạng sở hữu nhà do người vay cung cấp trong quá trình đăng ký. RENT, OWN, MORTGAGE, OTHER. |
|  | annual\_inc | 9.0 | Thu nhập hàng năm tự báo cáo do người vay cung cấp trong quá trình đăng ký. |
|  | verification\_status | 10.0 | Cho biết liệu thu nhập đã được Lending Club xác minh chưa, chưa được xác minh hoặc nguồn thu nhập đã được xác minh hay chưa |
|  | purpose | 11.0 | Một danh mục được cung cấp bởi người vay cho yêu cầu vay vốn để xem mục đích của người muốn vay vào việc gì. |
|  | dti | 12.0 | Tỷ lệ được tính bằng cách sử dụng tổng số tiền thanh toán nợ hàng tháng của người đi vay trên tổng nghĩa vụ nợ, không bao gồm khoản thế chấp và khoản vay LC được yêu cầu, chia cho thu nhập hàng tháng mà người đi vay tự báo cáo. |
|  | delinq\_2yrs | 13.0 | Số trường hợp quá hạn trên 30 ngày trong hồ sơ tín dụng của người đi vay trong 2 năm qua |
|  | inq\_last\_6mths | 14.0 | Số lần kiểm tra thông tin trong 6 tháng gần đây |
|  | open\_acc | 15.0 | Số hạn mức tín dụng mở trong hồ sơ tín dụng của người đi vay. |
|  | pub\_rec | 16.0 | Số bản ghi công cộng xấu |
|  | revol\_bal | 17.0 | Tổng dư nợ quay vòng tín dụng |
|  | revol\_util | 18.0 | Tỷ lệ sử dụng dòng quay vòng, hoặc số tiền tín dụng mà người vay đang sử dụng so với tất cả tín dụng quay vòng có sẵn. |
|  | total\_acc | 19.0 | Tổng số hạn mức tín dụng hiện có trong hồ sơ tín dụng của người vay |
|  | last\_pymnt\_amnt | 20.0 | Tổng số tiền thanh toán cuối cùng nhận được |
|  | loan\_status | 21.0 | Tình trạng hiện tại của khoản vay |

Để hiểu Dữ liệu, chúng ta sử dụng một số hàm trong thư viện Sklearn của Python:

- Xem cấu trúc Dữ liệu: Sử dụng hàm ***data.head(),*** ***data.info()*** và ***data.describe()***

******

Hình 3. 2: Tổng quan về dữ liệu

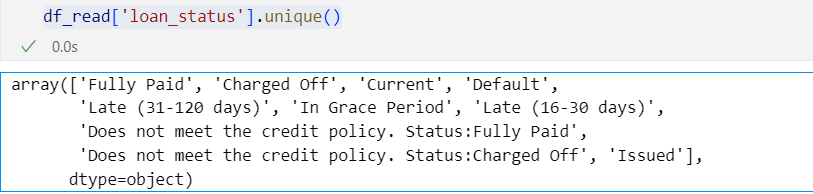


Hình 3. 3:Thông tin tập dữ liệu



Hình 3. 4: Bản tóm tắt thống kê

-Khám phá Các giá trị trong cột 'loan\_status', các trạng thái khác nhau mà một khoản vay có thể có.



Hình 3. 5: Các trạng thái đầy đủ của khoản vay

Ta có thể thấy các trạng thái khoản vay bao gồm:

 **Fully Paid**: Khoản vay đã được thanh toán hoàn toàn.

 **Charged Off**: Khoản vay đã bị công nhận là không thể thu hồi được do khả năng thanh toán kém.

 **Current**: Khoản vay hiện đang được trả tiền theo hợp đồng, không có vấn đề pháp lý hay vi phạm hợp đồng nào.

 **Default**: Khoản vay đã vượt quá thời hạn thanh toán và đã bị xem là mặc định.

 **Late (31-120 days)**: Khoản vay đã quá hạn từ 31 đến 120 ngày.

 **In Grace Period**: Khoản vay đang trong thời gian ân hạn thanh toán.

 **Late (16-30 days)**: Khoản vay đã quá hạn từ 16 đến 30 ngày.

**Paid**: Khoản vay đã thanh toán hoàn toàn nhưng không đáp ứng đủ các tiêu chuẩn tín dụng của tổ chức cho vay.

**Off**: Khoản vay bị công nhận là không thể thu hồi được và không đáp ứng đủ các tiêu chuẩn tín dụng của tổ chức cho vay.

 **Issued**: Khoản vay đã được phê duyệt nhưng chưa được chi tiêu.

Ta thấy có tất cả 10 trạng thái khoản vay. Nhưng để giải quyết bài toán đã đặt ra từ đầu là dự đoán xem người cho vay có nên phê duyệt khoản vay hay không và bài toán hồ sơ người vay có đủ điều kiện hay không thì ta không cần sử dụng đến 10 trạng thái khoản vay mà ta chỉ cần sử dụng 2 trạng thái đó là ‘**Charged Off**’ và “**Fully Paid**”để đưa ra dự đoán. Do vậy ta sẽ cần sử lý trạng thái khoản vay trong bước tiếp theo

### Tiền xử lý dữ liệu

Như trình bày ở trên dữ liệu đầu vào của bài toán mục tiêu của chúng ta sẽ là người cho vay có nên phê duyệt khoản vay hay không và bài toán hồ sơ người vay có đủ điều kiện hay không. Các thông tin này sẽ giúp chúng ta xây dựng một mô hình dự đoán có thể phân loại một hồ sơ vay mới vào một trong hai nhóm: được phê duyệt hoặc không được phê duyệt. Quá trình này không chỉ giúp tăng tính khả dụng vốn cho người vay mà còn giảm thiểu rủi ro cho nhà đầu tư hoặc người cho vay. Để có thể cài đặt được thuật toán, cài đặt được chương trình chúng ta cần xử lí dữ liệu đầu vào

- Lấy ra 2 trạng thái trong cột trạng thái khoản vay để làm mục tiêu dự đoán.

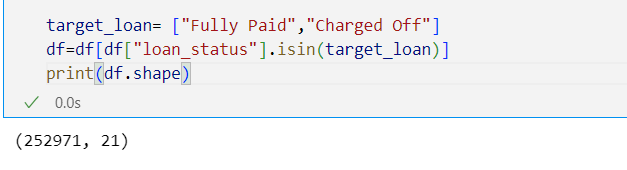
- Làm sạch dữ liệu dưới hình thức tìm và xử lý các giá trị null trong các hàng.

- Xem vào phân phối của các thuộc tính liên tục và nếu chúng không phân phối bình thường thì xử lý nó.

- Thực hiện một số phân tích thống kê trên dữ liệu để có được một số suy luận

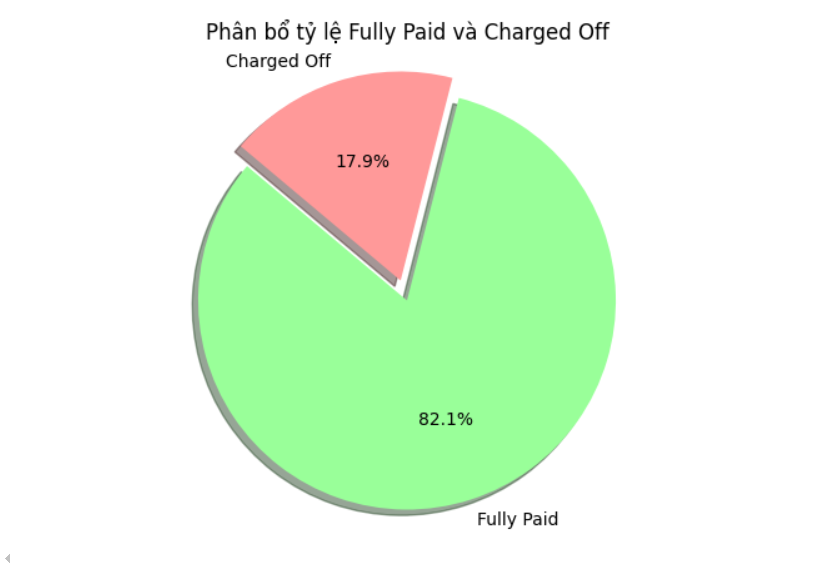
- Chuyển đổi biến phân loại của kiểu dữ liệu đối tượng thành kiểu int.

