ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC KINH TẾ - LUẬT
KHOA HỆ THỐNG THÔNG TIN

ĐỒ ÁN MÔN HỌC
KHAI PHÁ DỮ LIỆU & ỨNG DỤNG

Tên đề tài:

Default of credit card clients Data Set

(DỰ ĐOÁN VỠ NỢ CỦA KHÁCH HÀNG)

Giáo viên bộ môn: ThS. Nguyễn Thị Anh Thư

Danh sách sinh viên:

Tên	**MSSV**
Trịnh Thu Huyền Trang – Nhóm Trưởng	K184060811
Nguyễn Trúc Loan	K184060793
Trần Huyền Mơ Mơ	K184060794
Nguyễn Thị Thu Thảo	K184060804
Nguyễn Vũ Minh Thùy	K184060806

TP.HCM, Ngày 10 Tháng 01 Năm 2021

MỤC LỤC

MỤC LỤC HÌNH ẢNH

MỤC LỤC BIỂU ĐỒ

MỤC LỤC BẢNG

CHƯƠNG I: TỔNG QUAN

1.1 Giới thiệu

Đề tài được thực hiện nhằm kiểm định các nhân tố tài chính tác động đến rủi ro vỡ nợ của các khách hàng. Dữ liệu được thu thập thông qua việc thu thập số liệu của các ngân hàng cung cấp theo từng năm. Dữ liệu gồm nhiều yếu tố khác nhau về việc thu chi của khách hàng thông qua ngân hàng. Thuật toán Random Forest được sử dụng để xác định các nhân tố tác động rủi ro vỡ nợ, cũng như dự báo khả năng phát sinh rủi ro của các khách hàng trong tháng tới.

1.2 Mục tiêu

1.2.1 Mục tiêu của bài toán

Nghiên cứu này nhằm vào trường hợp thanh toán vỡ nợ của khách hàng ở Đài Loan và so sánh độ chính xác dự đoán của xác suất vỡ nợ giữa sáu phương pháp khai thác dữ liệu. Từ góc độ quản lý rủi ro, kết quả dự đoán chính xác về xác suất vỡ nợ ước tính sẽ có giá trị hơn dựa vào kết quả phân loại nhị phân - khách hàng đáng tin cậy hoặc không đáng tin cậy. Bởi vì xác suất vỡ nợ thực sự là không xác định, nghiên cứu này đã trình bày cuốn tiểu thuyết - Phương pháp làm mịn phân loại - để ước tính xác suất vỡ nợ thực.

1.2.2 Mục tiêu kiến thức

- Vận dụng kiến thức môn học khai phá dữ liệu trong đồ án môn học
- Biết cách sử dụng ngôn ngữ python trong lĩnh vực khai phá dữ liệu
- Hiểu rõ các thuật toán nhằm mục đích phân lớp dự đoán kết quả
- Nắm được các phương pháp thao tác, tiền xử lý dữ liệu
- Hiểu thêm các kiến thức trong lĩnh vực khai phá dữ liệu

1.3 Đối tượng nghiên cứu

- Tên tập dữ liệu : Default of credit card clients (Sự vỡ nợ của khách hàng sử dụng thẻ tín dụng).
- Nguồn dữ liệu:
    - UCI Machine Learning Repository
    - Name: I-Cheng Yeh

- Email: (1) icyeh '@' chu.edu.tw (2) 140910 '@' mail.tku.edu.tw
- Institutions: (1) Department of Information Management, Chung Hua University, Taiwan. (2) Department of Civil Engineering, Tamkang University, Taiwan.
- Other contact information: 886-2-26215656 ext. 3181

Link download: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

1.4 Phát biểu bài toán:

1.4.1 Input

Input của bài toán là tập dữ liệu về nhân khẩu học (giới tính, trình độ học vấn, tình trạng hôn nhân, tuổi tác), hạn mức tín dụng, tình trạng trả nợ, lịch sử thanh toán,bảng sao kê hóa đơn của khách hàng sử dụng thẻ tín dụng ở Đài Loan từ tháng 4 năm 2005 đến tháng 9 năm 2005 được thu thập để dự đoán chính xác xác suất vỡ nợ của khách hàng.

