

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN CHUYÊN NGÀNH KHOA HỌC DỮ LIỆU

DỰ ĐOÁN KHOẢN VAY

Giảng viên hướng dẫn	:	Th.S ĐẶNG NHƯ PHÚ
Sinh viên thực hiện	:	LÊ VÕ QUỐC HUY
MSSV	:	2000003954
Lớp	:	20DTH1D
Ngành	:	CÔNG NGHỆ THÔNG TIN

Tp HCM, tháng 9 năm 2023

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN CHUYÊN NGÀNH KHOA HỌC DỮ LIỆU

DỰ ĐOÁN KHOẢN VAY

Giảng viên hướng dẫn	:	Th.S ĐẶNG NHƯ PHÚ
Sinh viên thực hiện	:	LÊ VÕ QUỐC HUY
MSSV	:	2000003954
Lớp	:	20DTH1D
Ngành	:	CÔNG NGHỆ THÔNG TIN

Tp HCM, tháng 9 năm 2023

LỜI CẢM ƠN

Trước hết, xin bày tỏ lòng biết ơn và sự tận tâm với Thầy Cô và người hướng dẫn trong suốt quá trình thực hiện đồ án. Đặc biệt, chúng em muốn gửi lời cảm ơn chân thành tới Thầy Th.S Đặng Như Phú (ngành Công Nghệ Thông Tin, chuyên ngành Khoa Học Dữ Liệu, Trường Đại Học Nguyễn Tất Thành) vì sự hỗ trợ, chỉ dẫn, và định hướng nghiên cứu rất tận tâm từ Thầy. Thầy đã giúp chúng em chọn đề tài phù hợp, hướng dẫn chúng trong việc thực hiện và trình bày tiểu luận một cách chi tiết và cẩn thận.

Không chỉ riêng Thầy, mà chúng em còn có lòng biết ơn đối với tất cả Thầy Cô Giáo trong trường Đại học Nguyễn Tất Thành. Những kiến thức mà Thầy Cô đã chia sẻ và truyền đạt giúp chúng em có được nền tảng kiến thức vững chắc, chuẩn bị tốt cho cuộc hành trình trong tương lai.

Cuối cùng, bày tỏ lòng biết ơn sâu sắc tới các bạn cùng khóa K20 đã luôn ủng hộ và khuyến khích cùng nhau trong suốt thời gian học tập. Những người bạn này đã góp phần tạo nên môi trường học tập tích cực.

TP.HCM , ngày 01 tháng 09 năm 2023

Sinh viên

TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH
TRUNG TÂM KHẢO THÍ

KỲ THI KẾT THÚC HỌC PHẦN
HỌC KỲ NĂM HỌC -

PHIẾU CHẤM THI TIỂU LUẬN/BÁO CÁO

Môn thi: đồ án chuyên ngành khoa học dữ liệuLớp học phần: 20DTH1D

Nhóm sinh viên thực hiện :

- 1.Lê Võ Quốc Huy Tham gia đóng góp: 100%.....
- 2.Võ Tuấn Kiệt..... Tham gia đóng góp: 100%.....
- 3..... Tham gia đóng góp:
- 4..... Tham gia đóng góp:
- 5..... Tham gia đóng góp:.....
- 6..... Tham gia đóng góp:.....
- 7..... Tham gia đóng góp:.....
- 8..... Tham gia đóng góp:.....

Ngày thi: 19/09/2023Phòng thi: L.508

Đề tài tiểu luận/báo cáo của sinh viên :

.....

Phản đánh giá của giảng viên (căn cứ trên thang rubrics của môn học):

Tiêu chí (theo CDR HP)	Đánh giá của GV	Điểm tối đa	Điểm đạt được
Cấu trúc của báo cáo		
Nội dung			
- Các nội dung thành phần		
- Lập luận		
- Kết luận		
Trình bày		
TỔNG ĐIỂM			

Giảng viên chấm thi
(ký, ghi rõ họ tên)

LỜI MỞ ĐẦU

Trong bối cảnh mạnh mẽ của cuộc cách mạng công nghiệp 4.0, sự phát triển của khoa học dữ liệu đã mở ra những cánh cửa mới về việc hiểu và tận dụng dữ liệu trong mọi khía cạnh cuộc sống. Trong lĩnh vực kinh doanh, đặc biệt là trong ngành bán lẻ và siêu thị, dữ liệu bán hàng đang trở thành một nguồn tài nguyên quý báu, định hình cách thức doanh nghiệp tương tác với khách hàng và phản ánh xu hướng tiêu dùng.

Đề tài “Dự đoán khoản vay” đặt ra mục tiêu quan trọng là tìm hiểu cách mà khoa học dữ liệu có thể được áp dụng để nắm bắt, chuyển đổi và tận dụng dữ liệu nợ xấu trong ngữ cảnh vay mượn. Phân tích dữ liệu đang trở thành công cụ quyết định quan trọng, giúp doanh nghiệp đào sâu vào những thông tin quan trọng như mức độ ưa thích của khách hàng, sự tương tác với các sản phẩm cụ thể, và thậm chí là dự đoán xu hướng tương lai. Từ những thông tin này, doanh nghiệp có thể điều chỉnh chiến lược kinh doanh, cải thiện trải nghiệm khách hàng và tối ưu hóa hoạt động để đáp ứng nhu cầu thị trường một cách hiệu quả.

Ngoài việc phân tích, việc trực quan hóa dữ liệu cũng đóng một vai trò quan trọng trong việc tạo nên sự thông tin một cách trực quan và dễ hiểu. Thông qua việc biểu đồ hóa, biểu đạt thông tin dưới dạng hình ảnh, doanh nghiệp có thể nhanh chóng nhận biết các mô hình và xu hướng tiềm ẩn trong dữ liệu. Điều này không chỉ giúp cho việc ra quyết định dựa trên dữ liệu trở nên hiệu quả hơn mà còn giúp doanh nghiệp truyền đạt thông tin đến đội ngũ quản lý và nhân viên một cách rõ ràng và nhanh chóng. Trong bối cảnh cạnh tranh khốc liệt của ngành ngân hàng, việc áp dụng khoa học dữ liệu để phân tích và trực quan hóa dữ liệu đối tượng khách hàng ghi nợ không chỉ là một xu hướng mà còn là một yếu tố cần thiết để tồn tại và phát triển. Khả năng hiểu rõ hơn về khách hàng, đáp ứng nhanh chóng các biến đổi thị trường và tối ưu hóa hoạt động kinh doanh là những lợi thế mà công nghệ này mang lại. Qua đề tài này, chúng ta sẽ đi sâu vào thế giới phức tạp của dữ liệu nợ xấu và khám phá cách mà khoa học dữ liệu có thể là chìa khóa đưa doanh nghiệp vượt qua những thách thức và đạt được sự thành công trong tương lai.

MỤC LỤC

CHƯƠNG I: GIỚI THIỆU	1
1. Giới thiệu đề tài	1
2. Lý do chọn đề tài:	1
3. Mục tiêu của đề tài:	1
4. Phương pháp của đề tài	2
5. Đối tượng và phạm vi nghiên cứu:	2
6. Công nghệ áp dụng:	2
CHƯƠNG II: CƠ SỞ LÝ THUYẾT	3
1. Khoa học dữ liệu:	3
1.1 Tổng quan về khoa học dữ liệu.....	3
2. Định nghĩa về khoa học dữ liệu.....	4
2.1 Khoa học dữ liệu là gì?	4
2.2 Ưu điểm & nhược điểm của khoa học dữ liệu.....	4
2.3 ANN là gì.....	6
2.4 Dữ liệu và tiền xử lý dữ liệu là gì	7
2.4.2 Thu thập dữ liệu:	8
CHƯƠNG III: THỰC NGHIỆM	10
1. Mô tả bài toán	10
2. Xây dựng mô hình:	11
CHƯƠNG IV: KẾT LUẬN.....	27
1. Tóm tắt kết quả:	27
2. Kết quả đạt được:	27
3. Hạn chế	27
4. Hướng phát triển.....	28
TÀI LIỆU THAM KHẢO	29

DANH MỤC HÌNH

Hình 1. Phương pháp thu thập dữ liệu	8
Hình 2 : Tải gói scikit learn.....	11
Hình 3: import các thư viện	12
Hình 4 : Đọc dữ liệu	12
Hình 5: Khám phá dữ liệu	13
Hình 6: Biểu đồ heatmap (sơ đồ nhiệt)	14
Hình 7	15
Hình 8	16
Hình 9	17
Hình 10	18
Hình 11	19
Hình 12	20
Hình 13	20
Hình 14	21
Hình 15	22
Hình 16	23
Hình 17	24
Hình 18	24
Hình 19	25
Hình 20	25
Hình 21	26
Hình 22	26
Hình 23	26

CHƯƠNG I: GIỚI THIỆU

1. Giới thiệu đề tài

Dự đoán khoản vay là một vấn đề quan trọng đối với các ngân hàng và công ty tài chính. Bằng cách sử dụng mạng nơ-ron nhân tạo (Artificial Neural Network - ANN) để xây dựng một mô hình dự đoán khả năng vay tiền của khách giúp các tổ chức giảm thiểu rủi ro cho khoản vay.

2. Lý do chọn đề tài:

Tính thực tế:

Một trong những vấn đề quan trọng đối với nhiều tổ chức như ngân hàng, công ty tài chính và các công ty bảo hiểm đang gặp phải là khách hàng thực hiện các khoản vay nhưng không có khả năng hoàn trả đúng thời hạn hoặc là không hoàn trả. Mô hình của chúng em có thể giúp các tổ chức giảm thiểu những rủi ro cho các khoản vay, cải thiện hiệu quả hoạt động và giúp tăng lợi nhuận.

Tính thách thức và độ phức tạp:

Dự đoán người vay không có khả năng trả được khoản vay là một đề tài phức tạp và có tương đối nhiều biến số xảy ra, chẳng hạn như là thu nhập, lịch sử tín dụng các khoản nợ khác,... Nghiên cứu về chủ đề này giúp em hiểu rõ hơn về các yếu tố đã ảnh hưởng đến khả năng trả nợ của người vay.

