TRƯỜNG ĐẠI HỌC KINH TẾ - LUẬT

KHOA TÀI CHÍNH – NGÂN HÀNG

❖❖❖

BÁO CÁO CUỐI KỲ

MÔ HÌNH RỦI RO TÍN DỤNG TRONG R/PYTHON

Đề tài:

Đánh giá rủi ro tín dụng của khách hàng

GVHD: Phạm Thị Thanh Xuân

Nhóm sinh viên	**MSSV**
Nguyễn Đức Minh Tấn	K194141745
Hà Mỹ Duyên	K194141719
Huỳnh Mỹ Nga	K194141734
Nguyễn Vũ Thiên	K194141750
Nguyễn Việt Thường	K194141754

Thành phố Hồ Chí Minh, ngày 31 tháng 5 năm 2022

Mục lục

1. Tổng quan dữ liệu

1.1. Mô tả dữ liệu

Bộ dữ liệu ban đầu 8 biến, bao gồm:

Biến mục tiêu:

+ **Default (Khả năng trả nợ)**
  1. Trả nợ đúng hạn
  2. Trả nợ không đúng hạn

Biến phụ thuộc:

+ **Work status (Tình trạng công việc)**
  1. Không chuyển việc trong vòng 2 năm
  2. Chuyển việc 1 lần trong vòng 2 năm
  3. Chuyển việc 2 lần trong vòng 2 năm
+ **Collateral (Tài sản đảm bảo)**
  1. Không có tài sản đảm bảo
  2. Có 1 phần tài sản đảm bảo
  3. Có đủ tài sản đảm bảo được bảo lãnh
  4. Có đủ tài sản đảm bảo của bản thân
+ **Age (Độ tuổi)**
  1: $X \leq 30$
  2: $30 < X \leq 50$
  3: $X > 50$
+ **Marriage (Tình trạng hôn nhân)**
  1. Độc thân
  2: Đã kết hôn
  3: Ly dị
+ **Loan purpose (Mục đích vay)**
  1: Vay phục vụ nhu cầu đời sống
  2: Vay phục vụ SXKD
+ **Sex (Giới tính)**
  0: Nam
  1: Nữ
+ **Income (Thu nhập):** triệu VND

1.2. Thống kê mô tả

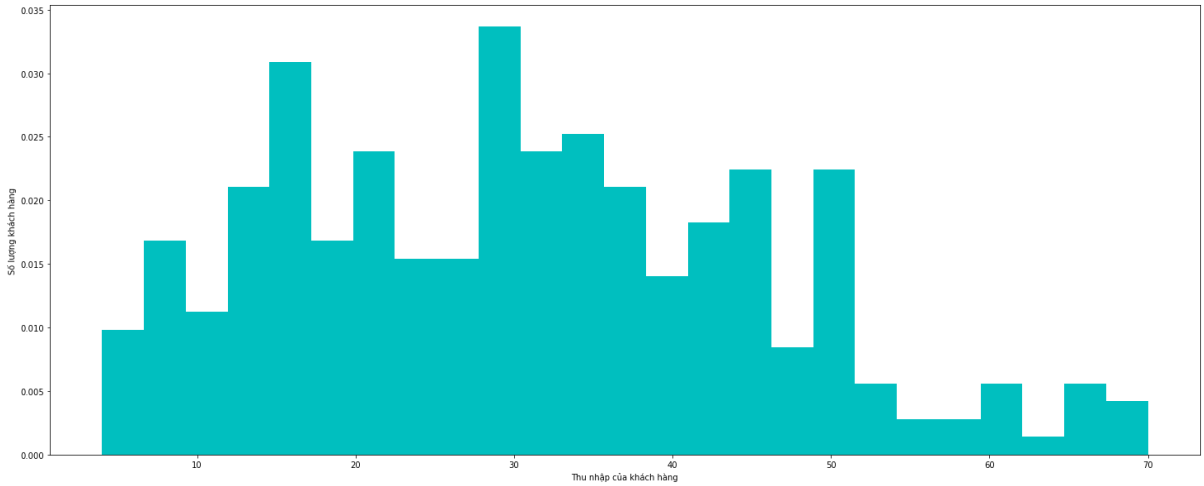
Bộ dữ liệu với 270 quan sát, có 1 biến liên tục là Income và 6 biến rời rạc là Work status, Collateral, Age, Marriage, Loan purpose, Sex. Tại biến liên tục có chỉ số thống kê như sau:

Bảng 1: Chỉ số thống kê của biến Income

Chỉ số	Income
Count	270

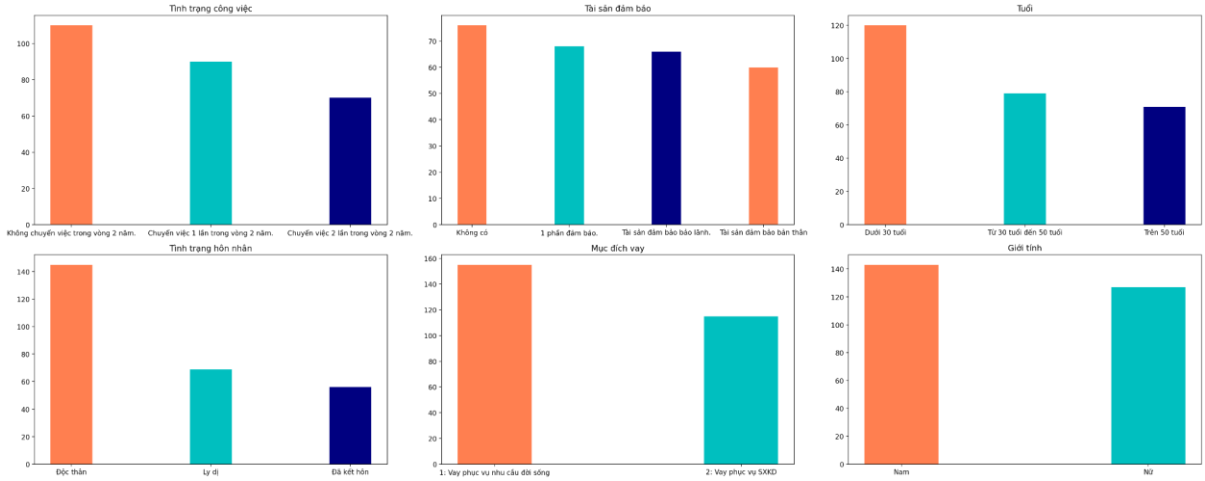
Mean	30,7
Std	15,06
Min	4
Max	70

và tổng quát hóa dữ liệu như hình dưới.