Tập dữ liệu có:

- 30000 mẫu dữ liệu.
- Mỗi mẫu có 24 thuộc tính.
- Ứng với mỗi thuộc tính có các thông tin sau đây:

STT	Tên thuộc tính	Ý nghĩa
1	ID	Số thứ tự
2	LIMIT_BAL	Hạn mức tín dụng
3	SEX	Giới tính
4	EDUCATION	Trình độ học vấn
5	MARRIAGE	Trình trạng hôn nhân
6	AGE	Tuổi tác
7	PAY_0	Tình trạng trả nợ trong tháng 9/2005

8	PAY_2	Tình trạng trả nợ trong tháng 8/2005
9	PAY_3	Tình trạng trả nợ trong tháng 7/2005
10	PAY_4	Tình trạng trả nợ trong tháng 6/2005
11	PAY_5	Tình trạng trả nợ trong tháng 5/2005
12	PAY_6	Tình trạng trả nợ trong tháng 4/2005
13	BILL_ATM1	Số tiền của bảng sao kê hóa đơn tháng 9/2005
14	BILL_ATM2	Số tiền của bảng sao kê hóa đơn tháng 8/2005
15	BILL_ATM3	Số tiền của bảng sao kê hóa đơn tháng 7/2005
16	BILL_ATM4	Số tiền của bảng sao kê hóa đơn tháng 6/2005
17	BILL_ATM5	Số tiền của bảng sao kê hóa đơn tháng 5/2005
18	BILL_ATM6	Số tiền của bảng sao kê hóa đơn tháng 4/2005
19	PAY_ATM1	Số tiền thanh toán vào tháng 9/2005
20	PAY_ATM2	Số tiền thanh toán vào tháng 8/2005
21	PAY_ATM3	Số tiền thanh toán vào tháng 7/2005
22	PAY_ATM4	Số tiền thanh toán vào tháng 6/2005
23	PAY_ATM5	Số tiền thanh toán vào tháng 5/2005
24	PAY_ATM6	Số tiền thanh toán vào tháng 4/2005

Bảng 1: Thông tin thuộc tính

1.4.2 Output

- Output là dữ liệu về tình trạng vỡ nợ tháng tới của khách hàng sử dụng thẻ tín dụng (Default payment in next month), có 2 giá trị: 1=có vỡ nợ vào tháng tới, 0=không vỡ nợ vào tháng tới.

STT	Tên thuộc tính	Ý nghĩa
1	Default payment in next mouth	Tình trạng vỡ nợ vào tháng tới

Bảng 2: Thuộc tính output

CHƯƠNG II: CƠ SỞ LÝ THUYẾT

2.1 Khai phá dữ liệu

2.1.1 Khai phá dữ liệu là gì?

−Khai phá dữ liệu (data mining) Là quá trình tính toán để tìm ra các mẫu trong các bộ dữ liệu lớn liên quan đến các phương pháp tại giao điểm của máy học, thống kê và các hệ thống cơ sở dữ liệu.

−Mục tiêu tổng thể của quá trình khai thác dữ liệu là trích xuất thông tin từ một bộ dữ liệu và chuyển nó thành một cấu trúc dễ hiểu để sử dụng tiếp.

2.1.2 Bài toán phân lớp:

Là dạng phân tích dữ liệu nhằm rút trích các mô hình mô tả các lớp dữ liệu hoặc dự đoán xu hướng dữ liệu.

