TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO MÔN HỌC

Đề Tài: Phân Tích Và Dự Đoán Rủi Ro Tín Dụng

Môn học: Phân tích dữ liệu
GVHD: ThS. Hồ Hướng Thiên
Lớp: DH22IM-01

SINH VIÊN THỰC HIỆN:
Lương Minh Thông – 2254050064
Nguyễn Thị Hạnh Quyên – 2254050056
Nguyễn Thị Phương Oanh – 2254052057

Thành phố Hồ Chí Minh, 2024

LỜI MỞ ĐẦU

Trong bối cảnh nền kinh tế hiện đại, việc quản lý rủi ro tín dụng đóng vai trò quan trọng đối với các tổ chức tài chính, đặc biệt là ngân hàng và công ty cho vay. Với sự gia tăng số lượng giao dịch tín dụng và các khoản vay, các tổ chức này ngày càng phải đối mặt với thách thức trong việc xác định khách hàng có nguy cơ vỡ nợ, từ đó đưa ra quyết định tín dụng phù hợp nhằm giảm thiểu tổn thất tài chính.

Phân tích dữ liệu đang trở thành công cụ mạnh mẽ giúp giải quyết bài toán này. Việc áp dụng các kỹ thuật phân tích dữ liệu và học máy không chỉ giúp cải thiện khả năng dự đoán rủi ro tín dụng mà còn cung cấp thông tin chi tiết về các yếu tố ảnh hưởng đến hành vi tín dụng của khách hàng. Đây là bước tiến quan trọng trong việc chuyển đổi từ các phương pháp truyền thống dựa trên kinh nghiệm sang các giải pháp dựa trên dữ liệu, chính xác và minh bạch hơn.

Ngoài ra, đề tài này còn mang tính thực tiễn cao, góp phần hỗ trợ các tổ chức tài chính tối ưu hóa quy trình ra quyết định, giảm thiểu rủi ro hệ thống và cải thiện trải nghiệm khách hàng. Việc nghiên cứu và phát triển mô hình dự đoán rủi ro tín dụng cũng tạo ra cơ hội lớn cho việc áp dụng các kỹ thuật tiên tiến như khai phá dữ liệu, trực quan hóa dữ liệu và học máy trong lĩnh vực tài chính, từ đó nâng cao khả năng cạnh tranh trên thị trường.

Với ý nghĩa thực tiễn và tiềm năng ứng dụng cao, đề tài "Credit Risk Analysis and Prediction" mà nhóm chúng em lựa chọn tìm hiểu là một hướng nghiên cứu có thể phát triển xa hơn và áp dụng được ở thị trường ngân hàng Việt Nam trong tương lai gần.

Với sự hướng dẫn tận tình qua các bài giảng của thầy, nhóm đã thực hiện được bản báo cáo này. Trong quá trình hoàn thành bài báo cáo, chắc chắn không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự thông cảm và đóng góp ý kiến từ thầy để nhóm có những bài báo cáo hoàn thiện hơn trong tương lai.

I) Phân chia nhiệm vụ

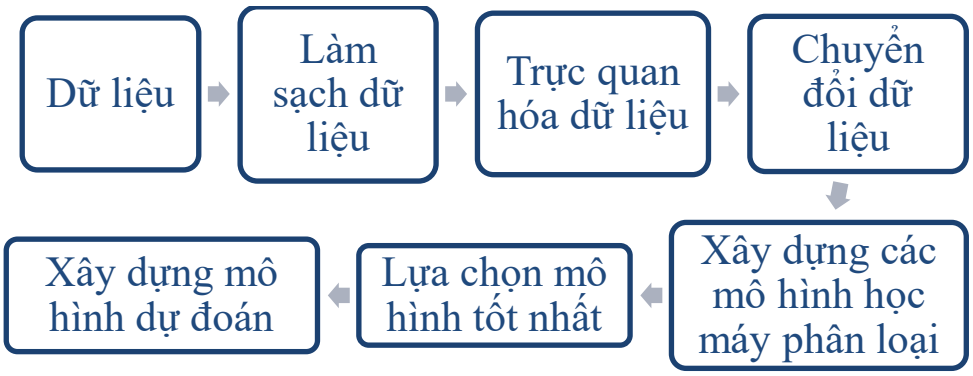
Tên SV	MSSV	Nhiệm vụ	Mức độ hoàn thành
Lương Minh Thông	2254050064	Tiền xử lý dữ liệu, xây dựng mô hình dự đoán	100%
Nguyễn Thị Hạnh Quyên	2254050056	Trực quan hóa dữ liệu	100%
Nguyễn Thị Phương Oanh	2254052057	Xây dựng các thuật toán machine learning	100%

II) Thông tin về bài báo cáo

1. Mục tiêu nghiên cứu

- Tìm hiểu các yếu tố ảnh hưởng xấu đến rủi ro tín dụng.
- Phân tích xu hướng và hành vi tín dụng của người vay dựa trên bộ dữ liệu nghiên cứu để xác định các đặc điểm chính liên quan đến rủi ro tín dụng từ đó hỗ trợ việc ra quyết định phê duyệt khoản vay.
- Đánh giá hiệu suất của các thuật toán học máy (machine learning) áp dụng cho dự đoán rủi ro tín dụng.
- Xây dựng mô hình dự đoán người vay có khả năng vỡ nợ hay không dựa trên việc sử dụng phương pháp học máy machine learning.

2. Quy trình thực hiện các phương pháp phân tích trong bài nghiên cứu



3. Thông tin về bộ dữ liệu nghiên cứu

"Credit Risk Dataset " là bộ dữ liệu mô phỏng từ các ngân hàng tín dụng về các yếu tố liên quan đến rủi ro tín dụng được thu thập từ trang Kaggle với 32,582 dòng dữ liệu và 12 biến đặc trưng. Dưới đây là bảng mô tả dữ liệu, liệt kê tên các đặc trưng (feature) và giải thích ý nghĩa của từng đặc trưng:

Cột	Mô tả
person_age	Tuổi của cá nhân nộp đơn xin vay.
person_income	Thu nhập hàng năm của cá nhân.
person_home_ownership	Tình trạng sở hữu nhà của người vay. **RENT**: Thuê nhà **OWN**: Sở hữu nhà **MORTGAGE**: Có thế chấp nhà
person_emp_length	Số năm người nộp đơn đang làm việc.
loan_intent	Mục đích vay vốn của người nộp đơn: **EDUCATION**: Giáo dục **MEDICAL**: Y tế **PERSONAL**: Cá nhân **VENTURE**: Khởi nghiệp **HOMEIMPROVEMENT**: Cải thiện nhà cửa **DEBTCONSOLIDATION**: Hợp nhất nợ (Thanh toán cho nhiều khoản nợ khác)

loan_grade	Điểm tín dụng được chỉ định cho khoản vay dựa trên mức độ tín nhiệm (xét với các yếu tố: lịch sử tín dụng, chất lượng tài sản thế chấp, khả năng trả nợ,...) của người vay: **A**: Người vay có độ tín nhiệm cao, cho thấy rủi ro thấp. **B**: Người vay có rủi ro tương đối thấp, nhưng không có độ tín nhiệm cao như mức A. **C**: Độ tín nhiệm của người vay ở mức trung bình. **D**: Người vay được coi là có rủi ro cao hơn so với các mức trước đó. **E**: Độ tín nhiệm của người vay thấp hơn, cho thấy rủi ro cao hơn. **F**: Người vay có rủi ro tín dụng đáng kể. **G**: Độ tín nhiệm của người vay là thấp nhất, biểu thị rủi ro cao nhất.
loan_amnt	Số tiền vay.
loan_int_rate	Lãi suất áp dụng cho khoản vay.
loan_percent_income	Tỷ lệ phần trăm số tiền vay theo tổng thu nhập.
cb_person_default_on_file	Lịch sử vỡ nợ của cá nhân theo hồ sơ của cơ quan tín dụng: **Y**: Cá nhân có lịch sử nợ xấu trong hồ sơ tín dụng. **N**: Cá nhân này không có tiền sử vi phạm nợ xấu.
cb_person_cred_hist_length	Số năm lịch sử cá nhân kể từ khoản vay đầu tiên.
loan_status	Trạng thái của khoản vay (biến mục tiêu): **0:** Không vỡ nợ - Người vay trả nợ thành công theo đúng thỏa thuận và không xảy ra vỡ nợ. **1:** Vỡ nợ - Người vay không trả nợ đúng hạn theo các điều khoản đã thỏa thuận và vỡ nợ khoản vay.

III) Nội dung báo cáo

1. Import thư viện và dữ liệu

- Thư viện:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns

from imblearn.over_sampling import SMOTE
from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import roc_curve, auc
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import auc

from catboost import CatBoostClassifier
from xgboost import XGBClassifier
import warnings
warnings.filterwarnings("ignore")
```

- Tập dữ liệu:

```
df = pd.read_csv('credit_risk_dataset.csv', skipinitialspace=True)
df
```

	person_age	person_income	person_home_ownership	person_emp_length	loan_intent	loan_grade	loan_amnt	loan_int_rate	loan_status	loan_percent_income	cb_person_default_on_file	cb_person_cred_hist_length
0	22	59000	RENT	123.0	PERSONAL	D	35000	16.02	1	0.59	Y	3
1	21	9600	OWN	5.0	EDUCATION	B	1000	11.14	0	0.10	N	2
2	25	9600	MORTGAGE	1.0	MEDICAL	C	5500	12.87	1	0.57	N	3
3	23	65500	RENT	4.0	MEDICAL	C	35000	15.23	1	0.53	N	2
4	24	54400	RENT	8.0	MEDICAL	C	35000	14.27	1	0.55	Y	4
...	...	...	...	...	...	...	...	...	...	...	...	...
32576	57	53000	MORTGAGE	1.0	PERSONAL	C	5800	13.16	0	0.11	N	30
32577	54	120000	MORTGAGE	4.0	PERSONAL	A	17625	7.49	0	0.15	N	19
32578	65	76000	RENT	3.0	HOMEIMPROVEMENT	B	35000	10.99	1	0.46	N	28
32579	56	150000	MORTGAGE	5.0	PERSONAL	B	15000	11.48	0	0.10	N	26
32580	66	42000	RENT	2.0	MEDICAL	B	6475	9.99	0	0.15	N	30

32581 rows × 12 columns

2. Thống kê mô tả dữ liệu

- Thuộc tính của trường dữ liệu:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32581 entries, 0 to 32580
Data columns (total 12 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   person_age                  32581 non-null  int64
 1   person_income               32581 non-null  int64
 2   person_home_ownership       32581 non-null  object
 3   person_emp_length           31686 non-null  float64
 4   loan_intent                 32581 non-null  object
 5   loan_grade                  32581 non-null  object
 6   loan_amnt                   32581 non-null  int64
 7   loan_int_rate               29465 non-null  float64
 8   loan_status                 32581 non-null  int64
 9   loan_percent_income         32581 non-null  float64
 10  cb_person_default_on_file   32581 non-null  object
 11  cb_person_cred_hist_length  32581 non-null  int64
dtypes: float64(3), int64(5), object(4)
memory usage: 3.0+ MB
```

- Mô tả thống kê:
  + **Biến định lượng:**

	count	mean	std	min	25%	50%	75%	max
person_age	32581.0	27.734600	6.348078	20.00	23.00	26.00	30.00	144.00
person_income	32581.0	66074.848470	61983.119168	4000.00	38500.00	55000.00	79200.00	6000000.00
person_emp_length	31686.0	4.789686	4.142630	0.00	2.00	4.00	7.00	123.00
loan_amnt	32581.0	9589.371106	6322.086646	500.00	5000.00	8000.00	12200.00	35000.00
loan_int_rate	29465.0	11.011695	3.240459	5.42	7.90	10.99	13.47	23.22
loan_status	32581.0	0.218164	0.413006	0.00	0.00	0.00	0.00	1.00
loan_percent_income	32581.0	0.170203	0.106782	0.00	0.09	0.15	0.23	0.83
cb_person_cred_hist_length	32581.0	5.804211	4.055001	2.00	3.00	4.00	8.00	30.00