- Chuẩn hóa biến liên tục bằng cách sử dụng MinMaxScaling



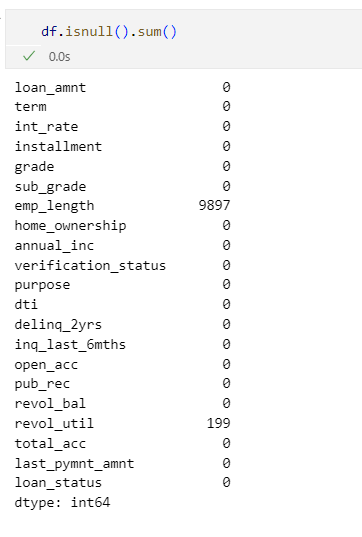
Hình 3. 6: Kiểm tra kích thước của cột trạng thái khoản vay

Sau đó ta kiểm tra xem giá trị trong cột trạng thái khoản vay có bị mất cân bằng hay không.



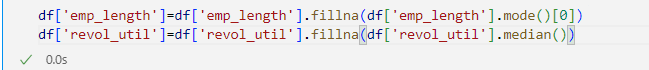
Hình 3. 7: Biểu đồ phân bố tỉ lệ

-Tính tổng giá trị null cho từng cột



Hình 3. 8 Kiểm tra có giá trị nào null hay không

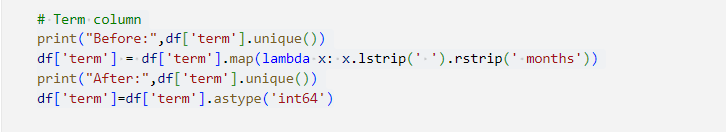
Nhận xét: ta thấy cột ‘emp\_length’ có 9897 giá trị null và cột ‘revol\_until’ có 199 các giá trị null. Một giải pháp đưa ra ta sẽ thay thế tất cả các giá trị null của cột ‘emp\_length’ thành các giá trị phổ biến của cột, đối với cột ‘revol\_until’ ta sẽ xử lý bằng cách thay thế các giá trị thiếu trong cột bằng giá trị trung vị của cột này.



Hình 3. 9: Điền đầy giá trị còn thiếu

-Tiếp đến chúng ta sẽ chuẩn hóa các dữ liệu của từng cột:

-Chúng ta loại bỏ các khoảng trắng và chữ "months" từ các giá trị trong cột ‘term’ , rồi chuyển đổi các giá trị này từ kiểu chuỗi (string) sang kiểu số nguyên (int). Kết quả là các giá trị trong cột term sẽ chỉ còn lại phần số.

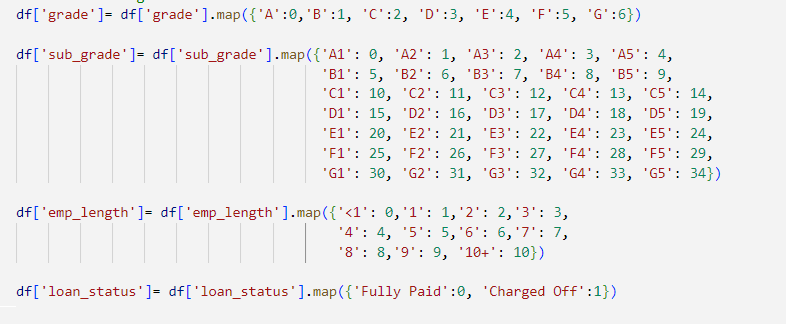


Hình 3. 10: Chuẩn hóa cột term

Bảng 3. 2: Cột term

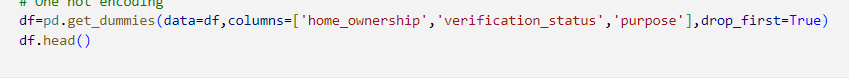
|  |  |  |
| --- | --- | --- |
| **STT** | **tearm** | **Chuẩn hoá về dạng số** |
| 1 | 36 months | 36 |
| 2 | 60 months | 60 |

-Ta tiếp tục thực hiện quá trình mã hóa(encoding) các cột phân loại ‘grade’, ‘sub\_grade’, ‘emp\_length’ và cột ‘loan\_status’



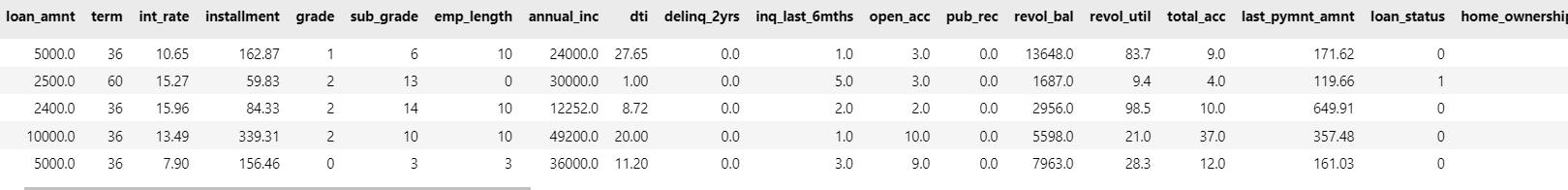
Hình 3. 11: Encoding

-Sau đó ta dùng kĩ thuật one hot encoding đối với những cột ‘home\_ownership’, ‘verificatin\_status’, ‘purpose’



Hình 3. 12: One hot encoding

-Kết quả ta đạt được:

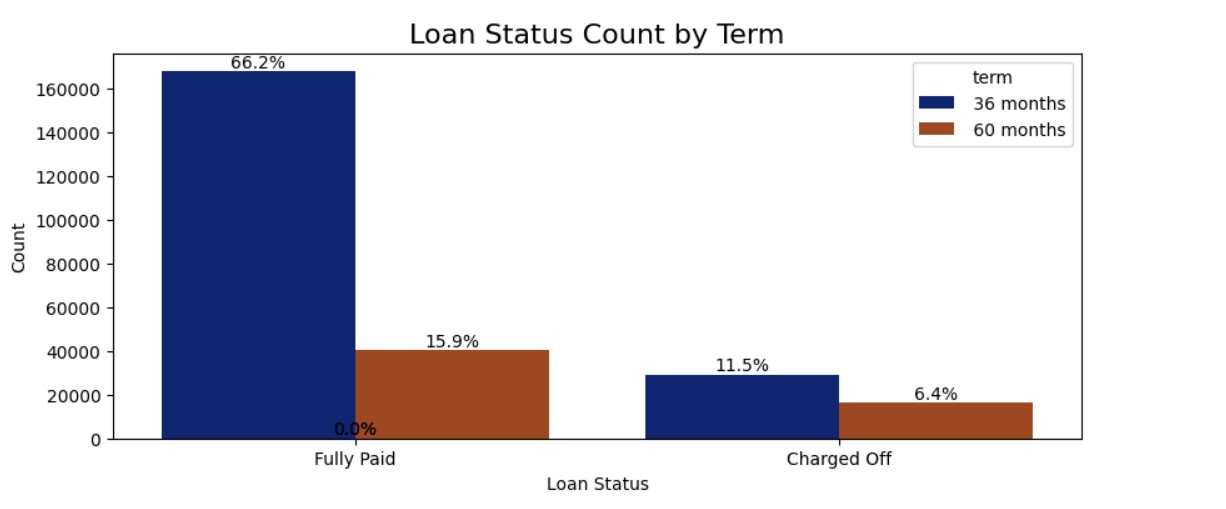


Hình 3. 13: Dữ liệu sao Encode

## Xây dựng mô hình học máy

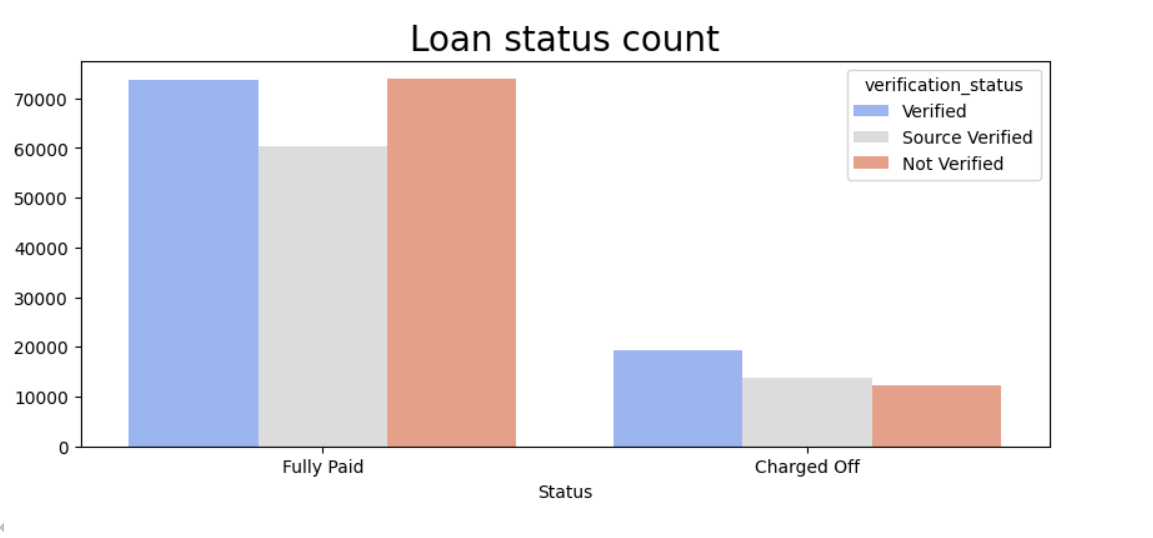
### Quan sát dữ liệu

Ta xem tỉ lệ dữ liệu của 2 giá trị của trạng thái khoản vay trong điều kiện khoản vay trong 36 tháng và 60 tháng.



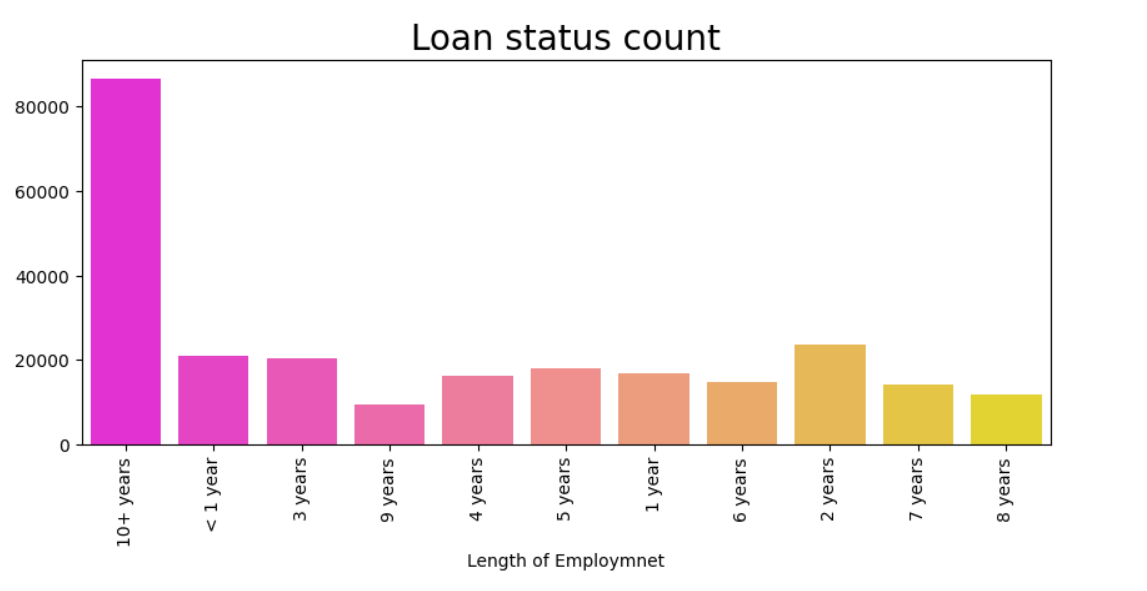
Hình 3. 14: Tỉ lệ Fully Paid Và Charge Off trong cột term

-Tỉ lệ trạng thái khoản vay của các mức thu nhập được xác nhận, được bên thứ 3 xác nhận và chưa được xác nhận.



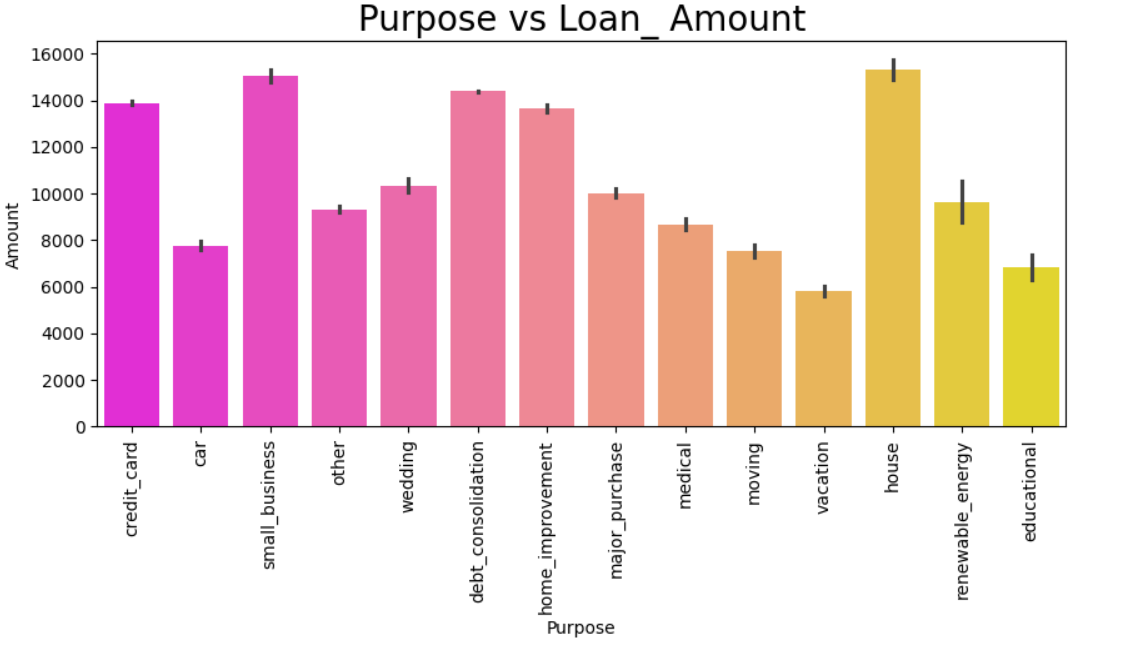
Hình 3. 15: Tỉ lệ Fully Paid và Charge Off trong cột verification\_status

Tỉ lệ của các tài khoản được phê duyệt trả được nợ của những người đi làm lâu năm cao hơn so với những người mới đi làm.



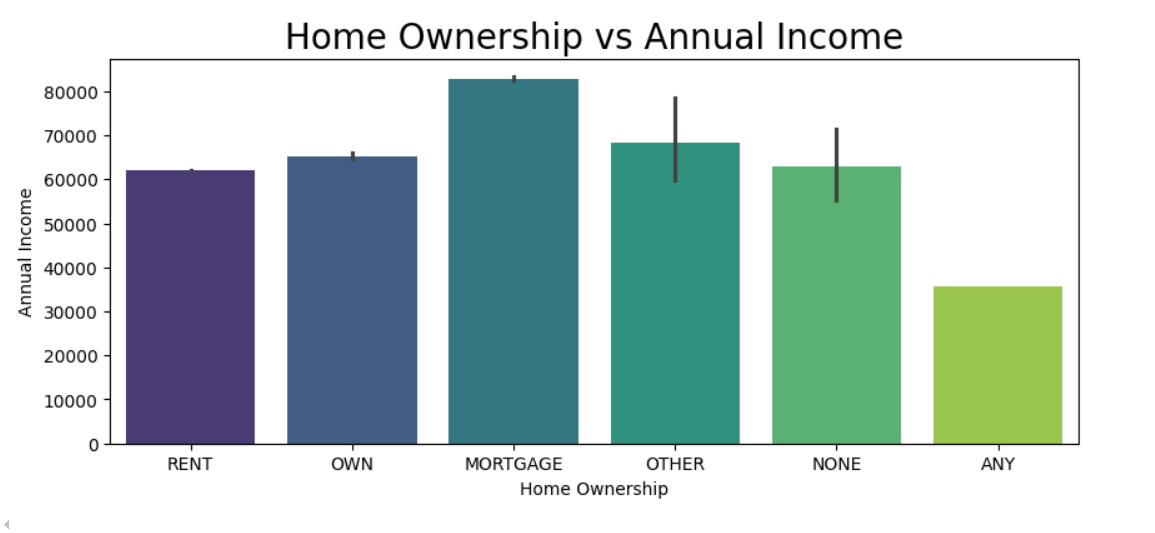
Hình 3. 16: Tỉ lệ Fully Paid Biểu diễn theo Employee

So với các mục tiêu thì mục tiêu thì mục tiêu mua nhà có tỉ lệ hoàn trả nợ là cao nhất.