− Quá trình gồm hai bước:

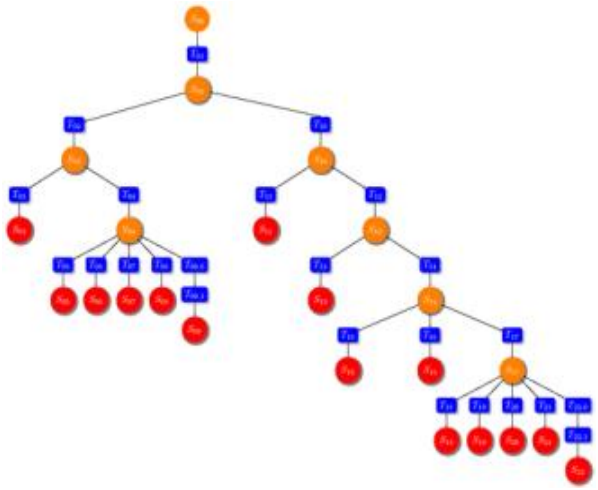
+ Bước học (giai đoạn huấn luyện): xây dựng bộ phân

+ Lớp (classifier) bằng việc phân tích/học tập huấn luyện

+ Bước phân lớp (classification): phân lớp dữ liệu/đối tượng mới nếu độ chính xác của bộ phân lớp được đánh giá là có thể chấp nhận được (acceptable)

2.2. Các thuật toán sử dụng

2.2.1 Decision trees



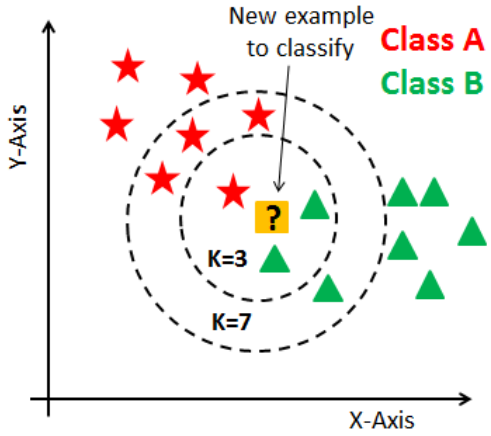
Hình 1: Mô tả thuật toán Decision Tree

Cây quyết định này phụ thuộc vào các câu trả lời của các câu hỏi được đưa ra. Dựa trên các câu trả lời để đưa ra quyết định. Thông thường được đưa vào bài toán phân

lớp nhị phân. Nút đầu tiên gọi là nút gốc, nút cuối cùng gọi là nút con. Các nút còn lại là nút lá. Mỗi nút lá đại diện cho giá trị dự đoán của biến mục tiêu, cho trước các giá trị của các biến được biểu diễn bởi đường đi từ nút gốc tới nút lá đó.

Nhiệm vụ là đi tìm ranh giới đơn giản giúp phân chia hai class này. Hay nói cách khác, đây là một bài toán classification, ta cần xây dựng một bộ phân lớp để quyết định việc một điểm dữ liệu mới thuộc vào class nào.

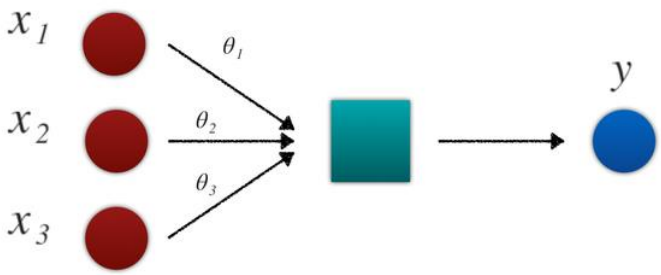
2.2.2 K-Nearest



Hình 2: Mô tả thuật toán K-nearest

KNN dựa trên giả định là những thứ tương tự hay có tính chất gần giống nhau sẽ nằm ở vị trí gần nhau, với giả định như vậy, KNN được xây dựng trên các công thức toán học phục vụ để tính khoảng cách giữa 2 điểm dữ liệu (gọi là Data points) để xem xét mức độ giống nhau của chúng.