Biểu đồ 1. Biểu đồ tần số của biến liên tục Income

Nguồn: Tính toán của nhóm tác giả

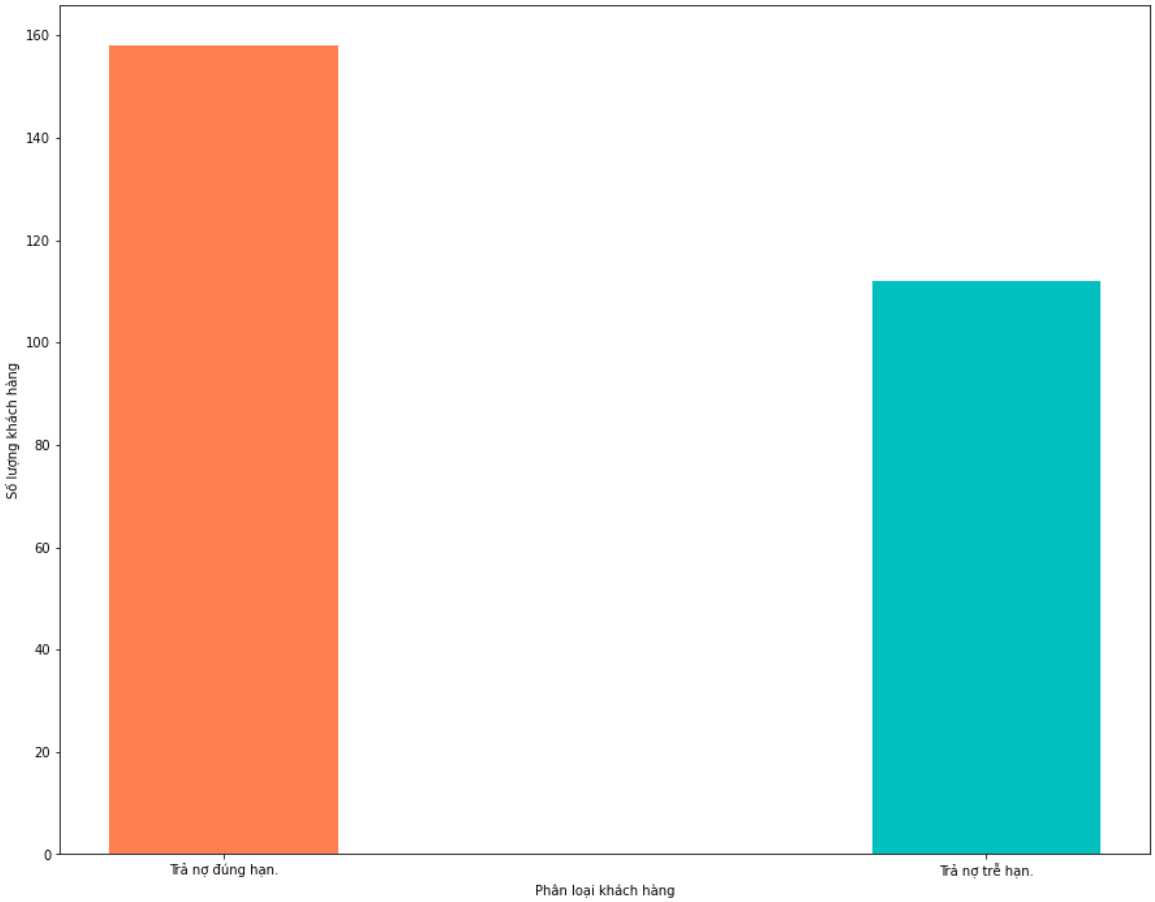


Biểu đồ 2. Tổng hợp biểu đồ mô tả của các biến rời rạc

Nguồn: Tính toán của nhóm tác giả

Biến Work status với tiêu chí Chuyển việc 2 lần trong vòng 2 năm xuất hiện nhiều nhất, tương tự với các biến Collateral là Có đủ tài sản đảm bảo của bản thân, Age là X > 50, Marriage là Ly dị, Loan purpose là Vay phục vụ SXKD và Sex là Nam có tần suất xuất hiện nhiều nhất.

Với biến mục tiêu Default có chỉ số thống kê như sau:



Biểu đồ 3. Biểu đồ mô tả của biến Default

Dựa theo hình trên, số lượng Trả nợ đúng hạn là 158 quan sát và Trả nợ trễ hạn là 112 quan sát.

Các mô hình trong bài báo cáo sẽ học dựa trên bộ train (80%) với các biến phụ thuộc, và dự đoán kết quả của biến mục tiêu với bộ test (20%). Sau đó, so sánh kết quả dự đoán với dữ liệu thực tế.

2. Kết quả ước tính các mô hình

2.1. Các tiêu chí đánh giá mô hình

2.1.1. Độ chính xác (Accuracy)

Cách đánh giá này đơn giản tính tỉ lệ giữa số điểm được dự đoán đúng và tổng số điểm trong tập dữ liệu kiểm thử. Accuracy càng cao thì mô hình càng có hiệu quả.

2.1.2. Ma trận bối rối (Confusion matrix)

Cách tính sử dụng accuracy như ở trên chỉ cho chúng ta biết được bao nhiêu phần trăm lượng dữ liệu được phân loại đúng mà không chỉ ra được cụ thể mỗi loại được phân loại như thế nào, lớp nào được phân loại đúng nhiều nhất, và dữ liệu thuộc lớp nào thường bị phân loại nhầm vào lớp khác. Để có thể đánh giá được các giá trị này, chúng ta sử dụng một ma trận được gọi là confusion matrix.

2.1.3. Precision và Recall

Với bài toán phân loại mà tập dữ liệu của các lớp là chênh lệch nhau rất nhiều, có một phép đó hiệu quả thường được sử dụng là Precision-Recall.

Trước hết xét bài toán phân loại nhị phân. Ta cũng coi một trong hai lớp là positive, lớp còn lại là negative.

Với một cách xác định một lớp là positive, Precision được định nghĩa là tỉ lệ số điểm true positive trong số những điểm được phân loại là positive (TP + FP).

$$Precision = \frac{True\ Positive}{(True\ Positive + False\ Positive)}$$

Recall được định nghĩa là tỉ lệ số điểm true positive trong số những điểm thực sự là positive (TP + FN).

$$Recall = \frac{True\ Positive}{(True\ Positive + False\ negative)}$$

Precision cao đồng nghĩa với việc độ chính xác của các điểm tìm được là cao. Recall cao đồng nghĩa với việc True Positive Rate cao, tức tỉ lệ bỏ sót các điểm thực sự positive là thấp.

Khi Precision = 1, mọi điểm tìm được đều thực sự là positive, tức không có điểm negative nào lẫn vào kết quả. Tuy nhiên, Precision = 1 không đảm bảo mô hình là tốt, vì câu hỏi đặt ra là liệu mô hình đã tìm được tất cả các điểm positive hay chưa. Nếu một mô hình chỉ tìm được đúng một điểm positive mà nó chắc chắn nhất thì ta không thể gọi nó là một mô hình tốt.

Khi Recall = 1, mọi điểm positive đều được tìm thấy. Tuy nhiên, đại lượng này lại không đo liệu có bao nhiêu điểm negative bị lẫn trong đó. Nếu mô hình phân loại mọi điểm là positive thì chắc chắn Recall = 1, tuy nhiên dễ nhận ra đây là một mô hình cực tồi.

Một mô hình phân lớp tốt là mô hình có cả Precision và Recall đều cao, tức càng gần 1 càng tốt.

2.1.4. F1-score

F1-score là trung bình điều hòa (harmonic mean) của precision và recall (giả sử hai đại lượng này khác 0). F1-score được tính theo công thức:

$$\frac{2}{f1\ score} = \frac{1}{precision} + \frac{1}{recall}$$

F1-score có giá trị nằm trong nửa khoảng (0,1], F1càng cao, bộ phân lớp càng tốt. Khi cả recall và precision đều bằng 1 (tốt nhất có thể), F1=1 khi cả recall và precision đều thấp.

