ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN - CƠ - TIN HỌC

Nguyễn Trung Hiếu

RỦI RO TÍN DỤNG VỚI NGÔN NGỮ PYTHON

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Toán Tin
(Chương trình đào tạo chuẩn)

Hà Nội - 2023

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN - CƠ - TIN HỌC

Nguyễn Trung Hiếu

RỦI RO TÍN DỤNG VỚI NGÔN NGỮ PYTHON

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Toán Tin
(Chương trình đào tạo chuẩn)

Cán bộ hướng dẫn: TS. HOÀNG THỊ PHƯƠNG THẢO

Hà Nội - 2023

LỜI CẢM ƠN

Khóa luận này được hoàn thành dưới sự hướng dẫn của giảng viên TS. Hoàng Thị Phương Thảo. Qua đây, em xin bày tỏ lòng biết ơn sâu sắc tới cô đã trực tiếp truyền thụ kiến thức, định hướng đề tài và tận tình hướng dẫn cho em trong suốt quá trình hoàn thành khóa luận.

Em cũng xin chân thành cảm ơn các thầy, cô giáo trong Khoa Toán - Cơ - Tin học, Trường Đại học Khoa học Tự nhiên - Đại học Quốc gia Hà Nội, những người đã giảng dạy, chỉ bảo cho em những bài học và kinh nghiệm quý giá trong quá trình học tập tại trường.

Do giới hạn về thời gian nghiên cứu và kiến thức còn hạn chế nên trong khóa luận của em không tránh khỏi những thiếu sót. Vì vậy, em rất mong nhận được ý kiến đóng góp của quý thầy (cô) và bạn đọc để khóa luận của em được hoàn thiện hơn nữa.

Em xin chân thành cảm ơn!

Hà Nội, ngày 17 tháng 05 năm 2023

Sinh viên

Nguyễn Trung Hiếu

MỤC LỤC

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

CHƯƠNG 1. GIỚI THIỆU

1.1 Tín dụng

Tín dụng là khái niệm thể hiện mối quan hệ giữa người cho vay và người vay, là việc một bên (bên cho vay) cung cấp nguồn tài chính cho đối tượng khác (bên đi vay) trong đó bên đi vay sẽ hoàn trả tài chính cho bên cho vay trong một thời hạn thỏa thuận và thường kèm theo lãi suất.

1.2 Rủi ro tín dụng

Rủi ro tín dụng (credit risk) là người đi vay không trả được nợ với người cho vay khi đến thời hạn thanh toán. Tuy nhiên, định nghĩa này chỉ xem xét trường hợp tiêu cực nhất trong đó người nợ mất khả năng thanh toán. Việc mất giá trị tín dụng cũng có thể xuất phát từ tình hình tài chính của con nợ xấu đi mà không nhất thiết phải trở nên mất khả năng thanh toán. Do đó, một định nghĩa đầy đủ hơn về thuật ngữ rủi ro tín dụng là ảnh hưởng mà sự thay đổi bất ngờ về khả năng trả nợ của con nợ có thể gây ra đối với giá trị tín dụng.

Đánh giá rủi ro tín dụng là quá trình đánh giá rủi ro vỡ nợ đối với khoản vay hoặc sản phẩm tín dụng của người đi vay. Đây là một khía cạnh quan trọng của việc cho vay và đi vay, vì nó giúp người cho vay đưa ra quyết định về việc có cấp tín dụng cho người vay hay không và với những điều khoản nào. Đánh giá rủi ro tín dụng cũng rất quan trọng đối với người đi vay, vì nó quyết định khả năng tiếp cận tín dụng và chi phí đi vay của họ.

Mô hình hóa rủi ro tín dụng là một khía cạnh quan trọng của tài chính và ngân hàng. Với nền kinh tế toàn cầu phức tạp như ngày nay, số vụ vỡ nợ và khủng hoảng tài chính ngày càng tăng. Mô hình rủi ro tín dụng hiệu quả có thể giúp các tổ chức tài chính xác định và quản lý rủi ro, tránh các trường hợp vỡ nợ tốn kém và đưa ra các quyết định cho vay tốt hơn.

1.3 Sự phát triển và tầm quan trọng của mô hình rủi ro tín dụng

Bắt nguồn từ khi người cho vay sử dụng các hệ thống dựa trên quy tắc đơn giản để đánh giá mức độ tin cậy của người đi vay, các hệ thống này dựa trên các yếu tố như thu nhập, lịch sử việc làm và tài sản của người đi vay và thường mang tính chủ quan và không nhất quán. Mặc dù những hệ thống ban đầu này cung cấp mức đánh giá rủi ro cơ bản, nhưng chúng không thể dự đoán chính xác khả năng vỡ nợ của khoản vay,

đặc biệt đối với các sản phẩm tài chính phức tạp hoặc danh mục cho vay lớn.

Trong những năm 1970 và 1980, các ngân hàng và các tổ chức tài chính khác bắt đầu áp dụng các phương pháp định lượng hơn để đánh giá rủi ro tín dụng. Các mô hình đầu tiên được phát triển và sử dụng để dự đoán khả năng vỡ nợ của khoản vay dựa trên dữ liệu thống kê và khoa học thống kê. Các mô hình này đã sử dụng các kỹ thuật thống kê như hồi quy logistic để phân tích dữ liệu lịch sử về hiệu suất cho vay và đánh giá khả năng vỡ nợ. Điều này đánh dấu một sự thay đổi đáng kể trong cách đánh giá rủi ro tín dụng, vì các mô hình này có thể đưa ra đánh giá khách quan và nhất quán hơn về rủi ro.

Vào cuối những năm 1990 và đầu những năm 2000, mô hình rủi ro tín dụng tiếp tục phát triển với sự phát triển của các thuật toán mới và sự sẵn có của các tập dữ liệu lớn hơn và đa dạng hơn. Việc sử dụng ngày càng nhiều ngân hàng kỹ thuật số và sự phát triển của các nền tảng cho vay trực tuyến đã cung cấp quyền truy cập vào lượng dữ liệu khổng lồ về hành vi của người tiêu dùng, có thể được sử dụng để phát triển các mô hình rủi ro tín dụng tinh vi hơn. Đồng thời, ngành tài chính có sự tăng trưởng đáng kể, dẫn đến tăng nhu cầu về các mô hình rủi ro tín dụng chính xác và đáng tin cậy hơn.

Cuộc khủng hoảng tài chính năm 2008 đã nhấn mạnh tầm quan trọng của việc quản lý rủi ro tín dụng hiệu quả và gây chú ý đến những hạn chế của các mô hình hiện có. Cuộc khủng hoảng cho thấy nhiều mô hình được sử dụng dựa trên các giả định phi thực tế về hành vi của người đi vay và sự ổn định của thị trường tài chính, đồng thời không thể dự đoán chính xác khả năng vỡ nợ của khoản vay khi đối mặt với suy thoái kinh tế nghiêm trọng. Điều này dẫn đến việc tăng cường tập trung vào việc phát triển các mô hình mới có thể nắm bắt tốt hơn sự phức tạp của rủi ro tín dụng trong bối cảnh tài chính đang thay đổi nhanh chóng.

Với sự ra đời của thời đại máy tính, mô hình rủi ro tín dụng đã phát triển để kết hợp các mô hình thống kê phức tạp hơn. Các thuật toán như mạng thần kinh nhân tạo, cây quyết định và SVM, đã được phát triển và áp dụng cho mô hình rủi ro tín dụng. Các mô hình này đã sử dụng một lượng lớn dữ liệu để tìm hiểu mối quan hệ giữa các yếu tố khác nhau và khả năng vỡ nợ của khoản vay, đồng thời có thể đưa ra dự đoán với độ chính xác cao hơn các hệ thống dựa trên quy tắc trước đó.

1.4 Mục tiêu khóa luận

Việc xin cấp tín dụng mà không được chấp thuận có thể làm giảm điểm tín dụng của bạn bởi vì các tổ chức tín dụng sẽ xem xét lịch sử của bạn và bao gồm cả các yêu cầu tín dụng bị từ chối. Nếu đã nhiều lần yêu cầu vay tiền mà không được chấp thuận, điều này có thể làm giảm khả năng của bạn để được chấp nhận cho các khoản vay trong tương lai. Mục tiêu của khóa luận này đó là sử dụng bộ dữ liệu của các khách hàng đăng ký cấp thẻ tín dụng, xây dựng các mô hình đánh giá rủi ro vỡ nợ của khách hàng từ đó dự đoán trước khách hàng có được cấp tín dụng không mà không làm ảnh hưởng đến điểm tín dụng. Đối với vấn đề này, em sử dụng ba mô hình: Hồi quy Logistic, Decision Tree, Random Forest, sau đó so sánh hiệu suất và đánh giá các mô hình. Cuối cùng xây dựng một ứng dụng cho phép người dùng điền các thông tin cá nhân và kiểm tra xem đơn xin cấp thẻ tín dụng có được chấp thuận hay không.

CHƯƠNG 2. DỮ LIỆU

Dữ liệu về các bản đăng ký cấp tín dụng được thu thập từ trang Kaggle.com – một trang web chuyên cung cấp các bộ dữ liệu cộng đồng.

https://www.kaggle.com/datasets/laotse/credit-card-approval

Dữ liệu sử dụng trong nghiên cứu gồm 2 tệp csv:

a) application_record.csv: chứa thông tin cá nhân của người đăng ký cấp tín dụng.

b) credit_record.csv: ghi lại hành vi sử dụng tín dụng của người dùng.

Mô tả chi tiết về bộ dữ liệu:

a) application_record.csv: gồm 438558 dòng (tương ứng với khách hàng) và 18 thuộc tính sau:

ID – đánh số khách hàng, sử dụng để kết nối 2 bảng
CODE_GENDER – giới tính (F, M)
FLAG_OWN_CAR – sở hữu ô tô (Y/N)
FLAG_OWN_REALTY – có tài sản (Y/N)
CNT_CHILDREN – số con (No children/ 1 children/ 2+ children)
AMT_INCOME_TOTAL – thu nhập hằng năm
NAME_INCOME_TYPE – loại thu nhập
NAME_EDUCATION_TYPE – trình độ học vấn (Secondary / higher education/ incomplete higher /lower secondary/ Academic degree)
NAME_FAMILY_STATUS – tình trạng hôn nhân (Married/ Single/ Civil Marriage/ Separeted/ Widow)
NAME_HOUSING_TYPE – cách sống (house / with parents / municipa apartment / rented apartment / office apartment)
DAYS_BIRTH – tuổi tính theo ngày, đếm ngược từ ngày hiện tại (0), -1 có nghĩa là ngày hôm qua
DAYS_EMPLOYED – ngày bắt đầu làm việc, đếm ngược từ ngày hiện tại (0). Nếu là 0, nó có nghĩa là người hiện đang thất nghiệp.
FLAG_MOBIL – điện thoại di động
FLAG_WORK_PHONE – điện thoại cơ quan
FLAG_PHONE – điện thoại bàn

FLAG_EMAIL – email
OCCUPATION_TYPE – công việc (laborers/ core staff / sale staff / managers / driver)
CNT_FAM_MEMBERS – số thành viên trong gia đình

b) credit_record.csv: gồm 1048576 dòng và 3 thuộc tính:

ID – đánh số khách hàng, sử dụng để kết nối 2 bảng
MONTH_RECORD – thời gian, tính từ 0
STATUS – trạng thái - 0: Quá hạn 1-29 ngày, 1: Quá hạn 30-59 ngày, 2: Quá hạn 60-89 ngày, 3: Quá hạn 90-119 ngày, 4: Quá hạn 120-149 ngày, 5: Quá hạn hoặc nợ khó đòi, xóa nợ trên 150 ngày, C: đã trả hết tháng đó, X: Không vay tháng nào.

CHƯƠNG 3. LÝ THUYẾT

3.1 Linear Regression

Linear Regression là mô hình sử dụng để tìm mối quan hệ tuyến tính giữa biến dự đoán với biến phản hồi. Mối quan hệ tuyến tính được biểu thị bằng hàm Linear Regression. Hàm Linear Regression định nghĩa như sau:

$$y = w^T X + \varepsilon$$

trong đó y là véc tơ của các biến phản hồi, X là ma trận của các biến giải thích, w là véc tơ của các tham số và $epsilon$ là véc tơ của số hạng sai số.

3.2 Logistic Regression

Logistic Regression là phương pháp thống kê sử dụng cho phân loại nhị phân. Phân tích hồi quy mô hình hóa quan hệ giữa các biến đầu vào với đầu ra nhị phân, thường biểu thị bằng 0 hoặc 1.

Mô hình này có thể giúp ngân hàng xác định khả năng khách hàng có rủi ro vỡ nợ không (biến đầu ra) trên cơ sở sử dụng các nhân tố có ảnh hưởng đến khách hàng (biến đầu vào).

Biến độc lập X_i: gồm tất cả biến còn lại, rời rạc hoặc liên tục. Y đóng vai trò là biến phụ thuộc và là biến nhị phân, chỉ có thể nhận hai giá trị là 0 hoặc 1, cụ thể là:

- $Y = 0$ nếu không vỡ nợ
- $Y = 1$ nếu có vỡ nợ

X_i là biến độc lập, thể hiện các nhân tố ảnh hưởng đến khách hàng ví dụ: thu nhập hàng năm, tỉ lệ nợ trên thu nhập, ...

Hàm sigmoid logistic:

$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$

Khi đó, xác suất một khách hàng bị vỡ nợ (Y = 1) được tính như sau:

$$p(Y = 1|x) = \sigma(w^T x) = \frac{1}{1 + \exp(-w^T x)} = \frac{e^{w^T x}}{e^{w^T x} + 1}$$

Phương pháp ước lượng: Xác suất người đi vay i không trả được nợ là:

$$P_i = \frac{e^{w^T x}}{e^{w^T x} + 1} = \frac{\exp\left(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + ... + \beta_n X_n\right)}{1 + \exp\left(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + ... + \beta_n X_n\right)}$$

Để tính xác suất không trả được nợ của khách hàng, chúng ta tính các giá trị ước lượng $\hat{Y} = w^T x$. Mục tiêu cần ước lượng được hệ số β_0,β_1,...,β_n bằng cách sử dụng ước lượng hợp lý cực đại (maximum likelihood):

Y có phân bố nhị thức do chỉ nhận hai giá trị 0 hoặc 1 vì vậy hàm hợp lý với mẫu kích thước n có dạng như sau:

$$L(\beta) = \prod_{i=1}^{n} p_i^{Y_i}(1-p_i)^{1-Y_i} = \prod_{i=1}^{n}(\frac{\exp X_i\beta}{1+\exp X_i\beta})^{Y_i}(\frac{1}{1+\exp X_i\beta})^{1-Y_i} = \frac{\exp\left(\beta\Sigma_{i=1}^{n} X_i Y_i\right)}{\prod_{i=1}^{n}(1+\exp\left(X_t\beta\right))^2}$$

véc tơ tham số β phù hợp nhất là nghiệm của bài toán tối ưu hàm hợp lý:

$$\beta = \arg\max_{\beta} L(\beta)$$

Chuyển bài toán tối ưu tích về tối ưu tổng bằng cách sử dụng hàm logarit (log likelihood):

$$\beta = \arg\max_{\beta} log(L(\beta))$$

Lập một hệ phương trình với các đạo hàm riêng ứng với các β_i bằng 0. Giải hệ phương trình trên và thu được ước lượng của hệ số β.

3.3 Decision Tree

Cây quyết định là một kiểu mô hình dự đoán, nghĩa là từ các quan sát biến độc lập đưa ra kết luận về biến phụ thuộc trong tập dữ liệu.

Giả sử tập S là tập dữ liệu hiện tại với số phần tử là |S| = N, tập nhãn có C giá trị. Giả sử N_c là điểm thuộc lớp c. ($c = 1, 2, \ldots C$). Xác suất một điểm dữ liệu thuộc một lớp c là N_c/N. Để phân loại một mẫu ta sử dụng khái niệm entropy:

$$H(S) = -\Sigma_{c=1}^{C} \frac{N_c}{N} \log(\frac{N_c}{N})$$

Thuộc tính x được chọn để chia S thành K nút con $S_1, S_2, \ldots, S_K$ với số điểm trong mỗi nút con lần lượt là $m_1, m_2, \ldots, m_K$. Tổng có trọng số entropy của mỗi nút con

tính như sau:

$$H(x, S) = \Sigma_{k=1}^{K} \frac{m_k}{N} H(S_k)$$

Information Gain (IG) đánh giá khả năng của một thuộc tính khi được dùng để phân lớp các mẫu dựa vào số entropy. IG cho biết mức độ giảm của entropy khi phân nhánh mẫu. Định nghĩa information gain:

$$IG(x, S) = H(S) - H(x, S)$$

Ở mỗi nút, thuộc tính được chọn xác định dựa trên:

$$x^* = \arg\max_x IG(x, S) = \arg\min_x H(x, S)$$

Thuộc tính khiến cho information gain đạt giá trị lớn nhất sẽ được chọn để phân mảnh.

Điều kiện dừng của thuật toán:

Nếu tiếp tục phân chia các nút chưa tinh khiết sẽ thu được một cây mà mọi điểm trong tập huấn luyện đều được dự đoán đúng (giả sử rằng không có hai input giống nhau nào cho output khác nhau). Khi đó, cây có thể sẽ rất phức tạp (nhiều nút) khả năng overfitting sẽ xảy ra.

Để tránh overfitting, các phương pháp sau có thể được sử dụng. Tại một node, nếu một trong số các điều kiện sau đây xảy ra, ta không tiếp tục phân chia nút đó và coi nó là một nút lá:

- Nếu nút đó có entropy bằng 0, tức mọi điểm trong nút đều thuộc một lớp.

- Nếu nút đó có số phần tử nhỏ hơn một ngưỡng nào đó. Trong trường hợp này, ta chấp nhận có một số điểm bị phân lớp sai để tránh overfitting.

- Nếu khoảng cách từ nút đó đến nút gốc đạt tới một giá trị nào đó. Việc hạn chế chiều sâu của cây làm giảm độ phức tạp của cây và phần nào giúp tránh overfitting.

- Nếu tổng số nút lá vượt quá một ngưỡng nào đó.

- Nếu việc phân chia nút đó không làm giảm entropy quá nhiều (information gain nhỏ hơn một ngưỡng nào đó).

Khách hàng sẽ được phân nhóm vỡ nợ hoặc không vỡ nợ tuân theo các quy luật phân nhóm được biểu diễn dưới dạng cây quyết định với các nút trong biểu thị một

phép thử trên một thuộc tính và các nhánh mô tả kết quả của phép thử.

3.4 Random Forest

Random Forest là thuật toán có thể được sử dụng cho việc phân lớp. Thuật toán Random Forest sẽ xây dựng nhiều cây quyết định bằng thuật toán Decision Tree, tuy nhiên mỗi cây quyết định sẽ khác nhau (do có tính ngẫu nhiên). Kết quả dự đoán cuối cùng được tổng hợp từ các cây quyết định.