Độ thú vị và kinh nghiệm rút ra được:

Đây là một đề tài khá sát với thực tế nên là những gì rút ra được trong đề tài này cũng là một kinh nghiệm quý báu và kiến thức thực tế mà chúng em đã trải nghiệm qua.

Dựa vào những lý do trên mà chúng em thấy rằng đề tài “Dự đoán khoản vay” là một đề tài thú vị và có tính thực tế cao. Đây cũng là một cơ hội tốt để chúng em áp dụng các kiến thức đã học để giải quyết các vấn đề trong bài toán.

3. Mục tiêu của đề tài:

Mục tiêu của đề tài "Dự đoán khoản vay" là phát triển một mô hình dự đoán có thể xác định chính xác những người vay có nguy cơ cao không trả được khoản vay. Mô hình này sẽ được sử dụng để giúp các tổ chức tài chính giảm thiểu rủi ro cho khoản vay và cải thiện hiệu quả hoạt động.

4. Phương pháp của đề tài

Để đạt được mục tiêu này, đề tài sẽ thực hiện các bước sau:

Thu thập dữ liệu về người vay, bao gồm lịch sử tín dụng, thu nhập, các khoản nợ hiện tại, mục đích vay, ...

Phân tích dữ liệu để xác định các yếu tố có thể ảnh hưởng đến khả năng trả nợ của người vay.

Đào tạo một mô hình dự đoán dựa trên các yếu tố này.

Đánh giá hiệu quả của mô hình dự đoán.

5. Đối tượng và phạm vi nghiên cứu:

Đối tượng:

Đối tượng nghiên cứu là các bộ dữ liệu liên quan đến thông tin về người vay, bao gồm các yếu tố như thu nhập, số lần vay mượn trước đó, tuổi, công việc, thông tin tài chính.

Phạm vi nghiên cứu:

Phạm vi nghiên cứu sẽ tập trung vào việc phát triển mô hình dự đoán dựa trên dữ liệu đã có, nhằm xác định khả năng trả nợ của người vay.

6. Công nghệ áp dụng:

Mạng nơ-ron nhân tạo (ANN): Sử dụng ANN là một trong những công nghệ quan trọng trong dự đoán khoản vay. ANN có khả năng học từ dữ liệu và tìm ra các mối quan hệ phức tạp giữa các biến để dự đoán khả năng vay tiền của khách hàng. ANN có khả năng xử lý dữ liệu phi tuyến và có thể mô hình hóa các mẫu phức tạp.

Tiền xử lý dữ liệu: Trước khi áp dụng ANN, cần tiền xử lý dữ liệu để chuẩn bị cho quá trình huấn luyện mô hình. Các bước tiền xử lý có thể bao gồm loại bỏ dữ liệu thiếu, mã hóa các biến hạng mục thành biến số, chuẩn hóa dữ liệu và chia thành các tập huấn luyện và kiểm tra.

Machine learning là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc phát triển các mô hình máy tính có thể học hỏi từ dữ liệu. Machine learning sẽ được dùng để phát triển mô hình dự đoán.

Data mining là một lĩnh vực của khoa học dữ liệu liên quan đến việc khám phá và phân tích dữ liệu để tìm ra các mẫu và xu hướng ẩn. Data mining sẽ được dùng để xác định các yếu tố có thể ảnh hưởng đến khả năng trả nợ của người vay

CHƯƠNG II: CƠ SỞ LÝ THUYẾT

1. Khoa học dữ liệu:

1.1 Tổng quan về khoa học dữ liệu

Khoa học dữ liệu là một lĩnh vực nghiên cứu và ứng dụng liên quan đến việc thu thập, xử lý, phân tích và trích xuất thông tin từ dữ liệu để tạo ra hiểu biết và thông tin hữu ích. Khoa học dữ liệu kết hợp các phương pháp, công cụ và thuật toán từ nhiều lĩnh vực như thống kê, máy học, học sâu, khai phá dữ liệu và trí tuệ nhân tạo.

Trong ngành công nghệ hiện đại, khoa học dữ liệu đóng vai trò quan trọng trong nhiều lĩnh vực, bao gồm trí tuệ nhân tạo, hệ thống thông tin, marketing, y tế, tài chính và nhiều lĩnh vực khác.:

Kinh doanh và Tài chính: khoa học dữ liệu được sử dụng để phân tích thị trường, dự đoán biến động giá cả, quản lý rủi ro tài chính và tối ưu hóa chuỗi cung ứng.

Y tế: Trong lĩnh vực y tế, dữ liệu từ bệnh nhân và thử nghiệm y tế giúp phát triển mô hình dự đoán bệnh, quản lý bệnh viện và hiểu rõ hơn về yếu tố di truyền.

Khoa học xã hội: khoa học dữ liệu hỗ trợ phân tích dữ liệu xã hội, dự đoán xu hướng xã hội và tạo mô hình mô phỏng tác động của chính sách xã hội.

Tiếp thị và Quảng cáo: Dữ liệu từ chiến dịch tiếp thị trực tuyến hỗ trợ tối ưu hóa quảng cáo, phân tích phản hồi khách hàng và phát triển chiến lược tiếp thị.

Giao thông và Đô thị thông minh: Dữ liệu từ hệ thống giao thông và cảm biến hỗ trợ tối ưu hóa luồng giao thông, dự đoán tình trạng giao thông và phát triển đô thị thông minh.

Nông nghiệp: khoa học dữ liệu được áp dụng trong quản lý nông nghiệp, từ dự đoán mùa vụ đến tối ưu hóa sử dụng tài nguyên như nước và phân bón.

Khám phá dược phẩm: Trong lĩnh vực nghiên cứu dược phẩm, dữ liệu về cấu trúc phân tử và thử nghiệm dược phẩm giúp phát triển các loại thuốc mới và dự đoán tác dụng phụ..

2. Định nghĩa về khoa học dữ liệu

2.1 Khoa học dữ liệu là gì?

Khoa học dữ liệu là một lĩnh vực nghiên cứu và ứng dụng liên quan đến việc thu thập, xử lý, phân tích và trích xuất thông tin từ dữ liệu để tạo ra hiểu biết và thông tin hữu ích. Nó kết hợp các phương pháp, công cụ và thuật toán từ nhiều lĩnh vực như thống kê, máy học, học sâu, khai phá dữ liệu và trí tuệ nhân tạo.

Mục tiêu chính của khoa học dữ liệu là hiểu và tận dụng giá trị của dữ liệu để giải quyết các vấn đề và đưa ra quyết định thông minh. Khoa học dữ liệu đòi hỏi sự kết hợp giữa kiến thức về ngành và kỹ thuật xử lý dữ liệu để tìm ra mẫu, xu hướng và thông tin hữu ích từ dữ liệu.

Các bước chính trong quá trình khoa học dữ liệu bao gồm thu thập dữ liệu, tiền xử lý dữ liệu, khám phá dữ liệu (EDA), xây dựng mô hình, đánh giá và tinh chỉnh mô hình, và trình bày và truyền đạt kết quả. Qua các bước này, người ta có thể tìm ra các mẫu, quy luật và hành vi ẩn trong dữ liệu để đưa ra dự đoán, phân tích và giải thích.

Khoa học dữ liệu có ứng dụng rộng rãi trong nhiều lĩnh vực như kinh tế, tài chính, y tế, marketing, giao thông vận tải, xử lý ngôn ngữ tự nhiên và nhiều lĩnh vực khác. Nó cũng đóng vai trò quan trọng trong việc phát triển các công nghệ mới như trí tuệ nhân tạo, xe tự lái và Internet of Things (IoT).

2.2 Ưu điểm & nhược điểm của khoa học dữ liệu

Ưu điểm:

Trước tiên thì ưu điểm của khoa học dữ liệu là phân tích thông tin: khoa học dữ liệu giúp phân tích và hiểu rõ thông tin từ các nguồn dữ liệu khác nhau. Nó cho phép phân loại, tổ chức và tìm ra mối liên kết giữa các dữ liệu để tạo ra thông tin hữu ích và nhận thức sâu hơn về các vấn đề cụ thể.

Kế đến chính là dự đoán và tiên đoán: với việc sử dụng các thuật toán học máy và kỹ thuật khai phá dữ liệu, khoa học dữ liệu có khả năng dự đoán và tiên đoán các xu hướng, mô hình và kết quả trong tương lai. Điều này giúp tăng cường khả năng ra quyết định và lập kế hoạch.

Cùng với khả năng tối ưu hóa hoạt động: khoa học dữ liệu có thể giúp tối ưu hóa hoạt động của một tổ chức, từ việc tăng cường hiệu suất làm việc cho đến giảm thiểu chi phí và tối ưu hóa quy trình sản xuất. Bằng cách phân tích dữ liệu, ta có thể tìm ra các cách cải thiện và tối ưu hóa tài nguyên hiện có.

Cuối cùng chính là phát hiện gian lận và rủi ro: khoa học dữ liệu có thể phát hiện các mô hình bất thường trong dữ liệu, giúp xác định các hoạt động gian lận hoặc nguy cơ tiềm ẩn. Điều này đặc biệt quan trọng trong lĩnh vực tài chính, bảo hiểm và an ninh mạng.

Nhược điểm:

Bên cạnh những ưu điểm thì khoa học dữ liệu cũng đối mặt với một số nhược điểm:

Dữ liệu không chính xác hoặc thiếu: Khoa học dữ liệu yêu cầu dữ liệu chính xác và đầy đủ để phân tích. Nếu dữ liệu không chính xác hoặc thiếu sót, kết quả của phân tích có thể bị sai lệch hoặc không chính xác.

Bảo mật và quyền riêng tư: chứa nhiều thông tin cá nhân và nhạy cảm, gây ra rủi ro về bảo mật và vi phạm quyền riêng tư. Việc bảo vệ và tuân thủ các quy định về quyền riêng tư trở nên phức tạp hơn khi xử lý dữ liệu lớn.