2.1.5. Receiver Operating Characteristic curve (ROC curve)

Dựa trên ROC curve, ta có thể chỉ ra rằng một mô hình có hiệu quả hay không. Một mô hình hiệu quả khi có False Positive Rate thấp và True Positive Rate cao, tức

tồn tại một điểm trên ROC curve gần với điểm có toạ độ (0, 1) trên đồ thị (góc trên bên trái). Curve càng gần thì mô hình càng hiệu quả.

2.2. Logistic

Hồi quy logistic là một thuật toán phân loại máy học được sử dụng để dự đoán xác suất của biến phụ thuộc phân loại. Trong hồi quy logistic, biến phụ thuộc là một biến nhị phân chứa dữ liệu được mã hóa là 1 (có, thành công, v.v.) hoặc 0 (không, thất bại, v.v.). Nói cách khác, mô hình hồi quy logistic dự đoán p (y = 1) là một hàm của các biến X.

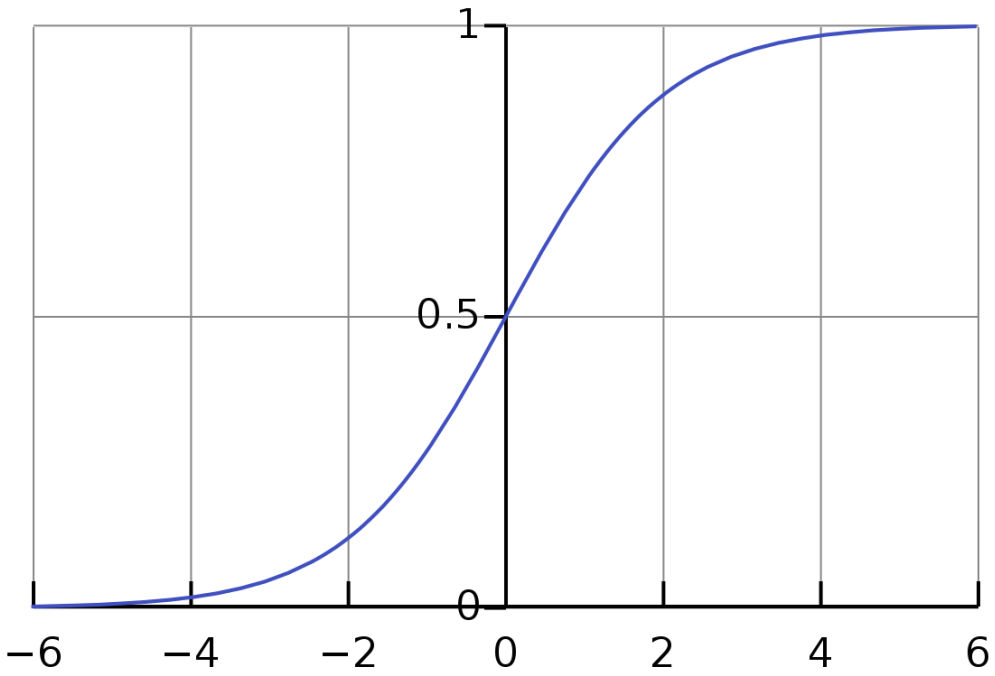
Là một kỹ thuật thống kê được giám sát để tìm xác suất của biến phụ thuộc (Các lớp có trong biến phụ thuộc). Hồi quy logistic sử dụng các hàm được gọi là hàm logit , giúp đưa ra mối quan hệ giữa biến phụ thuộc và các biến độc lập bằng cách dự đoán xác suất hoặc cơ hội xảy ra. Các hàm logistic hay còn được gọi là hàm sigmoid giúp chuyển đổi xác suất thành các giá trị nhị phân có thể được sử dụng thêm cho các dự đoán.

Ngay cả khi Hồi quy logistic thuộc về mô hình tuyến tính, nó không đưa ra bất kỳ giả định nào của mô hình hồi quy tuyến tính, như:

- Không yêu cầu mối quan hệ tuyến tính giữa các biến phụ thuộc và độc lập.
- Các điều khoản lỗi không cần phải được phân phối bình thường.
- Không bắt buộc phải có độ co giãn đồng nhất.

Tuy nhiên nó vẫn có các giả định riêng của một mô hình hồi quy Logistic hoàn chỉnh bao gồm:

- Hồi quy logistic nhị phân đòi hỏi biến phụ thuộc là nhị phân.
- Đối với hồi quy nhị phân, yếu tố cấp 1 của biến phụ thuộc phải thể hiện kết quả mong muốn.
- Chỉ có các biến có ý nghĩa nên được đưa vào.
- Mô hình nên có ít hoặc không có đa cộng tuyến.
- Các biến độc lập có liên quan tuyến tính với tỷ lệ cược logarit.
- Hồi quy logistic yêu cầu kích thước mẫu lớn.



Biểu đồ 4. Hàm Sigmoid Logistic

Nguồn: Internet

Đó là hàm sigmoid logistic tiêu chuẩn với L = 1, k = 1, x_0 = 0

Một hàm logistic hoặc đường cong logistic là một đường cong hình chữ S thông thường (đường cong sigmoid) với phương trình:

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$

Trong đó:
L: Giá trị tối đa của đường cong,
e: cơ số logarit tự nhiên (số Euler),
x_0: Giá trị x của điểm giữa của Sigmoid,
k: Tốc độ tăng trưởng của đường cong Logistic,

Về mặt thực tiễn, mô hình hồi quy logistic có 3 loại:

- Hồi quy logistic nhị phân.
- Hồi quy logistic đa hình.
- Hồi quy logistic thứ tự.

Và nhóm nhận thấy việc xây dựng mô hình dự báo rủi ro tín dụng này có sự kết hợp của các biến độc lập nhiều lớp (lớn hơn 2 lớp). Khi đó, Hồi quy logistic đa hình là giải pháp thích hợp cũng như là lí do để nhóm tác giả thử nghiệm và áp dụng cho mô hình dự báo lần này bởi loại hồi quy logistic này thích hợp để xử lý các biến mục tiêu hoặc biến độc lập có ba hoặc nhiều giá trị có thể.

Báo cáo phân loại cùng chỉ số accuracy để đánh giá hiệu suất chung của mô hình phân loại như sau:

Bảng 2: Confusion matrix của mô hình Logistic

	Predicted Negative	Predicted Positive
Actual Negative	20	0
Actual Positive	3	31

Nguồn: Tính toán của nhóm tác giả

Trong đó:

Negative: Trả nợ trễ hạn

Positive: Trả nợ đúng hạn

Trong bảng ma trận bối rối, giá trị đường chéo biểu thị các dự đoán chính xác, trong khi các phần tử không đường chéo là dự đoán không chính xác. Theo đó, 20 và 31 là dự đoán đúng với thực tế và 3 và 0 là dự đoán không chính xác.