Hoạt động thuật toán Random Forest:

1. Lấy ngẫu nhiên n dữ liệu từ bộ dữ liệu.
2. Chọn ngẫu nhiên k thuộc tính (k < n). Thu được bộ dữ liệu mới gồm n dữ liệu và mỗi dữ liệu có k thuộc tính.
3. Dùng thuật toán Decision Tree để xây dựng cây quyết định với bộ dữ liệu ở bước 2 và nhận kết quả từ mỗi cây dự đoán
4. Bỏ phiếu cho mỗi kết quả dự đoán
5. Chọn kết quả được dự đoán nhiều nhất là dự đoán cuối cùng

Điểm khác biệt giữa cây quyết định và rừng ngẫu nhiên:

1. Overfitting:

Cây quyết định có xu hướng overfit dữ liệu huấn luyện, nghĩa là chúng có thể thu được nhiễu trong dữ liệu và khái quát hóa kém đối với dữ liệu mới. Rừng ngẫu nhiên làm giảm nguy cơ overfitting bằng cách lấy trung bình các dự đoán của nhiều cây và chỉ sử dụng một tập hợp con các thuộc tính.

2. Đánh đổi độ lệch - phương sai:

Độ lệch là sai số trong việc mô hình hóa dữ liệu, tức là sai số giữa giá trị thực tế và giá trị dự đoán được bởi mô hình. Độ lệch thấp có nghĩa là mô hình có khả năng chính xác dự đoán dữ liệu, trong khi độ lệch cao có thể dẫn đến việc mô hình không thể dự đoán chính xác.

Phương sai là sự đo lường độ lệch giữa các giá trị dữ liệu và giá trị trung bình của chúng. Phương sai thể hiện sự độc lập của các biến đầu vào đối với mô hình, tức là sự thay đổi của kết quả dự đoán khi các biến đầu vào thay đổi. Một mô hình có phương sai cao có nghĩa là nó rất nhạy cảm với sự thay đổi của dữ liệu đầu vào, và có thể dẫn đến việc mô hình không thể dự đoán chính xác dữ liệu mới.

Cây quyết định có phương sai cao và độ lệch thấp, trong khi rừng ngẫu nhiên có phương sai thấp và độ lệch cao hơn. Điều này có nghĩa là cây quyết định nhạy cảm với nhiễu trong dữ liệu, trong khi rừng ngẫu nhiên ổn định hơn nhưng có thể bỏ sót một số mẫu tốt.

3. Khả năng diễn giải:

Cây quyết định có khả năng diễn giải cao và có thể dễ dàng hình dung, trong khi rừng ngẫu nhiên phức tạp hơn và khó diễn giải hơn.

4. Tốc độ:

Cây quyết định đào tạo và dự đoán nhanh, trong khi rừng ngẫu nhiên chậm hơn do có nhiều cây.

Cây quyết định đơn giản, nhanh chóng và dễ hiểu hơn nhưng có thể gây overfitting và có phương sai cao. Rừng ngẫu nhiên là kỹ thuật mạnh mẽ, chính xác và ổn định nhưng có thể chậm và khó diễn giải hơn cây quyết định.

3.5 SMOTE

SMOTE (Kỹ thuật lấy mẫu quá mức thiểu số tổng hợp) là kỹ thuật tăng cường dữ liệu phổ biến được sử dụng để giải quyết sự mất cân bằng lớp trong tập dữ liệu. Mất cân bằng lớp xảy ra khi số lượng mẫu trong một lớp nhỏ hơn đáng kể so với số lượng mẫu trong lớp khác, điều này có thể dẫn đến các dự đoán mô hình sai lệch và không chính xác.

Để gia tăng kích thước mẫu bằng kỹ thuật SMOTE, với mỗi một mẫu thuộc nhóm thiểu số ta sẽ lựa chọn ra k mẫu láng giềng gần nhất với nó và sau đó thực hiện tổ hợp tuyến tính để tạo ra mẫu giả lập. Quá trình này được lặp lại cho đến khi tạo ra số lượng mẫu tổng hợp mong muốn.

3.6 Confussion Matrix

Ma trận nhầm lẫn hay confusion matrix là số liệu đánh giá các mô hình phổ biến. Các cột của ma trận tương ứng với các lớp thực tế và các hàng tương ứng với các lớp dự đoán. TN (Phủ định Đúng) là số lượng mẫu âm tính được phân loại chính xác, FN (Phủ định Sai) là số lượng mẫu dương tính được phân loại không chính xác là âm tính, FP (Dương tính sai) là số lượng mẫu âm tính được phân loại không chính xác là dương tính và TP (Dương tính đúng) là số mẫu dương tính được phân loại chính xác.

Accuracy là độ chính xác tổng thể của mô hình và được tính bằng tổng số cách

phân loại đúng chia cho tổng số cách phân loại.

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}$$

Các chỉ số khác có thể được tính toán là tỷ lệ dương tính thực (TPR), còn được gọi là khả năng thu hồi hoặc độ nhạy, cho biết có bao nhiêu kết quả dương tính chính xác xảy ra trong số tất cả các mẫu dương tính.

$$TPR = \frac{TP}{TP + FN}$$

Tỷ lệ dương tính giả (FPR), còn được gọi là tỷ lệ rơi ra, cho biết có bao nhiêu kết quả dương tính không chính xác xảy ra trong số tất cả các mẫu âm tính.

$$FPR = \frac{FP}{TN + FP}$$

3.7 ROC

Đường cong ROC (Receiver Operating Characteristic) là biểu đồ biểu thị khả năng phân loại của một mô hình trong phân loại nhị phân. Đường cong ROC biểu diễn mối quan hệ giữa tỷ lệ dương tính giả (FPR) và tỷ lệ dương tính thực (TPR) của mô hình trên các ngưỡng (thresholds) khác nhau.

Trên đường cong ROC, trục tung biểu thị TPR (còn gọi là sensitivity hoặc recall) - tỷ lệ các điểm dữ liệu thuộc positive class mà được mô hình phân loại đúng. Trục hoành biểu thị FPR - tỷ lệ các điểm dữ liệu thuộc negative class mà bị mô hình phân loại sai thành positive class.

Một mô hình phân loại tốt có đường cong ROC gần với góc trên bên trái của biểu đồ, có TPR cao và FPR thấp ở nhiều ngưỡng khác nhau. Đường cong ROC hoàn toàn nằm trên đường chéo (tức TPR = FPR) cho thấy mô hình không có khả năng phân loại tốt hơn so với dự đoán ngẫu nhiên.

CHƯƠNG 4. THỰC NGHIỆM

4.1 Khám phá dữ liệu

4.1.1 Tổng quan về dữ liệu

Dữ liệu ban đầu gồm 2 bảng application_record.csv và credit_record.csv, hợp nhất với nhau theo ID khách hàng.

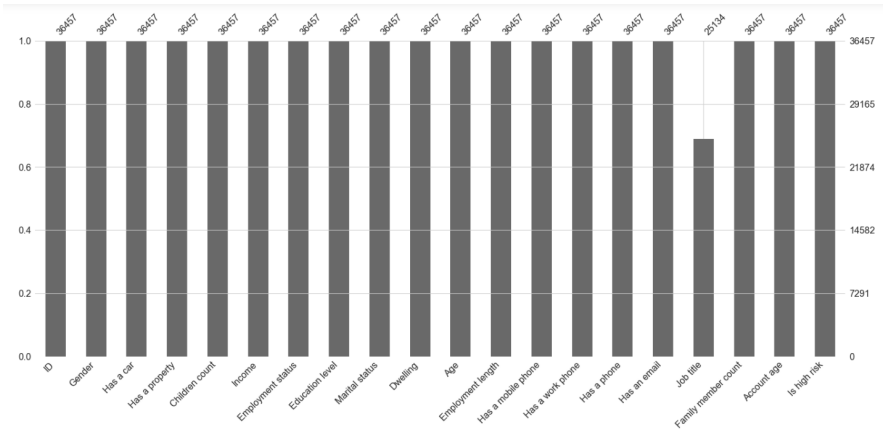
	Column	Non-Null Count	Dtype
0	ID	36457 non-null	int64
1	Gender	36457 non-null	object
2	Has a car	36457 non-null	object
3	Has a property	36457 non-null	object
4	Children count	36457 non-null	int64
5	Income	36457 non-null	float64
6	Employment status	36457 non-null	object
7	Education level	36457 non-null	object
8	Marital status	36457 non-null	object
9	Dwelling	36457 non-null	object
10	Age	36457 non-null	int64
11	Employment length	36457 non-null	int64
12	Has a mobile phone	36457 non-null	int64
13	Has a work phone	36457 non-null	int64
14	Has a phone	36457 non-null	int64
15	Has an email	36457 non-null	int64
16	Job title	25134 non-null	object
17	Family member count	36457 non-null	folat64
18	Account age	36457 non-null	float64
19	Is high risk	36457 non-null	int64

Bảng 4.1: Thông tin về dữ liệu

Thông tin về bảng dữ liệu (Bảng 4.1): Gồm 36457 dòng, 20 cột tương ứng 20 thuộc tính. Các kiểu dữ liệu trong bảng là: float64(3), int64(9), object(8).

Có thể thấy dữ liệu trong Job title đang bị thiếu 25134/36457 non-null. Để kiểm tra một cách trực quan hơn ta sử dụng msno.matrix - một hàm trong thư viện Python missingno, dùng để hiển thị dữ liệu bị thiếu trong tập dữ liệu. msno.matrix tạo một biểu đồ ma trận trong đó mỗi hàng biểu thị một biến trong tập dữ liệu và mỗi cột biểu thị một quan sát. Các ô của ma trận được tô màu tùy theo sự hiện diện hay vắng mặt

của các giá trị bị thiếu trong biến cụ thể đó cho quan sát cụ thể đó.



Hình 4.1

Dựa vào biểu đồ hình 4.1, tất cả các trường đều có đầy đủ dữ liệu ngoại trừ Job title là thiếu nhiều dữ liệu, mất đến gần 30%. Việc mất quá nhiều dữ liệu sẽ ảnh hưởng đến dự đoán vì vậy ta sẽ bỏ đi thuộc tính Job tiltle.

4.1.2 Phân tích đơn biến

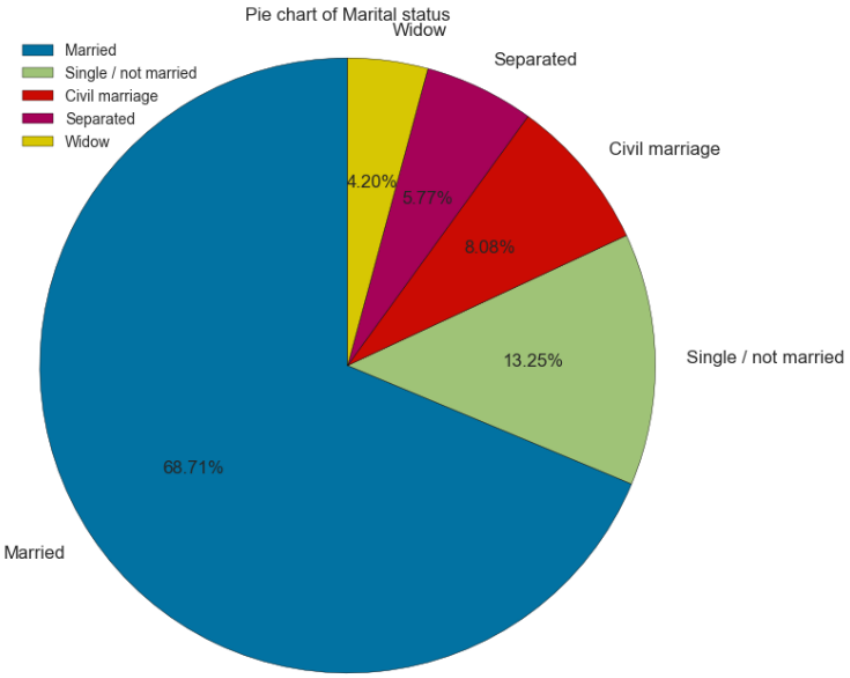
Các biến rời rạc:

- Về giới tính, các khách hàng nữ chiếm tỉ lệ nhiều hơn nam (67% so với 33%) (Hình 4.2)

```
Description:
count     36457
unique        2
top           F
freq      24430
Name: Gender, dtype: object
**************************************************
Object type:
object
**************************************************
Value count:
   Count  Frequency (%)
F  24430      67.010451
M  12027      32.989549
```

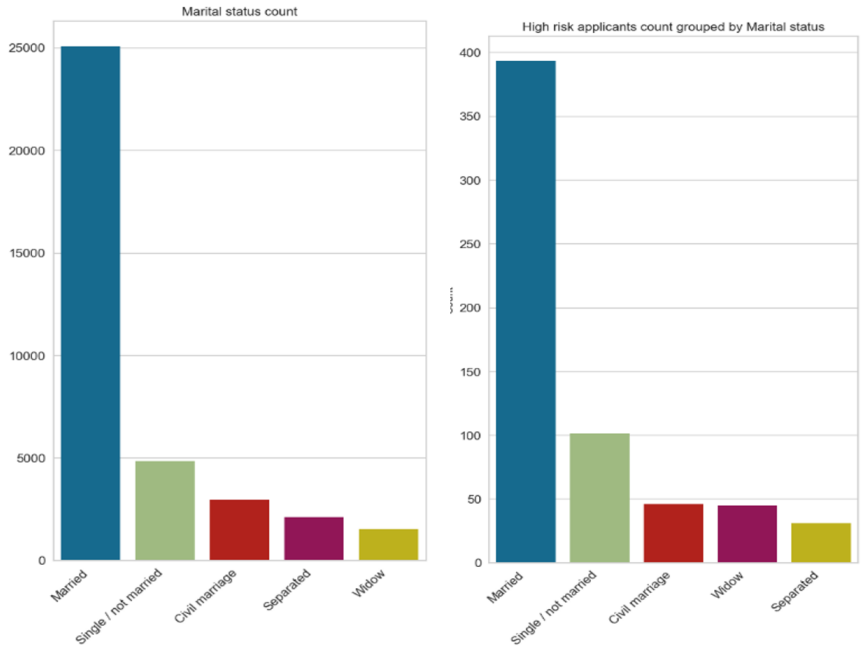
Hình 4.2

- Số lượng khách hàng đã kết hôn chiếm chủ yếu chiếm 68.71%, sau đó tới khách hàng độc thân / chưa kết hôn chiếm 13.25%.



Hình 4.3

- So sánh tình trạng hôn nhân của những người có khả năng vỡ nợ (is high risk) với tất cả tập dữ liệu, nhận thấy không có sự khác biệt nhiều. (Hình 4.4)



Hình 4.4

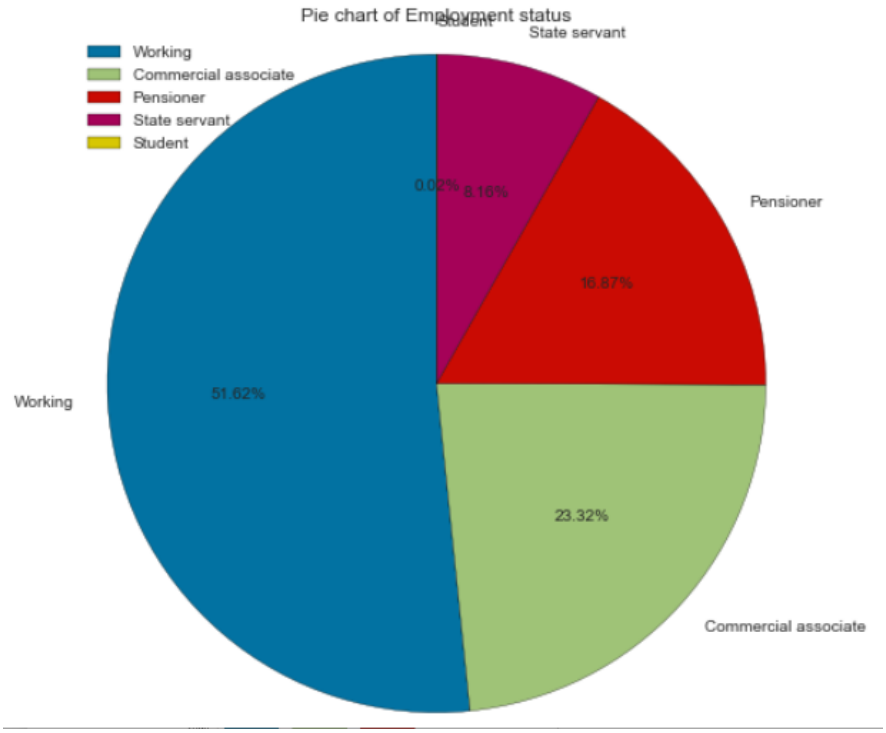
- Về loại nhà ở, loại cư trú chủ yếu nhất đó là Nhà/Căn hộ chiếm đến 89.27%

Value count:

	Count	Frequency (%)
House / apartment	32548	89.277779
With parents	1776	4.871492
Municipal apartment	1128	3.094056
Rented apartment	575	1.577201
Office apartment	262	0.718655
Co-op apartment	168	0.460817

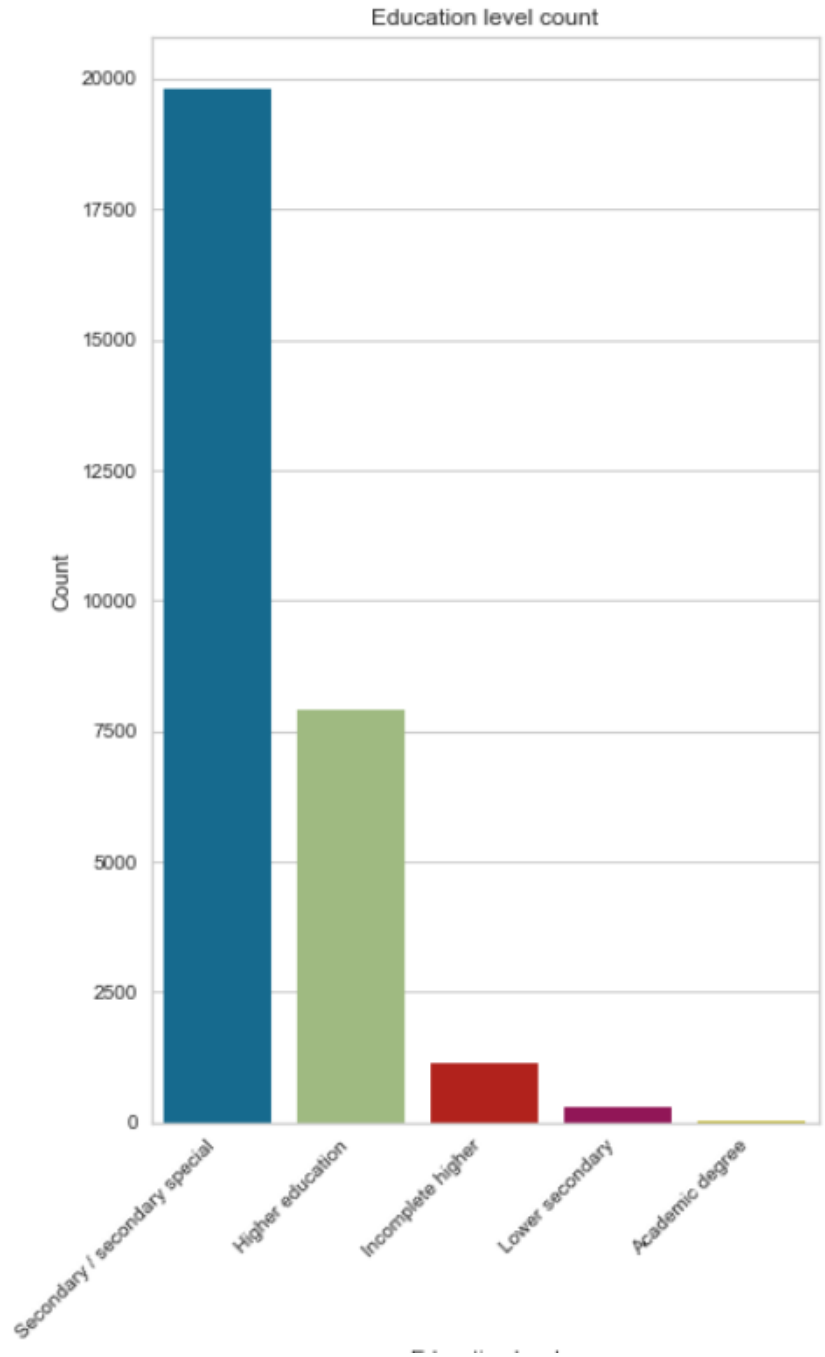
Hình 4.5

- Về kiểu lao động chủ yếu đó là đang làm việc (working) chiếm 51.62%, sau đó tới cộng tác viên (commercial associate) chiếm 23.32%. (Hình 4.6)