Khó khăn trong việc xác định thông tin có giá trị: Với khối lượng lớn dữ liệu, việc tìm ra thông tin có giá trị và ý nghĩa trở nên khó khăn hơn. Đòi hỏi sự kiểm soát kỹ thuật và sự hiểu biết để trích xuất insights chính xác từ dữ liệu lớn.

Quản lý và tổ chức dữ liệu: Dữ liệu lớn cần được tổ chức và quản lý một cách hiệu quả để tránh sự rối loạn và mất mát dữ liệu. Việc xác định và áp dụng các phương pháp phân loại, gắn nhãn và cấu trúc cho dữ liệu lớn là một thách thức. Ngoài ra, việc duy trì tính nhất quán và đồng bộ giữa các nguồn dữ liệu khác nhau cũng đòi hỏi quản lý chặt chẽ.

Khó khăn trong việc tìm kiếm thông tin cần thiết: Với lượng dữ liệu lớn, việc tìm kiếm và truy xuất thông tin cần thiết có thể trở nên khó khăn. Điều này yêu cầu phải

áp dụng các công nghệ và kỹ thuật tìm kiếm thông tin hiệu quả, bao gồm các hệ thống tìm kiếm thông minh và công cụ lọc dữ liệu phù hợp.

Sự không chính xác và nhiễu dữ liệu: Một vấn đề phổ biến khi làm việc với dữ liệu lớn là sự không chính xác và nhiễu dữ liệu. Những lỗi dữ liệu và sự sai sót có thể xuất hiện trong quá trình thu thập, xử lý hoặc nhập khẩu dữ liệu. Điều này đòi hỏi kiểm tra và làm sạch dữ liệu một cách cẩn thận để đảm bảo tính chính xác và đáng tin cậy của thông tin.

Phức tạp trong việc triển khai: Triển khai các hệ thống khoa học dữ liệu phức tạp và đòi hỏi kiến thức chuyên sâu về lĩnh vực này. Điều này có thể gây khó khăn cho những tổ chức không có nguồn lực và khả năng cần thiết để triển khai hiệu quả các giải pháp khoa học.

Tóm lại, khoa học dữ liệu mang lại nhiều ưu điểm về thông tin phong phú, phân tích tiên tiến, dự báo chính xác và tích hợp nguồn dữ liệu. Tuy nhiên, nhược điểm liên quan đến chi phí, bảo mật, khó khăn trong việc tìm kiếm và quản lý dữ liệu, cũng như sự không chính xác và thách thức về khả năng mở rộng cần được xem xét và giải quyết để tận dụng hết tiềm năng của dữ liệu lớn.

2.3 ANN là gì

ANN là viết tắt của Artificial Neural Network (Mạng Nơ-ron Nhân tạo), một phương pháp trong lĩnh vực học máy và khoa học dữ liệu. Nó được lấy cảm hứng từ cấu trúc và hoạt động của các mạng nơ-ron thần kinh trong não người.

Mô hình ANN bao gồm một tập hợp các nơ-ron nhân tạo, được tổ chức thành các lớp liên kết với nhau. Mỗi nơ-ron nhân tạo nhận đầu vào, xử lý thông tin và truyền tiếp giá trị đầu ra cho các nơ-ron khác trong mạng. Sự truyền thông tin giữa các nơ-ron diễn ra qua các trọng số kết nối, và kết quả cuối cùng được tính toán từ các giá trị đầu ra của nơ-ron ở lớp cuối cùng.

Ví dụ cụ thể về ANN có thể là một mô hình để nhận biết chữ viết tay. Trong mô hình này, dữ liệu đầu vào sẽ là hình ảnh của chữ viết tay, và mục tiêu là dự đoán chữ cái tương ứng. Mạng nơ-ron sẽ được huấn luyện với một tập dữ liệu lớn gồm các ví dụ hình ảnh kèm theo nhãn chữ cái tương ứng. Trong quá trình huấn luyện, các trọng số của mạng nơ-ron sẽ được điều chỉnh để giảm sai số giữa kết quả dự

đoán và nhận thực tế. Khi mô hình đã được huấn luyện, nó có thể được sử dụng để dự đoán chữ cái cho các hình ảnh chữ viết tay mới.

Một ví dụ khác về ANN là ứng dụng trong phân loại email. Mạng nơ-ron có thể được huấn luyện để phân loại email thành hai nhóm: "thư rác" và "thư gốc". Dữ liệu đầu vào là nội dung và các thuộc tính của email, và mục tiêu là dự đoán xem email đó thuộc nhóm nào. Qua quá trình huấn luyện, mạng nơ-ron sẽ học cách phân loại email dựa trên các mẫu và đặc điểm của từng nhóm. Sau khi huấn luyện, mạng nơ-ron có thể được sử dụng để phân loại tự động các email mới.

ANN là một công cụ mạnh mẽ trong khoa học dữ liệu, với khả năng học tập và hiểu các mô hình phức tạp từ dữ liệu. Điều này cho phép nó được sử dụng trong nhiều lĩnh vực như phân loại, dự đoán, xử lý ngôn ngữ tự nhiên, thị giác máy tính và nhiều ứng dụng khác.

2.4 Dữ liệu và tiền xử lý dữ liệu là gì

Thu thập dữ liệu là một trong những phần quan trọng nhất trước khi xây dựng mô hình Học máy bởi vì các mô hình của chúng ta được thiết kế tốt đến đâu cũng chẳng là gì, khi máy tính không học được bất cứ điều gì hữu ích từ dữ liệu chưa được xử lý.

Dữ liệu không phải lúc nào cũng hoàn hảo. Khi có nhiều dữ liệu lớn, có thể trong số đó chứa một số nhãn không chính xác, mặc dù vậy có thể giải quyết được, nhưng nếu có một số sai sót trong việc thu thập dữ liệu, chúng ta có thể kết thúc với dữ liệu hoàn toàn là rác.

Dữ liệu rác có nghĩa là tập dữ liệu chứa các giá trị bị thiếu, sai lệch dữ liệu và dữ liệu có tính tương quan cao. Loại dữ liệu này xảy ra sự cố trong khi xây dựng mô hình quyết định (decision model).

2.4.1 Khái niệm về dữ liệu:

Dữ liệu là tập hợp các giá trị của các biến định tính hoặc định lượng về một hoặc nhiều người hoặc đối tượng. Dữ liệu chỉ đơn giản là các đơn vị thông tin. Dữ liệu được đo lường, thu thập, báo cáo, phân tích và sử dụng để tạo dữ liệu hình ảnh hóa chẳng hạn như biểu đồ, bảng hoặc hình ảnh.

Dữ liệu đề cập đến các quan sát có thể đo lường được.

Định lượng - dựa trên các con số - 56% thanh niên 18 tuổi uống rượu ít nhất bốn lần một tuần - không biết tại sao, khi nào, như thế nào.

Dữ liệu định tính là dữ liệu có thể được sắp xếp thành các danh mục dựa trên đặc điểm thể chất, giới tính, màu sắc hoặc bất kỳ thứ gì không có số liên quan đến nó.

Định tính - liên quan đến nhiều chi tiết hơn cho biết tại sao, khi nào và như thế nào!

2.4.2 Thu thập dữ liệu:

Nhiệm vụ thu thập dữ liệu bắt đầu sau khi một vấn đề nghiên cứu đã được xác định. Đó là quá trình nhà nghiên cứu thu thập thông tin cần thiết để giải đáp vấn đề nghiên cứu.

Mục đích của thu thập dữ liệu:

Để có được thông tin

- Để lưu giữ hồ sơ
- Để đưa ra các quyết định các vấn đề quan trọng
- Truyền thông tin cho người khác

Có 2 phương pháp thu thập dữ liệu cơ bản:

- Dữ liệu chính: Dữ liệu chính là dữ liệu được thu thập lần đầu tiên và có tính chất nguyên gốc.
- Dữ liệu phụ: Dữ liệu thứ cấp là những dữ liệu đã được thu thập bởi người khác.

Dữ liệu chính	Dữ liệu phụ
<ul style="list-style-type: none">- Thời gian thực- Các nguồn bảo đảm- Có thể trả lời các câu hỏi- Giá trị và thời gian- Tránh thành kiến- Linh hoạt hơn	<ul style="list-style-type: none">- Dữ liệu cũ- Các nguồn không chắc chắn- Điều chỉnh các vấn đề nghiên cứu- Ít giá trị và thời gian- Không thể loại trừ thành kiến- Ít linh hoạt

Hình 1. Phương pháp thu thập dữ liệu

Thu thập dữ liệu chính thông qua: quan sát, khảo sát, phỏng vấn, bảng câu hỏi, lập lịch

- Phương pháp quan sát là phương pháp theo đó dữ liệu từ thực địa được thu thập với sự trợ giúp của quan sát viên hoặc bằng cách đích thân đến thực địa.
- Phương pháp khảo sát là phương pháp mà nhà nghiên cứu có thể thu thập thông tin bằng cách quan sát hoặc đặt các câu hỏi. 'Khảo sát' là một kỹ thuật thu thập thông tin bằng cách đặt câu hỏi cho những cá nhân là đối tượng nghiên cứu thuộc về một mẫu đại diện, thông qua quy trình đặt câu hỏi hoặc tiêu chuẩn hóa, với mục đích nghiên cứu mối quan hệ giữa các biến và/hoặc thu thập thông tin có thể mô tả toàn bộ quần thể.