Bên cạnh đó, trong 20 quan sát thực tế về các khách hàng trả nợ trễ hạn thì mô hình đã dự báo đúng hết 20 kết quả mà không có dự báo sai nào. Kết quả dự báo trả nợ trễ hạn đúng so với thực tế đạt được 100%. Ngược lại, trong 34 quan sát thực tế về các khách hàng có khả năng trả nợ đúng hạn thì có 31 dự báo đúng, 3 dự báo trả nợ trễ hạn. Kết quả dự báo trả nợ đúng hạn so với thực tế chiếm 91%. Từ đây, nhóm kỳ vọng vào việc phát hiện được những khách hàng có tiềm năng trả nợ trễ hạn để giúp ngân hàng có biện pháp phòng ngừa rủi ro khi cho vay. Sự kỳ vọng này sẽ được xem xét và đánh giá thông qua thống kê kết quả mô hình bên dưới.

Bảng 3: Báo cáo mô hình Logistic

	Precision	**Recall**	**f1-score**	**support**
0	0.87	1.00	0.93	20
1	1.00	0.91	0.95	34
accuracy			0.94	54
marco avg	0.93	0.96	0.94	54
weighted avg	0.95	0.94	0.95	54

Độ chính xác dự báo của mô hình Logistic là 0.9444.

Nhìn chung, có thể thấy rằng tuy mô hình Logistics có hiệu quả cao, đạt được hơn 94%, nhưng thứ ta cần quan tâm hơn bên cạnh đó chính là chỉ số Recall của việc báo khách hàng trả nợ trễ hạn. Thông qua bảng báo cáo, Recall đạt hiệu quả lên đến 100%, điều này rất đáng ngạc nhiên vì chứng tỏ rằng trong số các khách hàng trả nợ trễ hạn trong thực tế thì việc dự báo của chúng ta không bỏ sót bất kỳ khách hàng nào. Điều này vô cùng hữu ích cho ngân hàng trong việc rà soát cũng như xác định các khách hàng có tiềm năng tín dụng xấu. Cùng với đó, mô hình còn được củng cố thêm độ tin cậy với

chỉ số Precision cao khoảng 87%. Điều này có nghĩa là trong số các dự báo, mô hình dự báo đúng tới 87% các khách hàng trễ hạn. Hơn nữa, chỉ số f1-score còn là thước đo cho mức độ phù hợp của mô hình đối với bộ dữ liệu này, đạt được trên 93%.

Đối với tập dữ liệu trong bài báo cáo này, mô hình đã trả về kết quả dự báo đúng như kỳ vọng cũng như hiệu quả mô hình tương đối tốt. Điều này cũng được lý giải bởi bộ dữ liệu đầu vào đã được làm sạch và xử lý bằng nhiều cách khác nhau nhằm mục đích tối ưu kết quả dự báo. Tuy nhiên, nó vẫn đi kèm với những hạn chế của riêng nó. Đối với bộ dữ liệu mới hoàn toàn và chưa có những bước xử lý "làm đẹp", hồi quy Logistic sẽ không thể xử lý một lượng lớn các tính năng phân loại. Nhưng nếu các tính năng này quan trọng cho việc dự báo, ta sẽ buộc phải giữ lại chúng, nhưng sau đó mô hình sẽ không mang lại độ chính xác tốt. Hiện tượng quá khớp cũng là vấn đề thường gặp đối với mô hình Logistic. Nó không thể được áp dụng cho một bài toán phi tuyến tính, hoạt động kém với các biến độc lập không tương quan với mục tiêu và tương quan với nhau. Do đó, với những hạn chế như thế, ta sẽ phải đánh giá cẩn thận mức độ phù hợp của hồi quy Logistic đối với vấn đề đang cố gắng giải quyết.

2.3. Decision Tree

Mô hình được hình thành từ các tập con, mỗi tập con là một nhánh, các nhánh hoàn toàn độc lập với nhau, thể hiện giá trị cho một thuộc tính cần kiểm tra. Khi tập con tách thành các tập con nhỏ hơn sẽ hình thành nút quyết định. Nút đại diện cho toàn bộ tập dữ liệu là nút gốc (root node). Điểm đặc biệt của cây quyết định là có thể xử lý được dữ liệu số và dữ liệu phân loại. Cây quyết định thường dùng trong bài toán phân loại và hồi quy. Đối với phương pháp hồi quy, mô hình sẽ dự đoán một giá trị thay vì dự đoán một lớp trong mỗi nút như bài toán phân loại. Cốt lõi của phương pháp này là sử dụng thuật toán xác suất nhằm phân tách ra các trường hợp theo đúng điều kiện được đặt ra và giảm thiểu sai số. Trong Cây quyết định dạng hồi quy, thuật toán CART được sử dụng để nó cố gắng chia tập huấn luyện theo cách giảm thiểu MSE thay vì cố gắng phân chia tập huấn luyện theo cách giảm thiểu điểm dữ liệu không có ý nghĩa. Hàm chi phí mà thuật toán cố gắng giảm thiểu:

$$J(k, t_k) = \frac{m_{left}}{m} MSE_{left} + \frac{m_{right}}{m} MSE_{right}$$

$$Trong\ đó \begin{cases} MSE_{node} = \sum_{i \in node} (\hat{y}_{node} - y^{(i)})^2 \\ \hat{y}_{node} = \frac{1}{m_{node}} \sum_{i \in node} y^{(i)} \end{cases}$$

Bảng 4: Confusion matrix mô hình Decision Tree

	Predicted Negative	Predicted Positive
Actual Negative	17	3
Actual Positive	2	32

Trong bảng ma trận bối rối, giá trị đường chéo biểu thị các dự đoán chính xác, trong khi các phần tử không đường chéo là dự đoán không chính xác. Theo đó, 17 và 32 là dự đoán đúng với thực tế và 3 và 2 là dự đoán không chính xác.

Bên cạnh đó, trong 20 quan sát thực tế về các khách hàng trả nợ trễ hạn thì có 17 dự báo đúng, 3 dự báo là trả đúng hạn. Kết quả dự báo trả nợ trễ hạn đúng so với thực tế đạt được 85%. Ngược lại, trong 34 quan sát thực tế về các khách hàng có khả năng trả nợ đúng hạn thì có 32 dự báo đúng, 2 dự báo trả nợ trễ hạn. Kết quả dự báo trả nợ đúng hạn so với thực tế chiếm 94%. Tỷ lệ dự báo trả nợ trễ hạn thấp hơn những nhóm còn lại. Đây cũng chính là nguyên nhân làm giảm tỷ lệ chính xác so với mô hình Logistic vừa thực hiện.

Bảng 5: Báo cáo mô hình Decision Tree

	Precision	**Recall**	**f1-score**	**support**
0	0.89	0.85	0.87	20
1	0.91	0.94	0.93	34
accuracy			0.91	54
marco avg	0.90	0.90	0.90	54
weighted avg	0.91	0.91	0.91	54

Mô hình Decision tree có hiệu quả cao nhưng kém hơn Logistic, chỉ số accuracy đạt được 90,74%, bởi chỉ số Recall của việc dự báo khách hàng trả nợ trễ hạn nhưng chỉ số này lại kém hơn mô hình Logistic (85%<95% của Logistic), còn một vài dự báo sai. Cùng với đó, mô hình còn được củng cố thêm độ tin cậy với chỉ số Precision cao khoảng 89%. Điều này có nghĩa là trong số các dự báo, mô hình dự báo đúng tới 89% các khách hàng trễ hạn. Hơn nữa, chỉ số f1-score còn là thước đo cho mức độ phù hợp của mô hình đối với bộ dữ liệu này, đạt được trên 87%. Mô hình cần phải hiệu chỉnh để đạt kết quả tốt hơn.