  + **Biến định tính:**

	count	unique	top	freq
person_home_ownership	32581	4	RENT	16446
loan_intent	32581	6	EDUCATION	6453
loan_grade	32581	7	A	10777
cb_person_default_on_file	32581	2	N	26836

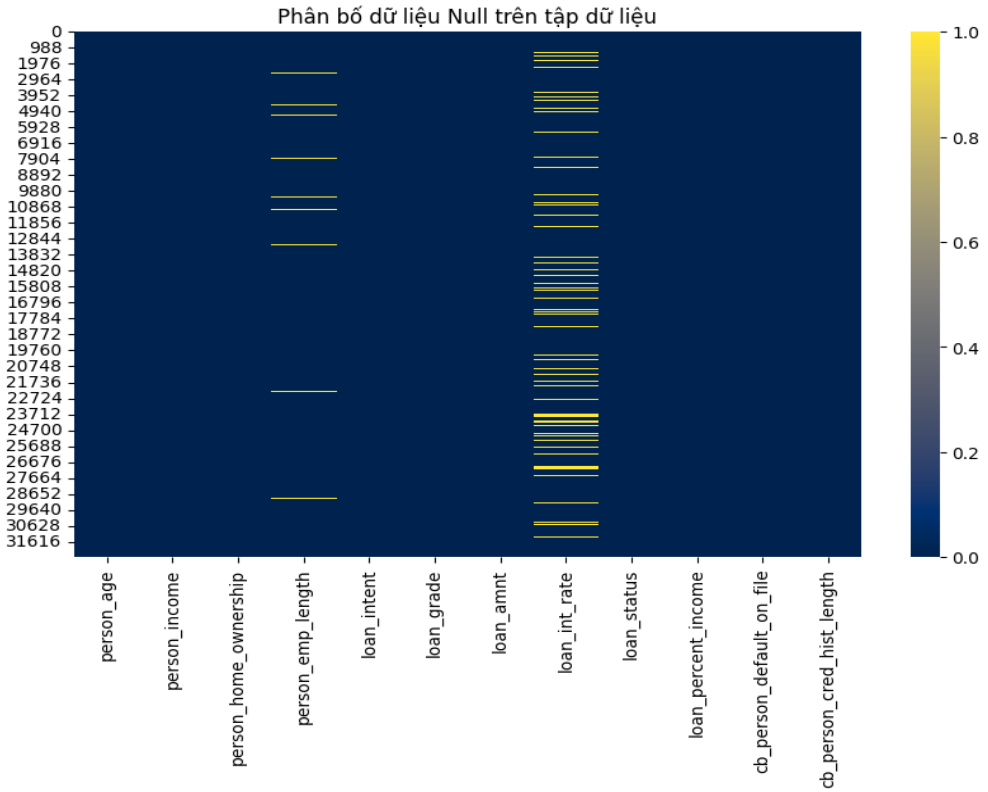
3. Tiền xử lý dữ liệu

3.1. Giá trị rỗng (Null Values) & trùng lặp (Duplicate Rows)

❖ *Kiểm tra giá trị Null:*

person_age	0
person_income	0
person_home_ownership	0
person_emp_length	895
loan_intent	0
loan_grade	0
loan_amnt	0
loan_int_rate	3116
loan_status	0
loan_percent_income	0
cb_person_default_on_file	0
cb_person_cred_hist_length	0

- *Phân bố của giá trị Null:*



Có nhiều phương pháp xử lý giá trị ngoại lệ nhưng do phân bố và số lượng của giá trị Null nằm rải rác tương đối ít trên tập dữ liệu, không ảnh hưởng nhiều đến quá trình phân tích dữ liệu nên nhóm lựa chọn phương pháp loại bỏ các giá trị này.

- *Sau khi loại bỏ giá trị Null:*

```
person_age                    0
person_income                 0
person_home_ownership         0
person_emp_length             0
loan_intent                   0
loan_grade                    0
loan_amnt                     0
loan_int_rate                 0
loan_status                   0
loan_percent_income           0
cb_person_default_on_file     0
cb_person_cred_hist_length    0
dtype: int64
```

```
df.shape
```

```
(28638, 12)
```

❖ *Kiểm tra hàng trùng lặp (Duplicate Rows)*

```
df.duplicated().sum()
```

```
137
```

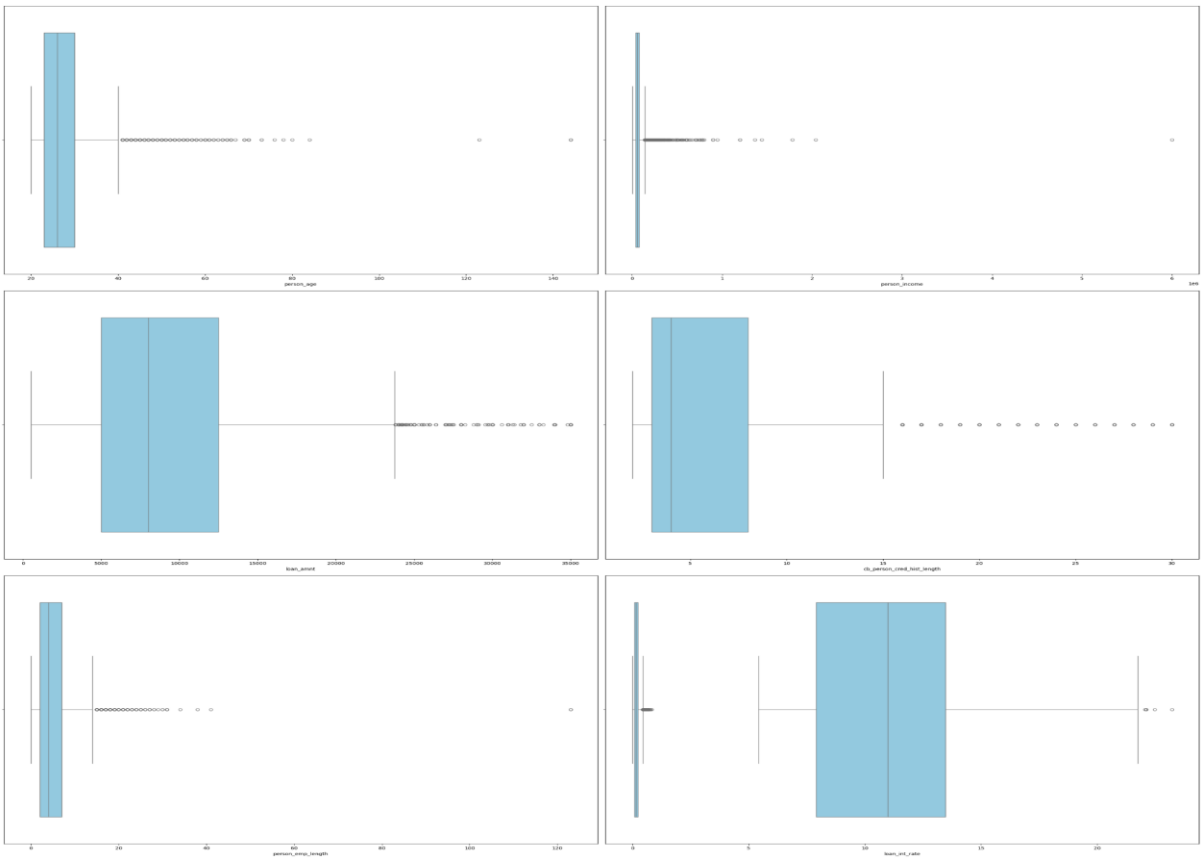
- Sau khi loại bỏ hàng trùng lặp:

```
df.shape
```

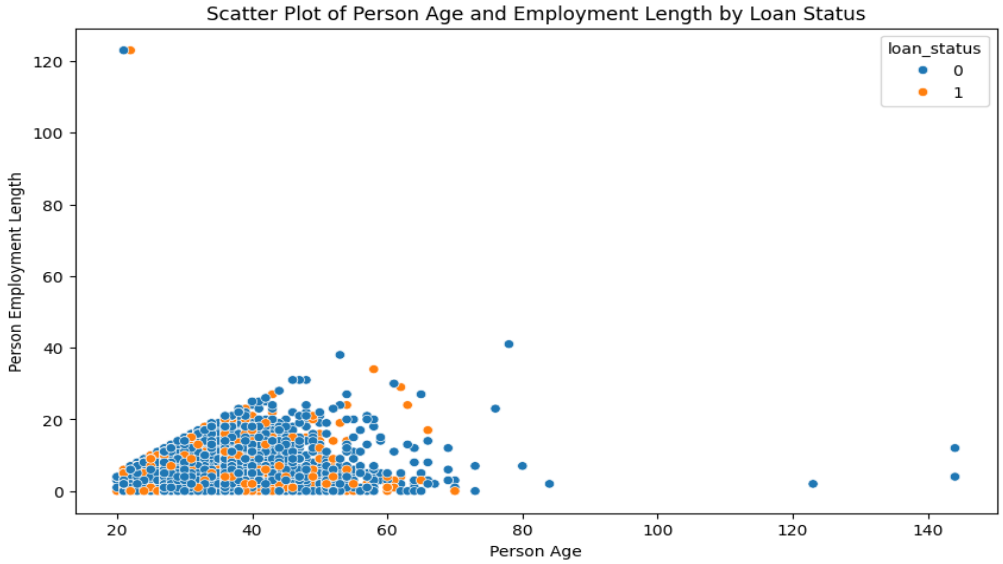
```
(28501, 12)
```

3.2. Giá trị ngoại lệ (Outlier Values)

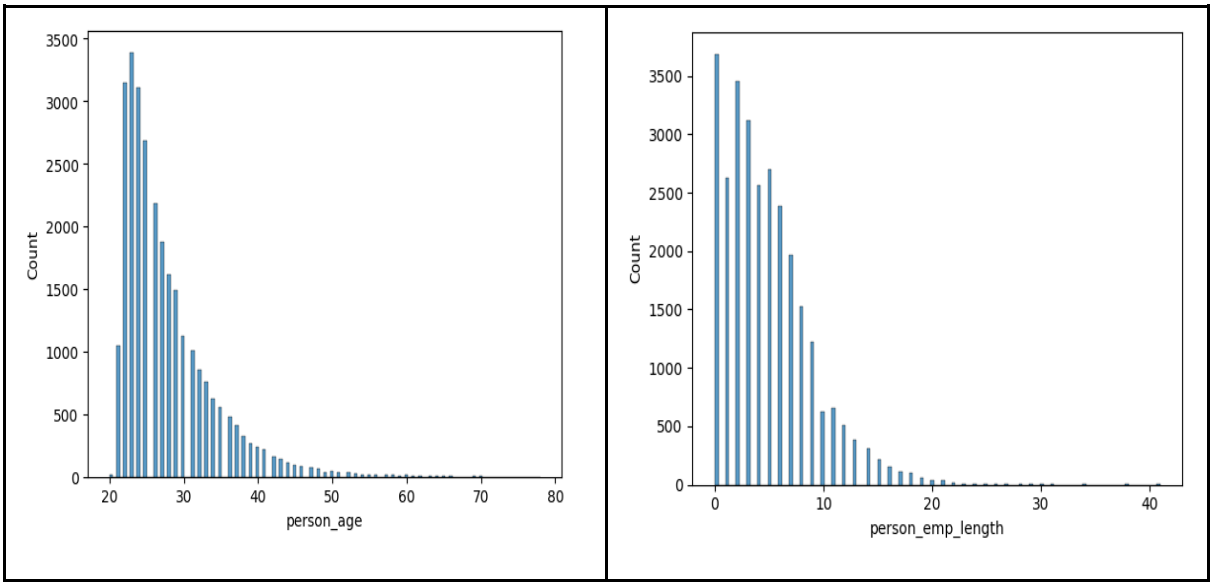
- *Phân bố của các giá trị định lượng*:



Hai đặc trưng person_age và person_emp_length có phân bố ở các khoảng giá trị chưa hợp lý cần loại bỏ:

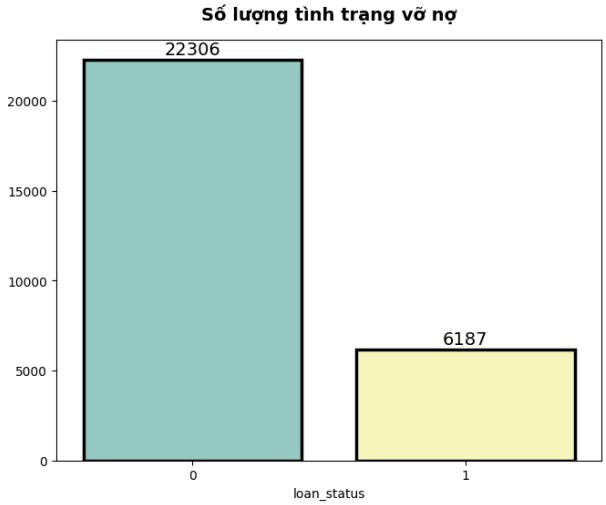


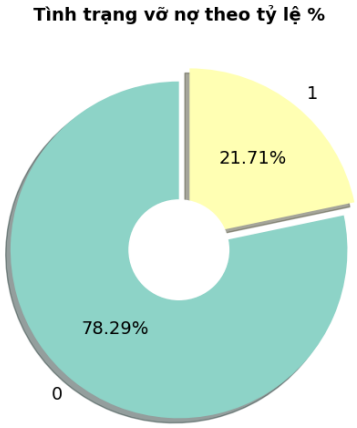
- Sau khi loại bỏ giá trị Outliers:



4. Trực quan hóa dữ liệu

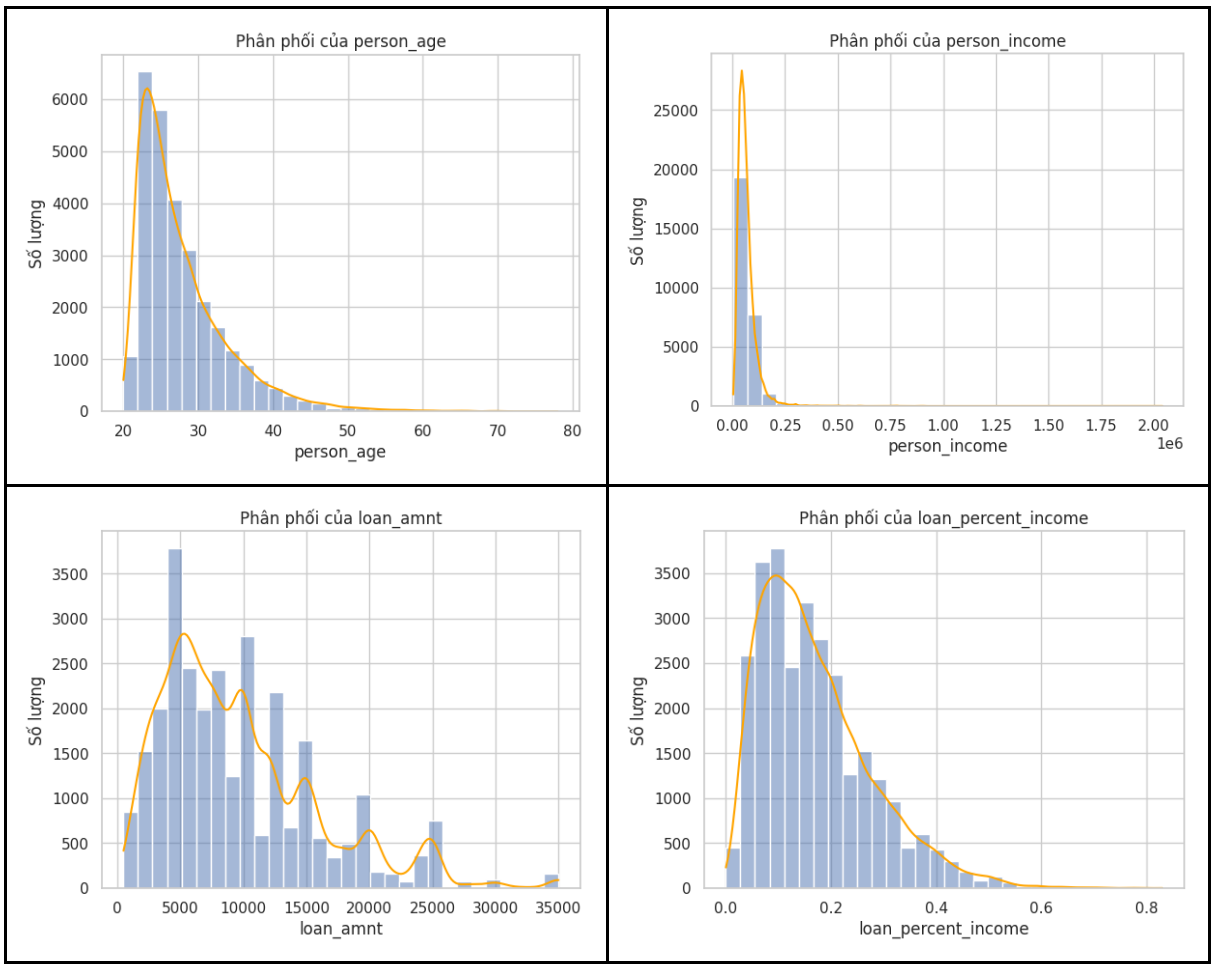
4.1. Tổng quan về biến mục tiêu

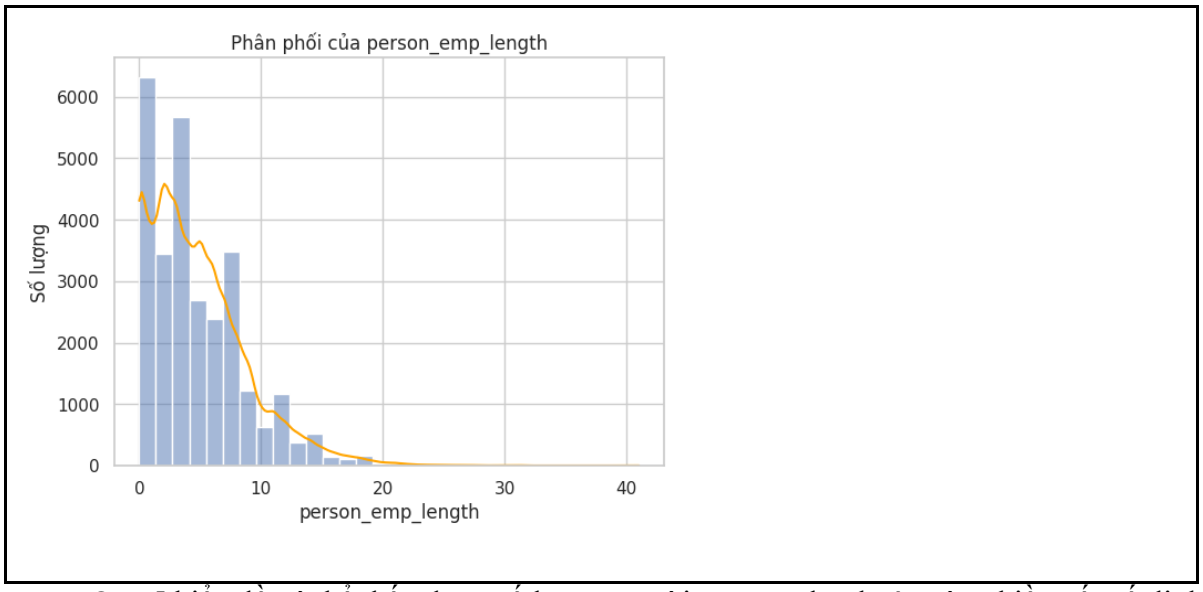




- Dữ liệu có sự mất cân bằng tỷ lệ gần 4:1. Điều này ảnh hưởng đến độ chính xác của mô hình dự báo, cần xử lý dữ liệu mất cân bằng.

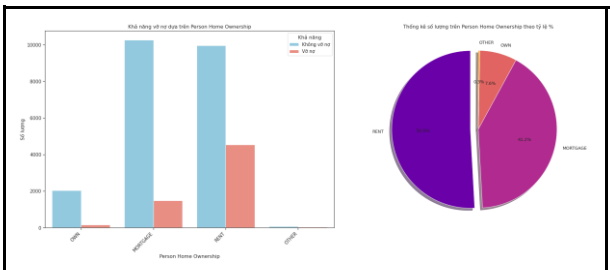
4.2. Phân phối của biến định lượng





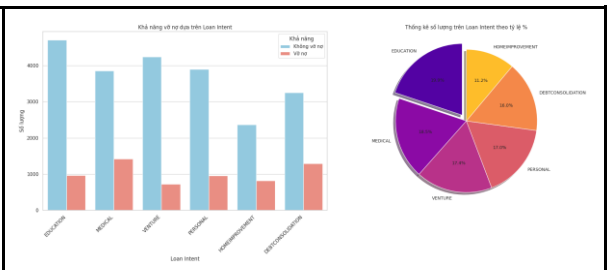
- Qua 5 biểu đồ có thể thấy được số lượng người vay nợ phụ thuộc vào nhiều yếu tố định lượng như : độ tuổi, thu nhập ,giá trị khoản vay, tỷ lệ thu nhập dành cho khoản vay, độ dài năm làm việc.
- Cụ thể
    + Nhóm tuổi 20-40 chiếm đa số, trong khi nhóm >50 ít vay hơn.
    + Thu nhập dưới 50.000 USD/năm và khoản vay nhỏ hơn 5.000 USD chiếm tỷ lệ lớn nhất.
    + Tỷ lệ vay/tổng thu nhập thường dưới 0.3, hiếm có người vay vượt 30% thu nhập.
    + Độ dài làm việc của những người đi vay chủ yếu dưới 10 năm.

4.3. Mức độ ảnh hưởng của các biến phân loại đối với rủi ro tín dụng



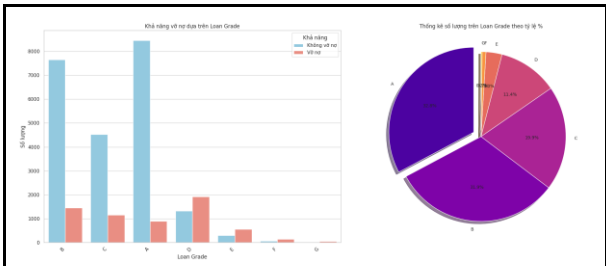
Bảng số lượng thống kê trên Person Home Ownership:

person_home_ownership	Số lượng	Phần trăm
RENT	14494	50.87
MORTGAGE	11732	41.18
OWN	2174	7.63
OTHER	93	0.33



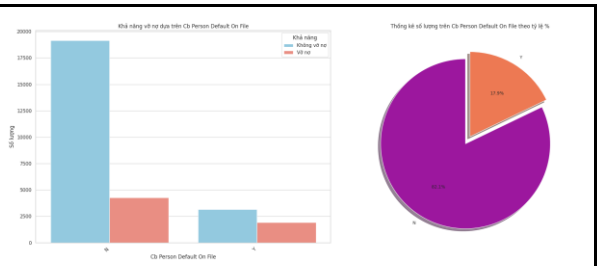
Bảng số lượng thống kê trên Loan Intent:

loan_intent	Số lượng	Phần trăm
EDUCATION	5668	19.89
MEDICAL	5268	18.49
VENTURE	4967	17.43
PERSONAL	4856	17.04
DEBTCONSOLIDATION	4547	15.96
HOMEIMPROVEMENT	3187	11.19



Bảng số lượng thống kê trên Loan Grade:

loan_grade	Số lượng	Phần trăm
A	9343	32.79
B	9091	31.91
C	5680	19.93
D	3242	11.38
E	869	3.05
F	209	0.73
G	59	0.21