- Phương pháp phỏng vấn thu thập dữ liệu liên quan đến việc trình bày bằng lời nói và trả lời dưới dạng phản hồi bằng lời nói. Các câu hỏi được hỏi trực tiếp cho người trả lời. Người phỏng vấn đặt câu hỏi cho người trả lời. (nhằm mục đích lấy thông tin cần thiết cho nghiên cứu).
- Bảng câu hỏi dùng để chỉ một công cụ thu thập dữ liệu, thường ở dạng viết, bao gồm các câu hỏi mở/đóng và các câu hỏi khác yêu cầu đối tượng trả lời. Bảng câu hỏi được gửi (qua đường bưu điện hoặc qua mail) đến những người có liên quan với yêu cầu trả lời các câu hỏi và gửi lại Bảng câu hỏi. Bảng câu hỏi bao gồm một số câu hỏi được in theo thứ tự xác định trên một biểu mẫu.
- Lập lịch rất giống với phương pháp Bảng câu hỏi sự khác biệt chính là một lịch trình được điền bởi điều tra viên, người được chỉ định đặc biệt cho mục đích này. Điều tra viên đến gặp người trả lời, hỏi họ các câu hỏi từ Bảng câu hỏi theo thứ tự được liệt kê và ghi lại các câu trả lời vào khoảng trống được cung cấp. Điều tra viên phải được đào tạo về quản lý lịch trình.

2.4.3 Tiền xử lý dữ liệu:

Tiền xử lý dữ liệu là một bước rất quan trọng trong việc giải quyết bất kỳ vấn đề nào trong lĩnh vực Học Máy. Hầu hết các bộ dữ liệu được sử dụng trong các vấn đề liên quan đến Học Máy cần được xử lý, làm sạch và biến đổi trước khi một thuật toán Học Máy có thể được huấn luyện trên những bộ dữ liệu này. Các kỹ thuật tiền xử lý dữ liệu phổ biến hiện nay bao gồm: xử lý dữ liệu bị khuyết (missing data), mã hóa các biến nhóm (encoding categorical variables), chuẩn hóa dữ liệu (standardizing data), co giãn dữ liệu (scaling data),... Những kỹ thuật này tương đối dễ hiểu nhưng sẽ có nhiều vấn đề phát sinh khi chúng ta áp dụng vào các dữ liệu thực tế. Bởi lẽ các bộ dữ liệu ứng với các bài toán trong thực tế rất khác nhau và mỗi bài toán thì đối mặt với những thách thức khác nhau về mặt dữ liệu. Trong bài viết này, chúng ta sẽ cùng nhau tìm hiểu về các kỹ thuật tiền xử lý dữ liệu và cách áp dụng chúng trong các bài toán thực tế.

Tiền xử lý dữ liệu (data preprocessing) là quá trình chuẩn bị, kiểm tra và biến đổi dữ liệu ban đầu thành một dạng thích hợp để sử dụng trong các ứng dụng phân tích dữ liệu hoặc học máy. Tiền xử lý dữ liệu có thể bao gồm nhiều bước khác nhau như loại bỏ giá trị ngoại lai, chuẩn hóa dữ liệu, rút trích đặc trưng, giảm chiều dữ liệu... Mục đích của tiền xử lý dữ liệu là cải thiện chất lượng dữ liệu để đạt được kết quả phân tích tốt hơn, giảm thiểu sai sót và tối ưu hoá hiệu suất của thuật toán phân tích hoặc học máy.

CHƯƠNG III: THỰC NGHIỆM

1. Mô tả bài toán

Tiền xử lý dữ liệu: Thu thập dữ liệu liên quan đến các khách hàng đã vay trong quá khứ, bao gồm các biến số như thu nhập, tuổi, số lượng tài sản, lịch sử vay nợ trước đó,... Chuẩn bị và tiền xử lý dữ liệu để chuẩn hóa, mã hóa và loại bỏ dữ liệu thiếu, không cần thiết hoặc nhiễu.

Xây dựng mô hình dự đoán:

Xác định kiến trúc mạng nơ-ron: Chọn kiến trúc mạng nơ-ron phù hợp với bài toán dự đoán khoản vay, bao gồm số lượng lớp ẩn, số nơ-ron trong mỗi lớp, và các hàm kích hoạt.

Huấn luyện mô hình: Sử dụng dữ liệu huấn luyện đã được tiền xử lý, huấn luyện mô hình ANN. Quá trình này bao gồm việc tối ưu hóa các trọng số và ngưỡng của mạng nơ-ron thông qua việc giảm thiểu sai số giữa dự đoán và giá trị thực tế.

Đánh giá mô hình: Đánh giá hiệu suất của mô hình ANN bằng cách sử dụng dữ liệu kiểm tra. Các độ đo như độ chính xác, độ phân loại chính xác, độ nhạy và độ đặc trưng của mô hình có thể được sử dụng để đánh giá hiệu suất.

Đánh giá mô hình:

Đánh giá hiệu suất: Xác định độ chính xác và độ tin cậy của mô hình dự đoán khoản vay trên dữ liệu kiểm tra. So sánh kết quả của mô hình với dữ liệu thực tế hoặc các mô hình khác để đánh giá hiệu suất.

Phân tích kết quả: Phân tích kết quả dự đoán khoản vay từ mô hình ANN để hiểu và giải thích quyết định của mô hình, xem xét các biến số quan trọng và tìm hiểu quy luật hoặc mối quan hệ trong dữ liệu.

Triển khai mô hình:

Triển khai mô hình: Áp dụng mô hình ANN vào một môi trường sản xuất thực tế để dự đoán khoản vay cho khách hàng mới.

Kiểm tra và cải thiện: Theo dõi hiệu suất của mô hình trong quá trình triển khai và tiến hành các điều chỉnh cần thiết để cải thiện kết quả dự đoán.

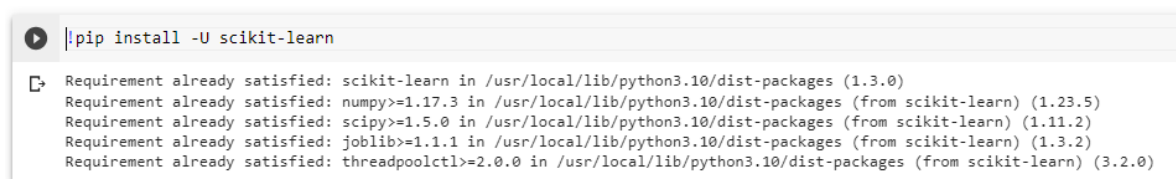
Theo dõi và cải thiện:

Theo dõi và đánh giá: Theo dõi hiệu suất của mô hình dự đoán khoản vay trong thời gian thực. Đánh giá các độ đo hiệu suất và so sánh với tiêu chuẩn hoặc mục tiêu đã đặt ra.

Cải thiện và tối ưu

2. Xây dựng mô hình:

Trước tiên để xây dựng mô hình thực nghiệm thì chúng mình đã sử dụng google colab để tiến hành tải xuống và cài đặt gói “scikit learn”, giúp cho chúng ta có thể sử dụng ANN trong mã Python của mình



```
▶ | pip install -U scikit-learn

Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.3.0)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.23.5)
Requirement already satisfied: scipy>=1.5.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.11.2)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.2.0)
```

Hình 2 : Tải gói scikit learn

import các thư viện và module trong Python để sử dụng trong phân tích dữ liệu và xây dựng mô hình học máy

```
import pandas as pd
import numpy as np
import seaborn as sns
from scipy import stats
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split, RandomizedSearchCV
from sklearn.preprocessing import MinMaxScaler

from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
from sklearn.metrics import auc
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import RocCurveDisplay
from sklearn.metrics import ConfusionMatrixDisplay
```

Hình 3: import các thư viện

Đọc dữ liệu từ tập tin

```
data = pd.read_csv("content/drive/MyDrive/DoinChuyenlganKHU/lending_club_loan_two.csv")
data.head()
```

	loan_amnt	term	int_rate	installment	grade	sub_grade	emp_title	emp_length	home_ownership	annual_inc	...	open_acc	pub_rec	revol_bal	revol_util	total_acc	initial_list_status	application_type	mort_acc	pub_rec_bankruptcies
0	10000.0	36 months	11.44	329.48	B	B4	Marketing	10+ years	RENT	117000.0	...	16.0	0.0	36369.0	41.8	25.0	w	INDIVIDUAL	0.0	0.0 GatewayV
1	8000.0	36 months	11.99	265.68	B	B5	Credit analyst	4 years	MORTGAGE	65000.0	...	17.0	0.0	20131.0	53.3	27.0	f	INDIVIDUAL	3.0	0.0 10761 347vinl
2	15600.0	36 months	10.49	506.97	B	B3	Statistician	< 1 year	RENT	43057.0	...	13.0	0.0	11987.0	92.2	26.0	f	INDIVIDUAL	0.0	0.0 87025 269vinl
3	7200.0	36 months	6.49	220.65	A	A2	Client Advocate	6 years	RENT	54000.0	...	6.0	0.0	5472.0	21.5	13.0	f	INDIVIDUAL	0.0	0.0 FordVnD
4	24375.0	60 months	17.27	609.33	C	C5	Destiny Management Inc.	9 years	MORTGAGE	55000.0	...	13.0	0.0	24584.0	69.8	43.0	f	INDIVIDUAL	1.0	0.0 RoadVnD