2.2.3 Phân tích hồi quy logistic



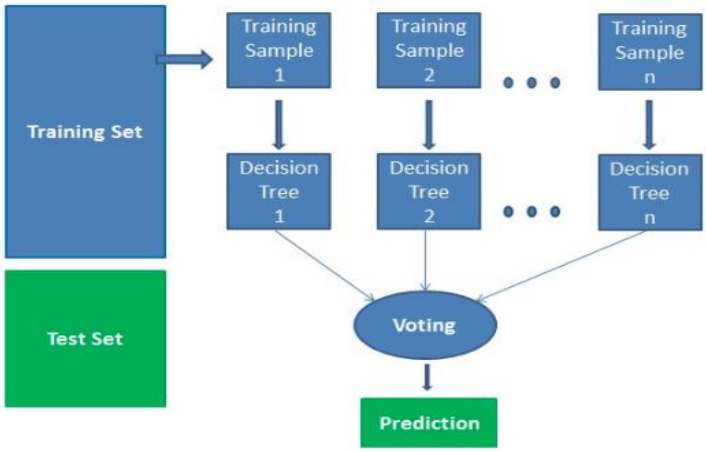
Hình 3: Mô hình thể hiện phương pháp Logistic regression

Logistic Regression là 1 thuật toán phân loại được dùng để gán các đối tượng

cho 1 tập hợp giá trị rời rạc (như 0, 1, 2, ...). Một ví dụ điển hình là phân loại Email, gồm có email công việc, email gia đình, email spam, ... Giao dịch trực tuyến có là an toàn hay không an toàn, khối u lành tính hay ác tình.

Sử dụng hàm sigmoid đưa về giá trị (0,1)

2.3.4 Random Forest



Hình 4: Thuật toán Random Forest

Cây quyết định riêng lẻ được tạo ra bằng cách sử dụng chỉ báo chọn thuộc tính như tăng thông tin, tỷ lệ tăng và chỉ số Gini cho từng thuộc tính. Mỗi cây phụ thuộc vào một mẫu ngẫu nhiên độc lập. Trong bài toán phân loại, mỗi phiếu bầu chọn và lớp phổ biến nhất được chọn là kết quả cuối cùng.

2.3 Các độ đo đánh giá mô hình

2.3.1 Precision:

Precision là tỉ lệ giữa các giá trị tích cực được dự đoán chính xác (TP) trên tổng giá trị tích cực được dự đoán(TP+FP). Chỉ số này làm nổi bật các dự đoán tích cực về tính chính xác trong số tất cả các dự đoán tích cực. Precision cao cho thấy tỉ lệ tích cực giả thấp.

$$\text{Precision} = \frac{\text{TP}}{(\text{TP}+\text{FP})}$$

2.3.2 Recall:

Recall là tỉ lệ giữa các giá trị tích cực được dự đoán chính xác(TP) so với giá trị

tích cực thực tế(TP+FN). Recall làm nổi bật độ nhạy cảm của thuật toán, tức là trong số tất cả các giá trị tích cực thực tế thì chương trình bắt gặp được bao nhiêu.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$$

2.3.3 F1-score:

F1-score là tỷ lệ model phân loại, dự báo đúng cho các trường hợp Negative và Positive. F1 là phương pháp hay để xem xét cả 2 độ đo Precision và Recall để hiểu rõ hơn về cách xử lý một phân lớp.

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

2.3.4 Accuracy:

Accuracy là số lượng các dự đoán chính xác được thực hiện theo tỉ lệ của tất cả các dự đoán được thực hiện. Nó là một trong những số liệu phổ biến nhất và chỉ phù hợp khi tập dữ liệu được cân bằng chứa số lượng quan sát bằng nhau trong mỗi lớp.

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

2.3.5 Error rate:

Error rate là tỉ lệ model phân loại, dự báo sai cho tất cả trường hợp Negative và Positive.

$$Error - rate = 1 - accuracy = \frac{1 - \text{TP+TN}}{\text{TP+ FP+TN+FN}}$$

! CHÚ THÍCH:

TP: True Positive	TN: True Negative
FP: False Positive	FN: False Negative

2.4 Công cụ và thư viện

2.4.1 Công cụ sử dụng

- Python 3.8*:* được cài đặt cho việc sử dụng ngôn ngữ lập trình Python, với ưu điểm là có nhiều thư viện được sử dụng cho các thuật toán như scikit learn,

pandas,... và cấu trúc dòng lệnh ngắn gọn dễ hiểu.