Một số cách cải thiện độ chính xác của mô hình là tăng số cấp độ của cây: Độ chính xác của cây quyết định có thể thay đổi dựa trên độ sâu của cây quyết định. Trong nhiều trường hợp, lá của cây là các nút thuần túy. Khi một nút thuần túy, điều đó có nghĩa là tất cả dữ liệu trong nút đó thuộc về một lớp duy nhất. Cây sâu hơn có thể ảnh hưởng đến thời gian chạy theo cách tiêu cực. Nếu một thuật toán phân loại nhất định đang được sử dụng, thì một cây sâu hơn có thể có nghĩa là thời gian chạy của thuật toán phân loại này chậm hơn đáng kể.- Cũng có khả năng là thuật toán thực sự xây dựng cây quyết định sẽ chậm hơn đáng kể khi cây càng sâu. Nếu thuật toán xây dựng cây đang được sử dụng chia tách các nút thuần túy, thì độ chính xác tổng thể sẽ giảm của bộ phân loại cây có thể được trải nghiệm. Đôi khi, việc đi sâu hơn vào cây có thể làm giảm độ

chính xác nói chung, vì vậy điều rất quan trọng là kiểm tra việc sửa đổi độ sâu của cây quyết định và chọn độ sâu tạo ra kết quả tốt nhất.

Dùng chỉ số lá tối đa "max-depth" để xác định độ sâu tối đa của cây. Thông số độ sâu là một trong những cách mà chúng ta có thể điều chỉnh cây hoặc giới hạn cách nó phát triển để ngăn chặn overfitting.

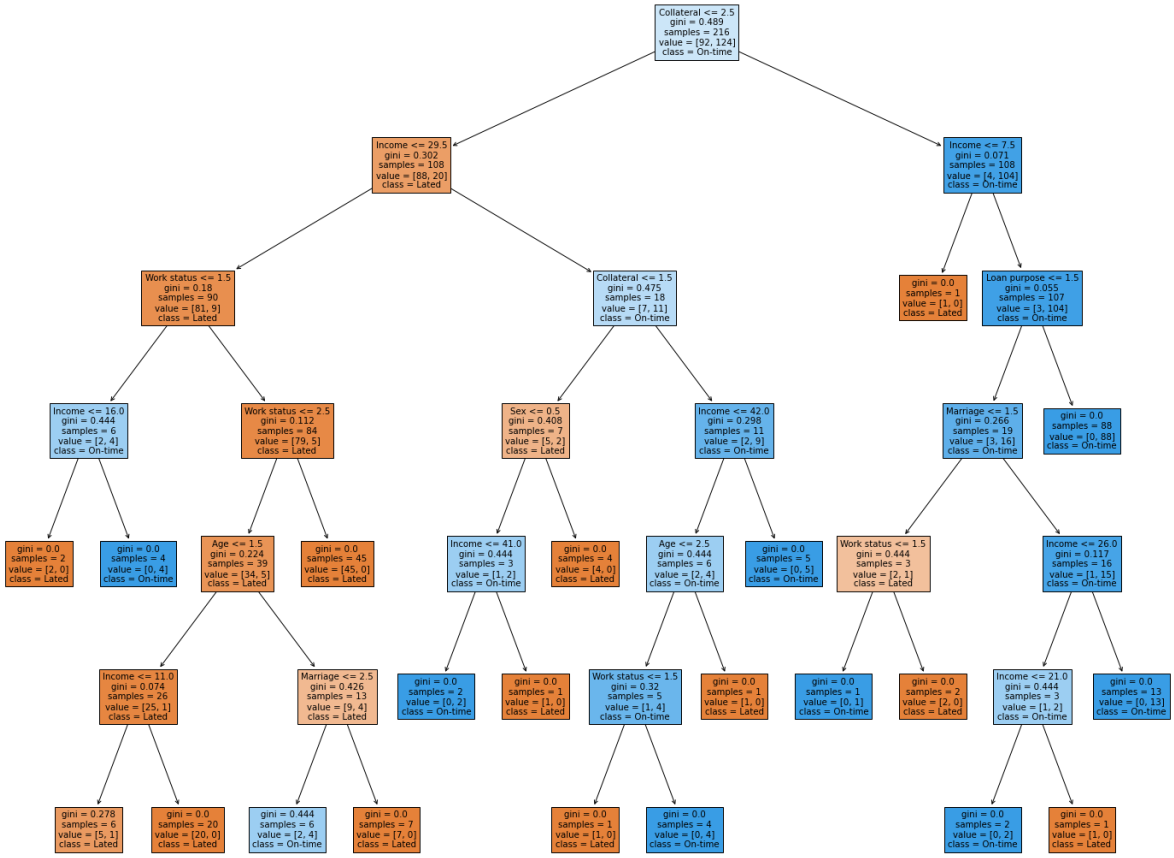
Bảng 6: Confusion matrix mô hình Decision Tree sau khi hiệu chỉnh

	Predicted Negative	Predicted Positive
Actual Negative	17	3
Actual Positive	1	33

Bảng 7: Báo cáo mô hình Decision Tree sau khi hiệu chỉnh

	Precision	**Recall**	**f1-score**	**support**
0	0.94	0.85	0.89	20
1	0.92	0.97	0.94	34
accuracy			0.93	54
marco avg	0.93	0.91	0.92	54
weighted avg	0.93	0.93	0.93	54

Ta sẽ thu được chỉ số accuracy với sáu lá đối đa "max_depth" là 92.59% cao hơn kết quả khi chưa thực hiện hiệu chỉnh. Tuy cải thiện được độ chính xác nhưng chỉ số recall vẫn giữ nguyên, tức mô hình này vẫn chưa là mô hình tốt nhất để dự báo bộ dữ liệu rủi ro trong tín dụng tài chính - ngân hàng.



Biểu đồ 5. Mô hình cây quyết định

Nguồn: Tính toán của nhóm tác giả

Từ hình ảnh cây quyết định, chúng ta có thể xây dựng nên các nguyên tắc cho vay đến các nhân viên tín dụng như sau:

(1) Người có tài sản đảm bảo được bảo lãnh hoặc của bản thân, có thu nhập lớn hơn 7,5 triệu, vay vì mục đích sản xuất kinh doanh thì quyết định cho vay. Nếu họ vay vì mục đích nhu cầu đời sống thì họ phải đã kết hôn hoặc ly hôn và thu nhập lớn hơn 26 triệu thì mới cho vay.