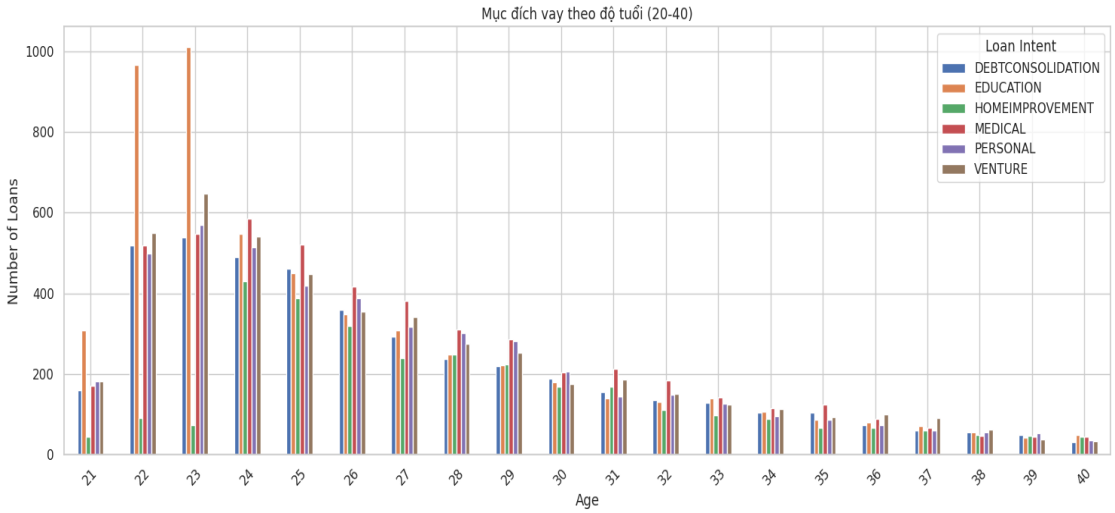
Bảng số lượng thống kê trên Cb Person Default On File:

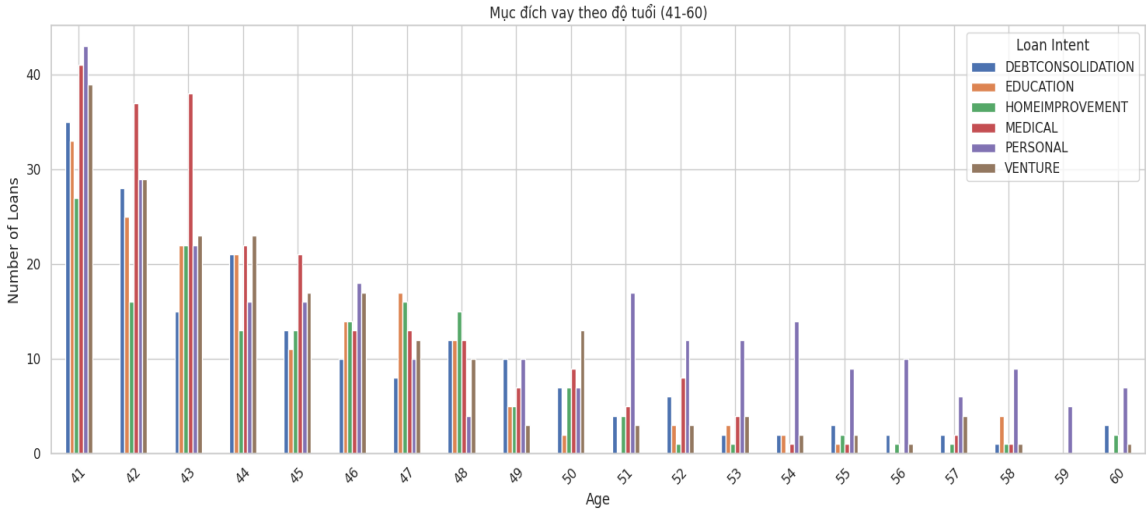
cb_person_default_on_file	Số lượng	Phần trăm
N	23404	82.14
Y	5089	17.86

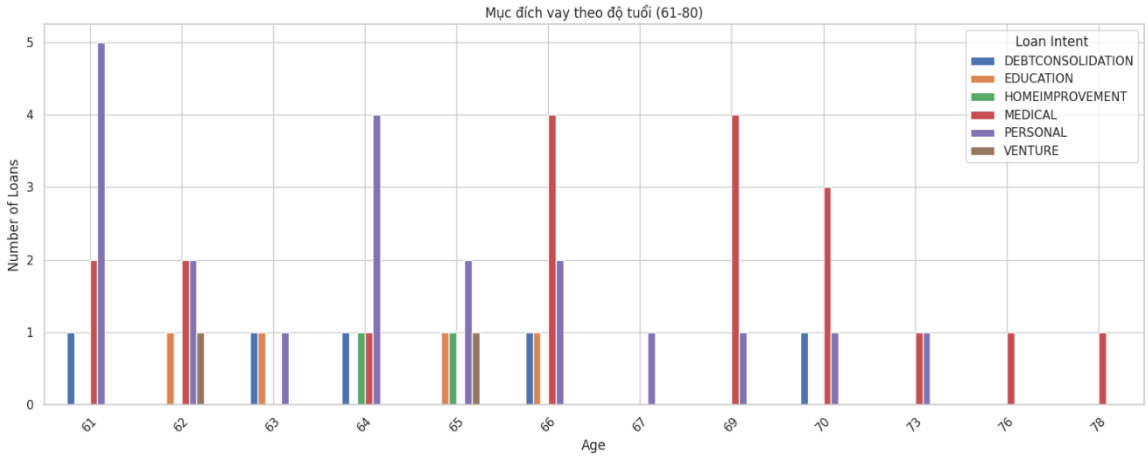
- Với 2 biểu đồ cột và tròn tương ứng cho mỗi biến phân loại có thể nhận thấy rằng rằng khả năng vỡ nợ của người vay có thể bị ảnh hưởng bởi nhiều tiêu chí như : Tình trạng sở hữu nhà, mục đích vay, điểm tín dụng, lịch sử vỡ nợ.
    + Người thuê nhà và thế chấp chiếm tổng tỷ lệ lớn (92%), là nhóm vay chính và có rủi ro vỡ nợ cao hơn.
    + Giáo dục (20%) là mục đích vay phổ biến nhất với tỷ lệ rủi ro thấp. Các khoản vay y tế và hợp nhất nợ có tỷ lệ vỡ nợ cao nhất.
    + Khoản vay xếp hạng A và B ít rủi ro nhất,rủi ro tăng dần từ hạng C trở xuống.
    + Người vay không có tiền sử vỡ nợ, duy trì trạng thái vay tốt, ít vỡ nợ.

4.4. Yếu tố cản trở khi xét duyệt khoản vay dựa trên phân tích các biến định lượng

- Mục đích vay thay đổi theo độ tuổi

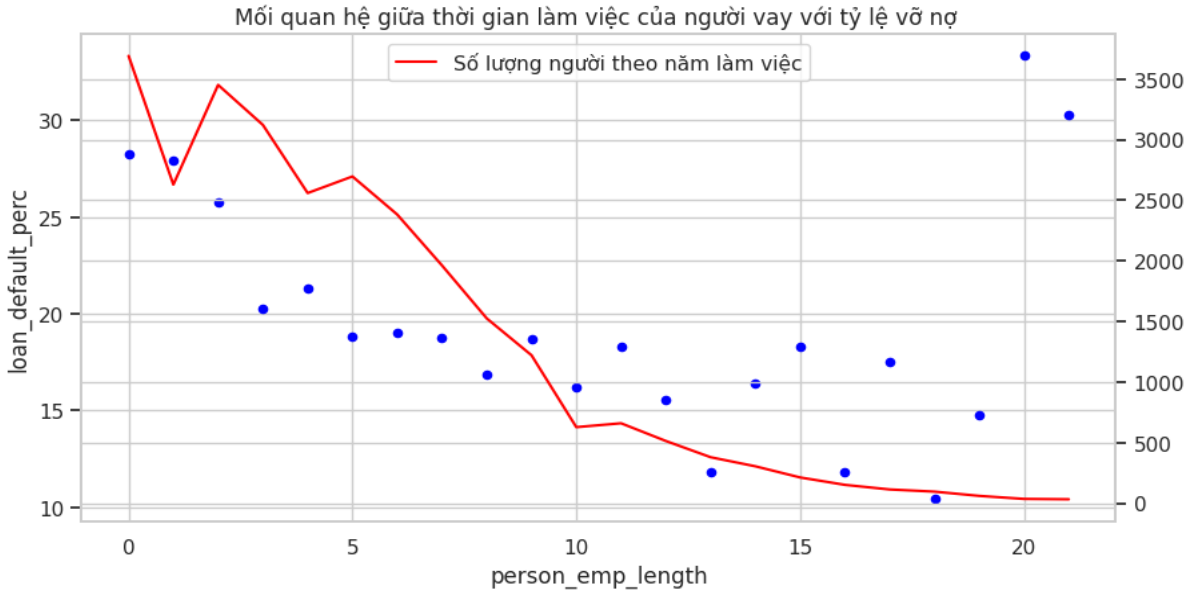






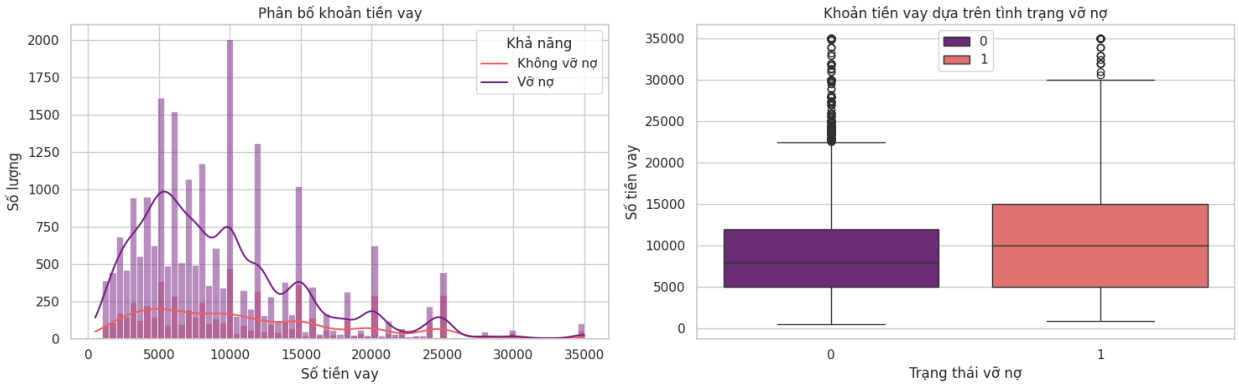
+ Người từ 20-40 tuổi: Mục đích vay đa dạng, tập trung vào giáo dục và cải thiện nhà cửa. Nhu cầu vay giảm dần khi tuổi tăng, nhưng nhu cầu cho y tế và cá nhân tăng lên.
+ Người từ 41-60 tuổi: Hợp nhất nợ và vay cá nhân là phổ biến nhất, cho thấy sự tập trung vào quản lý tài chính. Nhu cầu vay giảm dần khi độ tuổi tăng, nhưng cải thiện nhà cửa và y tế vẫn duy trì ổn định.
+ Người từ 61-80 tuổi: Nhu cầu vay giảm đáng kể ở độ tuổi này, chủ yếu vay cho mục đích y tế và cá nhân.

- Tác động của độ dài năm làm việc đến xu hướng gây ra vỡ nợ



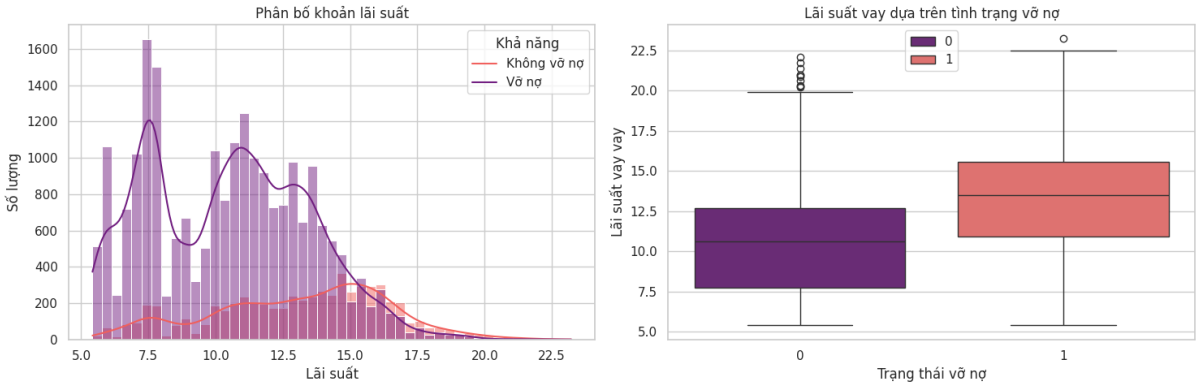
+ Thời gian làm việc là yếu tố quan trọng cần cân nhắc: khi tỷ lệ vỡ nợ giảm khi thời gian làm việc tăng, cho thấy thời gian làm việc dài hơn giúp giảm rủi ro vỡ nợ.