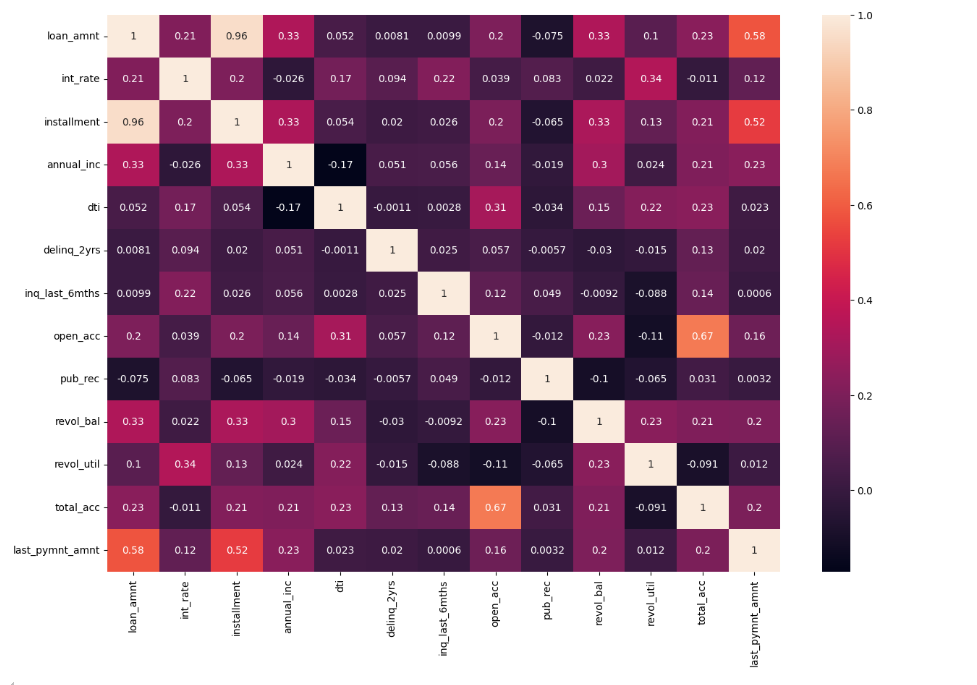
Hình 3. 17: Tỉ lệ Fully Paid phân bố theo PurPose

Và những người sở hữu nhà ở là những người có tỉ lệ trả nợ cao nhất.

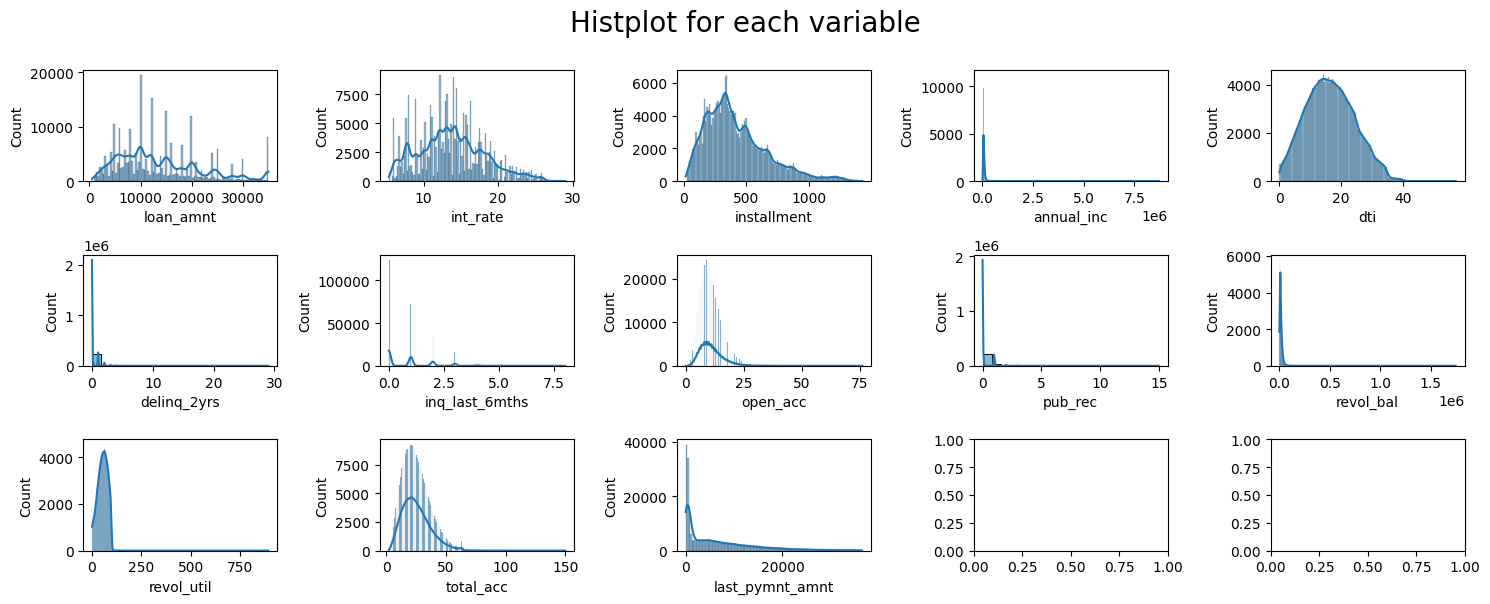


Hình 3. 19: Tỉ lệ Fully Paid Phân phối theo Home Ownership

Dưới đây là biểu đồ heatmap được sử dụng để trực quan hóa ma trận tương quan giữa các đặc trưng trong tập dữ liệu, nó thể hiện mối quan hệ tuyến tính giữa các đặc trưng.



Hình 3. 20: Biểu đồ heatmap



Hình 3. 21: Biều đồ Histogram cho từng biến số

### Xây dựng mô hình

Để thực hiện thực nghiệm đánh giá các mô hình, em sử dụng ngôn ngữ Python chạy trên môi trường Visual Studiocode để cài đặt chương trình.

Đối với bài toán hỗ trợ dự đoán trạng thái khoản vay mô hình dự đoán được xây dựng dựa trên ba thuật toán phân lớp đó là: RF, DT, LR Tất cả các thuật toán này được sử dụng trong thư viện Sklearn của Python, một thư viện rất phổ biến và được nhiều người sử dụng.

## Phân tích kết quả thực nghiệm

### Thước đo đánh giá

Trong quá trình xây dựng một mô hình machine learning, một phần không thể thiếu để biết được chất lượng của mô hình như thế nào đó chính là đánh giá mô hình. Việc đánh giá mô hình giúp chúng ta lựa chọn được mô hình phù hợp nhất đối với bài toán của mình. Tuy nhiên để tìm được thước đo đánh giá mô hình phù hợp thì chúng ta cần phải hiểu về ý nghĩa, bản chất và trường hợp áp dụng của từng thước đo. Trong luận văn này, tác giả đã sử dụng các metric đánh giá: Accuracy, Precision, Recall và F1-score. Cụ thể:

- Accuracy: là độ chính xác cho biết tỷ lệ các mẫu được chẩn đoán đúng trên tổng số mẫu.