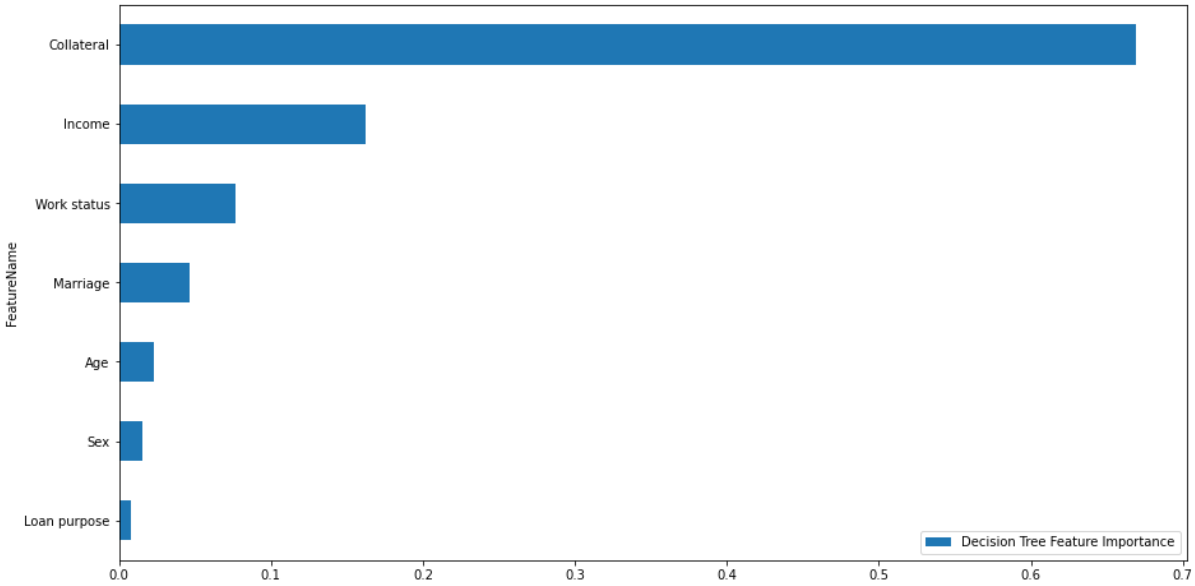
(2) Nếu người này không có tài sản đảm bảo hay chỉ có 1 phần tài sản đảm bảo, thu nhập nhỏ hơn 29,5 triệu, chuyển việc ít nhất 2 lần trong vòng 2 năm thì không cho vay. Hoặc chuyển việc ít nhất 1 lần trong vòng 2 năm và nhỏ hơn 30 tuổi thì cũng không cho vay.

Bảng 8: Các thuộc tính quan trọng của mô hình Decision Tree

	Feature Name	Decision Tree Feature Importance
1	Collateral	0.668919
6	Income	0.162360
0	Work status	0.076496
3	Marriage	0.046557

2	Age	0.022933
5	Sex	0.015043
4	Loan Purpose	0.007692

Nguồn: Tính toán của nhóm tác giả



Biểu đồ 6. Trực quan hóa các thuộc tính quan trọng của mô hình Decision tree

Nguồn: Tính toán của nhóm tác giả

Nhóm cũng xác định được tầm quan trọng của các biến: Collateral, Income, Work Status là 3 biến quan trọng, có ảnh hưởng mạnh mẽ đến việc dự báo kết quả của mô hình. Chỉ cần 3 biến này sai với quy luật dự báo thì kết quả dự báo sẽ sai.

Sau khi ra kết quả phân loại khách hàng, ta có thể so sánh kết quả dự báo và kết quả thực tế để tìm ra các nhược điểm mà mô hình còn tồn tại thông qua bảng 9.

Bảng 9: Khách hàng dự báo sai của mô hình Decision Tree

Feature Name	Khách hàng số 25	Khách hàng số 209	Khách hàng số 227	Khách hàng số 245
Default	1	0	0	0
Work status	2	1	2	3
Collateral	2	1	1	1
Age	1	1	3	2
Marriage	2	3	2	3
Loan Purpose	2	1	1	1
Sex	0	0	0	0

Income	29	17	12	37

Nguồn: Tính toán của nhóm tác giả

- Khách hàng số 25: Kết quả dự báo sai với tập test, khách hàng chỉ có 1 phần tài sản đảm bảo (Đây là yếu tố quyết định đầu tiên của cây quyết định), chỉ với một thuộc tính đã gây ra sự sai lệch kết quả dự báo, cho thấy sự quan trọng và tương quan mạnh mẽ của biến Collateral.
- Khách hàng số 209: Kết quả dự báo sai là do khách hàng không chuyển việc trong vòng 2 năm qua, đã ly hôn, và là nam. Với 3 thuộc tính sai, kết quả dự báo của mô hình sẽ bị sai lệch so với tập test.
- Khách hàng số 227: Kết quả dự báo sai với tập test, khách hàng này không có tài sản đảm bảo, chuyển việc ít nhất 1 lần trong vòng 2 năm, vay vì mục đích nhu cầu đời sống, thu nhập 12 triệu (nhỏ hơn 30 triệu) phù hợp một phần với quy luật dự báo. Tuy nhiên, có thể do độ tuổi, giới tính và tình trạng hôn nhân đã ảnh hưởng đến dự báo, nhưng 3 biến này có ảnh hưởng không lớn đến biến "default".
- Khách hàng số 245: Khách hàng không có tài sản đảm bảo, chuyển việc ít nhất 2 lần trong vòng 2 năm, vay vì mục đích nhu cầu đời sống phù hợp một phần với quy luật dư báo, nhưng khách hàng có thu nhập trên 29,5 triệu (cụ thể là 37 triệu), độ tuổi từ 30 đến 50 tuổi, đã ly hôn, và là nam. Dự báo này sai 4 thuộc tính, trong đó có thuộc tính tình trạng công việc và thu nhập tác động mạnh mẽ đến kết quả dự báo này.

Mô hình còn hạn chế khi dự báo sai các thuộc tính không quan trọng, khiến độ chính xác của mô hình giảm. Nhận định lại mô hình, ta thấy mô hình sẽ dự báo sai khi sai 3 thuộc tính trở lên, điều này là hợp lý vì dữ liệu chỉ dùng 6 biến nhưng sai trên một nửa. Tuy nhiên, mô hình quá phụ thuộc vào biến Collateral, chỉ cần thuộc tính này sai thì kết quả dự báo về khách hàng sẽ sai.

2.4. Random Forest

Random Forests là thuật toán học có giám sát "supervised learning". Nó có thể được sử dụng cho cả phân lớp và hồi quy. Nó cũng là thuật toán linh hoạt và dễ sử dụng nhất. Một khu rừng bao gồm cây cối. Người ta nói rằng càng có nhiều cây thì rừng càng mạnh. Random forests tạo ra cây quyết định trên các mẫu dữ liệu được chọn ngẫu nhiên, được dự đoán từ mỗi cây và chọn giải pháp tốt nhất bằng cách bỏ phiếu. Nó cũng cung cấp một chỉ báo khá tốt về tầm quan trọng của tính năng. Random forests có nhiều ứng dụng, chẳng hạn như công cụ đề xuất, phân loại hình ảnh và lựa chọn tính năng. Nó có thể được sử dụng để phân loại các ứng viên cho vay trung thành, xác định hoạt động gian lận và dự đoán các bệnh. Nó nằm ở cơ sở của thuật toán Boruta, chọn các tính năng quan trọng trong tập dữ liệu. Random Forest là một cách lấy trung bình nhiều cây quyết định sâu, được huấn luyện trên các phần khác nhau của cùng một tập huấn luyện, với mục tiêu giảm phương sai. Điều này phải trả giá bằng một sự gia tăng nhỏ trong độ

chệch và một số mất khả năng diễn giải, nhưng nói chung là tăng đáng kể hiệu suất trong mô hình cuối cùng. Rừng giống như sự kết hợp của các nỗ lực thuật toán cây quyết định. Thực hiện việc làm việc theo nhóm của nhiều cây do đó cải thiện hiệu suất của một cây ngẫu nhiên duy nhất. Mặc dù không hoàn toàn giống nhau, nhưng các khu rừng mang lại hiệu quả của xác thực chéo gấp K lần.