Hình 4.6

- Về trình độ học vấn, hầu hết các khách hàng đều hoàn thành chương trình cấp 2 và ¼ trong số đó hoàn thành chương trình cấp 3 hoặc cao hơn. (Hình 4.7)



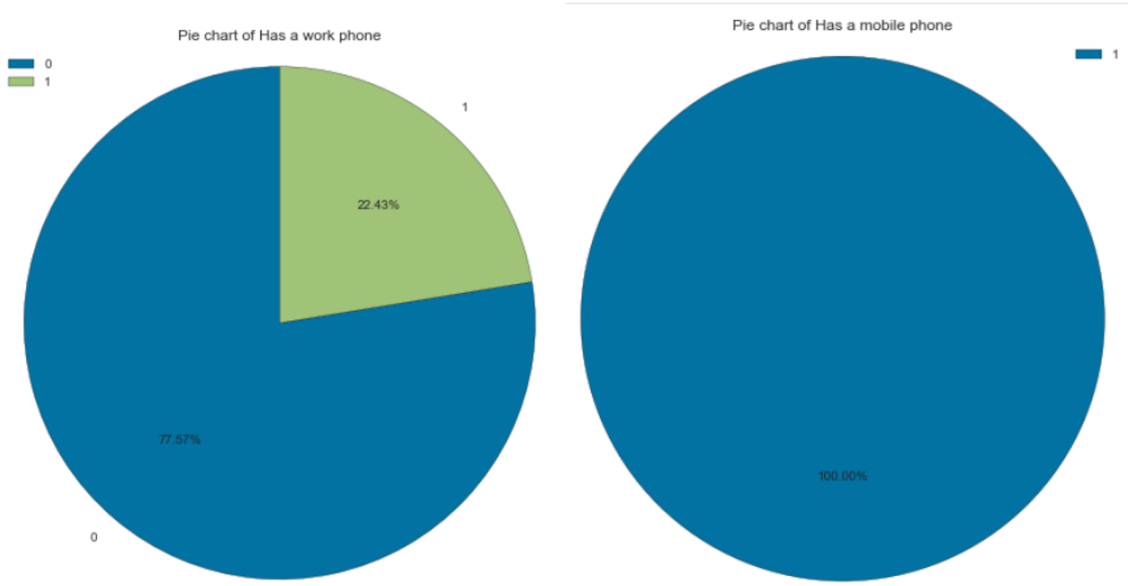
Hình 4.7

- Đa phần các khách hàng đều chưa có ô tô: 62.16% chưa có ô tô và 37.84% đã có ô tô.

- Đa số khách hàng đều sở hữu một tài sản có giá trị với tỉ lệ là 67.06%.

- Có nhiều hơn ¾ khách hàng không có điện thoại làm việc.

- 100% khách hàng có điện thoại di động, vì thế thuộc tính Has a mobile phone không giúp ích gì trong mô hình, có thể bỏ thuộc tính này đi.



Hình 4.8

- Tương tự 70% khách hàng có điện thoại bàn, 90% khách hàng không sử dụng tài khoản email.

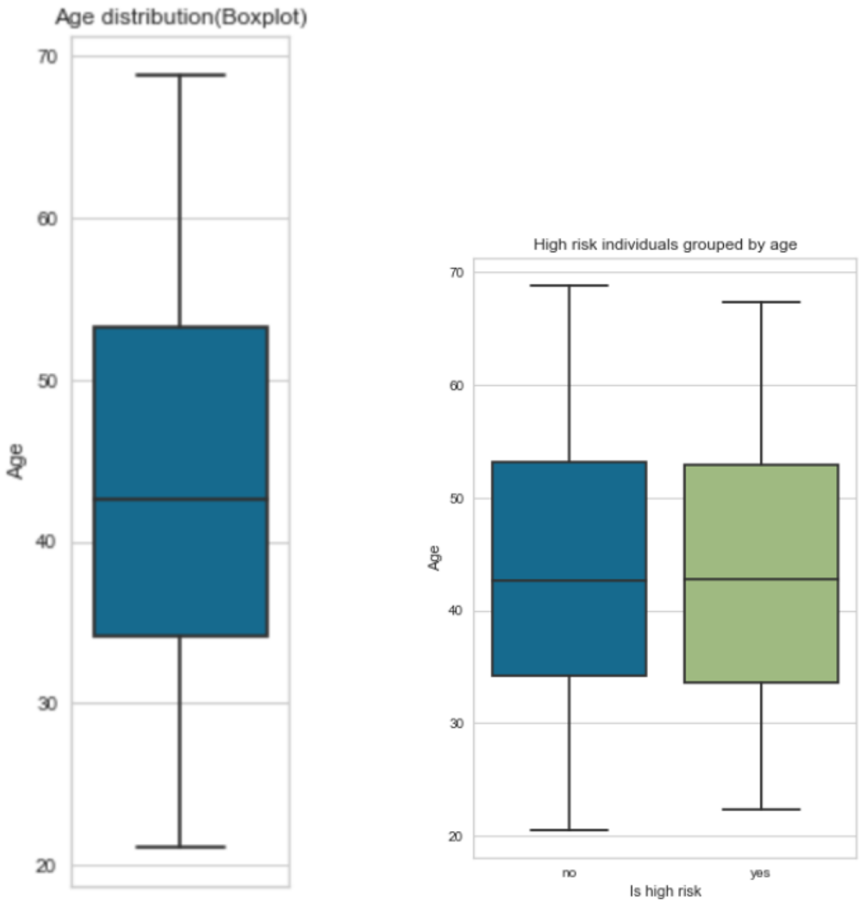
Các biến liên tục:

- Với biến tuổi 'Age':

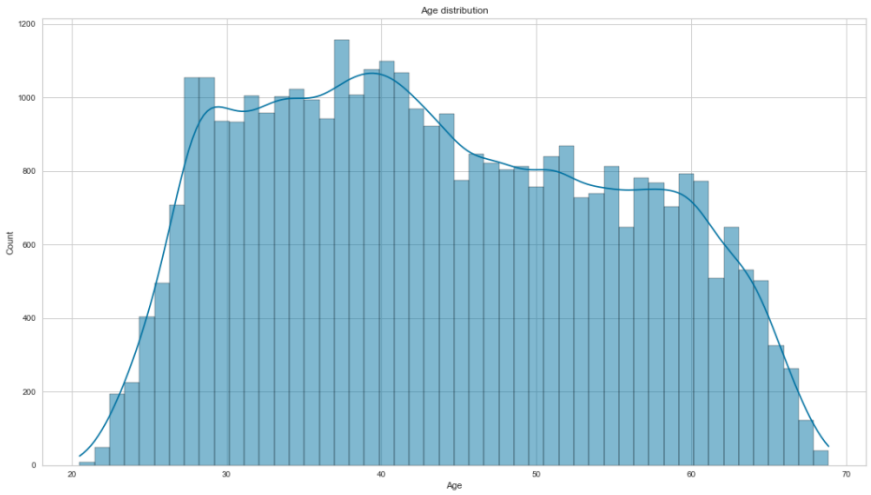
Dựa vào biểu đồ hộp (Hình 4.9) , có thể thấy người có độ tuổi trẻ nhất là 21 tuổi và già nhất có tuổi là 69 tuổi. Trung bình độ tuổi của tất cả khách hàng là 43 tuổi và có 75% khách hàng là trên 34 tuổi.

Nhóm khách hàng có rủi ro cao và nhóm khách hàng có rủi ro thấp không có sự khác nhau về mặt độ tuổi.

Phân bố của tuổi phân bố không bình thường, bị lệch về bên trái (Hình 4.10).



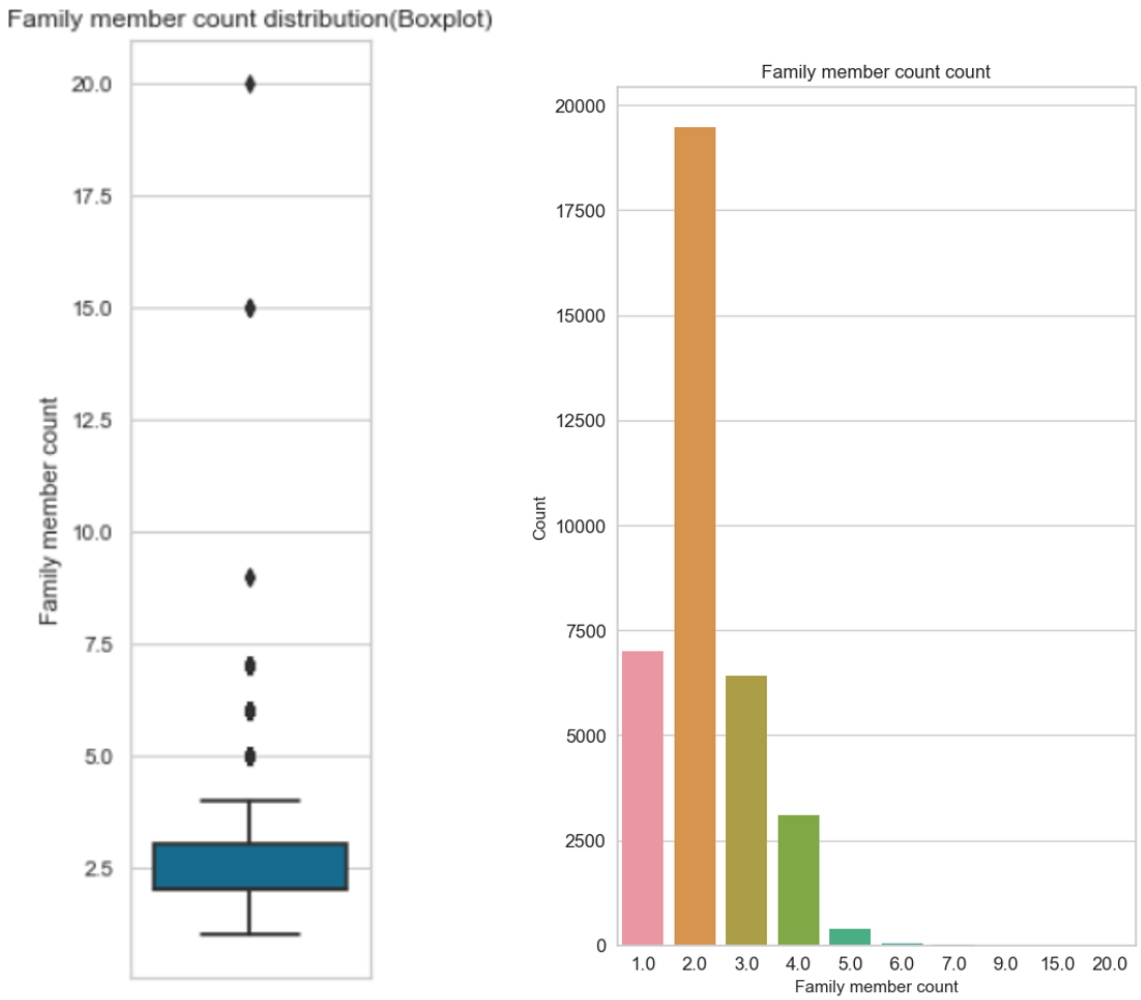
Hình 4.9



Hình 4.10

- Với biến ‘Family member count’:

Ở biểu đồ hộp hình 4.11 có 6 outlier rất lớn như có 15 - 20 người trong một hộ gia đình. Theo biểu đồ, gia đình hường có 2 người tức là tập trung chủ yếu ở gia đình chưa có con.

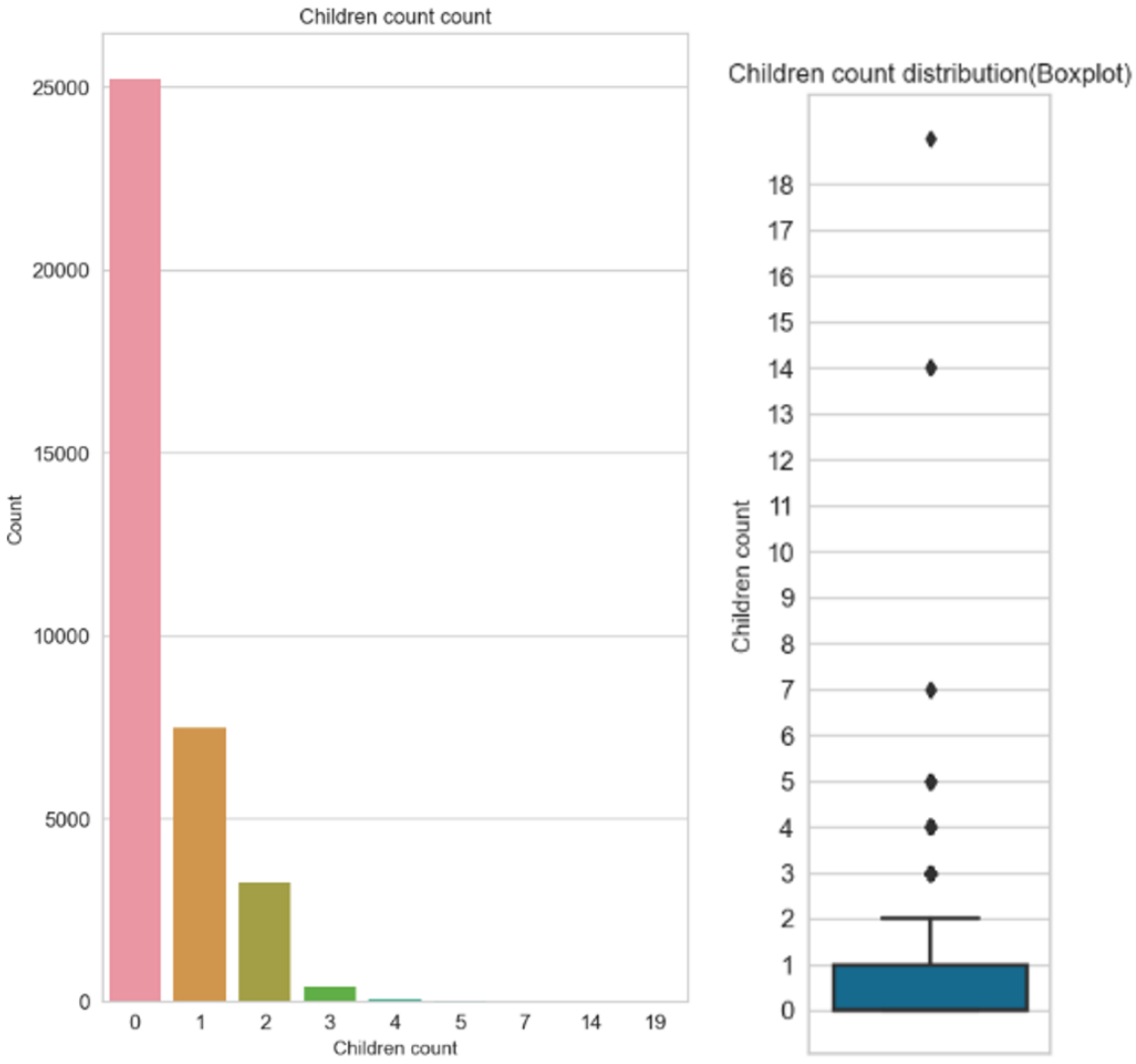


Hình 4.11

- Với biến ‘Children count’:

Các hộ gia đình có xu hướng không có con hoặc có 1 con. (Hình 4.12)

Tương tự như phần số lượng thành viên trong gia đình ta vẫn có 6 outlier.



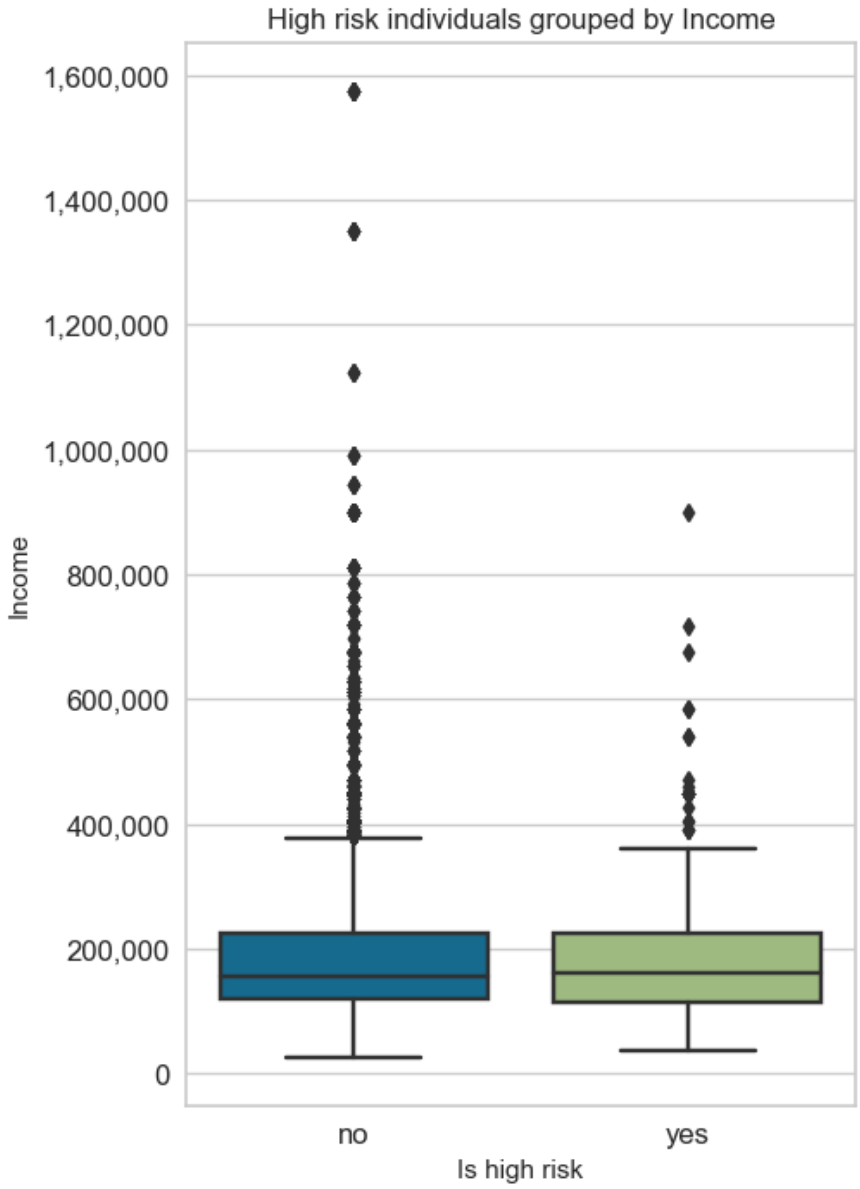
Hình 4.12

- Với biến 'Income':

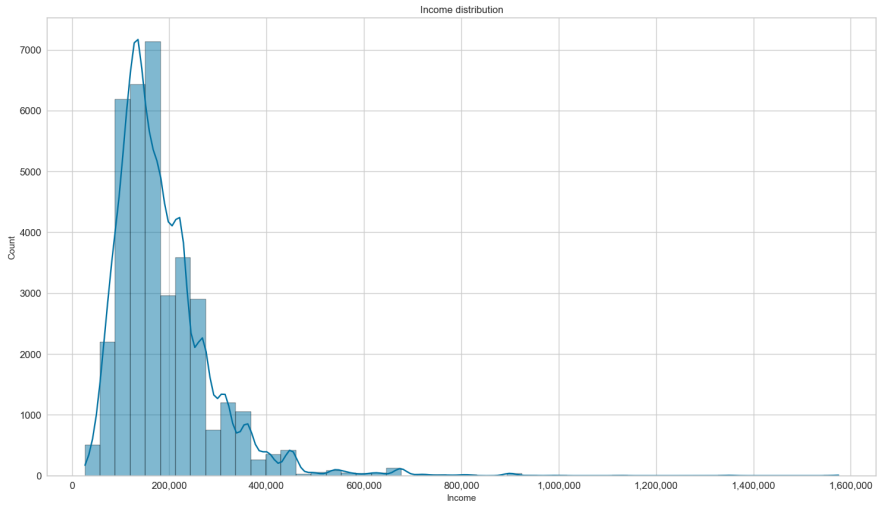
Dựa vào biểu đồ hộp (Hình 4.13) so sánh giữa nhóm high-risk và nhóm low-risk nhận thấy không có sự khác nhau giữa hai nhóm này.