- Precision: cho biết thực sự có bao nhiêu chẩn đoán Positive là thật sự True

- Recall: là tỷ lệ chẩn đoán chính xác các trường hợp positive trên toàn bộ các mẫu thuộc nhóm positive

- F1-score: là là trung bình điều hòa giữa precision và recall

Trong đó: + TP (True Positive): chẩn đoán đúng là Positive + TN (True Negative): chẩn đoán đúng là Negative + FP (False Positive): chẩn đoán sai là Positive. + FN (False Negative): chẩn đoán sai là Negative. + Positive và Negative ở đây tương ứng với hai nhãn 1 (không trả được nợ) và 0 (trả được nợ)

Ngoài ra, khái niệm ma trận nhầm lẫn cũng được tác giả đề cập đến trong luận văn. Ma trận nhầm lẫn có bố cục dạng bảng, cho biết bố cục dạng bảng cho biết hiệu suất của một thuật toán. Cụ thể: [TP FP]

[FN TN]

### Chuẩn bị thực nghiệm

Để tiến hành thực nghiệm, tác giả luận văn chia tập dữ liệu thành hai tập: Tập huấn luyện (X\_train, y\_train) và tập kiểm tra (X\_test, y\_test) với tỷ lệ 75% và 25% (X là đặc trưng, y là nhãn).

Chương trình dưới đây sẽ chia tập dữ liệu thành hai tập: Tập huấn luyện (X\_train, y\_train) và tập kiểm tra (X\_test, y\_test).



Hình 3. 22: Chương trình chia tập dữ liệu thành hai tập huấn luyện và kiểm tra

### Tiến hành thực nghiệm

Dựa vào ta tiến hành thực nghiệm mô hình với các trường hợp khác nhau. Cụ thể:

-Thực nghiệm mô hình trước khi chuẩn hóa MinMax Scaler.

- Để kiểm tra tính hiệu quả, ta sẽ thực hiện thực nghiệm sau khi chuẩn hóa để so sánh.

- Thực nghiệm mô hình với kỹ thuật Undersampling (Giảm số lượng mẫu của lớp đa số để cân bằng với lớp thiểu số).

- Thực nghiệm mô hình với kỹ thuật Oversampling(Làm giàu dữ liệu của lớp thiểu số bằng cách tạo thêm các mẫu dữ liệu tổng hợp).

- Thực nghiệm mô hình với kỹ thuật SMOTETomek (Kết hợp kỹ thuật Undersampling và Oversampling).

### Kết quả thực nghiệm

#### Thực nghiệm khi chưa chuẩn hóa với mô hình RandomForest và sau khi chuẩn hóa

-Kết quả mô hình RF trước khi chuẩn hóa:

Bảng 3. 3: Kết quả mô hình RF trước khi chuẩn hóa

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| 0 | 0.94 | 0.96 | 0.95 | 52015 |
| 1 | 0.80 | 0.70 | 0.74 | 11228 |
| Accuracy |  |  | 0.91 | 63243 |
| Macro avg | 0.87 | 0.83 | 0.85 | 63243 |
| Weighted avg | 0.91 | 0.91 | 0.91 | 63243 |

-Kết quả của mô hình sau khi đã được chuẩn hóa

Bảng 3. 4: -Kết quả của mô hình sau khi đã được chuẩn hóa

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| 0 | 0.94 | 0.96 | 0.95 | 52015 |
| 1 | 0.80 | 0.69 | 0.74 | 11228 |
| Accuracy |  |  | 0.91 | 63243 |
| Macro avg | 0.87 | 0.83 | 0.85 | 63243 |
| Weighted avg | 0.91 | 0.91 | 0.91 | 63243 |

Nhận xét : Việc sử dụng MinMax Scaling không cải thiện đáng kể hiệu suất của mô hình. Điều này cho thấy rằng đối với mô hình cụ thể này và dữ liệu cụ thể này, chuẩn hóa dữ liệu không có tác động lớn.

#### Thực nghiệm mô hình khi xử lý cân bằng

-Kết quả thực nghiệm mô hình RF khi xử lý bằng kỹ thuật Undersampling:

Bảng 3. 5: Kết quả thực nghiệm mô hình RF khi xử lý bằng kỹ thuật Undersampling:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Pecision | Recall | F1-Score | Support |
| 0 | 0.96 | 0.71 | 0.82 | 52015 |
| 1 | 0.39 | 0.87 | 0.54 | 11228 |
| Accuracy |  |  | 0.74 | 63243 |
| Macro avg | 0.68 | 0.79 | 0.68 | 63243 |
| Weighted avg | 0.86 | 0.74 | 0.77 | 63243 |

- Kết quả thực nghiệm mô hình RF khi xử lý bằng kỹ thuật Oversampling:

Bảng 3. 6: Kết quả thực nghiệm mô hình RF khi xử lý bằng kỹ thuật Oversampling

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Pecision | Recall | F1-Score | Support |
| 0 | 0.95 | 0.94 | 0.95 | 52015 |
| 1 | 0.73 | 0.79 | 0.76 | 11228 |
| Accuracy |  |  | 0.91 | 63243 |
| Macro avg | 0.84 | 0.86 | 0.85 | 63243 |
| Weighted avg | 0.91 | 0.91 | 0.91 | 63243 |

- Kết quả thực nghiệm mô hình RF khi xử lý bằng kỹ thuật SMOTETomek:

Bảng 3. 7: Kết quả thực nghiệm mô hình RF khi xử lý bằng kỹ thuật SMOTETomek

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Pecision | Recall | F1-Score | Support |
| 0 | 0.95 | 0.94 | 0.95 | 52015 |
| 1 | 0.73 | 0.79 | 0.76 | 11228 |
| Accuracy |  |  | 0.91 | 63243 |
| Macro avg | 0.84 | 0.86 | 0.85 | 63243 |
| Weighted avg | 0.91 | 0.91 | 0.91 | 63243 |

-Ma trận nhầm lẫn:

Bảng 3. 8: Ma trận nhầm lẫn

|  |  |
| --- | --- |
| Kỹ thuật | Ma trận nhầm lẫn |
| Undersampling | [[36888 15127]  [ 1408 9820]] |
| Oversampling | [[48794 3221]  [ 2380 8848]] |
| SMOTETomek | [[48115 3900]  [ 2058 9170]] |

Nhận xét: ta thấy với kỹ thuật oversampling đã cung cấp sự cân bằng tốt nhất giữa độ chính xác và khả năng học các đặc trưng của mô hình (Bảo toàn thông tin, Không mất mẫu), dẫn đến hiệu suất cao hơn so với các kỹ thuật khác. Do vậy ta sẽ dùng kỹ kết quả của kỹ thuật này để làm thực nghiệm lần lượt với 3 mô hình RF, LR, DC.

#### Thực nghiệm mô hình với 3 thuật toán RF, LR, DC

-Mô hình DC:

Bảng 3. 9: Mô hình DC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Pecision | Recall | F1-Score | Support |
| 0 | 0.94 | 0.94 | 0.94 | 52015 |
| 1 | 0.72 | 0.71 | 0.71 | 11228 |
| Accuracy |  |  | 0.90 | 63243 |
| Macro avg | 0.83 | 0.82 | 0.83 | 63243 |
| Weighted avg | 0.90 | 0.90 | 0.90 | 63243 |

-Mô hình LR:

Bảng 3. 10: Mô hình LR

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Pecision | Recall | F1-Score | Support |
| 0 | 0.99 | 0.82 | 0.90 | 52015 |
| 1 | 0.54 | 0.97 | 0.69 | 11228 |
| Accuracy |  |  | 0.85 | 63243 |
| Macro avg | 0.76 | 0.89 | 0.79 | 63243 |
| Weighted avg | 0.91 | 0.85 | 0.86 | 63243 |

-Mô hình RF:

Bảng 3. 11: Mô hình RF

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Pecision | Recall | F1-Score | Support |
| 0 | 0.95 | 0.94 | 0.95 | 52015 |
| 1 | 0.73 | 0.79 | 0.76 | 11228 |
| Accuracy |  |  | 0.91 | 63243 |
| Macro avg | 0.84 | 0.86 | 0.85 | 63243 |
| Weighted avg | 0.91 | 0.91 | 0.91 | 63243 |

-Ma trận nhầm lẫn:

Bảng 3. 12: Ma trận nhầm lẫn

|  |  |
| --- | --- |
| Model | Ma trận nhầm lẫn |
| RF | [[48808 3207]  [ 2413 8815]] |
| LR | [[42633 9382]  [ 360 10868]] |
| DC | [[48864 3151]  [ 3321 7907]] |

Nhận xét: Dựa trên các thông số đánh giá như accuracy, precision, recall và f1-score, mô hình tốt nhất trong ba mô hình đã đề cập là **Random Forest**. Nó có độ chính xác cao nhất (91.1%).