5 rows × 27 columns

Hình 4 : Đọc dữ liệu

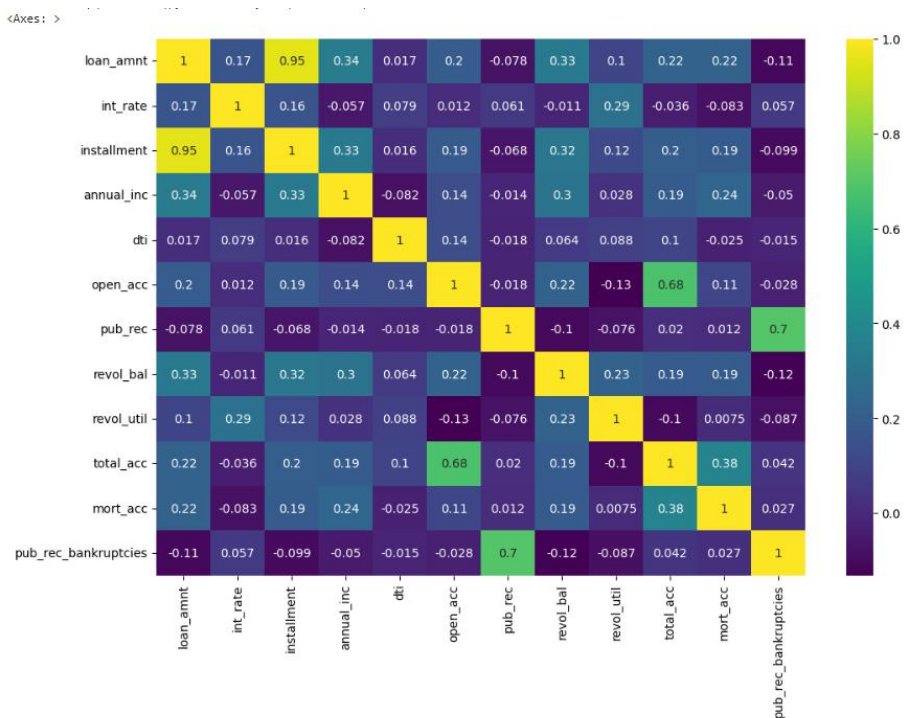
Sau khi đã khai báo các thư viện cần thiết và đọc dữ liệu thì chúng ta bắt đầu phân tích và thăm dò các dữ liệu, khám phá dữ liệu bằng dòng lệnh `data.info()` trước khi tiến hành các phân tích hoặc xây dựng mô hình machine learning

```
✓ 3 play data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 396030 entries, 0 to 396029
Data columns (total 27 columns):
#   Column                Non-Null Count  Dtype
---  -
0   loan_amnt             396030 non-null float64
1   term                  396030 non-null object
2   int_rate              396030 non-null float64
3   installment           396030 non-null float64
4   grade                 396030 non-null object
5   sub_grade             396030 non-null object
6   emp_title             373103 non-null object
7   emp_length            377729 non-null object
8   home_ownership        396030 non-null object
9   annual_inc            396030 non-null float64
10  verification_status    396030 non-null object
11  issue_d               396030 non-null object
12  loan_status           396030 non-null object
13  purpose               396030 non-null object
14  title                 394275 non-null object
15  dti                   396030 non-null float64
16  earliest_cr_line      396030 non-null object
17  open_acc              396030 non-null float64
18  pub_rec               396030 non-null float64
19  revol_bal             396030 non-null float64
20  revol_util            395754 non-null float64
21  total_acc             396030 non-null float64
22  initial_list_status    396030 non-null object
23  application_type       396030 non-null object
24  mort_acc              358235 non-null float64
25  pub_rec_bankruptcies  395495 non-null float64
26  address               396030 non-null object
dtypes: float64(12), object(15)
memory usage: 81.6+ MB
```

Hình 5: Khám phá dữ liệu

tạo ra một biểu đồ heatmap (sơ đồ nhiệt) dựa trên ma trận tương quan của dữ liệu, hiển thị các giá trị số trong ma trận theo cấp độ nhiệt (màu sắc). Các ô có giá trị cao hơn có màu sắc khác so với các ô có giá trị nhỏ hơn, điều này giúp chúng ta dễ dàng nhận ra các mẫu và mối quan hệ trong dữ liệu.



Hình 6: Biểu đồ heatmap (sơ đồ nhiệt)

“`data.groupby(by='loan_status')['loan_amnt'].describe()`” là một phương thức trong pandas để tạo ra một báo cáo thống kê mô tả cho cột "loan_amnt" dựa trên nhóm của cột "loan_status" trong DataFrame "data".

Cụ thể, phương thức này sẽ nhóm các hàng của DataFrame theo giá trị của cột "loan_status". Sau đó, nó sẽ áp dụng phép toán `describe()` lên cột "loan_amnt" trong từng nhóm. Phép toán `describe()` tính toán các thông số thống kê chính như số lượng quan sát (count), giá trị trung bình (mean), độ lệch chuẩn (standard deviation), giá trị tối thiểu (minimum), các phân vị (percentiles) và giá trị tối đa (maximum). Kết quả của câu lệnh này là một báo cáo thống kê mô tả cho cột "loan_amnt", được tổ chức thành từng nhóm dựa trên giá trị của cột "loan_status".

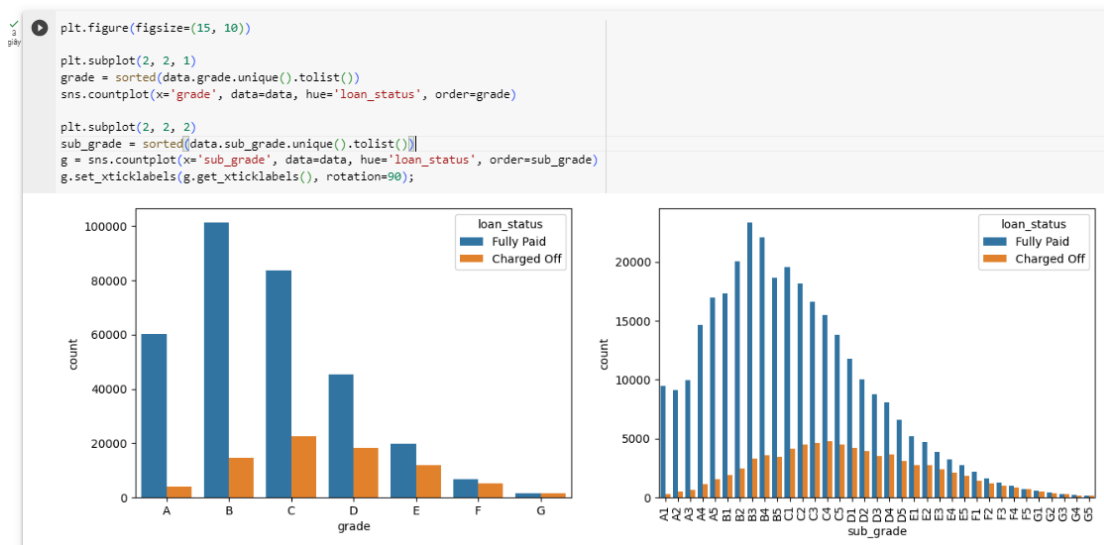
```
data.groupby(by='loan_status')['loan_amnt'].describe()
```

	count	mean	std	min	25%	50%	75%	max
loan_status								
Charged Off	77673.0	15126.300967	8505.090557	1000.0	8525.0	14000.0	20000.0	40000.0
Fully Paid	318357.0	13866.878771	8302.319699	500.0	7500.0	12000.0	19225.0	40000.0

Hình 7

Trong đoạn mã trên, chúng ta sử dụng thư viện `matplotlib.pyplot` để tạo ra một hình vẽ. Dòng đầu tiên `plt.figure(figsize=(15, 10))` xác định kích thước của hình vẽ là 15 inches theo chiều rộng và 10 inches theo chiều cao. Tiếp theo, chúng ta sử dụng hai lệnh `plt.subplot(2, 2, 1)` và `plt.subplot(2, 2, 2)` để tạo ra một lưới gồm hai hàng và hai cột. Lệnh đầu tiên (`plt.subplot(2, 2, 1)`) chỉ ra rằng chúng ta muốn vẽ biểu đồ trong ô (hàng:1,cột:1) của lưới. Lệnh thứ hai (`plt.subplot(2, 2, 2)`) chỉ ra rằng chúng ta muốn vẽ biểu đồ trong ô (hàng:1,cột:2) của lưới.

Ở mỗi ô trong lưới này: Đối với ô (hàng:1,cột:1), chúng ta sắp xếp các giá trị duy nhất của cột 'grade' thành danh sách được sắp xếp (`sorted(data.grade.unique().tolist())`). Sau đó chúng ta sử dụng `sns.countplot()` từ thư viện Seaborn để hiển thị số lượng các mẫu dữ liệu theo từng giá trị của 'grade'. Biểu đồ được tạo ra là một biểu đồ cột (countplot) với trục x là 'grade', trục y là số lượng và các thanh cột được phân loại theo 'loan_status'. Đối với ô (hàng:1,cột:2), chúng ta sắp xếp các giá trị duy nhất của cột 'sub_grade' thành danh sách được sắp xếp (`sorted(data.sub_grade.unique().tolist())`). Sau đó chúng ta sử dụng `sns.countplot()` để hiển thị số lượng các mẫu dữ liệu theo từng giá trị của 'sub_grade'. Biểu đồ này cũng là một biểu đồ cột với trục x là 'sub_grade', trục y là số lượng và các thanh cột được phân loại theo 'loan_status'. Cuối cùng, dòng cuối `g.set_xticklabels(g.get_xticklabels(), rotation=90)` được sử dụng để xoay nhãn của trục x trong biểu đồ thứ hai đi 90°.



Hình 8

``df = data[(data.grade == 'F') | (data.grade == 'G')]`: Tạo một DataFrame mới có tên là ``df`` từ DataFrame gốc ``data``. DataFrame mới chỉ chứa các hàng có giá trị cột "grade" là "F" hoặc "G".

``plt.figure(figsize=(15, 10))``: Tạo một hình vẽ với kích thước 15x10.