- Ảnh hưởng của giá trị khoản vay đến khả năng vỡ nợ



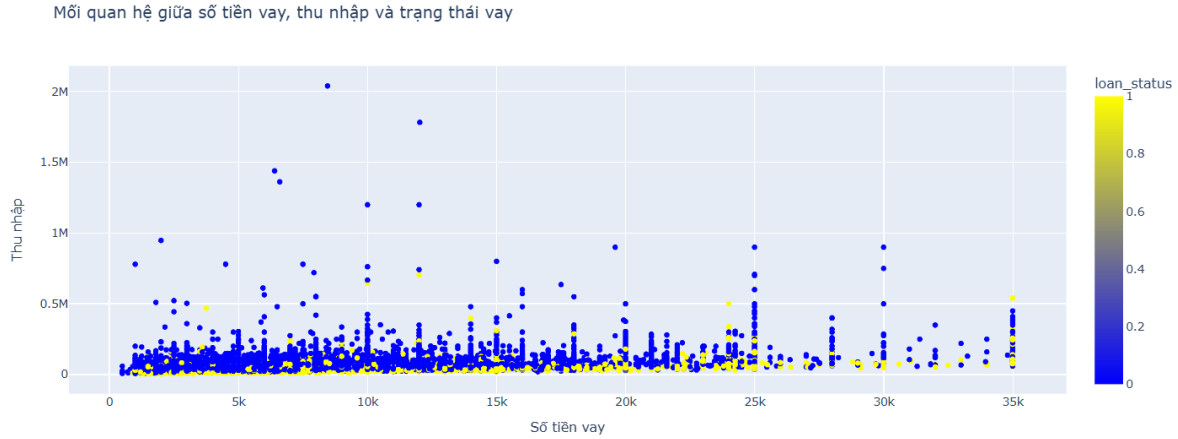
+ Biểu đồ cho thấy các khoản vay vỡ nợ có xu hướng tập trung ở mức khoản vay cao hơn đặc biệt là những khoản vay trên 10.000USD.

- Ảnh hưởng của lãi suất đến khả năng vỡ nợ trong phê duyệt khoản vay



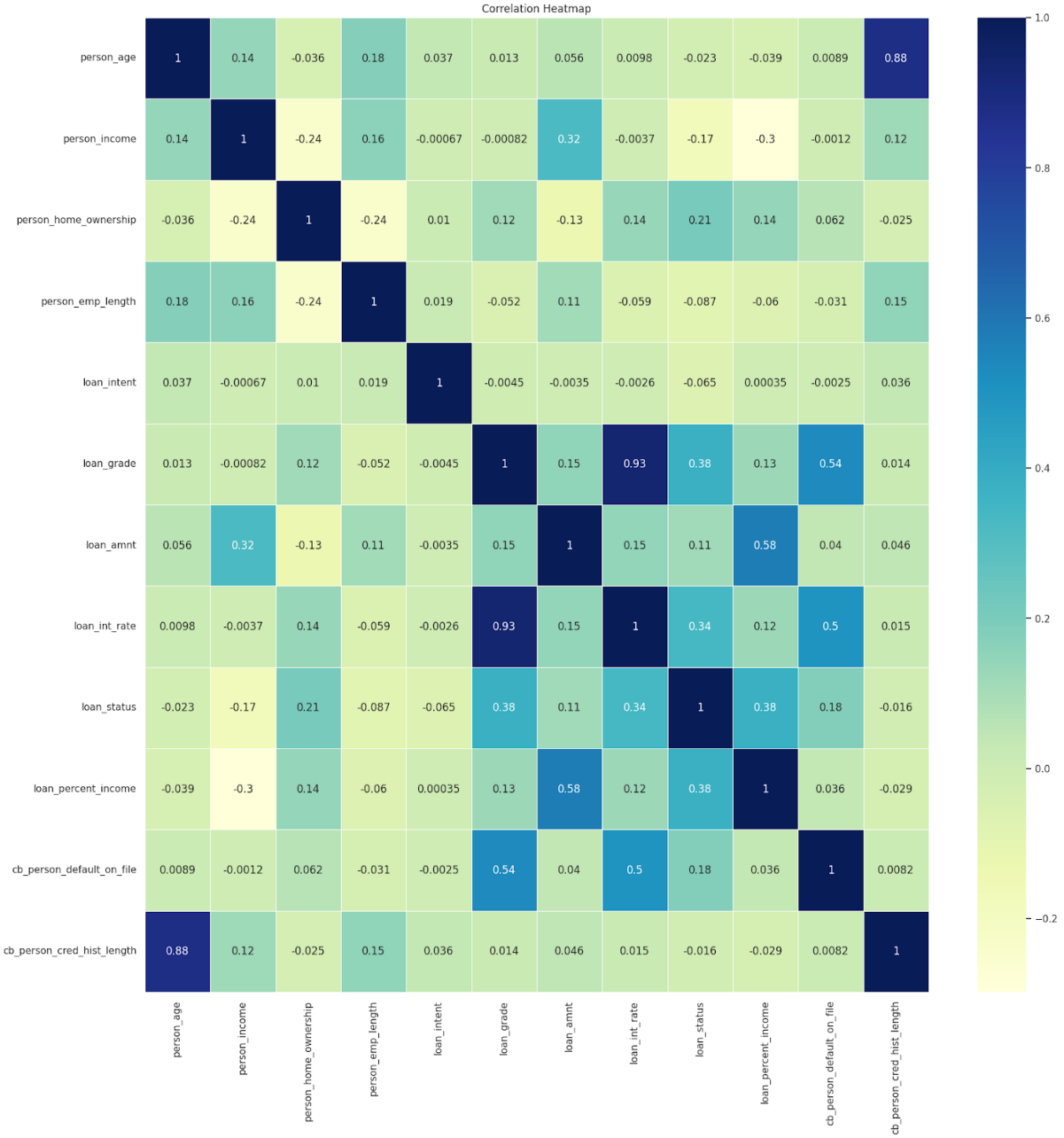
+ Để giảm nguy cơ vỡ nợ, lãi suất cho vay nên duy trì dưới 10%. Lãi suất này giúp giảm thiểu rủi ro vỡ nợ so với các mức lãi suất cao hơn.

- Mối liên hệ giữa thu nhập và mức tiền vay đối với rủi ro vỡ nợ



+ Phần lớn các khoản vay có giá trị dưới 25,000 USD, và thu nhập cá nhân tập trung dưới 500,000 USD. Trung bình các khoản vay vỡ nợ cao hơn, đặc biệt là ở mức trên 15,000 USD. Những người có thu nhập trên 500,000 USD ít có dấu hiệu vỡ nợ hơn, dù vay các khoản lớn.

5. Correlation (Ma trận tương quan)



- Độ tuổi và lịch sử tín dụng có mối tương quan mạnh (0.88), người lớn tuổi có lịch sử tín dụng dài hơn. Số tiền vay và tỷ lệ thu nhập dành cho vay có mối tương quan dương mạnh (0.58). Nhưng các mối tương quan này có thể dẫn đến đa cộng tuyến.
- Lãi suất và trạng thái khoản vay có mối tương quan dương (0.34), lãi suất cao liên quan đến nợ xấu.
- Thu nhập và số tiền vay có mối tương quan dương (0.32), người thu nhập cao vay số tiền lớn hơn.
- Một số tương quan đáng chú ý khác: loan_ Grade với *cb_person_default_on_file*, l*oan_int_rate và Loan Int Rate và cb_person_default_on_file*

⇨ Các thuộc tính trong ma trận tương quan cung cấp thông tin quan trọng, bổ sung cho nhau trong dự đoán rủi ro tín dụng. Việc sử dụng đầy đủ 11 thuộc tính (trừ loan_status) giúp mô hình toàn diện hơn. Các thuật toán như CatBoost, Decision Tree, XGBoost, và

Random Forest xử lý tốt mối quan hệ phức tạp, cải thiện độ chính xác và hiệu suất dự đoán.

6. Chuyển đổi dữ liệu

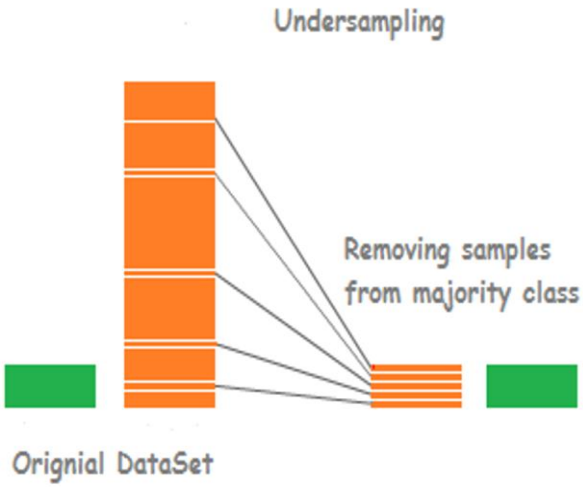
6.1. Chuyển các dữ liệu phân loại sang dạng mã hóa

	person_age	person_income	person_emp_length	loan_amnt	loan_int_rate	loan_status	loan_percent_income	cb_person_cred_hist_length	person_home_ownership_MORTGAGE	person_home_ownership_OTHER	...	loan_intent_VENTURE
1	21	9600	5.0	1000	11.14	0	0.10	2	False	False	...	False
2	25	9600	1.0	5500	12.87	1	0.57	3	True	False	...	False
3	23	65500	4.0	35000	15.23	1	0.53	2	False	False	...	False
4	24	54400	8.0	35000	14.27	1	0.55	4	False	False	...	False
5	21	9900	2.0	2500	7.14	1	0.25	2	False	False	...	True

5 rows × 27 columns

6.2. Xử lý dữ liệu mất cân bằng

Mất cân bằng dữ liệu là một trong những hiện tượng phổ biến của bài toán phân loại, cụ thể là với bài toán phân loại nhị phân như phân loại email spam, phát hiện gian lận, dự báo vỡ nợ, chuẩn đoán bệnh lý,….Có nhiều phương pháp xử lý dữ liệu mất cân bằng nhưng nhóm tìm hiểu được hai phương pháp tập trung vào thay đổi dữ liệu đầu vào để cân bằng tỷ lệ giữa các lớp đó là Undersampling và Oversampling.



❖ **Undersampling**

- Phương pháp Undersampling tập trung vào việc giảm số lượng mẫu của lớp đa số để cân bằng với lớp thiểu số.

+ Ưu điểm là làm cân bằng mẫu một cách nhanh chóng, dễ dàng tiến hành thực hiện mà không cần đến thuật toán giả lập.

+ Nhược điểm của phương pháp này là số lượng mẫu sẽ bị giảm đi đáng kể. Tập huấn luyện mới sau khi Undersampling khá nhỏ, không đại diện cho phân phối của toàn bộ tập dữ liệu và thường dễ dẫn tới hiện tượng Overfitting.

❖ **Oversampling**

- Phương pháp Oversampling tập trung vào giải quyết hiện tượng mất cân bằng mẫu bằng cách gia tăng kích thước mẫu thuộc lớp thiểu số bằng các kỹ thuật khác nhau.
- Có 2 phương pháp chính để để thực hiện Oversampling đó là:
- **Lập lại mẫu hiện có để tăng số lượng mẫu**: Sao chép lại ngẫu nhiên các dữ liệu ban đầu của lớp thiểu số để tăng số lượng mẫu, cân bằng với lớp đa số.
- **Tạo mẫu mới dựa trên tổng hợp của các mẫu cũ** (SMOTE - Synthetic Minority Over-sampling Technique): Thuật toán chọn 2 hay nhiều trường hợp giống nhau (sử dụng thước đo khoảng cách để so sánh) và xáo trộn một cá thể một thuộc tính tại một thời điểm bằng một lượng ngẫu nhiên trong khoảng chênh lệnh với các trường hợp lân cận.