Biểu đồ hist (Hình 4.14) cho thấy phân bố thu nhập có xu hướng lệch bên trái, phân bố thu nhập chủ yếu trong khoảng 0 – 400000 $ một năm.

Biểu đồ hộp cũng chỉ ra rất nhiều ngoại lệ và có tới 3 người có thu nhập trên 1,000,000 $ một năm.


High risk individuals grouped by Income
1,600,000
1,400,000
1,200,000
1,000,000
800,000
600,000
400,000
200,000
0
Income
no
yes
Is high risk

Hình 4.13

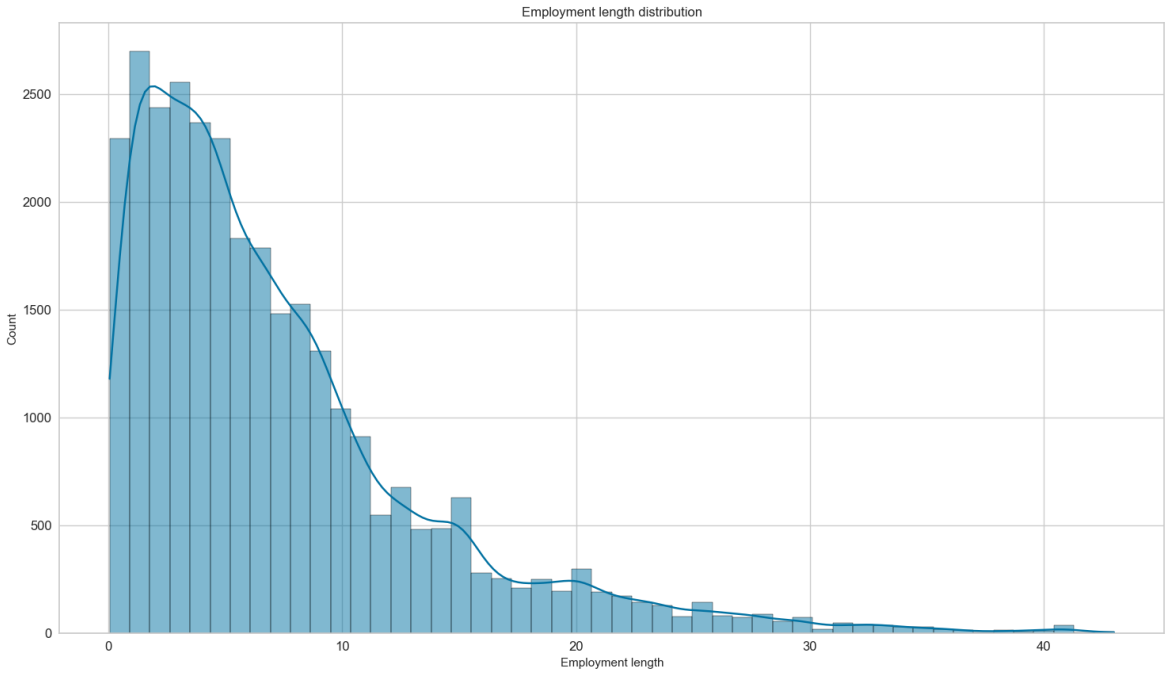


Hình 4.14

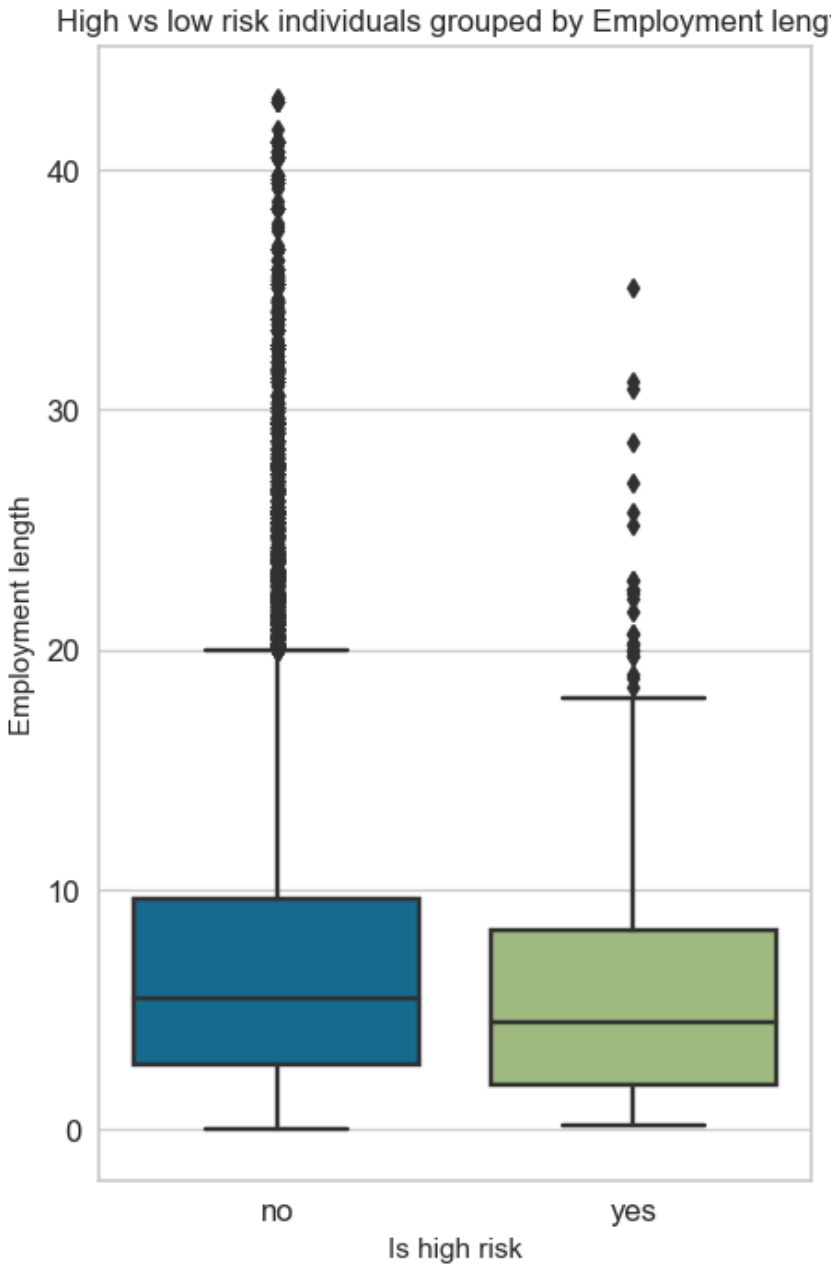
- Với biến 'Employment length':

Phân phối thời gian làm việc có xu hướng lệch trái nhiều, tập trung nhiều nhất ở nhóm thời gian làm việc từ 0 đến 10 năm. (Hình 4.15)

Có khá nhiều ngoại lệ làm việc nhiều hơn 20 năm. Nhóm rủi ro cao vỡ nợ có xu hướng làm việc ít hơn so với nhóm rủi ro thấp. (Hình 4.16)



Hình 4.15

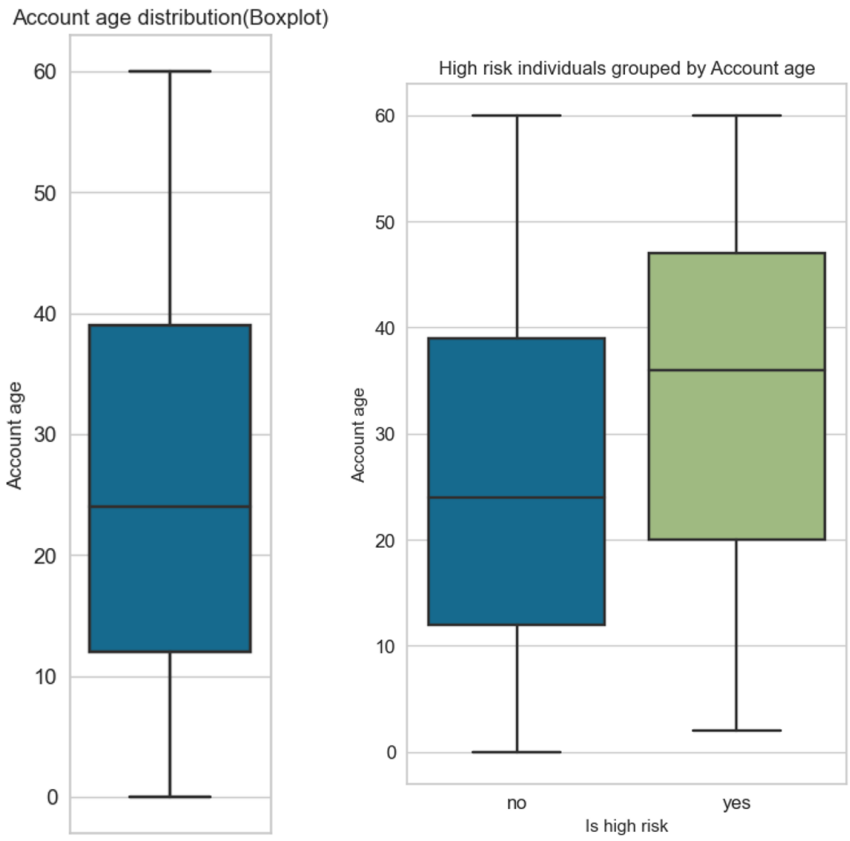


Hình 4.16

- Với biến 'Account age':

Trung bình tuổi của các tài khoản là 26 tháng. Không có ngoại lệ ở biến này (Hình 4.17).

Các tài khoản có độ tuổi trung bình 34 tháng có độ rủi ro cao hơn.

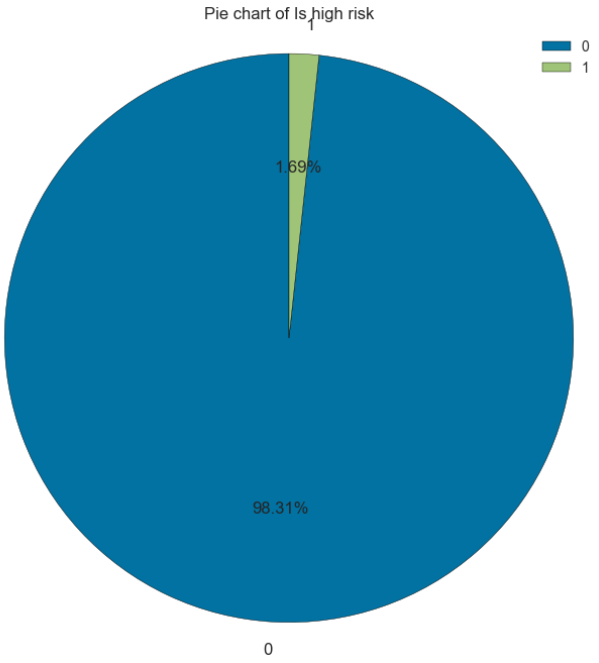


Hình 4.17

- Biến mục tiêu ‘Is high risk’:

- Dữ liệu bị lệch quá nhiều, có quá ít dữ liệu high-risk.

- Để khắc phục lệch dữ liệu như vậy sử dụng kỹ thuật SMOTE ở phần tiền xử lí dữ liệu.



Hình 4.18

4.1.3 Phân tích đa biến

a, Phân tích tương quan giữa các biến liên tục

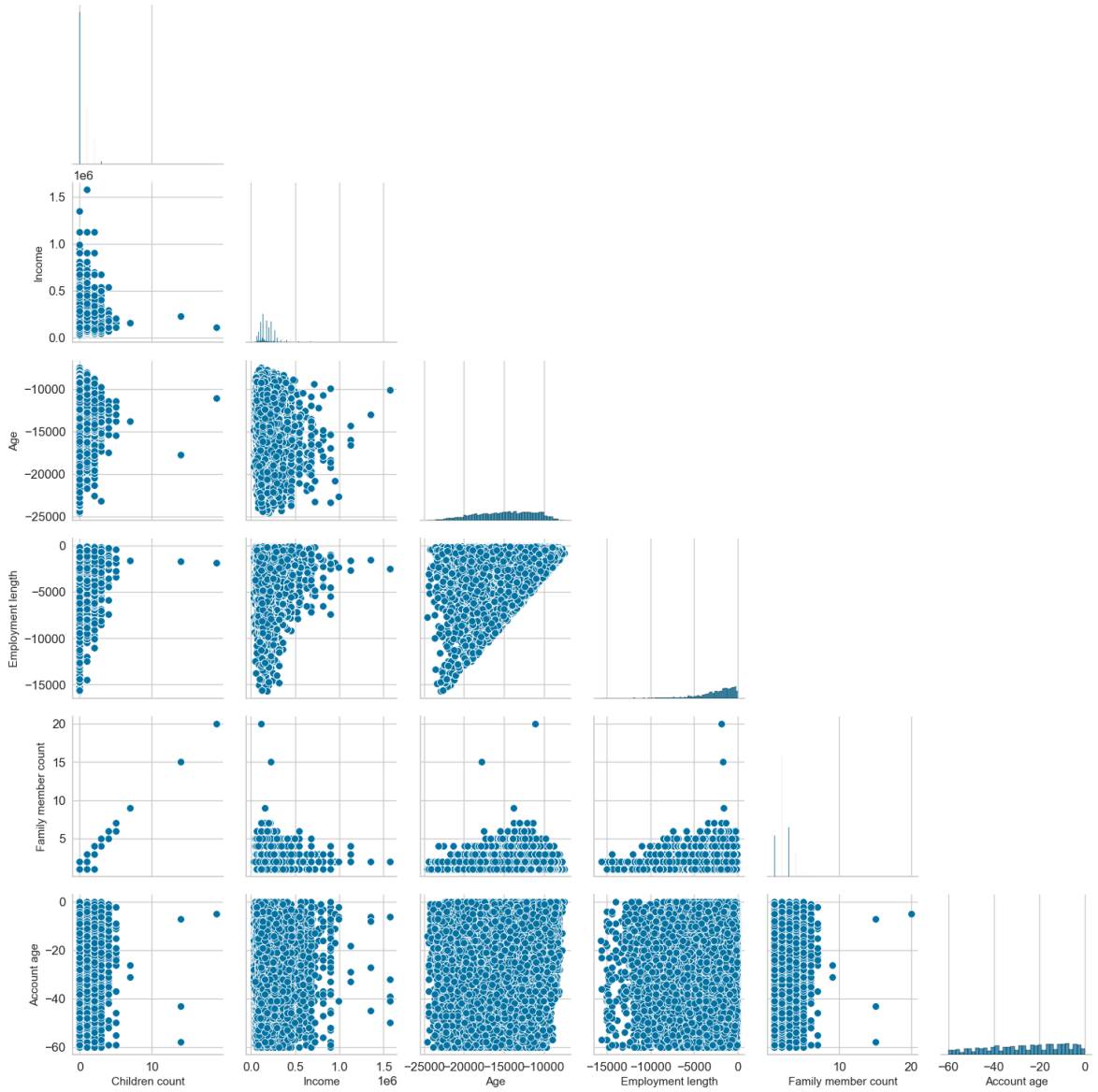
Biểu đồ phân tán:

Đầu tiên để xem xét quan hệ giữa các biến số, ta sẽ sử dụng biểu đồ phân tán.

Biểu đồ scatter plot (biểu đồ phân tán) là biểu đồ sử dụng để thể hiện mối quan hệ giữa hai biến định lượng. Trên trục hoành của biểu đồ là giá trị của biến đầu tiên, trên trục tung là giá trị của biến thứ hai. Mỗi điểm trên biểu đồ tương ứng với một cặp giá trị của hai biến này.

Trong thư viện Seaborn của Python, hàm 'sns.pairplot' được sử dụng để tạo ra một biểu đồ scatter plot matrix cho tập dữ liệu:

sns.pairplot(data,corner=True)



Hình 4.19

Nhận thấy một quan hệ tuyến tính giữa ‘Family member count’ và ‘Children count’. Điều này rất có lý vì thêm trẻ có thì thêm thành viên trong gia đình. Nghĩa là 2 cột trên có mối quan hệ tương quan rất lớn nên ta phải bỏ 1 trong hai cột đó đi.