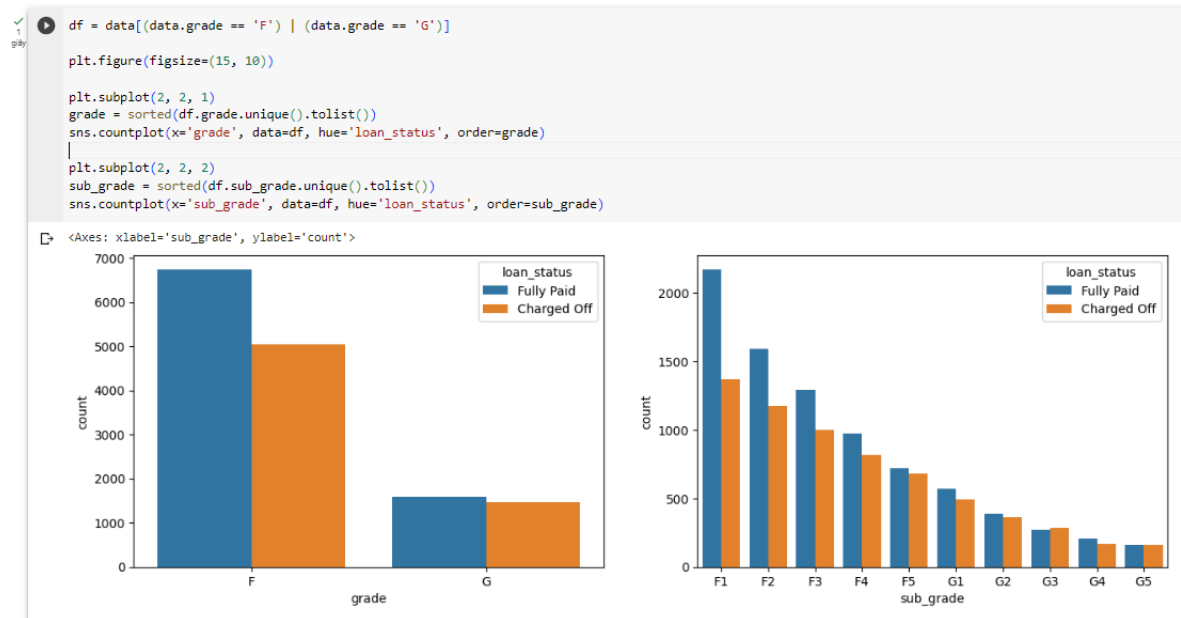
``plt.subplot(2, 2, 1)``: Tạo một subplot trong hình vẽ với lưới 2x2 và đặt nó là subplot thứ nhất.

``grade = sorted(df.grade.unique().tolist())``: Lấy danh sách các giá trị duy nhất trong cột "grade" của DataFrame ``df``, sắp xếp theo thứ tự và lưu vào biến ``grade``.

``sns.countplot(x='grade', data=df, hue='loan_status', order=grade)``: Vẽ biểu đồ đếm số lượng dữ liệu cho từng giá trị của cột "grade" trong DataFrame ``df``. Biểu đồ này được phân loại theo cột "loan_status". Thứ tự hiển thị các giá trị được xác định bởi danh sách đã sắp xếp trong biến ``order``.

Tiếp theo, tương tự như bước 3-5, ta tạo subplot thứ hai và vẽ biểu đồ cho cột "sub_grade" trong DataFrame ``df``.

Tổng cộng, đoạn mã trên tạo ra một hình vẽ gồm hai biểu đồ cột, mỗi biểu đồ hiển thị số lượng dữ liệu cho từng giá trị của cột "grade" và "sub_grade" trong DataFrame `df`. Các biểu đồ này được phân loại theo cột "loan_status".



Hình 9

`data['home_ownership'].value_counts()` là một phương thức trong pandas để đếm số lượng các giá trị duy nhất trong cột 'home_ownership' của DataFrame 'data'. Nó trả về một Series chứa số lần xuất hiện của mỗi giá trị duy nhất.

`data.loc[(data.home_ownership == 'ANY') | (data.home_ownership == 'NONE'), 'home_ownership'] = 'OTHER'` là một câu lệnh để thay đổi các giá trị trong cột 'home_ownership' của DataFrame 'data'. Nếu giá trị là "ANY" hoặc "NONE", nó sẽ được thay bằng "OTHER". Điều này được thực hiện bằng cách sử dụng phương thức `.loc` để chọn các hàng có điều kiện và sau đó gán lại giá trị mới cho cột tương ứng.

`data.home_ownership.value_counts()` được sử dụng sau khi đã thay đổi các giá trị trong cột. Nó tính toán lại số lần xuất hiện của từng giá trị duy nhất trong cột mới và trả về kết quả dưới dạng Series.


```

✓ [11] data['home_ownership'].value_counts()
0 giây
MORTGAGE    198348
RENT        159790
OWN         37746
OTHER        112
NONE         31
ANY          3
Name: home_ownership, dtype: int64

✓ [12] data.loc[(data.home_ownership == 'ANY') | (data.home_ownership == 'NONE'), 'home_ownership'] = 'OTHER'
0 giây
data.home_ownership.value_counts()

MORTGAGE    198348
RENT        159790
OWN         37746
OTHER        146
Name: home_ownership, dtype: int64

```

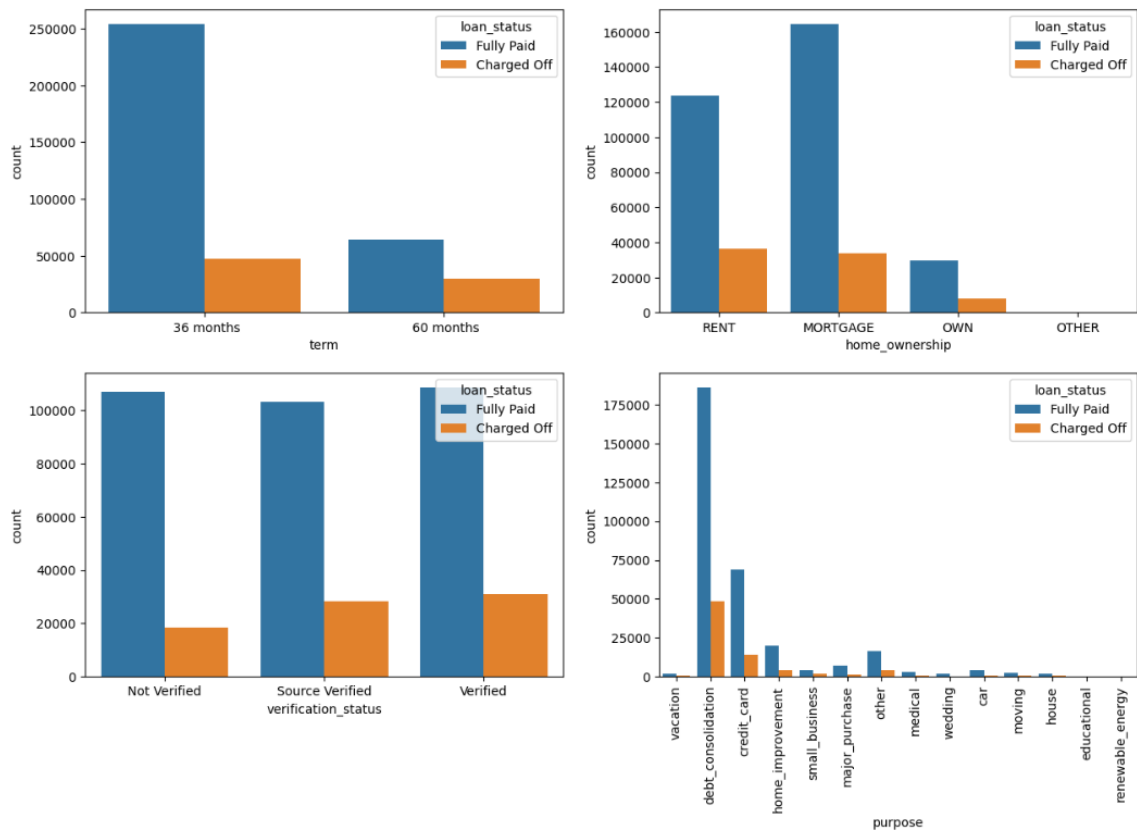
Hình 10

chúng ta sử dụng thư viện `matplotlib.pyplot` và `seaborn` để tạo ra một biểu đồ có nhiều hình con. Dòng đầu tiên `plt.figure(figsize=(15, 20))` xác định kích thước của hình vẽ chính. Trong trường hợp này, kích thước là 15 inch theo chiều ngang và 20 inch theo chiều dọc. Tiếp theo, chúng ta sử dụng các lệnh `plt.subplot()` để tạo ra các hình con trong biểu đồ chính. Có tổng cộng 4 hàng và 2 cột của các hình con. Mỗi lệnh `plt.subplot(4, 2, n)` xác định vị trí của mỗi hình con trong biểu đồ chính. Trong trường hợp này:

Hàng số nằm trong khoảng từ 1 đến 4.

Cột số nằm trong khoảng từ 1 đến 2.

Tham số cuối cùng là chỉ số của mỗi hình con (từ trái qua phải và từ trên xuống dưới). Sau khi đã xác định vị trí cho mỗi subplot, chúng ta sử dụng lệnh `sns.countplot()` để tạo ra biểu đồ cột cho từng thuộc tính được chỉ rõ (`x`) từ dữ liệu (`data`). Thêm vào đó, chúng ta sử dụng tham số `hue` để phân biệt các giá trị của thuộc tính `loan_status`. Cuối cùng, lệnh `g.set_xticklabels(g.get_xticklabels(), rotation=90)` được sử dụng để xoay nhãn trục x của biểu đồ cột cuối cùng (hình con thứ 4) đi 90 độ.