Ứng dụng vào bài toán dự đoán rủi ro tín dụng với biến mục tiêu loan_status, nhóm lựa chọn phương pháp Oversampling với kỹ thuật SMOTE để cân bằng dữ liệu vì bộ dữ liệu thu thập được có kích thước không quá lớn nên việc cắt bớt dữ liệu đi sẽ không còn nhiều dữ liệu đưa vào mô hình machine learning và quyết định sử dụng kỹ thuật SMOTE để dữ liệu đưa vào máy học có thể đưa ra độ tin cậy cao.

- Sau khi xử lý dữ liệu mất cân bằng:

loan_status	count
0	22306
1	22306

dtype: int64

6.3. Đào tạo và phân chia dữ liệu cho machine learning

```
X = df.drop('loan_status', axis=1)
y = df['loan_status']
```

```
from sklearn.model_selection import train_test_split
# split train and test sets
X_train, X_test, y_train, y_test = train_test_split(
                                                    df.drop(labels=['loan_status'], axis=1),
                                                    df['loan_status'],
                                                    test_size=0.3,
                                                    random_state=0)
```

```
X_train.shape, X_test.shape
```

```
((19945, 26), (8548, 26))
```

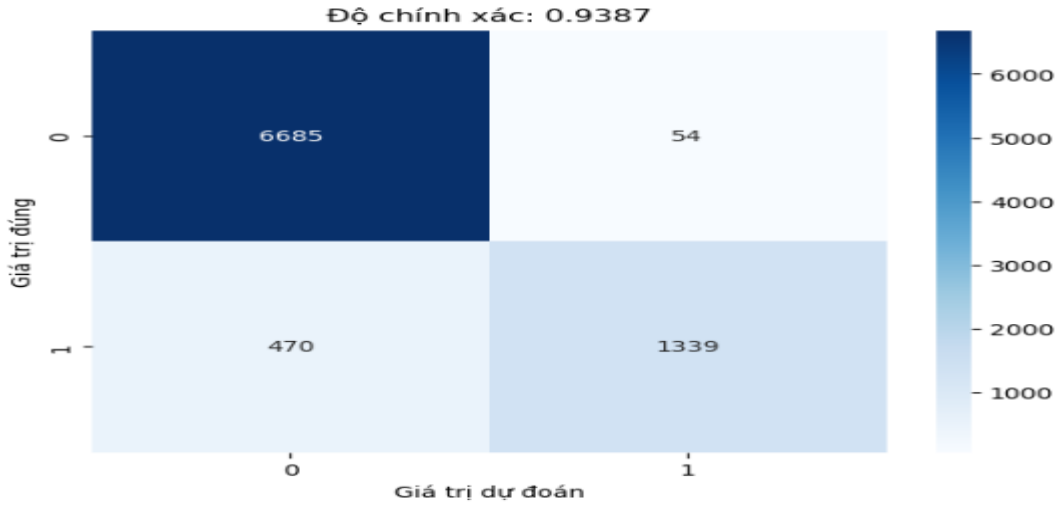
```
y_train.shape, y_test.shape
```

```
((19945,), (8548,))
```

7. Model

7.1. CatBoost

- **Định nghĩa:** CatBoost (Categorical Boosting) là một thuật toán học máy mạnh mẽ, đặc biệt hiệu quả trong việc xử lý dữ liệu phân loại mà không cần tiền xử lý phức tạp. Dựa trên phương pháp gradient boosting, CatBoost tự động xử lý các tính năng phân loại mà không cần phải chuyển đổi chúng thành dạng số. Nó giúp cải thiện độ chính xác mô hình, tránh overfitting và tăng tốc quá trình huấn luyện.
- **Biểu đồ so sánh kết quả thực tế với dự đoán của CatBoost**



Báo cáo phân loại:

	precision	recall	f1-score	support
0	0.93	0.99	0.96	6739
1	0.96	0.74	0.84	1809
accuracy			0.94	8548
macro avg	0.95	0.87	0.90	8548
weighted avg	0.94	0.94	0.94	8548

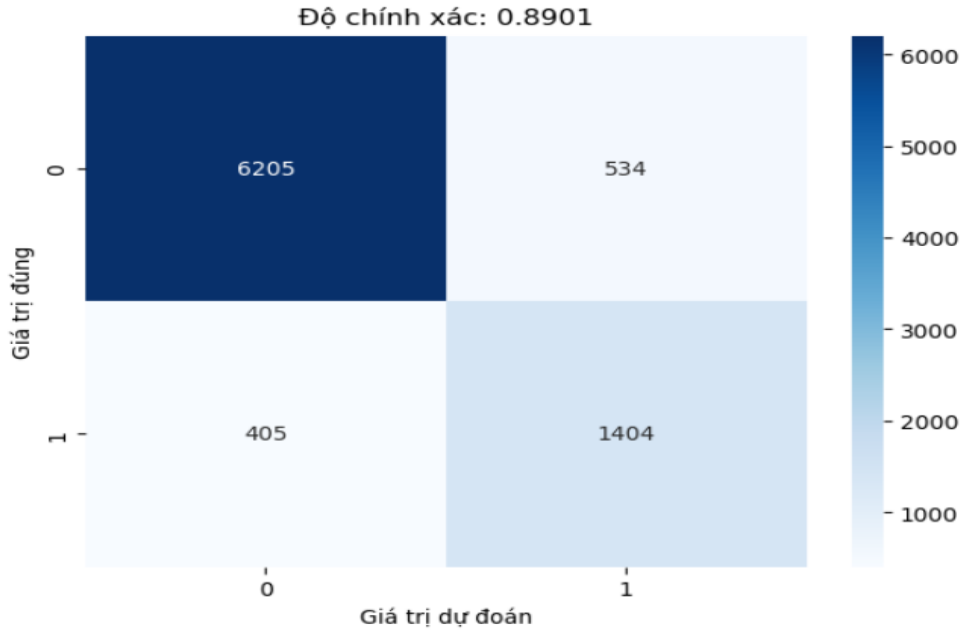
Nhận xét:

- **Lớp 0**: Độ chính xác 93% và recall 99%, phân loại tốt và ít bỏ sót.
- **Lớp 1**: Độ chính xác 96%, nhưng recall 74%, bỏ sót nhiều trường hợp rủi ro.
- **Chênh lệch score** giữa Train (0.9573) và Test (0.9391) là 2%, với độ chính xác 93.87%, cho thấy mô hình hiệu quả.

7.2. Decision Tree

❖ **Định nghĩa:** Decision Tree (Cây quyết định) là một thuật toán học máy phổ biến, được sử dụng trong cả các bài toán hồi quy và phân loại. Mô hình cây quyết định giúp phân loại hoặc dự đoán giá trị bằng cách chia nhỏ dữ liệu theo một cách có cấu trúc, dựa trên các câu hỏi hoặc điều kiện về các đặc trưng của dữ liệu.

❖ **Biểu đồ so sánh kết quả thực tế với dự đoán của Decision Tree**



Báo cáo phân loại:

	precision	recall	f1-score	support
0	0.94	0.92	0.93	6739
1	0.72	0.78	0.75	1809
accuracy			0.89	8548
macro avg	0.83	0.85	0.84	8548
weighted avg	0.89	0.89	0.89	8548

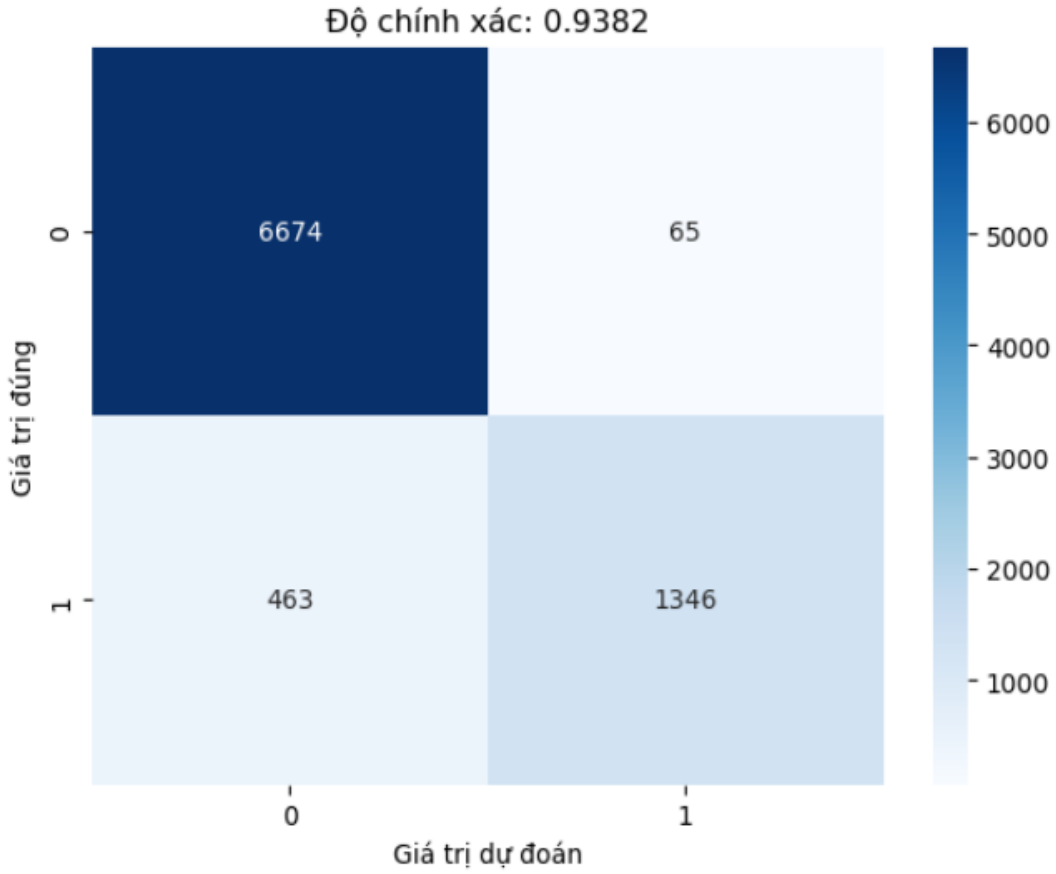
Nhận xét:

- **Lớp 0:** Precision 94% và recall 92%, phân loại tốt và ít bỏ sót.
- **Lớp 1:** Precision 72% và recall 78%, phát hiện khá nhưng vẫn bỏ sót một số trường hợp.
- **Chênh lệch score** giữa Train (1.0000) và Test (0.8901) là 11%, thể hiện overfitting, giảm độ tin cậy khi áp dụng thực tế.

7.3. XgBoost

❖ **Định nghĩa:** XGBoost (Extreme Gradient Boosting) là một thuật toán học máy mạnh mẽ và phổ biến, được phát triển dựa trên kỹ thuật Gradient Boosting, đặc biệt hiệu quả trong các bài toán phân loại và hồi quy. XGBoost được thiết kế để cải thiện hiệu suất và tốc độ của thuật toán Gradient Boosting, đồng thời giảm thiểu các vấn đề như overfitting (quá khớp) và tính toán chậm.