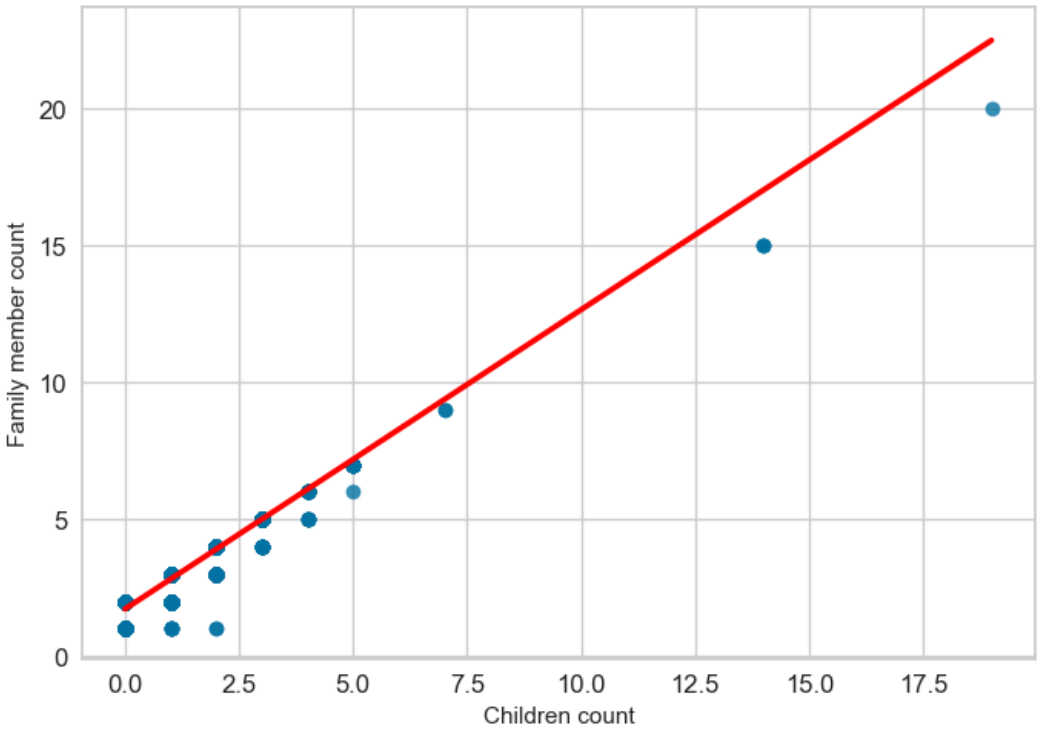
Quan hệ tuổi càng lớn thì thời gian lao động cũng càng cao. Điều này cũng có đúng vì khi càng thời gian làm việc càng dài thì độ tuổi cũng càng nhiều hơn.

‘Family member count’ và ‘children count’:

Sử dụng hàm ‘sns.regplot()’ trong thư viện Seaborn của Python để tạo biểu đồ scatter plot trực quan hóa mối tương quan giữa hai biến ‘Family member count’ và

‘children count’:

sns.regplot(x = "Children count", y = "Family member count", data= full_data)

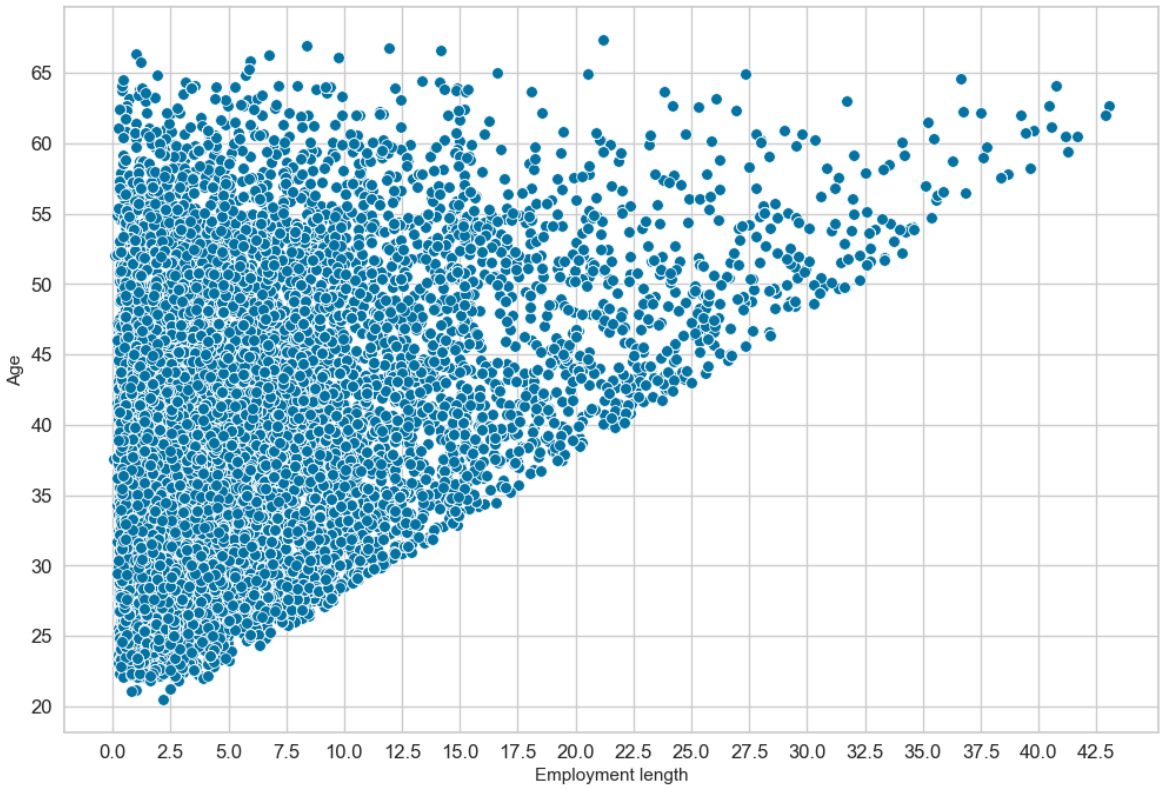


Hình 4.20

Dễ dàng nhận thấy số lượng con của một người tỉ lệ thuận số thành viên trong gia đình. Hai thuộc tính này có tương quan mạnh với nhau, vì vậy khi đào tạo mô hình ta sẽ bỏ đi một trong hai thuộc tính này.

‘Employment length’ và ‘Age’:

Vẽ biểu đồ Scatter của 2 biến ‘Employment length’ và ‘Age’:



Hình 4.21

Biểu đồ trên thể hiện hai biến 'Employment length' vs 'Age' là có tương quan.

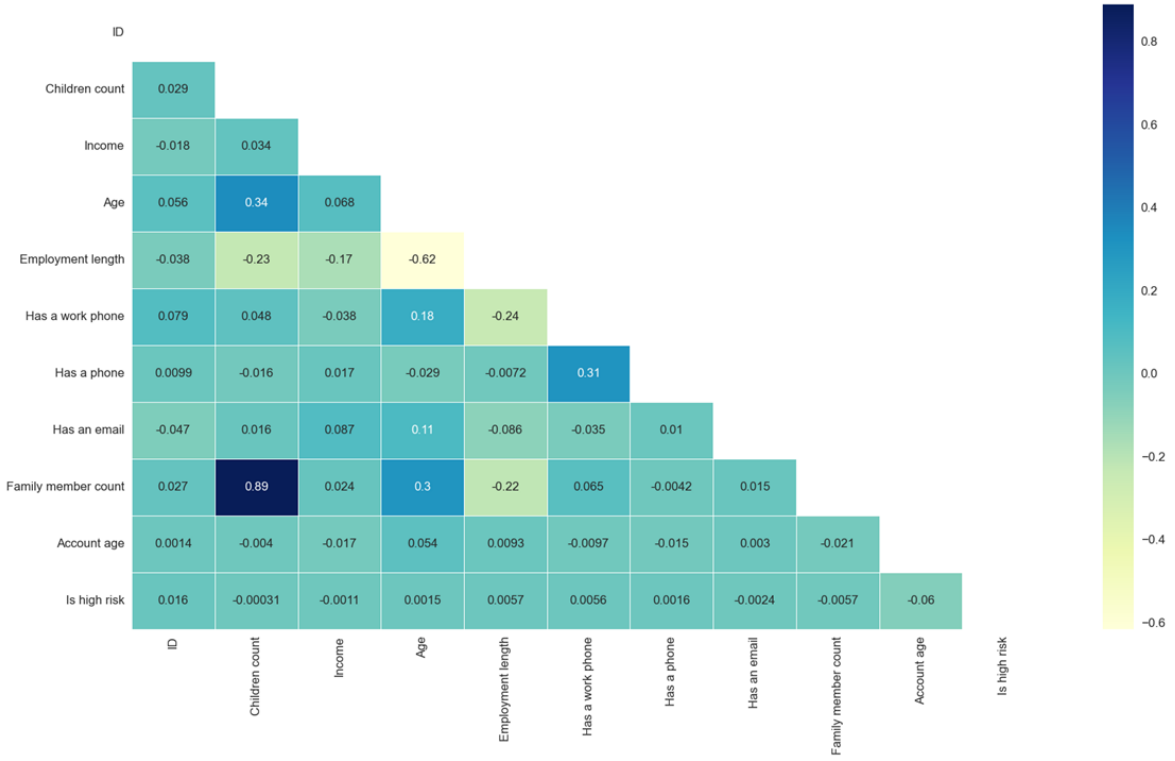
Biểu đồ trên cho ra một hình như một tam giác trên bởi vì không thể có thời gian làm việc lớn hơn độ tuổi.

Ma trận tương quan:

Ma trận tương quan sử dụng để mô tả mối tương quan giữa các biến trong một bộ dữ liệu, cung cấp một cái nhìn tổng quan về sự tương quan giữa các biến và có thể giúp lựa chọn các biến quan trọng để sử dụng trong mô hình.

Để vẽ ma trận tương quan cho tất cả các biến số, sử dụng hàm 'sns.heatmap()' trong thư viện Seaborn của Python. Hàm này cho phép trực quan hóa mối tương quan giữa các biến dưới dạng ma trận.

sns.heatmap(corr_matrix, annot=True)



Hình 4.22

Từ ma trận tương quan trên, rút ra một số nhận xét:

- Không có biến nào tương quan mạnh với biến mục tiêu là 'Is high risk'.

- Biến 'Family member count' và biến 'Children count' tương quan mạnh và có tỉ lệ thuận với nhau, điều này đã được nhận xét ở các biểu đồ trên.

- Biến 'Age' có tương quan yếu với biến 'Family member count', 'Children count'. Điều này có lý vì những người có độ tuổi lớn thì có xu hướng là có số thành viên lớn hơn.

- Biến 'Has a work phone' và biến 'Has a phone' có tính tương quan yếu và tỉ lệ thuận với nhau.

- Biến 'Employment length' và 'Age' tương quan trung bình với nhau, điều này đã được nhận xét ở trên.

b, Phân tích tương quan giữa biến liên tục với biến rời rạc

Tương quan giữa biến Age với các biến rời rạc:

Quan sát biểu đồ hộp của biến 'Age' với từng biến rời rạc (Hình 4.23), ta rút ra một

số nhận xét sau:

- Khách hàng Nữ có độ tuổi trung bình cao hơn so với khách hàng Nam.

- Những khách hàng không sở hữu xe hơi có xu hướng nhiều tuổi hơn.

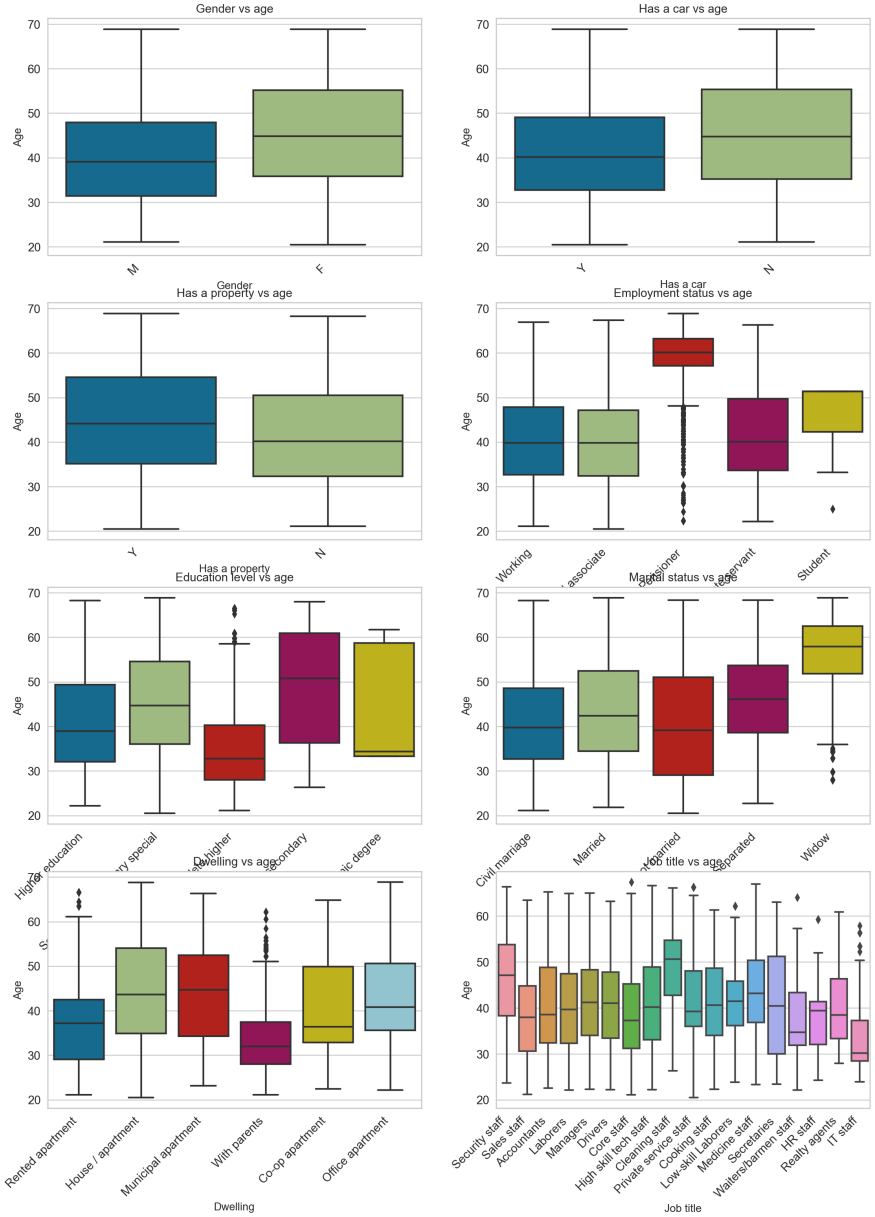
- Những khách hàng sở hữu một tài sản đảm bảo có xu hướng già hơn những người không.

- Khách hàng về hưu có độ tuổi trung bình cao hơn tất cả trạng thái khác. Điều này vô cùng hiển nhiên tuy nhiên ta thấy rất nhiều các outlier nghỉ hưu khi độ tuổi còn trẻ.

- Những khách hàng góa có số tuổi cao hơn so với khách hàng khác. Điều này khá dễ hiểu.

- Những khách hàng sống với bố mẹ thường có độ tuổi trẻ hơn so với nhóm khác. Điều này cũng khá dễ hiểu.

- Những khách hàng làm công việc dọn dẹp thì thường có độ tuổi cao nhất, những người làm IT hay có độ tuổi thấp nhất, nhân viên phục vụ/ nhân viên pha chế cũng có độ tuổi thấp.



Hình 4.23

Tương quan giữa biến Employment length với các biến rời rạc:

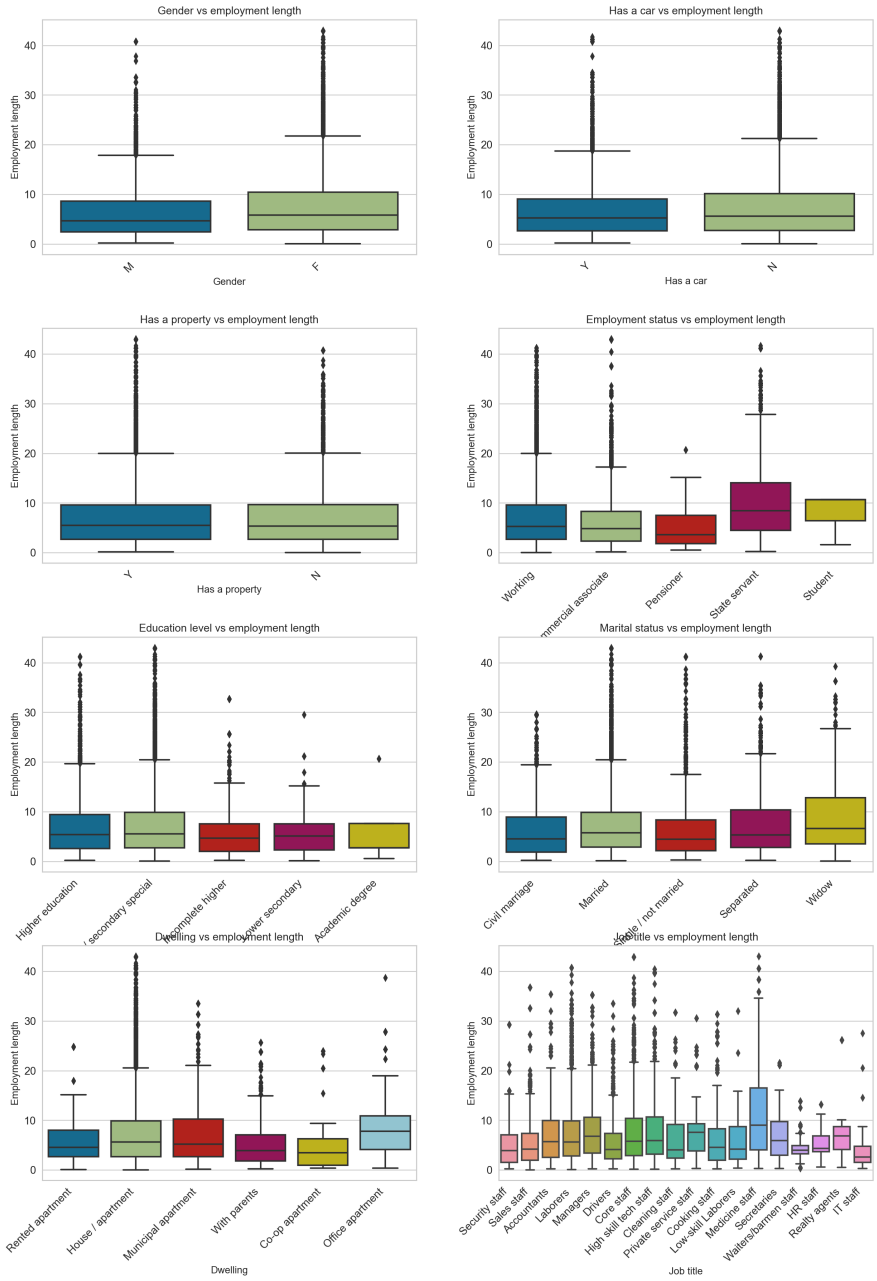
Quan sát biểu đồ hộp của biến 'Employment length' với từng biến rời rạc (Hình 4.24), ta rút ra một số nhận xét sau:

- Thời gian làm việc của khách hàng nữ có xu hướng nhiều hơn nam.

- Thời gian làm việc của công chức nhà nước (state servent) có xu hướng nhiều nhất so với các tình trạng làm việc khác.

- Những người có nơi ở là căn hộ văn phòng (office apartment) có thời gian làm việc nhiều hơn các kiểu nhà ở còn lại.

- Ngành nghề có thời gian lao động nhiều nhất đó là nhân viên y tế (medicine staff).