Thuật toán đào tạo cho các khu rừng ngẫu nhiên áp dụng kỹ thuật tổng hợp bootstrap hoặc đóng gói chung cho những người học cây. Cho một tập huấn luyện $X = x_1, ..., x_n$ với các phản hồi $Y = y_1, ..., y_n$, đóng gói lặp đi lặp lại (B lần) chọn một mẫu ngẫu nhiên thay thế tập huấn luyện và lắp các cây vào các mẫu:

Đối với $b = 1, ..., B$:

- Ví dụ huấn luyện mẫu, với thay thế, n từ X, Y; gọi chúng là X_b, Y_b.
- Huấn luyện cây phân loại hoặc hồi quy f_b trên X_b, Y_b.

Sau khi huấn luyện, có thể thực hiện dự đoán cho các mẫu chưa nhìn thấy x' bằng công thức:

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$$

hoặc bằng cách lấy đa số phiếu trong trường hợp phân loại cây.

Sau thuật toán Decision Tree, khi xây dựng cây quyết định nếu để độ sâu tùy ý thì cây sẽ phân loại đúng hết với các dữ liệu tận training dẫn đến mô hình có thể dự đoán không tốt trên tập test, khi đó mô hình bị overfitting (hay high variance). Nhóm sử dụng thuật toán Random Forest gồm nhiều cây quyết định, mỗi cây đều có yếu tố ngẫu nhiên nhằm cho ra kết quả dự đoán tốt nhất. Bên cạnh đó, Random forests cũng cung cấp một chỉ số lựa chọn tính năng tốt. Scikit-learn cung cấp thêm một biến với mô hình, cho thấy tầm quan trọng hoặc đóng góp tương đối của từng tính năng trong dự đoán. Vì thế, nhóm đã thực hiện thuật toán này nhằm tối ưu kết quả nghiên cứu.

Bảng 10: Ma trận bối rối mô hình Random Forest

	Predicted Negative	Predicted Positive
Actual Negative	20	0
Actual Positive	2	32

Nguồn: Tính toán của nhóm tác giả

Bảng 11: Báo cáo mô hình Random Forest

	Precision	**Recall**	**f1-score**	**support**
0	0.91	1	0.95	20

1	1	0.94	0.97	34
accuracy			0.96	54
marco avg	0.95	0.97	0.96	54
weighted avg	0.97	0.96	0.96	54

Nguồn: Tính toán của nhóm tác giả

Với tỷ lệ chính xác đạt 96.3%, rất cao. Nên kết quả dự đoán gần như giống với thực tế.

Hiệu chỉnh mô hình sao cho tìm được một giá trị lá tối đa "max dept" ngẫu nhiên để có độ chính xác "accuracy" cao nhất. Sau khi thực hiệu hiệu chỉnh mô hình, nhóm tác giả nhận thấy rằng độ chính xác của dự báo không thay đổi.

Nhóm đưa ra một vài nhận định đầu tiên về quy luật dự báo của mô hình Random Forest như ở mô hình thuật toán Decision Tree như sau : Đối với một khách hàng có các đặc điểm không chuyển việc trong vòng 2 năm, có tài sản đảm bảo được bảo lãnh hoặc của bản thân, độ tuổi nằm trong khoản 30 đến 50 tuổi hoặc sau 50 tuổi, độc thân hoặc đã kết hôn, khoản vay dùng vào mục đích sản xuất kinh doanh, giới tính nam, thu nhập trên 30 triệu thì trả nợ đúng hạn.

Bảng 12: Khách hàng dự báo sai của mô hình Random Forest

Feature Name	Khách hàng số 43	Khách hàng số 47
Default	1	1
Work status	2	2
Collateral	1	2
Age	2	3
Marriage	1	1
Loan Purpose	2	2
Sex	1	1
Income	15	16

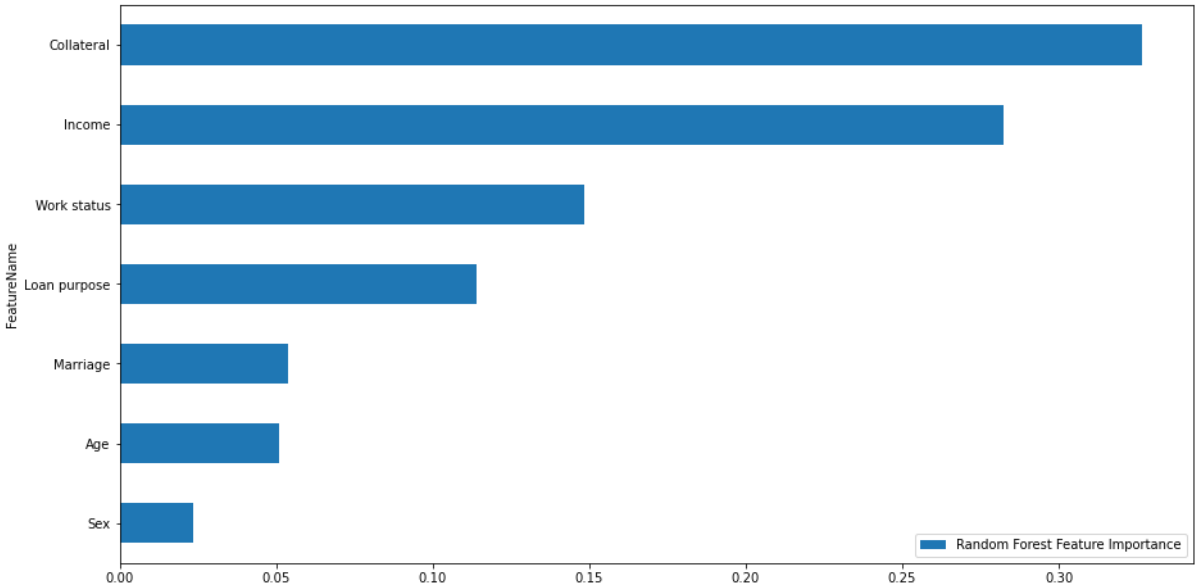
Nguồn: Tính toán của nhóm tác giả

Tìm vị trí dự báo sai trong bộ dữ liệu và in ra dữ liệu của cột dự báo sai đó. Mô hình có 2 vị trí dự báo sai ở vị trí khách hàng số 43 và khách hàng số 47. So với mô hình Decision tree với 4 vị trí dự đoán sai thì mô hình Random Forest đã giảm số lượng dự đoán sai xuống 50% chỉ còn 2 vị trí. Điều đó có thể nhìn thấy được phần nào tối ưu của mô hình thuật toán này.