### Thực nghiệm mô hình tinh chỉnh tham số cho từng mô hình

-Mô hình DC:

Bảng 3. 13: Mô hình DC tinh chỉnh

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Pecision | Recall | F1-Score | Support |
| 0 | 0.98 | 0.86 | 0.92 | 52015 |
| 1 | 0.59 | 0.92 | 0.72 | 11228 |
| Accuracy |  |  | 0.87 | 63243 |
| Macro avg | 0.79 | 0.89 | 0.82 | 63243 |
| Weighted avg | 0.91 | 0.87 | 0.88 | 63243 |

-Mô hình LR:

Bảng 3. 14: Mô hình LR tinh chỉnh

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Pecision | Recall | F1-Score | Support |
| 0 | 0.99 | 0.82 | 0.90 | 52015 |
| 1 | 0.54 | 0.97 | 0.69 | 11228 |
| Accuracy |  |  | 0.85 | 63243 |
| Macro avg | 0.76 | 0.89 | 0.79 | 63243 |
| Weighted avg | 0.91 | 0.85 | 0.86 | 63243 |
|  |  |  |  |  |

-Mô hình RF:

Bảng 3. 15: Mô hình RF tinh chỉnh

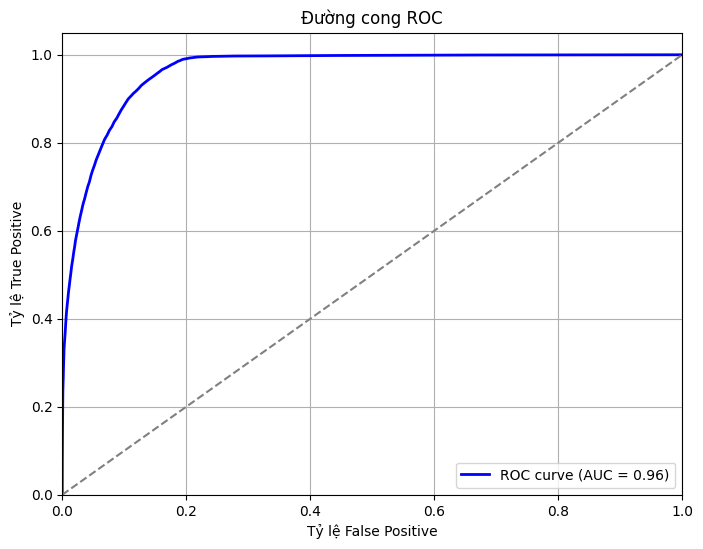
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Pecision | Recall | F1-Score | Support |
| 0 | 0.99 | 0.86 | 0.92 | 52015 |
| 1 | 0.59 | 0.94 | 0.72 | 11228 |
| Accuracy |  |  | 0.87 | 63243 |
| Macro avg | 0.79 | 0.90 | 0.82 | 63243 |
| Weighted avg | 0.92 | 0.87 | 0.88 | 63243 |

-Ma trận nhầm lẫn:

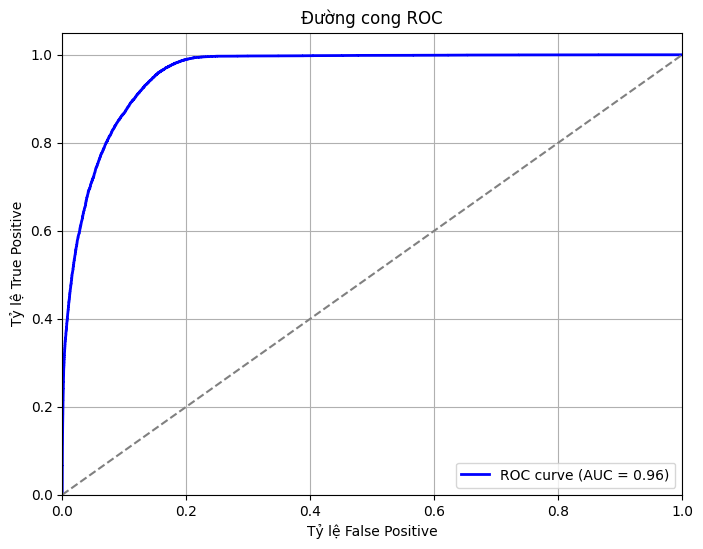
Bảng 3. 16: Ma trận nhầm lẫn

|  |  |
| --- | --- |
| Model | Ma trận nhầm lẫn |
| RF | [[44604 7411]  [ 649 10579]] |
| LR | [[42633 9382]  [ 360 10868]] |
| DC | [[44904 7111]  [ 866 10362]] |

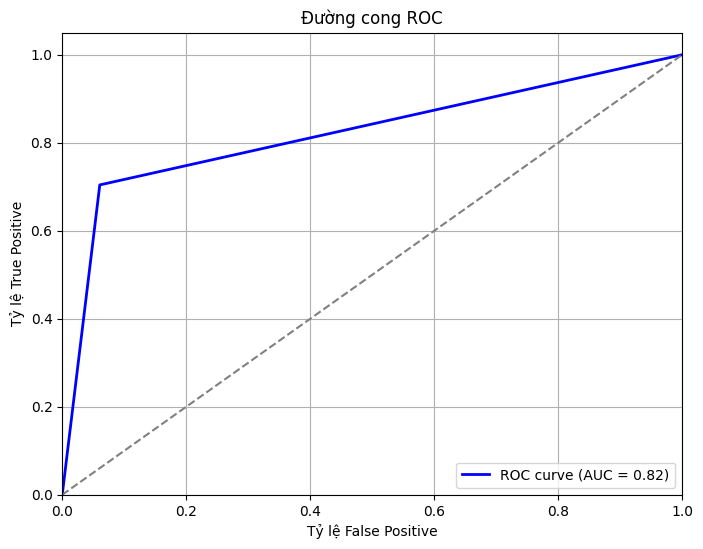
-Lần lượt là biểu đồ của ROC của 3 mô hình RF, DC, LR. Huấn luyện bình thường và với tinh chỉnh siêu tham số



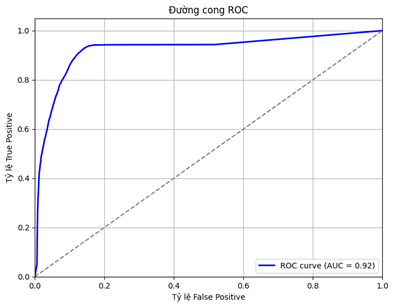
Hình 3. 23: ROC của Random Forest trước tinh chỉnh



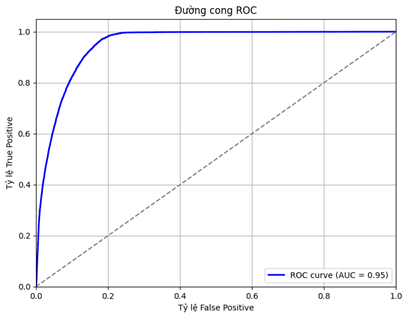
Hình 3. 24: ROC Random Forest sau khi tinh chỉnh



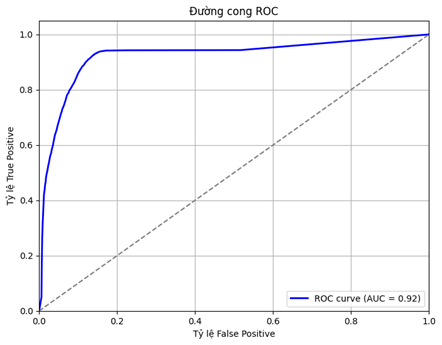
Hình 3. 25: ROC Decision Tree trước tinh chỉnh



Hình 3. 26: ROC Decision Tree sau khi tinh chỉnh



Hình 3. 27: ROC Linear trước tinh chỉnh



Hình 3. 28: ROC mô hình Linear sau khi tinh chỉnh

## Kết chương

Trong chương này, em đã tiến hành những công việc cụ thể để áp dụng các giải thuật Máy học để dự đoán trạng thái khoản vay:

Thứ nhất, em đã tiến hành thu thập được dữ liệu,phân tích và xử lý được để huấn luyện mô hình.