Hình 4.24

4.2 Tiền xử lí dữ liệu

4.2.1 Xử lí ngoại lệ

Với dạng số, dữ liệu ngoại lệ có thể là một giá trị phi thực tế như số tuổi, số ngày làm việc âm, hoặc một giá trị rất khác với phần còn lại của các giá trị ví dụ như 3 khách hàng có thu nhập tới trên 1 triệu $. Có hai nhóm các giá trị ngoại lệ:

1. Các giá trị không nằm trong miền xác định của dữ liệu. Ví dụ, tuổi, ngày làm việc không thể là số âm.

2. Các giá trị có khả năng xảy ra nhưng xác suất rất thấp. Những giá trị này có khả năng xảy ra nhưng thực sự hiếm có.

Đầu tiên ở nhóm 1, thực hiện chuẩn hóa thời gian chuyển hết các giá trị số ngày âm thành dương ở hai thuộc tính là ‘Employment length’ và ‘Age’:

data[[’Employment length’,’Age’]] = np.abs(data[[’Employment length’,’Age’]])

Ở nhóm 2 gồm các giá trị có thể xảy ra nhưng xác suất thấp. Dựa vào phân tích đơn biến, có thể thấy xuất hiện các ngoại lệ ở ba thuộc tính đó là: ’Family member count’,’Income’, ’Employment length’. Để xử lí ngoại lệ sử dụng phương pháp IQR:

IQR là sự khác biệt giữa tứ phân vị thứ nhất Q1 và tứ phân vị thứ ba Q3. Tính IQR: IQR = Q3 - Q1.

Sau đó tính các giá trị biên trên biên dưới:

Biên dưới: lower_iqr = Q1 – k*IQR

Biên trên: upper_iqr = Q3+ k*IQR

Hệ số k chọn trong trong bài nghiên cứu k = 3. Sau khi tính được biên, giữ lại tất cả các giá trị trong khoảng [Q1 – 3*IQR,Q3+ 3*IQR]. Các giá trị ngoài biên coi là ngoại lệ và bỏ các giá trị đó:

*df = df[((df[self.feat_with_outliers]< (Q1 - 3 * IQR))*
*|(df[self.feat_with_outliers] > (Q3 + 3 * IQR))).any(axis=1)]*

4.2.2 Xóa thuộc tính không cần thiết

Thực hiện bỏ đi các thuộc tính sau:

- ID : Không có giá trí khi ta áp dụng mô hình dự đoán, ID chỉ giúp merge các bảng.

- Has a mobile phone: tất cả mọi người đều có mobile phone nên thuộc tính này không còn quan trọng.

- Children count: do thuộc tính có tính tương quan rất cao với Family count nên ta chọn một trong hai thuộc tính mang tính tổng quát hơn.

- Job title: do bị thiếu quá nhiều dữ liệu nên ta cũng bỏ đi.

- Account age: dữ liệu bị overfitting.

df.drop(self.feature_to_drop,axis=1,inplace=True)

4.2.3 Giảm nhiễu giữa các khoảng dữ liệu

Xử lý độ lệch là bước tiền xử lý với mục tiêu là làm cho việc phân phối các tính năng trở nên đối xứng hơn để cải thiện hiệu suất của các mô hình. Bằng cách lấy căn bậc ba của các giá trị trong cột, phép biến đổi có thể làm giảm độ lệch của phân phối. Ở đây phân phối của Income và Age quá tương lệch và tập trung ở một khoảng lớn vì vậy ta thực hiện giảm nhiễu ở hai thuộc tính này.

df[self.feat_with_skewness] = np.cbrt(df[self.feat_with_skewness])

4.2.4 One hot encoding và Ordinal encoding

One hot encoding và Ordinal encoding là là hai kỹ thuật phổ biến được sử dụng để chuyển đổi dữ liệu phân loại thành dữ liệu số

One hot encoding là một kỹ thuật tạo cột nhị phân cho từng danh mục trong một thuộc tính phân loại. Mỗi cột đại diện cho một danh mục và nếu một điểm dữ liệu thuộc về danh mục đó, thì cột đó sẽ có giá trị là 1 và 0 nếu không.

Sử dụng one hot encoding với các thuộc tính: Gender, Matirial status, Dwelling type, Employment status, Has a car, Has a property, Has a work phone, Has a phone, Has an email.

Ordinal encoding là một kỹ thuật gán một giá trị số cho từng danh mục dựa trên thứ hạng hoặc thứ tự của chúng. Ví dụ: trong một đối tượng địa lý có ba danh mục - "thấp", "trung bình" và "cao" - có thể gán giá trị thấp là 1, trung bình là giá trị 2 và cao là giá trị 3. Mã hóa này hữu ích khi có một trật tự tự nhiên giữa các danh mục.

Sử dụng Ordinal encoding với thuộc tính: Education level.

4.2.5 Min-Max Scaler

MinMaxScaler là một kỹ thuật chuẩn hóa dùng để thu nhỏ khoảng giá trị mà không làm thay đổi hình dạng phân phối.

Công thức cho MinMax Scaler như sau:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

MinMax Scaler nhạy cảm với các giá trị ngoại lệ và có thể bị ảnh hưởng bởi sự hiện diện của các giá trị cực đoan trong dữ liệu, vì vậy trước khi thực hiện kỹ thuật MinMax Scaler ta loại bỏ các giá trị ngoại lệ trước.

Trong Python, MinMax Scaler có sẵn trong thư viện Scikit-learning, ở đây sử dụng MinMaxScaler với 3 thuộc tính: 'Age', 'Income', 'Employment length'.

4.2.6 Oversampling

SMOTE (Kỹ thuật lấy mẫu quá mức thiểu số tổng hợp) là một thuật toán thường được sử dụng để lấy mẫu quá mức trong trường hợp phân loại không cân bằng.

Ở mẫu dữ liệu trong bài luận, số lượng mẫu trong lớp thiểu số - số khách hàng vỡ nợ thấp hơn đáng kể so với số lượng mẫu trong lớp đa số - số khách hàng không vỡ nợ. Điều này có thể dẫn đến một bộ phân loại thiên về lớp đa số và hoạt động kém đối với lớp thiểu số. Các kỹ thuật lấy mẫu quá mức như SMOTE có thể được sử dụng để giải quyết vấn đề này.

Bằng cách tạo các mẫu tổng hợp, SMOTE giúp cân bằng phân bố lớp, làm cho bộ phân loại ít thiên vị hơn đối với lớp đa số. Điều này có thể dẫn đến hiệu suất tốt hơn trên lớp thiểu số và cải thiện hiệu suất mô hình tổng thể.

oversample = SMOTE(sampling_strategy='minority')

X_bal, y_bal = oversample.fit_resample (data.loc[:, data.columns != 'Is high risk'],data['Is high risk'])

4.3 Huấn luyện và đánh giá các mô hình

4.3.1 Logistic regression

Huấn luyện mô hình:

Khởi tạo mô hình hồi quy logistic với random_state = 42 và max_iter = 1000.

logistic_regression = LogisticRegression(random_state=42,max_iter=1000)

random_state = 42: đặt giá trị ngẫu nhiên cho mô hình hồi quy logistic là 42, nghĩa là nếu mã được chạy nhiều lần với cùng một dữ liệu, thì mô hình sẽ được khởi tạo với cùng một giá trị ngẫu nhiên, đảm bảo khả năng tái tạo kết quả.

max_iter=1000: chỉ định số lần lặp tối đa để mô hình hội tụ. Mặc định max_iter được đặt là 100. Tuy nhiên, trong một số trường hợp, mô hình có thể không hội tụ trong vòng 100 lần lặp và việc đặt giá trị cao hơn cho max_iter có thể giúp mô hình hội tụ. Ở đây ta sử dụng 1000 lần lặp.

Sau đó khớp mô hình hồi quy logistic (logistic_regression) với dữ liệu huấn luyện (X_train_prep và y_train_prep):

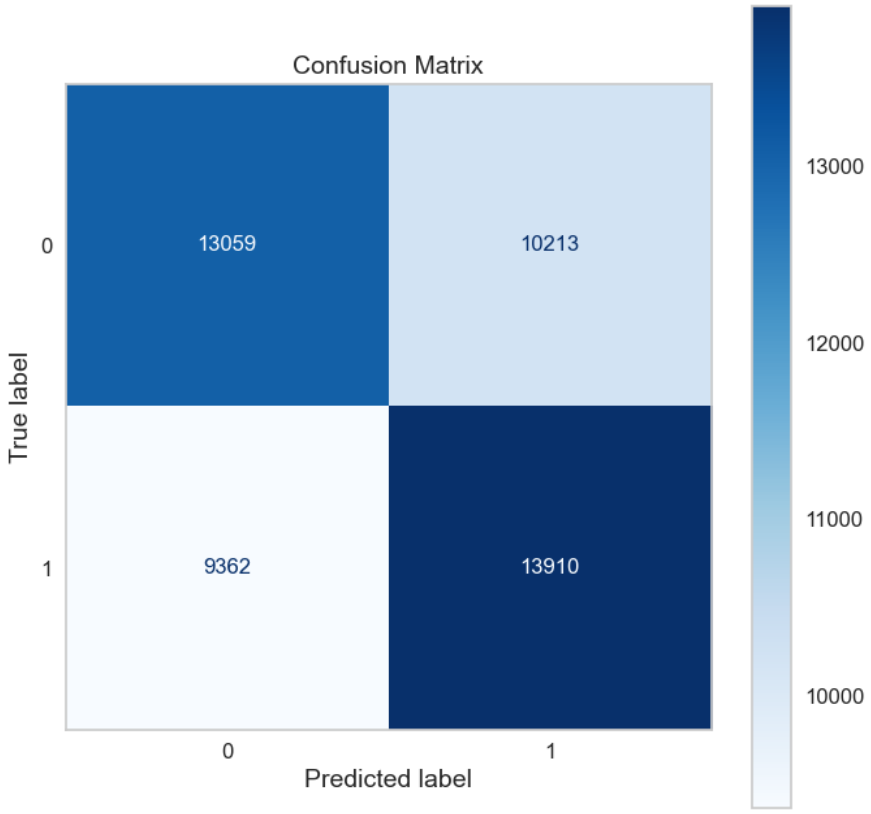
lr_model =logistic_regression.fit(X_train_prep,y_train_prep)

Đánh giá mô hình:

Xem xét Confussion matrix (Hình 4.25)

- Mô hình đã dự đoán đúng 13059 trường hợp thuộc lớp 0 (không vỡ nợ) và dự đoán đúng 13910 trường hợp thuộc lớp 1 (vỡ nợ).

- Tuy nhiên có thể thấy mô hình Logistic Regression phân loại sai khá nhiều, cụ thể: số lượng false negative (FN) là 9362, có nghĩa là mô hình đã dự đoán sai 9362 trường hợp vỡ nợ là không vỡ nợ. Và số lượng false positive (FP) là 10213, có nghĩa là mô hình đã dự đoán sai 10213 trường hợp không vỡ nợ là vỡ nợ.



Hình 4.25

	precision	recall	f1-score	support
0	0.58	0.56	0.57	23272
1	0.58	0.60	0.59	23272
accuracy			0.58	46544
macro avg	0.58	0.58	0.58	46544
weghted avg	0.58	0.58	0.58	46544

Bảng 4.2: Classification report cho mô hình Hồi quy Logistic

Từ confusion matrix, tính được các độ đo đánh giá hiệu suất của mô hình như accuracy, precision, recall và F1 score thể hiện trên bảng (Bảng 4.2):

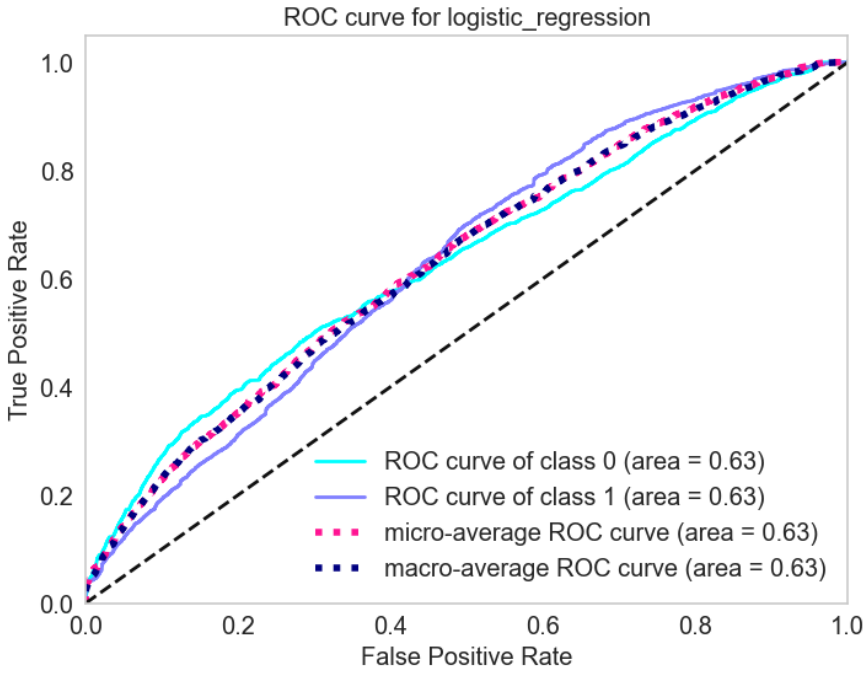
Độ chính xác: tỷ lệ các trường hợp tích cực được dự đoán thực sự tích cực là 0,58 đối với cả lớp 1 (những người vỡ nợ) và lớp 0 (những người không vỡ nợ).

Chỉ số Recall: tỷ lệ các trường hợp tích cực thực tế được mô hình xác định chính xác là 0.60 đối với lớp 1 và 0.56 đối với lớp 0.

Điểm F1: là 0.59 đối với lớp 1 và 0.57 đối với lớp 0.

Độ chính xác tổng thể của mô hình là 0.58, có nghĩa là mô hình dự đoán chính xác lớp cho 58% số trường hợp trong tập dữ liệu.

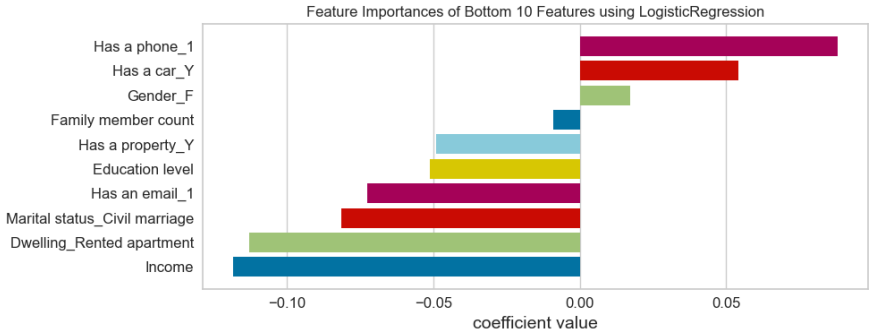
Tiếp theo, xét đến đồ thị ROC cho mô hình Logistic Regression (Hình 4.26):



Hình 4.26

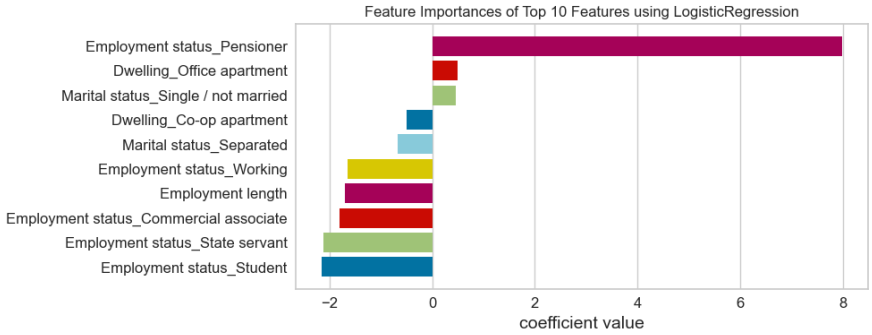
Mô hình này tốt hơn dự đoán ngẫu nhiên vì giá trị AUC-ROC = 0,63 > 0,5. Tuy nhiên mô hình này không đạt được hiệu suất cao, một mô hình được coi là tốt thì AUC-ROC cần lớn hơn hoặc bằng 0,8; vì vậy mô hình này thực tế chưa thể áp dụng cần phải cải thiện.

Sau đó ta xem xét về tầm quan trọng của các thuộc tính trong mô hình này.



Hình 4.27

Biểu đồ 4.27 chỉ ra 10 thuộc tính ít quan trọng nhất đối với mô hình Logistic Regression. Các thuộc tính này có ảnh hưởng ít quan trọng đến khả năng dự đoán vỡ nợ của khách hàng.



Hình 4.28

Hệ số có giá trị dương đại diện cho một mối quan hệ dương tuyến tính giữa thuộc tính và khả năng khách hàng vỡ nợ và ngược lại. Theo 10 thuộc tính quan trọng nhất của mô hình Logistic Regression (Hình 4.28) có thể rút ra một số nhận xét:

- Khách hàng có trạng thái lao động là nghỉ hưu (Employment status_Pensioner) có khả năng vỡ nợ cao hơn.

- Trong khi đó, khách hàng có trạng thái lao động là đang đi làm, cộng tác viên thương mại, công chức nhà nước, hay là học sinh có khả năng thấp hơn bị vỡ nợ.

- Khách hàng có thời gian lao động (Employment length) dài thì khả năng vỡ nỡ cũng thấp hơn.

- Khách hàng có kiểu nhà văn phòng (Dwelling_Office apartment) và có tình trạng hôn nhân độc thân (Maritial status_Single) cũng có mối quan hệ dương tuyến tính với khả năng vỡ nợ, nhưng tác động của chúng không lớn như Employment status_Pensioner.

- Mặt khác, khách hàng có kiểu nhà ở "hợp tác" (Dwelling_Co-op apartment) và tình trạng hôn nhân đã ly thân có khả năng vỡ nợ thấp hơn.

Nhìn chung, mô hình có hiệu suất gần tương tự cho cả hai lớp 0 và 1, với khả năng nhớ lại và điểm F1 chỉ cao hơn một chút đối của lớp 1 với lớp 0. Giá trị AUC-ROC value thấp đạt 0.63. Hiệu suất tổng thể của mô hình không cao, độ chính xác chỉ là 0.58.

4.3.2 Decision Tree

Huấn luyện mô hình:

Khởi tạo mô hình Decision Tree với random_state=42

decision_tree = DecisionTreeClassifier(max_depth=500, max_leaf_nodes=500, min_samples
min_samples_split=20, random_state=42)

max_depth = 500: giá trị này đặt độ sâu tối đa của cây là 500.

max_leaf_nodes = 500: đặt số nút lá tối đa mà cây có thể có là 500. Sử dụng để kiểm soát độ phức tạp của mô hình và ngăn chặn overfitting.

min_samples_leaf = 50: đặt số lượng mẫu tối thiểu cần có tại một nút lá là 50. Nếu số lượng mẫu tại một nút lá nhỏ hơn, thì nút đó không bị phân tách.

min_samples_split = 20: đặt số lượng mẫu tối thiểu cần thiết để phân chia một nút trong là 20. Nếu số lượng mẫu tại một nút nhỏ hơn giá trị này, thì nút đó không bị phân chia.