Hình 11

```

0 giây
▶ data.loc[data['home_ownership']=='OTHER', 'loan_status'].value_counts()

Fully Paid      123
Charged Off     23
Name: loan_status, dtype: int64

[15] print((data[data.annual_inc >= 250000].shape[0] / data.shape[0]) * 100)
print((data[data.annual_inc >= 1000000].shape[0] / data.shape[0]) * 100)

1.0294674645860162
0.018937959245511705

[16] data.loc[data.annual_inc >= 1000000, 'loan_status'].value_counts()

Fully Paid      65
Charged Off     10
Name: loan_status, dtype: int64

[17] data.loc[data.annual_inc >= 250000, 'loan_status'].value_counts()

Fully Paid      3509
Charged Off     568
Name: loan_status, dtype: int64

[18] print(data.emp_title.isna().sum())
print(data.emp_title.nunique())

22927
173105

```

Hình 12

```

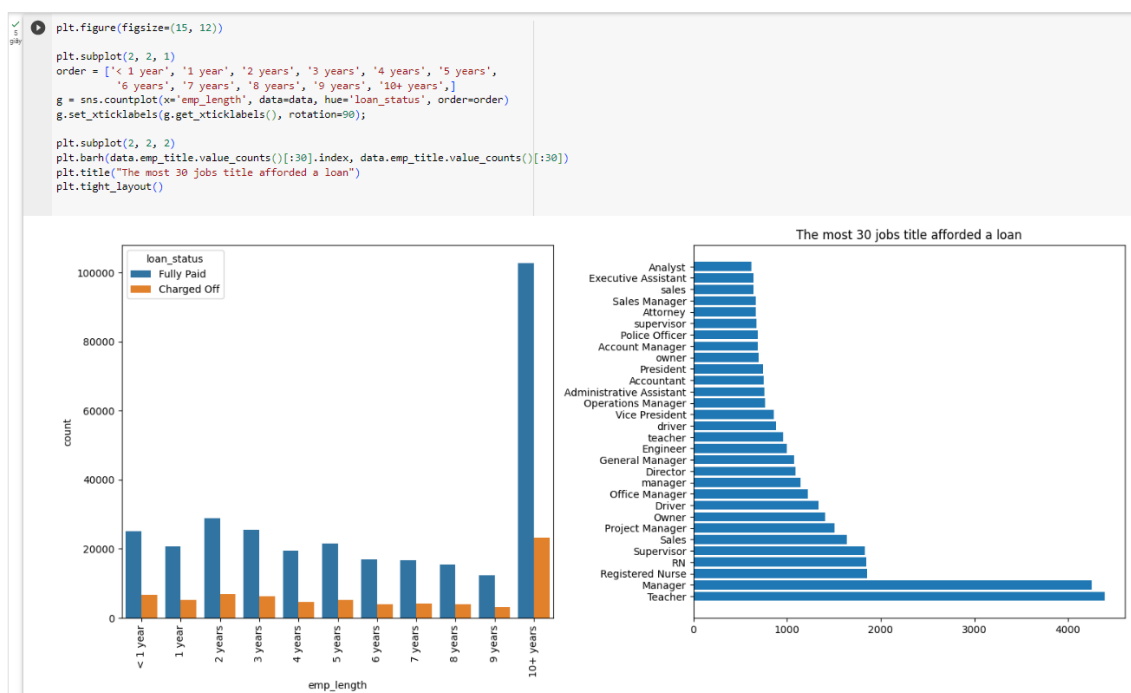
0 giây
▶ data['emp_title'].value_counts()[:20]

Teacher      4389
Manager      4250
Registered Nurse  1856
RN           1846
Supervisor   1830
Sales        1638
Project Manager 1505
Owner        1410
Driver       1339
Office Manager 1218
manager      1145
Director     1089
General Manager 1074
Engineer     995
teacher      962
driver       882
Vice President 857
Operations Manager 763
Administrative Assistant 756
Accountant   748
Name: emp_title, dtype: int64

```

Hình 13

`plt.figure(figsize=(15, 12))` được sử dụng để tạo một hình vẽ mới với kích thước rộng 15 inch và cao 12 inch. Tiếp theo, `plt.subplot(2, 2, 1)` được sử dụng để tạo ra một lưới hình chữ nhật có kích thước là 2 hàng và 2 cột. Trong trường hợp này, subplot đầu tiên (góc trên bên trái) được chọn. Sau đó, biến `order` được khởi tạo là một danh sách các chuỗi biểu diễn thứ tự của thuộc tính `'emp_length'`. Đây là thứ tự mong muốn khi hiển thị các thanh bar trong biểu đồ countplot. Đồng tiếp theo sử dụng seaborn (`'sns.countplot()'`) để vẽ biểu đồ cột (countplot) cho thuộc tính `'emp_length'` từ DataFrame `'data'`. Biểu đồ này hiển thị số lượng khoản vay (`loan_status`) cho từng giá trị của `emp_length`. Tham số `'hue='loan_status'` chỉ ra rằng ta muốn phân loại theo `loan_status`. Dòng cuối cùng (`'g.set_xticklabels(g.get_xticklabels(), rotation=90)'`) được sử dụng để xoay nhãn xác định của các thanh x trong countplot đi góc quay là 90 độ. Tiếp theo, `plt.subplot(2, 2, 2)` được sử dụng để tạo subplot thứ hai (góc trên bên phải). Sau đó, `plt.barh()` được sử dụng để vẽ biểu đồ thanh ngang (barh) cho thuộc tính `'emp_title'` từ DataFrame `'data'`. Biểu đồ này hiển thị số lượng các công việc (`jobs title`) trong danh sách `emp_title`. Tham số `'[:30]'` chỉ ra rằng ta chỉ muốn hiển thị 30 công việc phổ biến nhất. Cuối cùng, `plt.title("The most 30 jobs title afforded a loan...")` được sử dụng để thiết lập tiêu đề cho subplot này.

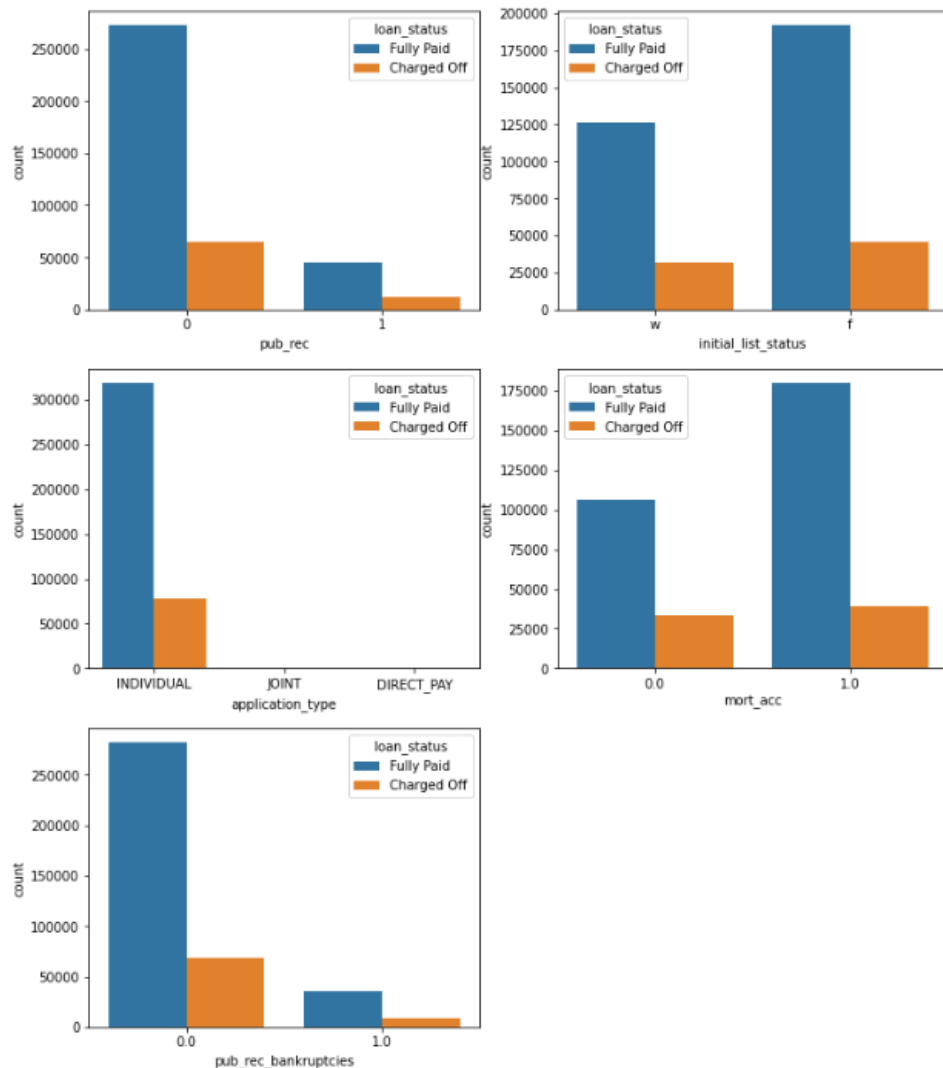


Hình 14

chúng ta sử dụng các hàm `plt.subplot()` để tạo ra các ô con (subplots) trong hình ảnh. Các ô con này được xếp thành lưới có kích thước là 6 hàng và 2 cột.

Mỗi ô con được sử dụng để vẽ biểu đồ countplot bằng cách sử dụng thư viện `seaborn`. Biểu đồ countplot hiển thị số lượng quan sát trong từng nhóm của một biến xác định (`x`) theo từng giá trị của biến khác (`hue`). Trong trường hợp này, chúng ta vẽ các countplot cho các biến `'pub_rec'`, `'initial_list_status'`, `'application_type'`, `'mort_acc'` và `'pub_rec_bankruptcies'`. Các countplot này được phân loại theo giá trị của biến `'loan_status'`.

Với cấu trúc lưới và các subplot như vậy, chúng ta có thể hiển thị nhiều biểu đồ trong cùng một hình ảnh để so sánh và phân tích dữ liệu.



Hình 15

phát hiện và loại bỏ ngoại lệ trong mô hình giúp cải thiện độ chính xác và hiệu suất của mô hình. Ngoại lệ (outliers) là các điểm dữ liệu có giá trị rời rạc hoặc không tuân theo quy luật chung của tập dữ liệu. Chúng có thể gây ảnh hưởng tiêu cực đến quá trình huấn luyện và dự đoán của mô hình.

Bước phát hiện ngoại lệ trong nhằm xác định các điểm dữ liệu không tuân theo quy luật thông qua việc so sánh giá trị thực tế với kết quả được dự đoán bởi mô hình.