Thứ hai, em đã cài đặt được các mô hình RF, DC, LR cho bài toán.

# XÂY DỰNG ỨNG DỤNG KHAI PHÁ DỮ LIỆU

## Triển khai các chức năng nghiệp vụ

Trang Loan Details bao gồm các chức năng sau:

* Loan Amount (Số tiền vay): Người dùng nhập số tiền mà họ muốn vay, được đo bằng đơn vị đô la ($).
* Term (Kỳ hạn vay): Người dùng nhập số tháng mà họ muốn vay, ví dụ: 36 tháng.
* Interest Rate (Lãi suất): Người dùng nhập lãi suất cho khoản vay, thường là dưới dạng phần trăm. Ví dụ: 0,00% là lãi suất rất thấp.
* Monthly Payment (Thanh toán hàng tháng): Số tiền mỗi tháng phải trả cho khoản vay.
* Grade (Loại): Phân loại của khoản vay, ví dụ như loại A.
* Sub-Grade (Loại phụ): Phân loại cụ thể hơn của khoản vay, ví dụ như A1.
* Loan Payment Amount (Số tiền thanh toán khoản vay): Số tiền cụ thể mà người vay phải thanh toán cho khoản vay.

Trang này giúp người dùng cung cấp thông tin chi tiết về khoản vay, cho phép họ nhập các thông tin liên quan và thực hiện các chức năng như tính toán khoản vay, kỳ hạn và đánh giá lãi suất và cho biết hồ sơ của họ có đủ điều kiện để được duyệt hay không.

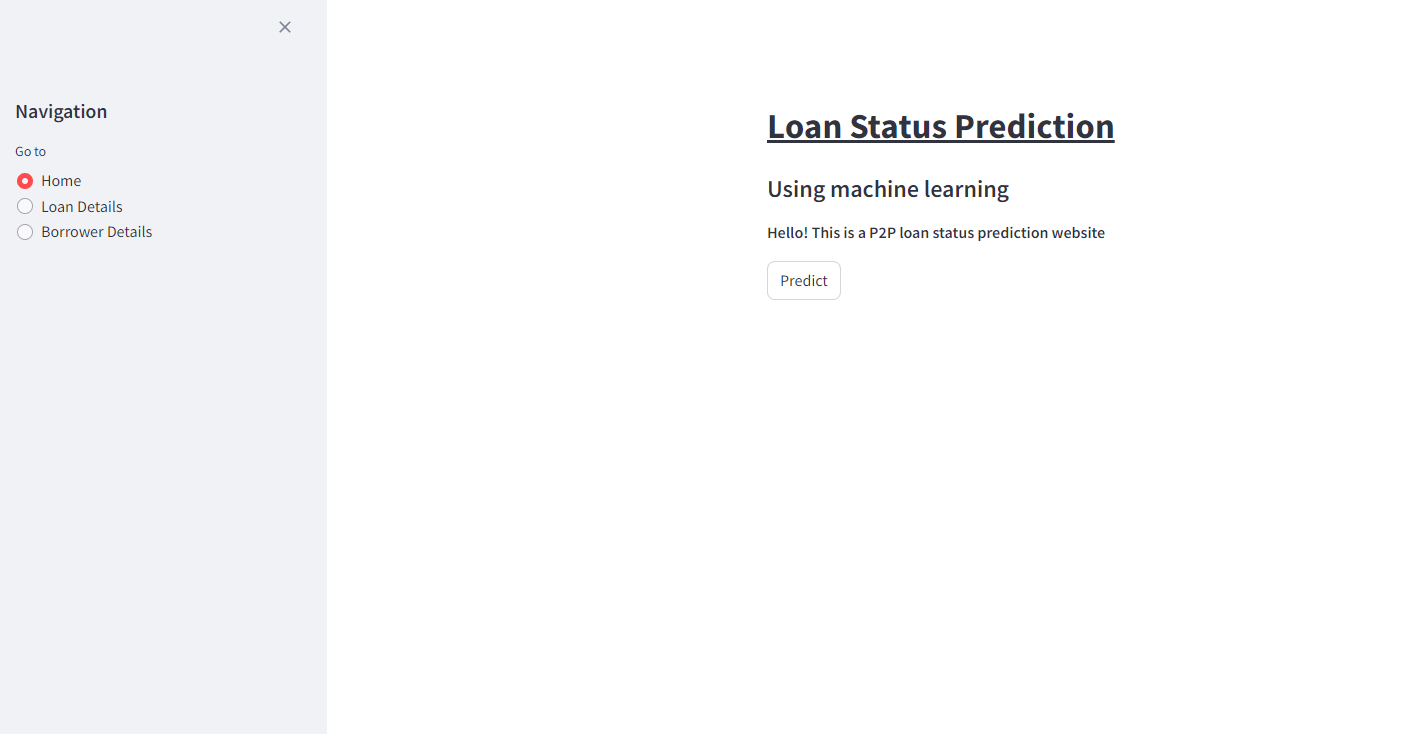
Trang Borrower Details cung cấp các chi tiết về người vay và bao gồm các thông tin sau:

* Employment Length (Thời gian làm việc): Người dùng nhập số năm họ đã làm việc, ví dụ như "<1" cho dưới 1 năm.
* Home Ownership (Sở hữu nhà ở): Người dùng chọn tình trạng sở hữu nhà, ví dụ "ANY" cho bất kỳ loại nào.
* Annual Income (Thu nhập hàng năm): Số tiền thu nhập hàng năm của người vay, được đo bằng đơn vị đô la ($).
* Income Verification Status (Tình trạng xác minh thu nhập): Người dùng chọn tình trạng xác minh thu nhập, ví dụ "Not Verified" cho chưa được xác minh.
* Purpose for the loan (Mục đích vay): Người dùng nhập mục đích vay tiền, ví dụ như "car" cho mua xe hơi.
* Debt-to-Income Ratio (Tỷ lệ nợ thu nhập - DTI): Tỷ lệ giữa tổng số nợ và thu nhập hàng tháng của người vay, tính dưới dạng phần trăm.
* Delinquency count in the past 2 years (Số lần nợ nần trong 2 năm qua): Số lần người vay đã bị trả nợ quá hạn trong 2 năm qua.
* Credit inquiry count in the last 6 months (Số lần kiểm tra tín dụng trong 6 tháng qua): Số lần các tổ chức đã kiểm tra tín dụng của người vay trong 6 tháng qua.
* Open credit account count (Số tài khoản tín dụng đang mở): Số lượng tài khoản tín dụng mà người vay đang sử dụng.
* Derogatory public record count (Số lần vi phạm kết án công khai): Số lần các hành vi vi phạm của người vay đã được công khai.
* Revolving balance (Số dư dòng dài hạn): Số tiền còn lại chưa trả trong các khoản vay quay vòng.
* Revolving credit utilization (%) (Tỉ lệ sử dụng tín dụng quay vòng): Tỷ lệ giữa số tiền đã sử dụng và giới hạn tín dụng quay vòng, tính dưới dạng phần trăm.
* Enter your total number of credit accounts (Nhập tổng số tài khoản tín dụng): Số lượng tổng số tài khoản tín dụng mà người vay có.

Trang này giúp người dùng cung cấp và quản lý các thông tin chi tiết về người vay, các yếu tố quan trọng liên quan đến tài chính và lịch sử tín dụng của họ, từ đó hỗ trợ quyết định về việc duyệt hồ sơ vay tiền.