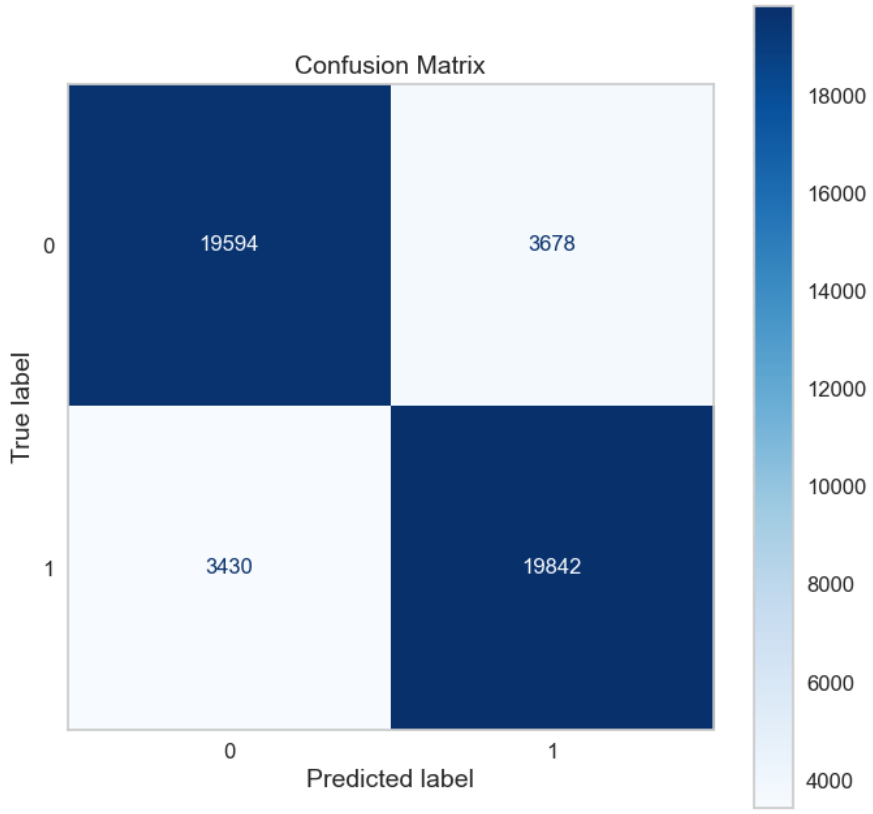
random_state = 42: đặt giá trị ngẫu nhiên cho mô hình là 42, đảm bảo mô hình sẽ được khởi tạo với cùng một giá trị ngẫu nhiên, đảm bảo khả năng tái tạo kết quả.

Sau đó khớp mô hình với dữ liệu huấn luyện (X_train_prep và y_train_prep):

dt_model = decision_tree.fit(X_train_prep,y_train_prep)

Đánh giá mô hình:

Trước hết xem xét Confusion matrix:



Hình 4.29

Nhận xét:

- Mô hình đã dự đoán đúng 19594 trường hợp thuộc lớp 0 (không vỡ nợ) và 19842 trường hợp thuộc lớp 1 (vỡ nợ).

- Có thể thấy mô hình Decision phân loại sai rất ít, cụ thể: số lượng false negative (FN) là 3430 và số lượng false positive (FP) là 3678.

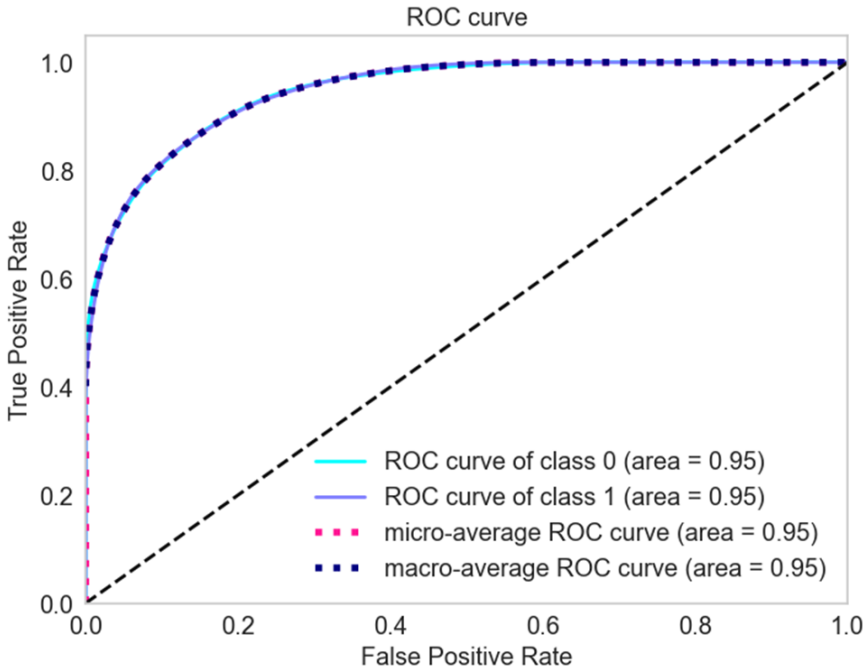
Từ confusion matrix, tính được các độ đo đánh giá hiệu suất của mô hình như accuracy, precision, recall và F1 score (Bảng 4.3).

	precision	recall	f1-score	support
0	0.85	0.84	0.85	23272
1	0.84	0.85	0.85	23272
accuracy			0.85	46544
macro avg	0.85	0.85	0.85	46544
weghted avg	0.85	0.85	0.85	46544

Bảng 4.3: Classification report cho mô hình Decision Tree

Mô hình Decision Tree này cho thấy kết quả khá tốt trên tập dữ liệu được đánh giá, với độ chính xác đạt 0.84 cho lớp 1 và 0.85 cho lớp 0. Precision và recall đều là 0.84 và 0.85 cho cả hai lớp, cho thấy mô hình có độ chính xác khá cao khi phân loại các trường hợp vỡ nợ và không vỡ nợ.

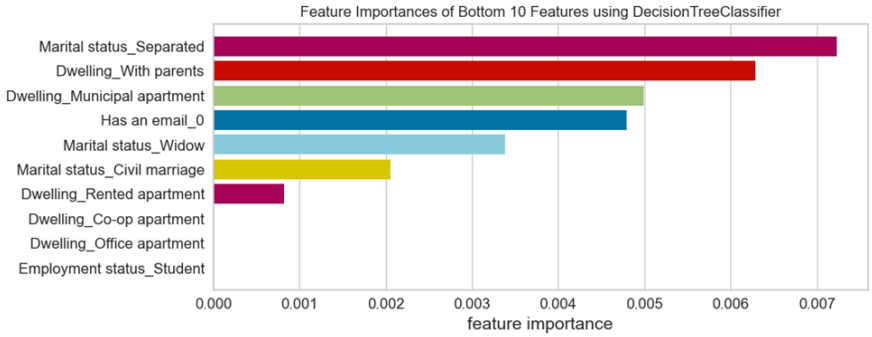
Đồ thị ROC cho mô hình Decision Tree (Hình 4.30):



Hình 4.30

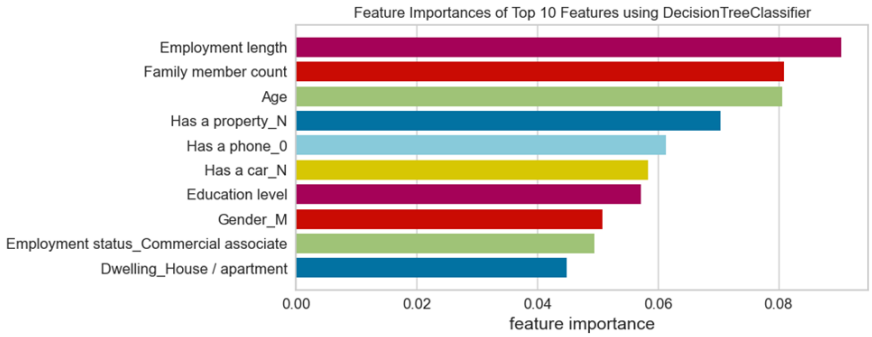
Diện tích của đường ROC sử dụng để đo lường độ chính xác của mô hình, trong đó diện tích càng gần 1 thì mô hình càng tốt trong việc phân loại. Đồ thị ROC của cả hai lớp (class 0 và class 1) đều có diện tích bằng 0.95, điều đó cho thấy rằng mô hình này rất tốt trong việc phân loại khả năng rủi ro tín dụng.

Xem xét các thuộc tính trong mô hình Decision Tree



Hình 4.31

Các thuộc tính như: Email, Tình trạng hôn nhân ... , có ảnh hưởng rất ít đến khả năng dự đoán vỡ nợ của khách hàng. Các thuộc tính này có thể bị loại bỏ mà không ảnh hưởng nhiều đến hiệu suất của mô hình dự đoán.



Hình 4.32

Biểu đồ 4.32 chỉ ra 10 thuộc tính quan trọng nhất của mô hình Decision Tree, có một số nhận xét:

- Thuộc tính quan trọng nhất là Employment length, tiếp đến là Family member count và Age, và các thuộc tính còn lại cũng đóng vai trò quan trọng trong mô hình.

- Nhìn chung, các thuộc tính quan trọng đều liên quan đến khả năng thanh toán của khách hàng, bao gồm thu nhập, tuổi tác, số lượng thành viên trong gia đình và tình trạng việc làm. Các thuộc tính về tài sản như Has a property_Y và Dwelling_house cũng đóng vai trò quan trọng, cho thấy rằng sở hữu tài sản ảnh hưởng đến khả năng vay vốn và thanh toán.

Tổng quan, mô hình Decision Tree này cho thấy hiệu suất khá tốt và có thể được sử dụng để dự đoán khả năng vỡ nợ trong tập dữ liệu.

4.3.3 Random Forest

Huấn luyện mô hình:

Đầu tiên khởi tạo mô hình Random Forest với random_state=42

random_forest = RandomForestClassifier(max_depth=500,max_leaf_nodes=500, min_samples_leaf=50,min_samples_split=20, random_state=42)

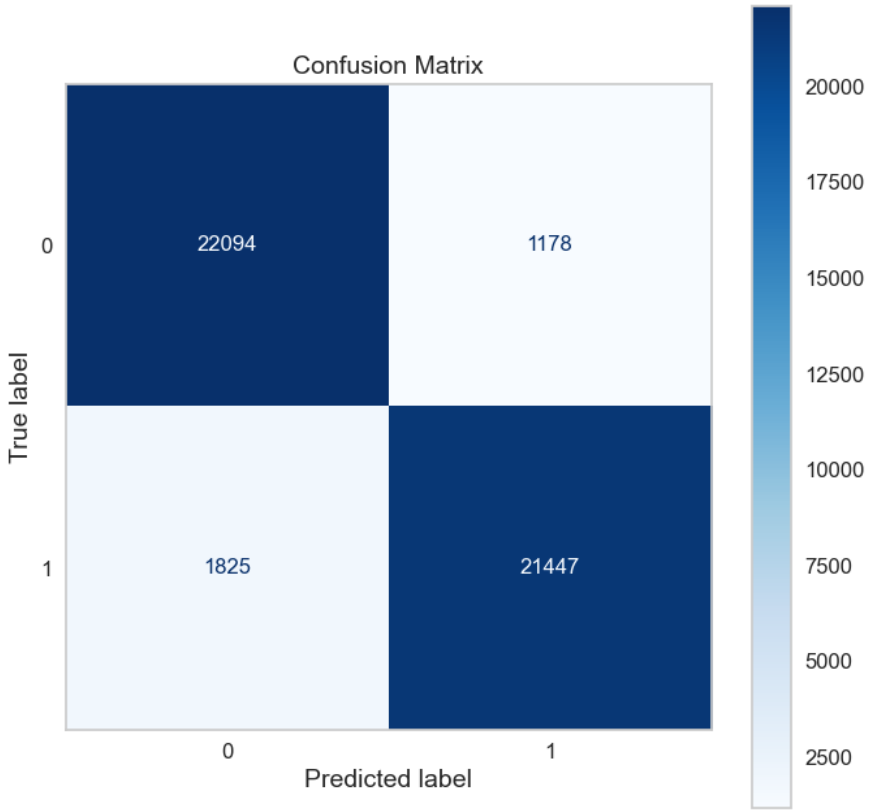
Các tham số khởi tạo tương tự như mô hình Decision Tree.

Sau đó khớp mô hình với dữ liệu huấn luyện (X_train_prep và y_train_prep):

rf_model =random_forest.fit(X_train_prep,y_train_prep)

Đánh giá mô hình:

Xem xét Confusion matrix:



Hình 4.33

Nhận xét:

- Mô hình đã dự đoán đúng 22094 trường hợp thuộc lớp 0 (không vỡ nợ) và dự

đoán đúng 21447 trường hợp thuộc lớp 1 (vỡ nợ).

- Có thể thấy mô hình Decision phân loại sai rất ít, cụ thể: số lượng false negative (FN) là 1825 và số lượng false positive (FP) là 1178.

	precision	recall	f1-score	support
0	0.92	0.95	0.94	23272
1	0.95	0.92	0.93	23272
accuracy			0.94	46544
macro avg	0.94	0.94	0.94	46544
weghted avg	0.94	0.94	0.94	46544

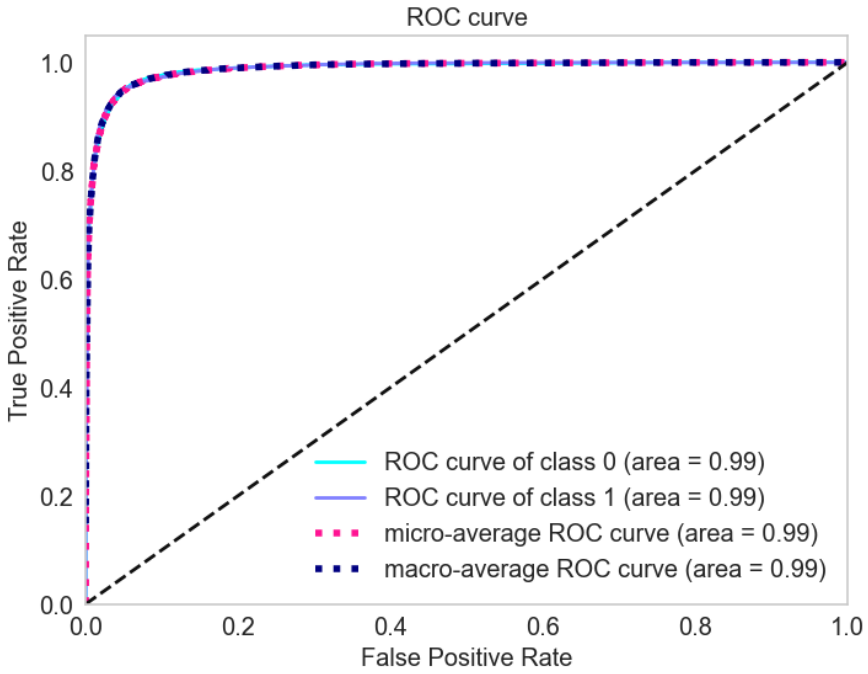
Bảng 4.4: Classification report cho mô hình Random Forest

Dựa trên các chỉ số precision, recall, accuracy và F1-score, mô hình Random Forest có hiệu suất rất cao trong việc dự đoán rủi ro tín dụng.

Cả precision và recall đều được đánh giá cao là 0.95 và 0.92 cho cả hai nhãn 0 và 1. Tỷ lệ dự đoán chính xác của mô hình cũng là rất cao đạt 0.94.

F1-score là phương pháp kết hợp giữa precision và recall, và cũng đạt giá trị 0.93 cho lớp 1 và 0.94 cho lớp 0. Điều này cho thấy mô hình dự đoán chính xác và đầy đủ cả những trường hợp có nhãn 0 và 1.

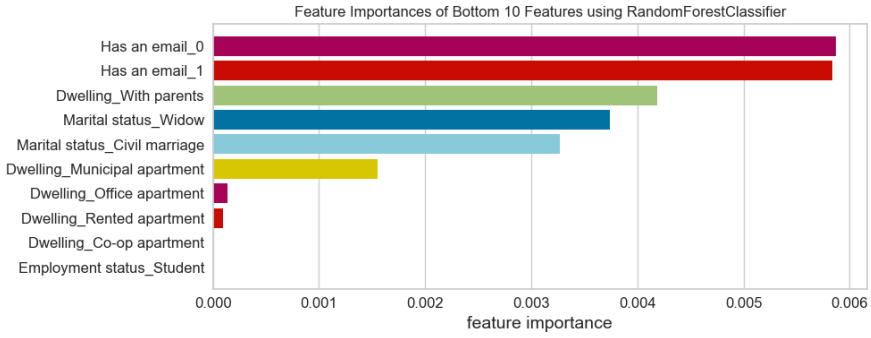
Tiếp theo, xét đồ thị ROC cho mô hình Random Forest (Hình 4.34):



Hình 4.34

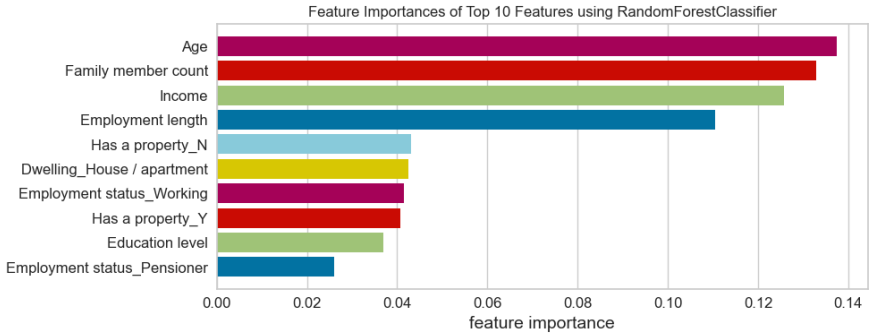
Giá trị AUC-ROC của đồ thị là 0.99, đây là dấu hiệu của một mô hình tốt. Điều này cũng chứng minh là mô hình gần như không có lỗi phân loại, và độ chính xác rất cao.

Cuối cùng ta xem xét về tầm quan trọng của các thuộc tính trong mô hình này.



Hình 4.35

Các thuộc tính trong hình 4.35 có ảnh hưởng không quan trọng hoặc ít quan trọng đến khả năng dự đoán vỡ nợ của khách hàng. Các thuộc tính của khách hàng như: Email, Dwelling, Marital status... có thể bị loại bỏ mà không ảnh hưởng nhiều đến hiệu suất của mô hình dự đoán.



Hình 4.36

Với 10 thuộc tính quan trọng nhất của mô hình Decision Tree (Hình 4.36), có các quan sát sau:

- Thuộc tính quan trọng nhất là tuổi (Age) và số thành viên trong gia đình (Family member count) tiếp đó đến thu nhập (Income), thời gian làm việc (Employment length), và các thuộc tính còn lại cũng đóng vai trò quan trọng trong mô hình.

- Các thuộc tính quan trọng liên quan đến thu nhập, tuổi và thời gian làm việc có thể cho thấy khả năng thanh toán của khách hàng. Số lượng thành viên trong gia đình và trình độ học vấn cũng có tác động đến khả năng thanh toán của khách hàng.

Nhìn chung, có thể kết luận rằng mô hình Random Forest là mô hình có hiệu suất cao nhất với các thông số đánh giá vượt trội. Thực tế mô hình này có thể rất tốt trong việc dự đoán khả năng vỡ nợ của khách hàng.