Bảng 13: Các thuộc tính quan trọng của mô hình Random Forest

	FeatureName	Decision Tree Feature Importance
1	Collateral	0.326468
6	Income	0.282443
0	Work status	0.148481
4	Loan purpose	0.113922
3	Marriage	0.053955
2	Age	0.051038
5	Sex	0.023692

Nguồn: Tính toán của nhóm tác giả



Biểu đồ 7. Trực quan hóa các thuộc tính quan trọng của mô hình Random Forest

Nguồn: Tính toán của nhóm tác giả

Hình dung tầm quan trọng của các thuộc tính bằng hình ảnh dễ quan sát và hiểu hơn. Ta sử dụng thư viện Matplotlib cho phần visualizations. Điều này giúp ta có được cái nhìn tổng quan hơn đối với các thuộc tính. Đối với những thuộc tính có độ ảnh hưởng rất nhỏ hoặc không đáng kể, ta có thể loại bỏ các tính năng ít quan trọng nhất từ đó giúp độ chính xác tăng lên vì đã xóa dữ liệu gây hiểu lầm và bias, dẫn đến độ chính xác tăng lên và cũng giảm thời gian đào tạo mô hình. Từ biểu đồ trên, có thể thấy được 3 biến quan trọng sẽ ảnh hưởng trực tiếp đến kết quả mô hình : Biến Collateral, biến Income và biến Work status.

3. Dự báo khách hàng mới

Nhóm tác giả dự báo kết quả các mô hình dựa trên tập dữ liệu khách hàng mới nhằm so sánh kết quả dự báo rằng có sự khác nhau giữa các mô hình hay không.

Bảng 14: Kết quả dự báo tập hợp dữ liệu khách hàng mới

Khách hàng	Logistic regression	Desition tree	Random forest
0	1	1	1
1	0	0	0
2	0	0	0
3	1	1	1
…..			
45	1	1	1
46	0	0	0
47	1	1	1
48	1	1	1

Nguồn: Tính toán của nhóm tác giả

Bảng 15: Kết quả dự báo khác biệt giữa 3 mô hình

Khách hàng	Logistic regression	Decision tree	Random forest
8	1	0	0
13	1	0	0
37	0	1	1

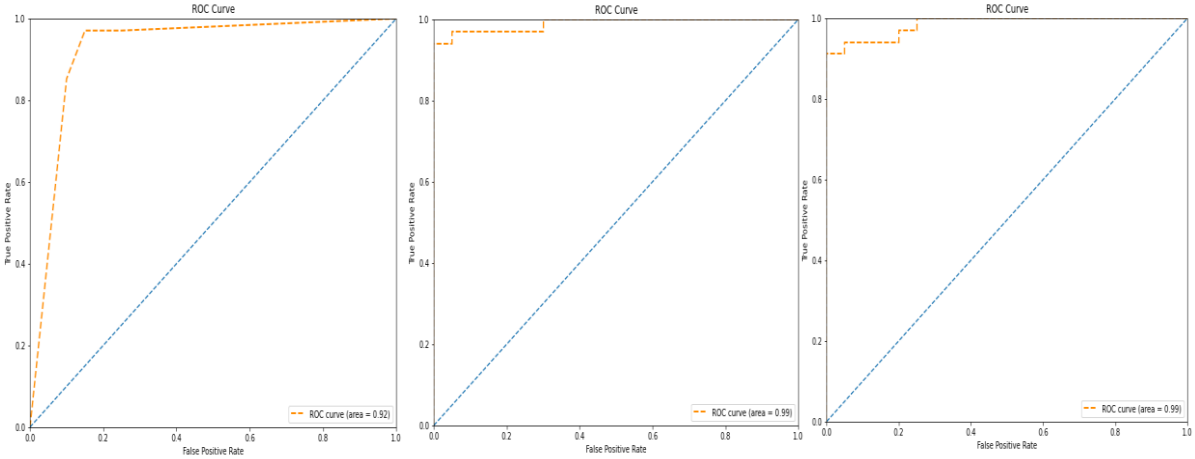
Nguồn: Tính toán của nhóm tác giả

Kết quả cho thấy rằng có sự khác biệt giữa các giá trị dự báo giữa ba mô hình. Tuy nhiên sự khác biệt này là tương đối nhỏ. Trong đó kết quả của mô hình cây quyết định và rừng ngẫu nhiên cho ra kết quả là giống nhau. Điều này là hoàn toàn hợp lý vì mô hình rừng ngẫu nhiên được xây dựng dựa trên thuật toán của cây quyết định.

4. Kết luận

Nhóm nghiên cứu sử dụng đa dạng các mô hình thuật toán nhằm cân nhắc lựa chọn mô hình tối ưu nhất để dự đoán khả năng trả nợ của khách hàng cá nhân.

Dựa vào các kết quả tính toán ROC curve của ba mô hình, nhóm tác giả đưa ra một số nhận định tổng quan về hiệu năng dự báo của mô hình Hồi quy Logistic, Decision Tree và Random Forest.



Biểu đồ 8. ROC curve lần lượt của Decision Tree, Logistic, Random Forest

Từ ba biểu đồ đường cong ROC của ba mô hình trên, có thể thấy hai mô hình Hồi quy Logistic và Random Forest gần như hoàn hảo rằng tất cả chúng đều cách rất xa đường cơ sở. Cả hai thuật toán này đều rất tốt, với điểm AUC là 0,99. Trong khi Decision Tree có phần kém hiệu quả hơn nhưng nhìn chung vẫn mang lại kết quả không tệ với điểm AUC chỉ 0,92.

Dựa vào bảng kết quả chạy mô hình, nhóm tác giả lựa chọn mô hình có độ chính xác cao nhất với tỉ lệ 96.3% của mô hình thuật toán Random Forest. Điều đó đồng nghĩa với việc tỷ lệ dự đoán sai khả năng trả nợ của khách hàng cá nhân là thấp nhất. Tuy nhiên, việc ứng dụng kết quả của mô hình vào thực tế thì ta cần quan tâm đặc biệt đến độ chính xác của việc dự đoán khả năng trả nợ không đúng hạn của khách hàng nhằm giảm thiểu rủi ro không đáng có cho ngân hàng khi quyết định cho vay. Từ đó, nhóm tác giả quan tâm đến tỉ lệ Recall, Precision của lớp 0 (Khả năng trả nợ không đúng hạn) nào là cao nhất trong tất cả các mô hình đã chạy để lựa chọn mô hình tối ưu nhất.

Bảng 16: Tổng hợp kết quả ba mô hình

Mô hình	Precision	Recall	Accuracy
Logistic (1)	0.87	1	94.44%
Decision Tree (2)	0.94	0.85	92.59%
Random Forest (3)	0.91	1	96.30%

Nguồn : Kết quả nhóm tác giả tổng hợp dựa trên bộ dữ liệu xử lý

Vậy với mô hình Random Forest có tỉ lệ "recall" = 1 là cao nhất và độ chính xác 96.3% là mô hình phù hợp và cho kết quả dự đoán tốt nhất.