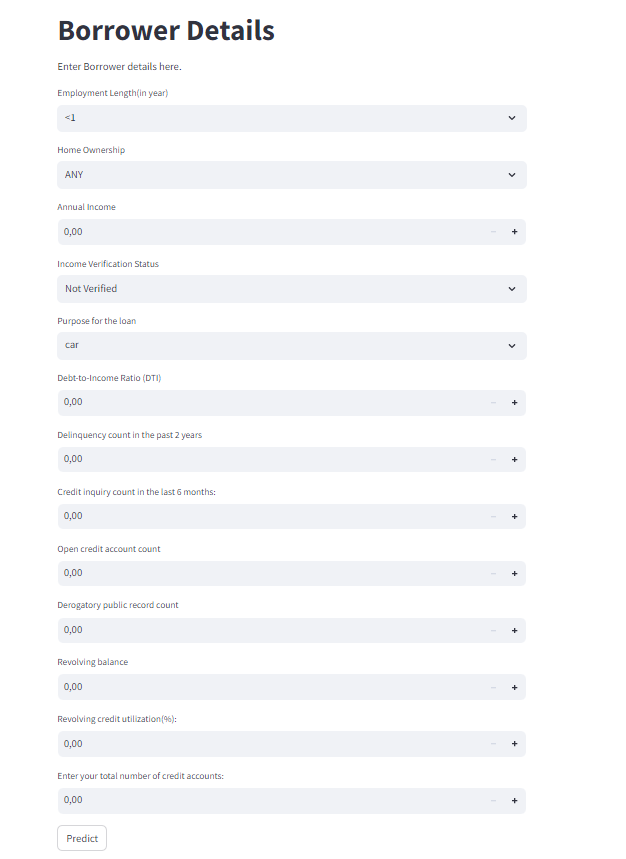
## Kiểm thử đơn vị và triển khai hệ thống

* Trang chủ của hệ thống

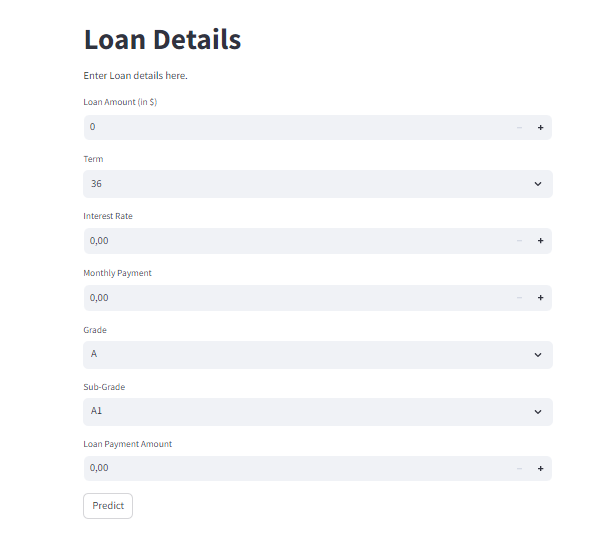


Hình 4. 1: Trang chủ

- Trang Chi tiết Người vay cung cấp thông tin chi tiết về người muốn vay và cho biết hồ sơ của họ có đủ điều kiện để được duyệt hay không.



Hình 4. 2: Trang chi tiết người đi vay

Trang Chi tiết Khoản vay cung cấp dự đoán về rủi ro của khoản vay, giúp người cho vay quyết định liệu họ có nên cấp vốn cho hồ sơ đó hay không.  


Hình 4. 3: Trang chi tiết trạng tháy khoản vay

## Kết chương

# KẾT LUẬN

**Kết quả đạt được**

### Kỹ năng đã học:

**Xử lý dữ liệu:**

Thành thạo các kỹ thuật làm sạch dữ liệu, xử lý giá trị thiếu, và chuẩn bị dữ liệu cho mô hình học máy.

**Tiền xử lý dữ liệu:**

* Áp dụng các phương pháp như mã hóa nhãn (label encoding)

**Hiểu biết về huấn luyện mô hình và thực nghiệm:**

* Biết cách chia dữ liệu thành tập huấn luyện và tập kiểm tra, huấn luyện mô hình học máy và đánh giá hiệu suất của mô hình.

**Chuẩn hóa dữ liệu:**

* Sử dụng các kỹ thuật như Min-Max Scaling để chuẩn hóa dữ liệu, đảm bảo mô hình hoạt động hiệu quả hơn.

### Công việc đã hoàn thành:

* **Xây dựng hệ thống Trang Chi tiết Người vay:** Hệ thống dự đoán liệu hồ sơ của người vay có đủ điều kiện để được duyệt hay không, dựa trên các tiêu chí và mô hình học máy
* **Xây dựng Trang Chi tiết Khoản vay:** Trang này cung cấp dự đoán về rủi ro của khoản vay, giúp người cho vay đưa ra quyết định cấp vốn.

**Hạn chế của đề tài**

 Hạn chế về kiến thức:

* Do hạn chế về kiến thức, mô hình chưa được tối ưu.

 Triển khai mô hình:

* Mới chỉ triển khai được mô hình lên Streamlit, chưa có giải pháp triển khai toàn diện.

 Dự đoán mức tiền vay:

* Chưa dự đoán được mức tiền tối đa có thể cho vay cho mỗi hồ sơ.

**Hướng phát triển**

#### **Nâng cao kiến thức và tối ưu hóa mô hình:**

* **Học thêm về các thuật toán học máy nâng cao**: Tiếp tục học về các thuật toán phức tạp hơn như Gradient Boosting, Random Forest, và Neural Networks để cải thiện hiệu suất của mô hình.
* **Tối ưu hóa tham số (Hyperparameter Tuning)**: Sử dụng các kỹ thuật như Grid Search, Random Search, hoặc Bayesian Optimization để tìm ra bộ tham số tối ưu cho mô hình.
* **Kỹ thuật tiền xử lý nâng cao**: Tìm hiểu và áp dụng các kỹ thuật tiền xử lý như Feature Engineering, xử lý outliers và khắc phục vấn đề imbalance trong dữ liệu (nếu cần).

#### **Triển khai mô hình chuyên nghiệp:**

* **Triển khai trên các nền tảng cloud**: Sử dụng các dịch vụ như AWS, GCP hoặc Azure để triển khai mô hình, đảm bảo khả năng mở rộng và độ tin cậy.
* **Tích hợp CI/CD**: Thiết lập các pipeline CI/CD để tự động hóa quá trình triển khai và cập nhật mô hình.

#### . **Phát triển thêm tính năng:**

* **Dự đoán mức tiền tối đa có thể cho vay**: Xây dựng và tích hợp mô hình dự đoán số tiền tối đa có thể cho vay dựa trên hồ sơ của người vay.
* **Giao diện người dùng (UI) thân thiện**: Cải tiến giao diện Streamlit để dễ sử dụng hơn, bổ sung các biểu đồ và báo cáo chi tiết.
* **Thông báo tự động**: Thiết lập hệ thống thông báo tự động qua email hoặc SMS khi có quyết định duyệt vay hoặc khi có thay đổi về trạng thái hồ sơ.

#### **Phân tích và giám sát mô hình:**

* **Theo dõi hiệu suất mô hình**: Sử dụng các công cụ như TensorBoard, MLflow hoặc các dịch vụ giám sát khác để theo dõi hiệu suất của mô hình theo thời gian.
* **Phân tích lỗi (Error Analysis)**: Thực hiện phân tích lỗi để hiểu rõ nguyên nhân các dự đoán sai và cải thiện mô hình.

#### **Bảo mật và tuân thủ:**

* **Bảo mật dữ liệu**: Đảm bảo rằng dữ liệu người vay được bảo mật và tuân thủ các quy định về bảo vệ dữ liệu cá nhân.
* **Tuân thủ pháp lý**: Đảm bảo rằng hệ thống và quy trình của bạn tuân thủ các quy định pháp luật và quy định tài chính liên quan.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | K. CNTT, Phân tích thiết kế phần mềm, Hưng Yên: Đại học SPKT Hưng Yên, 2009. |
| [2] | K. CNTT, Cơ sở dữ liệu, Hưng Yên: Đại học SPKT Hưng Yên, 2012. |
| [3] | K. CNTT, Lập trình ứng dụng windows forms, Hưng Yên: Đại học SPKT Hưng Yên, 2019. |
| [4] | W3schools, "W3schools," [Online]. Available: https://www.w3schools.com/. |
| [5] | K. CNTT, Kiểm thử phần mềm, Hưng Yên: Đại học SPKT Hưng Yên, 2016. |