❖ **Biểu đồ so sánh kết quả thực tế với dự đoán của XgBoost**



Báo cáo phân loại:

	precision	recall	f1-score	support
0	0.94	0.99	0.96	6739
1	0.95	0.74	0.84	1809
accuracy			0.94	8548
macro avg	0.94	0.87	0.90	8548
weighted avg	0.94	0.94	0.94	8548

Nhận xét:

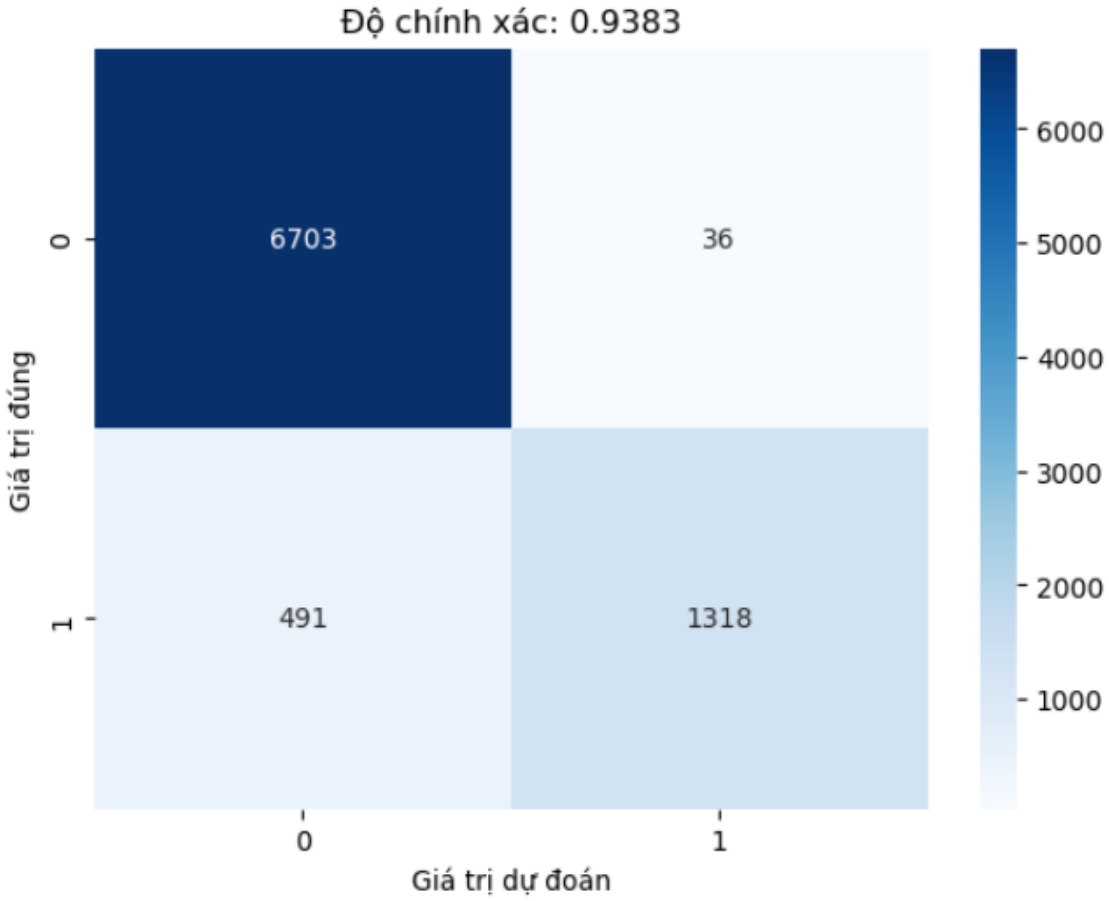
- **Lớp 0**: Độ chính xác 94% và recall 99%, phân loại tốt và ít bỏ sót.
- **Lớp 1**: Độ chính xác 95%, nhưng recall 74%, bỏ sót nhiều trường hợp rủi ro.
- **Chênh lệch score** giữa Train (0.9601) và Test (0.9382) là 2%, với độ chính xác 93.82%, mô hình phù hợp để phân tích.

7.4. Random Forest

❖ **Định nghĩa:** Random Forest là một thuật toán học máy mạnh mẽ và phổ biến, thuộc nhóm các phương pháp ensemble learning (học máy tổ hợp), được sử dụng

cho cả bài toán phân loại và hồi quy. Random Forest kết hợp nhiều cây quyết định để tạo ra một mô hình mạnh mẽ và chính xác hơn.

❖ **Biểu đồ so sánh kết quả thực tế với dự đoán của Random Forest**



Báo cáo phân loại:

	precision	recall	f1-score	support
0	0.93	0.99	0.96	6739
1	0.97	0.73	0.83	1809
accuracy			0.94	8548
macro avg	0.95	0.86	0.90	8548
weighted avg	0.94	0.94	0.93	8548

Nhận xét:

- **Lớp 0:** Mô hình phân loại tốt với độ chính xác và recall cao.
- **Lớp 1:**: Độ chính xác cao (97%) nhưng recall 73%, bỏ sót nhiều trường hợp rủi ro.
- **Chênh lệch score** giữa Train (1.0000) và Test (0.9383) là 6%, cho thấy overfitting, nhưng độ chính xác tổng thể (93.83%) vẫn cao.

8. Đánh giá các mô hình đã xây dựng

8.1. So sánh các chỉ số đánh giá hiệu suất của các mô hình

	Độ chính xác (Accuracy)	Độ chuẩn xác (Precision)	Độ nhạy (Recall)	Điểm F1 (F1 Score)
CatBoost	0.9387	0.9612	0.7402	0.8364
Decision Tree	0.8901	0.7245	0.7761	0.7494
XgBoost	0.9382	0.9539	0.7441	0.8360
Random Forest	0.9383	0.9734	0.7286	0.8334

Nhận xét:

- Top 3 mô hình có độ chính xác (Accuracy) cao nhất là Random Forest, XGBoost và CatBoost với tỷ lệ 93.8%.
- Random Forest đạt Precision cao nhất (97.34%), nghĩa là nó có khả năng giảm thiểu false positives tốt nhất, phù hợp khi mục tiêu là giảm thiểu sai lầm khi dự đoán ai sẽ không vỡ nợ.
- Decision Tree có Recall cao nhất (77.61%), nghĩa là thuật toán phát hiện được nhiều trường hợp vỡ nợ nhất.
- CatBoost đạt F1 Score cao nhất (83.64%), cân bằng tốt giữa Precision và Recall. Decision Tree có F1 Score thấp nhất (74.94%), cho thấy hiệu suất tổng quan yếu hơn.

➔ Do đó, trong project này, model phù hợp nhất cho dự báo Credit Risk là Cat Boost vì đạt hiệu suất cao trên cả Accuracy, Precision, Recall, và F1 Score. Mô hình này cân bằng giữa phát hiện rủi ro (Recall) và dự đoán chính xác (Precision).

8.2. Đánh giá hiệu quả các mô hình qua giá trị AUC của ROC

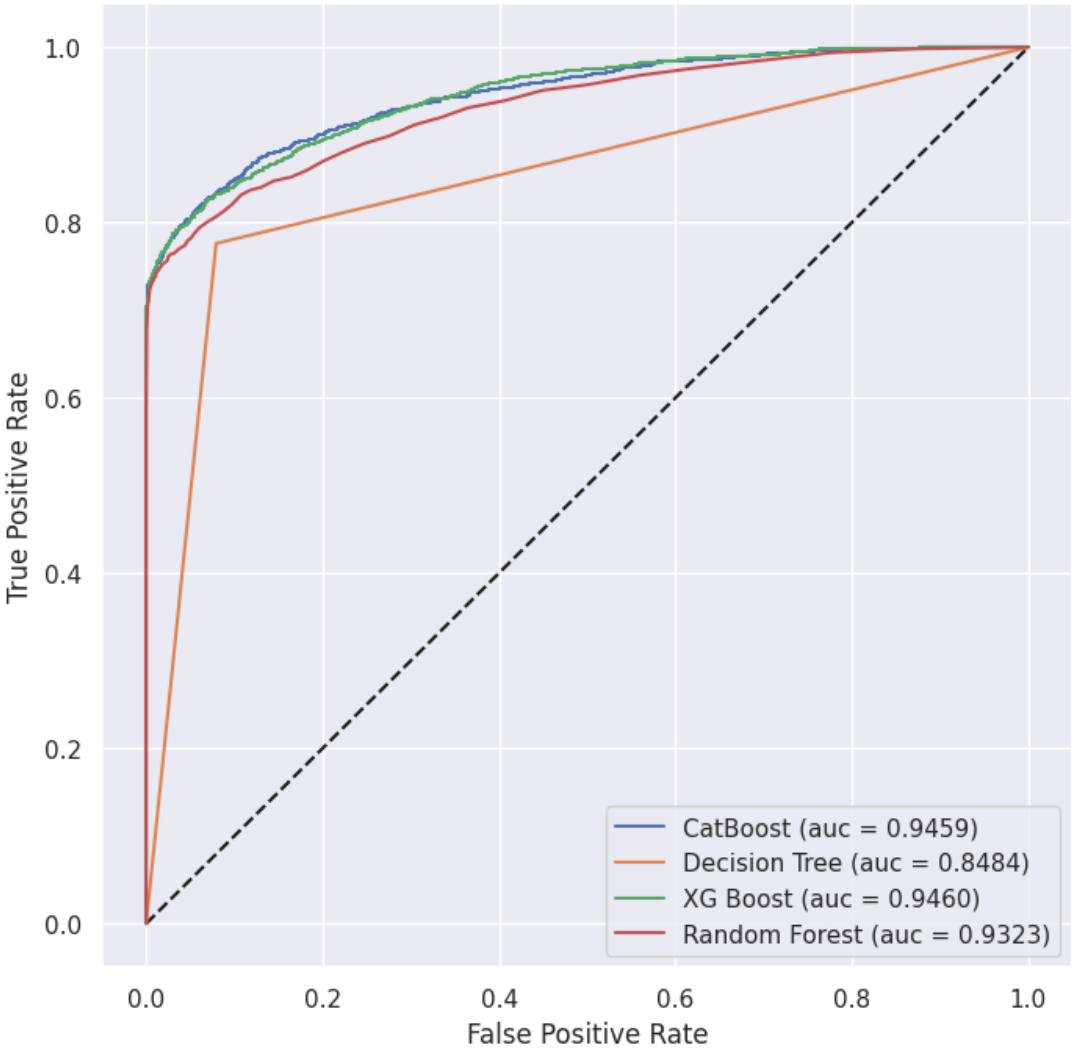
AUC là diện tích dưới đường cong ROC (Receiver Operating Characteristic), biểu diễn khả năng phân biệt giữa hai lớp (positive và negative). Áp dụng vào việc đánh giá cho các mô hình đã xây dựng, AUC có thể đưa ra đánh giá tốt hơn vì chỉ số này tập trung vào khả năng phân biệt, độc lập với tỷ lệ mất cân bằng, giúp đánh giá chính xác hiệu quả của mô hình.

➢ **Ý nghĩa giá trị AUC:**

AUC cao **(> 0.9)**	- Mô hình rất tốt trong việc phân biệt giữa các lớp. - Đảm bảo khả năng phát hiện đúng các trường hợp positive (ví dụ: khách hàng có nguy cơ vỡ nợ)