4.4 So sánh các mô hình

Sử dụng phương pháp xác thực chéo k-fold cross validation, ta đã đánh giá được hiệu suất của các mô hình, với Decision Tree và Random Forest đạt độ chính xác cao nhất với 85% và 94% tương ứng. Trong khi đó, mô hình Logistic Regression đạt độ chính xác thấp hơn khá nhiều, khoảng 59%. Thể hiện trên đồ thị ROC cho kết quả rất tốt ở mô hình Decision Tree và Random Forest. Đường cong ROC của hai mô hình này gần sát với góc trên bên trái của biểu đồ, có TPR cao và FPR thấp ở nhiều ngưỡng khác nhau và diện tích dưới đường cong lần lượt là 0.95 và 0.99 . Ở mô hình Logistic Regression, đường cong tiến tới thành đường chéo 45 độ trong không gian ROC, diện tích dưới đường cong là 0.63, độ chính xác thấp.

Các mô hình Decision Tree và Random Forest cho thấy độ chính xác đồng đều cho cả hai lớp, và có Precision, Recall và F1-score cao cho cả hai lớp. Trong khi đó,

mô hình Logistic Regression có các chỉ số đều thấp hơn 0.7. Các thuộc tính quan trọng cho mỗi mô hình cũng khác nhau, tuy nhiên có những thuộc tính trùng lặp như 'Income', 'Age', 'Employment length', 'Family member count'. Các thuộc tính này đóng vai trò quan trọng trong việc dự đoán khả năng vỡ nợ của khách hàng.

Thực nghiệm trên bộ dữ liệu test với P1 P2 P3 là bộ dữ liệu với tỷ lệ train/test lần lượt là 90/10, 80/20 và 75/25, kết quả accuracy tổng hợp ở bảng 4.5:

	P1	P2	P3
Logistic Regression	51.75%	53.74%	54.13%
Decision Tree	84.07%	80.41%	80.07%
Random Forest	93.50%	88.79%	86.08%

Bảng 4.5: So sánh mô hình trên tập dữ liệu test

Kiểm tra trên tập dữ liệu mới (unseen), mô hình Logistic Regression có độ chính xác rất thấp: chỉ nhỉnh hơn phân loại ngẫu nhiên một chút, độ chính xác khoảng 51% - 55%. Để sử dụng trong thực tế, mô hình cần phải cải thiện rất nhiều. Có một số điều chỉnh có thể thực hiện: tăng kích thước tập dữ liệu - tập dữ liệu lớn hơn có thể giúp cho mô hình học được các mối quan hệ phức tạp hơn giữa các đặc trưng và kết quả dự đoán; thay đổi tham số mô hình như hệ số điều chỉnh (regularization parameter) hay giá trị ngưỡng quyết định (decision threshold)... Với dữ liệu mới, độ chính xác của mô hình Decision Tree khá cao xấp xỉ 80-84%, và 88-93% đối với mô hình Random Forest. Nếu chỉ quan tâm đến độ chính xác của mô hình thì nên chọn Random Forest vì nó có độ chính xác cao hơn so với Decision Tree. Tuy nhiên, khi đánh giá mô hình, ta cần cân nhắc nhiều yếu tố khác như thời gian chạy, khả năng mở rộng, tính ổn định và khả năng giải thích kết quả. Nếu yêu cầu cần thời gian chạy nhanh, thì Decision Tree có thể là sự lựa chọn tốt hơn, bởi vì nó có thời gian huấn luyện và dự đoán nhanh hơn so với Random Forest. Tuy nhiên, nếu dữ liệu là rất lớn và phức tạp, thì Random Forest có thể có kết quả chính xác hơn và giải quyết các vấn đề như overfitting. Quyết định chọn mô hình nào phụ thuộc vào mục đích sử dụng và yêu cầu của ứng dụng. Ở khóa luận này, sau khi đánh giá các mô hình, em sẽ sử dụng mô hình Random Forest để xây dựng ứng dụng dự đoán chấp nhận thẻ tín dụng.

4.5 Ứng dụng

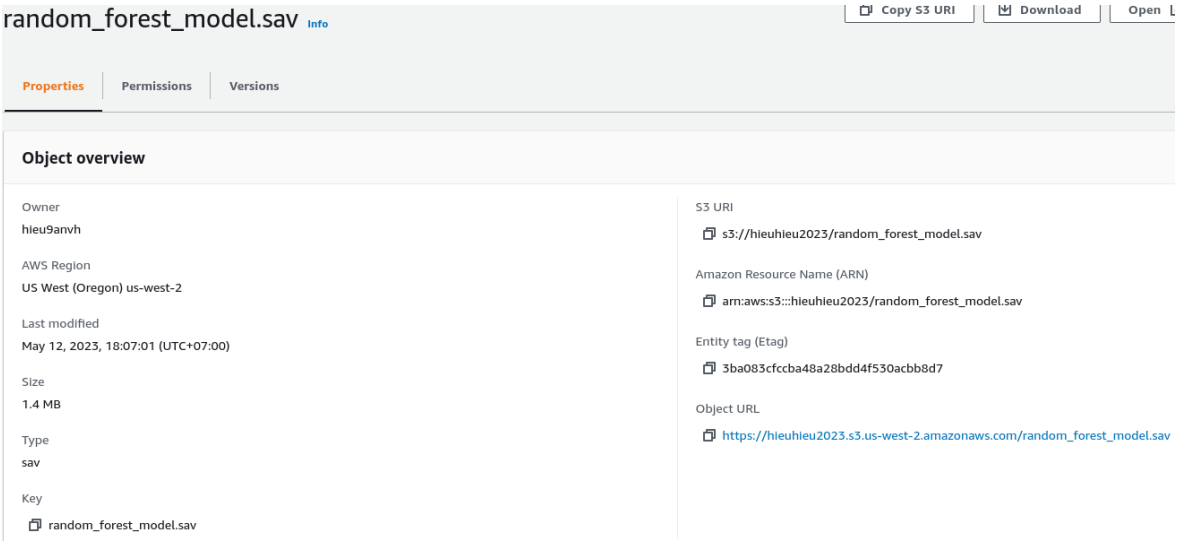
a) Triển khai mô hình trên AWS3

Mô hình được chọn sử dụng đó là RandomForest, lưu lại mô hình ta sử dụng dump

trong thư viện joblib như sau:

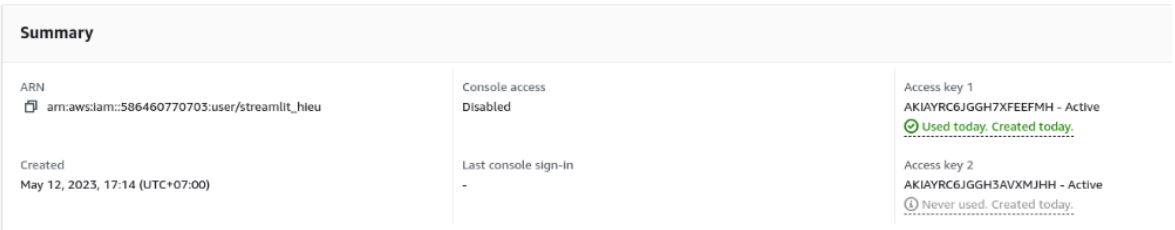
dump(randomforest_model, 'randomforest_model.sav')

Đầu tiên cần triển khai mô hình Random Forest đã lưu trên máy cục bộ lên AWS S3. AWS S3 (Amazon Web Services Simple Storage Service) là một dịch vụ lưu trữ đám mây cho phép người dùng lưu trữ và truy xuất các tệp tin trong một không gian lưu trữ trực tuyến. Tạo một bucket và tải mô hình lên vùng lưu trữ, trạng thái sau khi tải lên thành công như sau:



Hình 4.37

Mô hình đã được tải lên S3, ta có thể truy cập nó và đưa ra dự đoán bằng cách sử dụng các khóa truy cập bí mật và truy cập. Các khóa này sẽ được sử dụng khi liên kết ứng dụng web Streamlit với mô hình được lưu trữ trên AWS. Tạo tài IAM user mới sẽ được cung cấp hai khóa Access key ID và Secret access key như hình sau:



Hình 4.38

Sau đó cần tạo một hàm để đưa ra dự đoán. Hàm make_prediction() có chức năng

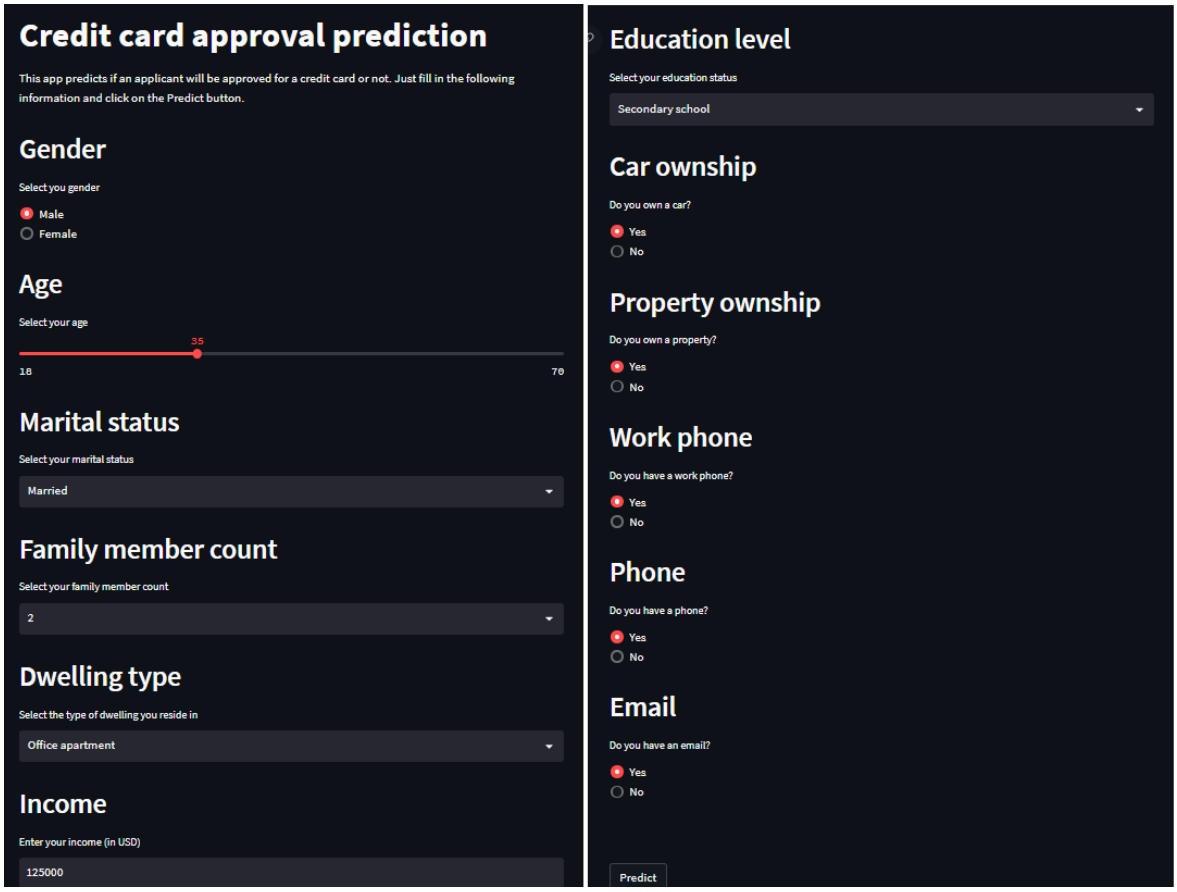
kết nối đến một bucket trên Amazon S3 và tải mô hình từ bucket đó. Để thực hiện chức năng này, ta cần cung cấp thông tin xác thực cho AWS là Access key ID và Secret access key đã tạo ở phần trên.

```
def make_prediction():
client = boto3.client( 's3', aws_access_key_id=st.secrets["access_key"],
aws_secret_access_key=st.secrets["secret_access_key"])
bucket_name = "creditrisk"
key = "random_forest_model.sav"
with tempfile.TemporaryFile() as fp:
client.download_fileobj(Fileobj=fp, Bucket=bucket_name, Key=key)
fp.seek(0)
model = joblib.load(fp)
return model.predict(profile_to_pred_prep)
```

Mô hình được lưu dưới dạng file có tên là random_forest_model.sav' trong bucket. Sau khi tải mô hình về, hàm sử dụng module joblib để load mô hình vào bộ nhớ và sử dụng hàm predict() để dự đoán kết quả dựa trên đầu vào profile_to_pred_prep. Kết quả trả về từ hàm này chính là dự đoán của mô hình, có thể là 0 hoặc 1.

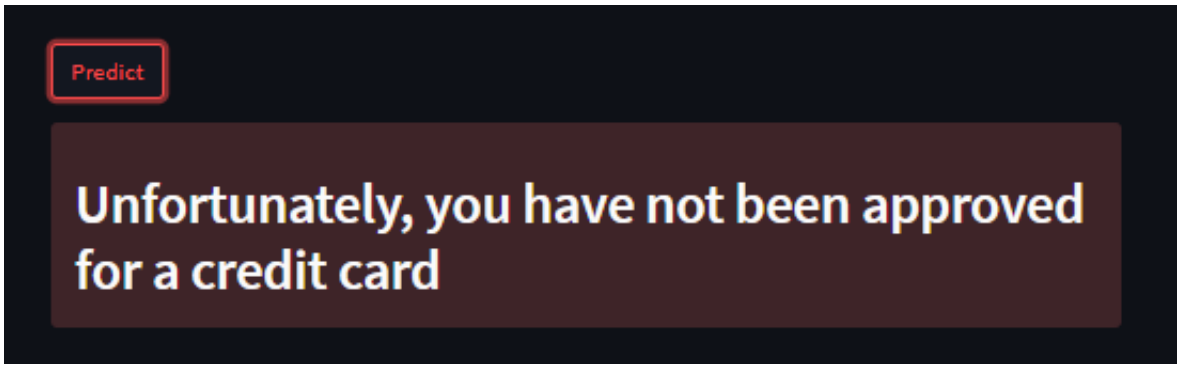
b) Giao diện Streamlit

Sau khi lưu trữ được mô hình đào tạo trên AWS S3, ta xây dựng một giao diện cho mô hình để người dùng có thể nhập thông tin của họ dưới một dạng biểu mẫu (là hồ sơ để dự đoán) và xem liệu họ có được chấp thuận cấp thẻ tín dụng hay không. Giao diện người dùng được thiết kế bằng framework Streamlit như hình 4.39. Khi đã điền các thông tin hồ sơ cấp thẻ tín dụng, nhấp vào nút predict để dự đoán kết quả.



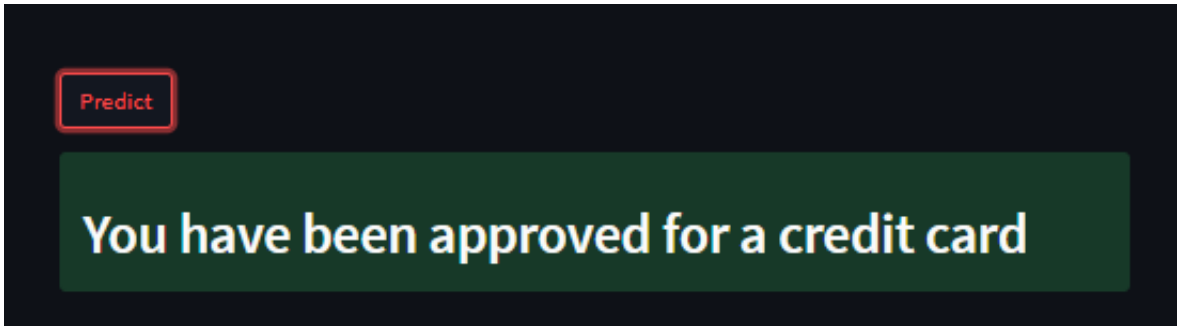
Hình 4.39

Nếu yêu cầu lập thẻ tín dụng không được chấp thuận, màn hình sẽ thông báo:



Hình 4.40

Nếu yêu cầu mở thẻ tín dụng được chấp thuận, màn hình sẽ thông báo:



Hình 4.41

KẾT LUẬN

Việc sử dụng các mô hình rủi ro tín dụng cũng ngày càng trở nên quan trọng trong bối cảnh toàn cầu hóa tài chính. Khi dòng vốn xuyên biên giới và thị trường tài chính trở nên kết nối hơn, điều cần thiết là phải có các phương pháp chính xác để đánh giá được rủi ro tín dụng. Bài luận này đã áp dụng kiến thức thống kê và phân tích dữ liệu, nhằm đánh giá rủi ro tín dụng với sự tiếp cận của ba mô hình chính là Logistic Regression, Decision Tree và Random Forest. So sánh ba mô hình với nhau về hiệu suất em nhận thấy rằng độ chính xác trong việc dự đoán khách hàng vỡ nợ giảm dần theo thứ tự Random Forest, Decision Tree, Logistic Regression. Ứng dụng dự đoán cấp thẻ tín dụng được xây dựng dựa theo mô hình đạt hiệu suất cao nhất là RandomForest. Ứng dụng này có thể được sử dụng bởi những người nộp đơn muốn tìm hiểu xem họ có được chấp thuận cấp thẻ tín dụng mà không ảnh hưởng đến điểm tín dụng của họ. Trong quá trình nghiên cứu, vấn đề khó khăn của bài toán gặp phải là tìm được bộ dữ liệu về thông tin khách hàng xin cấp tín dụng, có rất ít bộ dữ liệu loại này, điều này ảnh hưởng đến tính áp dụng thực tế của các mô hình. Bên cạnh đó, mô hình này chỉ dự đoán xem người đăng ký có được chấp thuận mở thẻ tín dụng hay không, có thể phát triển bằng cách kết hợp mô hình này với mô hình hồi quy để dự đoán mức giới hạn tín dụng mà người đăng ký sẽ được chấp thuận.

Trong thời gian tới, đề tài sẽ tập trung vào việc phát triển thêm các mô hình dự đoán rủi ro tín dụng mới sử dụng các kỹ thuật khác nhau hoặc tập trung vào việc nghiên cứu về các yếu tố khác ảnh hưởng đến rủi ro tín dụng. Ngoài ra, có thể sử dụng các phương pháp học sâu hoặc kết hợp các phương pháp học máy khác nhau để có thể cải thiện kết quả của mô hình.

TÀI LIỆU THAM KHẢO

Tiếng Việt

1. Đỗ Trung Tuấn (2018), *Phân tích thống kê và khai phá dữ liệu*, NXB ĐHQGHN.

2. Phạm Văn Kiều (1996), *Lý thuyết xác suất & thống kê toán học*, NXB ĐHQGHN.

Tiếng Anh

3. Ohri, Ajay (2018), *Python for R users : a data science approach*, Wiley.

4. Giulio Carlone (2021), *Introduction to Credit Risk, CRC Press*, Boca Raton United States of America.

5. Christian Bluhm, Ludger Over Beck and Christoph Wagner (2003), *An Introduction to Credit Risk Modeling*, CRC Press, Boca Raton United States of America.

6. Peter Bruce, Andrew Bruce and Peter Gedeck (2017), *Practical Statistics for data Scientists*, O'Reilly Media, United States of America.

7. David Lando (2009), *Credit Risk Modeling: Theory and Applications*, Princeton University Press, United States of America.

8. Christoph Molnar (2019), *Interpretable Machine Learning*, Lean Pub, Germany.