Sau khi phát hiện, các ngoại lệ này có thể được loại bỏ hoặc xử lý để không ảnh hưởng tiêu cực tới kết quả cuối cùng

```
[38] print(f"The Length of the data: {data.shape}")
The Length of the data: (396030, 27)

[39] for column in data.columns:
    if data[column].isna().sum() != 0:
        missing = data[column].isna().sum()
        portion = (missing / data.shape[0]) * 100
        print(f"{column}': number of missing values '{missing}' ==> '{portion:.3f}%'"

'emp_title': number of missing values '22927' ==> '5.789%'
'emp_length': number of missing values '18301' ==> '4.621%'
'title': number of missing values '1755' ==> '0.443%'
'revol_util': number of missing values '276' ==> '0.070%'
'mort_acc': number of missing values '37795' ==> '9.543%'
'pub_rec_bankruptcies': number of missing values '535' ==> '0.135%'

[40] data.emp_title.nunique()
173105

[41] data.drop('emp_title', axis=1, inplace=True)

[42] data.emp_length.unique()
array(['10+ years', '4 years', '< 1 year', '6 years', '9 years',
       '2 years', '3 years', '8 years', '7 years', '5 years', '1 year',
       nan], dtype=object)

[43] for year in data.emp_length.unique():
    print(f"{year} years in this position:")
    print(f"{data[data.emp_length == year].loan_status.value_counts(normalize=True)}")
    print('=====')
```

Hình 16

chuẩn hóa dữ liệu cải thiện hiệu suất, khả năng tổng quát hoá và tăng tốc quá trình huấn luyện của mô hình

▼ Chuẩn hóa dữ liệu

```
[75] X_train, y_train = train.drop('loan_status', axis=1), train.loan_status
      X_test, y_test = test.drop('loan_status', axis=1), test.loan_status
```

```
[76] X_train.dtypes
```

```
loan_amnt      float64
term           int64
int_rate       float64
installment    float64
annual_inc     float64
...
zip_code_30723  uint8
zip_code_48052  uint8
zip_code_70466  uint8
zip_code_86630  uint8
zip_code_93700  uint8
Length: 78, dtype: object
```

```
[77] scaler = MinMaxScaler()
      X_train = scaler.fit_transform(X_train)
      X_test = scaler.transform(X_test)
```

Hình 17

Và sau khi xong bước tiền xử lý dữ liệu thì chúng ta cùng nhau xây dựng mô hình ANN

```
def evaluate_nn(true, pred, train=True):
    if train:
        clf_report = pd.DataFrame(classification_report(true, pred, output_dict=True))
        print("Train Result:\n=====")
        print(f"Accuracy Score: {accuracy_score(true, pred) * 100:.2f}%")
        print("-----")
        print(f"CLASSIFICATION REPORT:\n{clf_report}")
        print("-----")
        print(f"Confusion Matrix: \n {confusion_matrix(true, pred)}\n")

    elif train==False:
        clf_report = pd.DataFrame(classification_report(true, pred, output_dict=True))
        print("Test Result:\n=====")
        print(f"Accuracy Score: {accuracy_score(true, pred) * 100:.2f}%")
        print("-----")
        print(f"CLASSIFICATION REPORT:\n{clf_report}")
        print("-----")
        print(f"Confusion Matrix: \n {confusion_matrix(true, pred)}\n")

def plot_learning_evolution(r):
    plt.figure(figsize=(12, 8))

    plt.subplot(2, 2, 1)
    plt.plot(r.history['loss'], label='Loss')
    plt.plot(r.history['val_loss'], label='val_Loss')
    plt.title('Loss evolution during trainig')
    plt.legend()

    plt.subplot(2, 2, 2)
    plt.plot(r.history['AUC'], label='AUC')
    plt.plot(r.history['val_AUC'], label='val_AUC')
    plt.title('AUC score evolution during trainig')
    plt.legend()

def nn_model(num_columns, num_labels, hidden_units, dropout_rates, learning_rate):
    inp = tf.keras.layers.Input(shape=(num_columns, ))
    x = BatchNormalization()(inp)
    x = Dropout(dropout_rates[0])(x)
    for i in range(len(hidden_units)):
        x = Dense(hidden_units[i], activation='relu')(x)
        x = BatchNormalization()(x)
        x = Dropout(dropout_rates[i + 1])(x)
    x = Dense(num_labels, activation='sigmoid')(x)

    model = Model(inputs=inp, outputs=x)
    model.compile(optimizer=Adam(learning_rate), loss='binary_crossentropy', metrics=[AUC(name='AUC')])
    return model
```

Hình 18

```

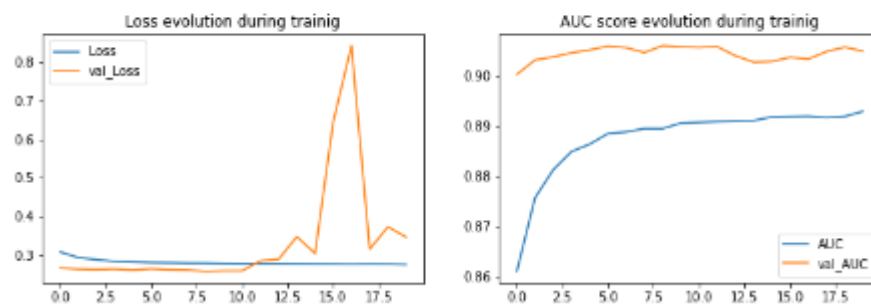
num_columns = X_train.shape[1]
num_labels = 1
hidden_units = [150, 150, 150]
dropout_rates = [0.1, 0, 0.1, 0]
learning_rate = 1e-3

model = nn_model(
    num_columns=num_columns,
    num_labels=num_labels,
    hidden_units=hidden_units,
    dropout_rates=dropout_rates,
    learning_rate=learning_rate
)
r = model.fit(
    X_train, y_train,
    validation_data=(X_test, y_test),
    epochs=20,
    batch_size=32
)

```

Hình 19

```
plot_learning_evolution(r)
```



Hình 20


```
y_train_pred = model.predict(X_train)
evaluate_nn(y_train, y_train_pred.round(), train=True)
```

Train Result:

=====

Accuracy Score: 88.84%

CLASSIFICATION REPORT:

	0.0	1.0	accuracy	macro avg	weighted avg
precision	0.95	0.88	0.89	0.92	0.90
recall	0.46	0.99	0.89	0.73	0.89
f1-score	0.62	0.93	0.89	0.78	0.87
support	51665.00	210478.00	0.89	262143.00	262143.00

Confusion Matrix:

```
[[ 23680 27985]
 [ 1281 209197]]
```

Hình 21

```
y_test_pred = model.predict(X_test)
evaluate_nn(y_test, y_test_pred.round(), train=False)
```

Test Result:

=====

Accuracy Score: 88.87%

CLASSIFICATION REPORT:

	0.0	1.0	accuracy	macro avg	weighted avg
precision	0.94	0.88	0.89	0.91	0.89
recall	0.46	0.99	0.89	0.73	0.89
f1-score	0.62	0.93	0.89	0.78	0.87
support	25480.00	104943.00	0.89	130423.00	130423.00

Confusion Matrix:

```
[[ 11682 13798]
 [   724 104219]]
```

Hình 22

```
:
scores_dict = {
    'ANNs': {
        'Train': roc_auc_score(y_train, model.predict(X_train)),
        'Test': roc_auc_score(y_test, model.predict(X_test)),
    },
}
```

Hình 23

CHƯƠNG IV: KẾT LUẬN

1. Tóm tắt kết quả:

Ta đã thực hiện một quá trình phân tích dữ liệu dự đoán khoản vay một cách chi tiết và cẩn kẽ. Từ việc tiền xử lý dữ liệu đến trực quan hóa.

2. Kết quả đạt được:

Ta đã thực hiện thành công việc tiền xử lý dữ liệu bằng cách loại bỏ dữ liệu không hợp lệ và chuyển đổi các kiểu dữ liệu phù hợp.

Xây dựng mô hình phân tích dựa trên các biểu đồ và trực quan hóa để tìm ra các xu hướng và thông tin quan trọng.

Đã thực hiện trực quan hóa dữ liệu bằng các biểu đồ, giúp hình dung rõ ràng về các khoản nợ.

Áp dụng ANN vào thực tiễn, ta đã tạo ra các kết quả và thông tin hữu ích cho quản lý và ra quyết định.

Kết luận:

Tóm lại, ta đã thể hiện tiềm năng của khoa học dữ liệu trong việc phân tích và trực quan hóa dữ liệu gợi mở cho nhiều hướng phát triển tương lai.

3. Hạn chế

Mặc dù dự án đã đạt được một số kết quả tích cực, còn một số hạn chế cần lưu ý:

Dữ liệu có thể không hoàn toàn đại diện cho mọi tình huống thực tế.

Các mô hình dự đoán có thể bị ảnh hưởng bởi các yếu tố không xác định, và cần được cải thiện và kiểm tra định kỳ.

Phân tích chỉ dựa trên dữ liệu lịch sử, không thể dự đoán một cách chính xác những biến đổi không mong đợi trong tương lai..

Kết luận:

Những hạn chế này cần được xem xét và giải quyết một cách cẩn thận trong quá trình nghiên cứu và thực hiện để đảm bảo tính khả thi và hiệu quả của mô hình.

4. Hướng phát triển

Mở rộng dự án sang các lĩnh vực khác, như dự đoán xu hướng thị trường, dự báo nguồn cung cấp và nhu cầu.

Kết hợp thêm các yếu tố bên ngoài như dữ liệu thời tiết, ngày lễ, sự kiện đặc biệt để cải thiện tính chính xác của mô hình.

Nâng cao mô hình dự đoán bằng cách sử dụng các thuật toán phức tạp hơn và tối ưu hóa tham số.

TÀI LIỆU THAM KHẢO

1. Smith, J. M. (2018). "Data Science for Business." Springer.
2. Zhang, Y., & Wu, Q. (2019). "Exploratory Data Analysis: Concepts and Methods." CRC Press.
3. McKinney, W. (2017). "Python for Data Analysis." O'Reilly Media.
4. Brownlee, J. (2021). "Better Machine Learning: Crash Course." Machine Learning Mastery.
5. Kelleher, J. D., Mac Namee, B., & D'Arcy, A. (2015). "Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies." MIT Press.
6. Raschka, S., & Mirjalili, V. (2019). "Python Machine Learning." Packt Publishing.

Mã nguồn chương trình hoàn chỉnh:

https://colab.research.google.com/drive/1sIr-d1x_rPJhIXewrnEuCttTbynF1w13?usp=sharing