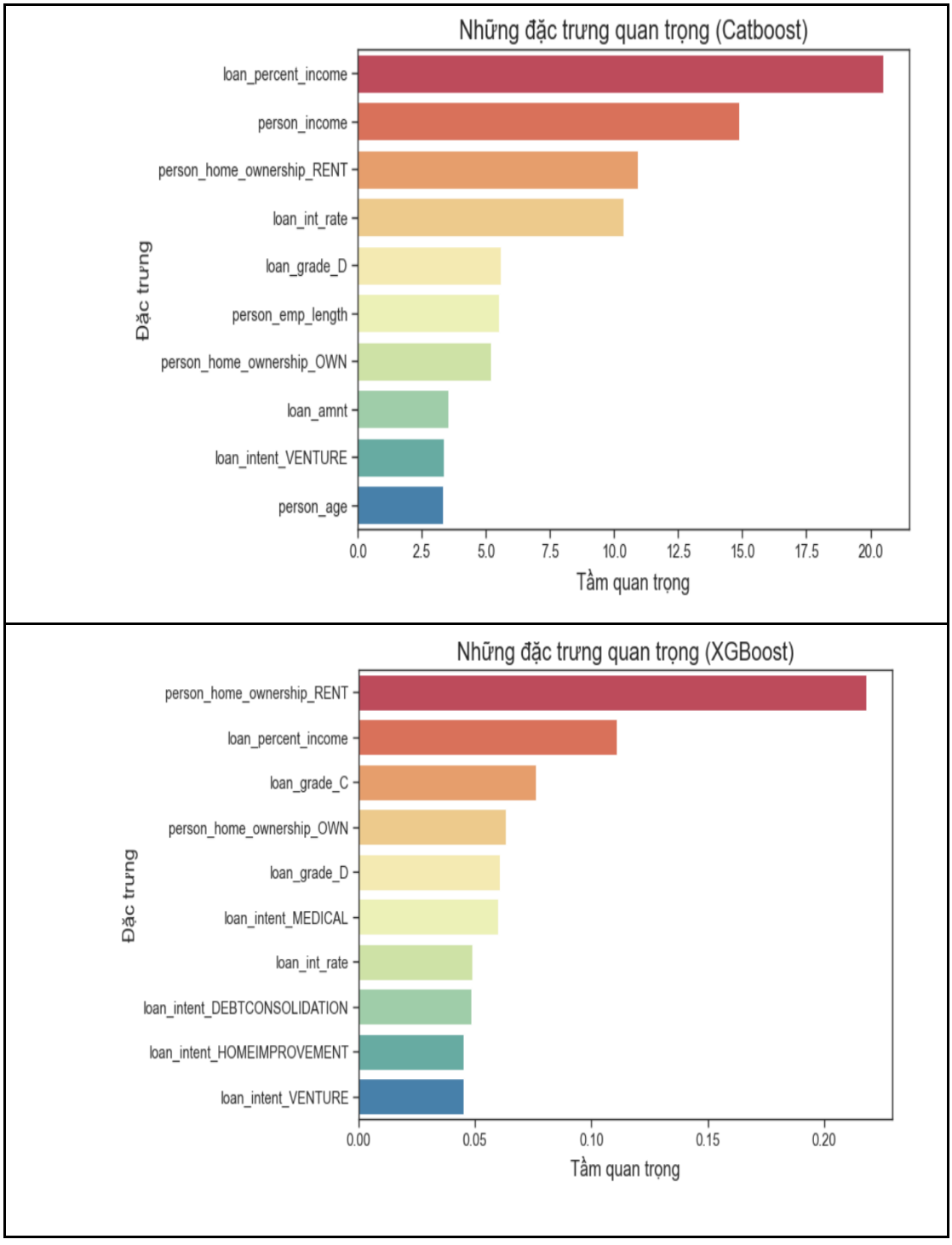
AUC trung bình **(0.7 - 0.9)**	- Mô hình có hiệu quả chấp nhận được. - Có thể cần tối ưu hóa thêm để tăng độ chính xác hoặc giảm lỗi phân loại.
AUC thấp **(< 0.7)**	- Hiệu suất mô hình kém, khó phân biệt rõ giữa các lớp. - Cần cải thiện bằng cách xử lý dữ liệu, chọn mô hình khác, hoặc điều chỉnh tham số.

➢ **Đánh giá AUC của các mô hình qua plotting graph:**



Nhận xét: Dựa trên biểu đồ, XGBoost và CatBoost là hai mô hình tối ưu nhất cho bài toán dự đoán rủi ro tín dụng nhờ giá trị AUC cao có thể phân biệt giữa các lớp một cách hiệu quả (rủi ro vỡ nợ và không rủi ro).

8.3. Đánh giá những đặc trưng quan trọng trong mô hình dự đoán



Nhận xét:

- **CatBoost** ưu tiên các đặc trưng liên quan đến tỷ lệ thu nhập và khoản vay hơn, trong khi **XGBoost** có xu hướng nhấn mạnh quyền sở hữu nhà và các biến liên quan đến mục đích vay.
- Mức độ phân tách tầm quan trọng giữa các đặc trưng của CatBoost rõ ràng hơn so với XGBoost, thể hiện qua việc một số đặc trưng chiếm ưu thế lớn.

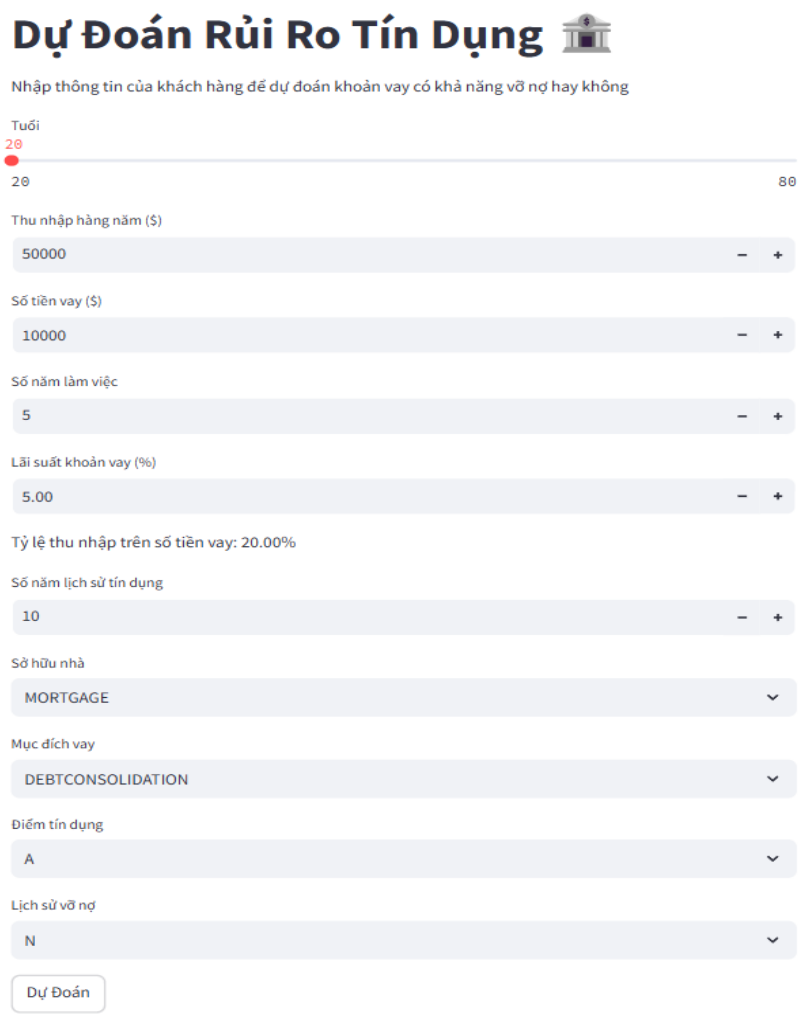
9. Xây dựng mô hình dự đoán rủi ro tín dụng trên Streamlit

- Lưu lại mô hình tốt nhất:

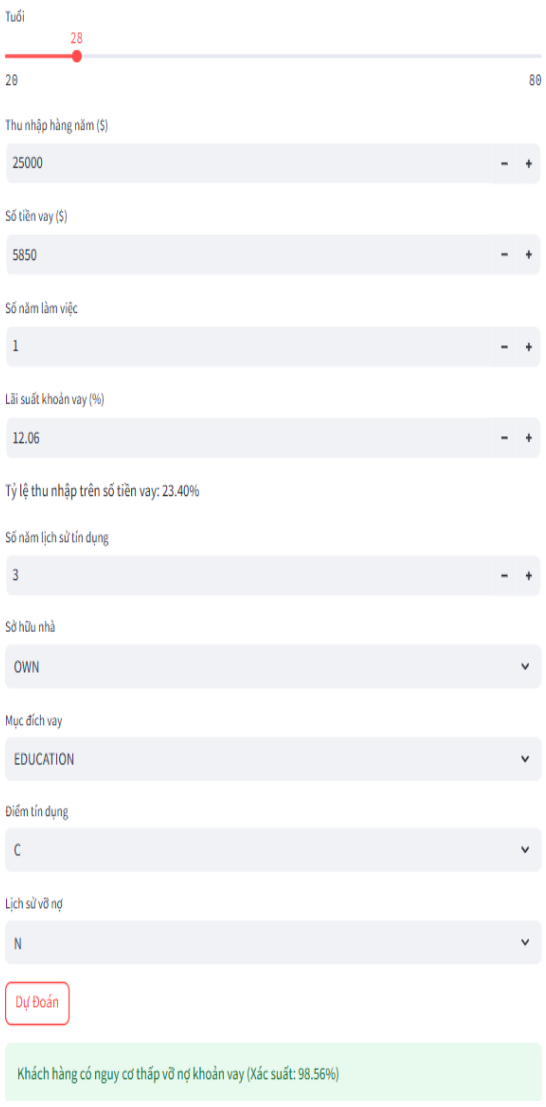
```
import joblib

# Lưu mô hình CatBoost
joblib.dump(CB_model, 'catboost_model.pkl')
```

```
['catboost_model.pkl']
```

- Áp dụng mô hình vào bài toán dự đoán rủi ro tín dụng (vỡ nợ khoản vay):
  + Giao diện xây dựng trên Streamlit:
  + Link web: https://creditriskapp-vsxpewncyjc3zkplnb5yhu.streamlit.app/



+ Nhập dữ liệu để dự đoán

Trường hợp vỡ nợ	Trường hợp không vỡ nợ
	

IV) Kết luận

- Về trực quan dữ liệu:

Phân tích các yếu tố định lượng và phân loại đã làm rõ nhiều khía cạnh quan trọng trong việc xét duyệt tín dụng. Một số điểm nổi bật gồm:

- Tập trung vào các yếu tố như giá trị khoản vay, tỷ lệ vay theo tổng thu nhập, và lãi suất để giảm thiểu rủi ro vỡ nợ.
- Cân nhắc thời gian làm việc và lịch sử tín dụng khi đánh giá hồ sơ vay.
- Cần chú ý trong việc xây dựng các mô hình dự báo chính xác hơn bằng cách xử lý dữ liệu mất cân bằng và đa cộng tuyến từ các biến có tương quan mạnh trong tập dữ liệu.

⇨ Phân tích này cung cấp cơ sở cho việc ra xây dựng chiến lược xét duyệt khoản vay hiệu quả hơn, đồng thời hỗ trợ trong việc xây dựng mô hình dự đoán rủi ro và nâng cao hiệu quả xét duyệt tín dụng.

- Về mô hình máy học:

Trong dự án này, một số kết quả quan trọng được rút ra như sau:

- Đề tài sử dụng các mô hình Machine Learning để tìm ra mô hình tối ưu cho việc dự đoán rủi ro tín dụng (Credit Risk). Mô hình hiệu quả nhất được xác định là CatBoost.
- Độ chính xác của mô hình này đạt 93,87%.
- Bên cạnh đó, ROC của mô hình CatBoost rất cao (auc = 95%).
- XGBoost cũng là một mô hình cho kết quả dự đoán tương đối tốt và không kém gì CatBoost trong dự án này.

Tóm lại, việc xây dựng mô hình dự đoán rủi ro tín dụng là rất quan trọng đối với các ngân hàng và tổ chức tín dụng. Vì vậy, khi thực hiện dự đoán rủi ro tín dụng, việc lựa chọn phương pháp phù hợp là rất cần thiết, nhằm cung cấp thông tin chính xác cho các dự báo, yêu cầu người thực hiện phải hiểu rõ cách áp dụng các phương pháp này một cách hiệu quả.

V) Tài liệu tham khảo

[1]
https://www.kaggle.com/code/dangvannam/methods-balance-data#B.2-K%E1%BB%B9-thu%E1%BA%ADt-Oversampling: Imbalanced data

[2] https://tapchinganhang.gov.vn/du-doan-rui-ro-tin-dung-su-dung-hoc-sau-11142.html

[3]
https://hvnh.edu.vn/medias/tapchi/vi/07.2024/system/archivedate/b2435167_2762-%20S%E1%BB%91%20266-T7CD.2024-%20Nguy%E1%BB%85n%20Minh%20Nh%E1%BA%ADt,%20Ng%C3%B4%20Ho%C3%A0ng%20Kh%C3%A1nh%20Duy-%20D%E1%BB%B1%20b%C3%A1o%20kh%E1%BA%A3%20n%C4%83ng%20v%E1%BB%A1%20n%E1%BB%A3%20c%E1%BB%A7a%20doanh%20nghi%E1%BB%87p%20nh%E1%BB%8F%20v%C3%A0%20v%E1%BB%ABa%20t%E1%BA%A1i%20Vi%E1%BB%87t%20Nam.pdf