- Heroku: là nền tảng đám mây hỗ trợ hosting web miễn phí và có hỗ trợ backend là ngôn ngữ python phù hợp với đề tài.
- Github: là nền tảng để lưu trữ source code dùng để kết nối với Heroku nhằm hosting web một cách dễ dàng và mượt mà.

2.4.2 Thư viện sử dụng trong python

2.4.2.1 Pandas

2.4.2.1.1 Định nghĩa

−Pandas là là thư viện mã nguồn mở với hiệu năng cao cho phân tích dữ liệu trong Python được phát triển bởi Wes McKinney trong năm 2008.

−Thư viện này sử dụng một cấu trúc dữ liệu riêng là Dataframe. Pandas cung cấp rất nhiều chức năng xử lý và làm việc trên cấu trúc dữ liệu này.

2.4.2.1.2 Mục đích

−Biểu diễn dữ liệu: Nó có thể dễ dàng biểu diễn tập dữ liệu một cách tự nhiên phù hợp với dữ liệu phân tích thông qua cấu trúc DataFrame và Series (Những khái niệm chúng ta sẽ tìm hiểu chi tiết ở các bài sau). Dùng với các ngôn ngữ khác mất nhiều công code hơn

−Tập hợp và lọc dữ liệu: Nó cung cấp các phương thức để dễ dàng tập hợp và lọc dữ liệu, những thủ tục gắn liền với việc phân tích dữ liệu

−Code ngắn gọn và clear hơn: những API ngắn gọn và clear cho phép người dùng tập trung vào những mục tiêu chính hơn là việc phải viết một đoạn code dài từ đầu để thực hiện công việc

2.4.2.2 Numpy

2.4.2.2.1 Định nghĩa

−NumPy là một gói Python là viết tắt của Numerical Python. Đây là thư viện cốt lõi cho scientific computing, nó chứa một đối tượng mảng n chiều mạnh mẽ, cung cấp các công cụ để tích hợp C, C ++, v.v. Nó cũng hữu ích trong đại số tuyến tính, random number capability, ...

2.4.2.2.2 Mục đích

− Hỗ trợ cho việc tính toán các mảng nhiều chiều, có kích thước lớn với các hàm đã được tối ưu áp dụng lên các mảng nhiều chiều đó.

2.4.2.3 Matplotlib

2.4.2.3.1 Định nghĩa

- Matplotlib là một thư viện vẽ đồ thị rất mạnh mẽ hữu ích cho những người làm việc với Python và NumPy.
- Một Matplotlib figure có thể được phân loại thành nhiều phần như dưới đây:
- Figure: Như một cái cửa sổ chứa tất cả những gì bạn sẽ vẽ trên đó.
- Axes: Thành phần chính của một figure là các axes (những khung nhỏ hơn để vẽ hình lên đó). Một figure có thể chứa một hoặc nhiều axes. Nói cách khác, figure chỉ là khung chứa, chính các axes mới thật sự là nơi các hình vẽ được vẽ lên.
- Axis: Chúng là dòng số giống như các đối tượng và đảm nhiệm việc tạo các giới hạn biểu đồ.
- Artist: Mọi thứ mà bạn có thể nhìn thấy trên figure là một artist như Text objects, Line2D objects, collection objects. Hầu hết các Artists được gắn với Axes.

4.2.3.2. Mục đích

Đơn giản hóa tối đa công việc vẽ biểu đồ để "chỉ cần vài dòng lệnh" Hỗ trợ rất nhiều loại biểu đồ, đặc biệt là các loại được sử dụng trong nghiên cứu hoặc kinh tế như biểu đồ dòng, đường, tần suất (histograms), phổ, tương quan, errorcharts, scatterplots,…

2.4.2.4 Seaborn

2.4.2.4.1 Định nghĩa

- Seaborn : trực quan hóa dữ liệu thống kê là một thư viện Python phổ biến để thực hiện EDA.
- Nó dựa trên matplotlib và cung cấp giao diện cấp cao để vẽ đồ họa thống kê hấp dẫn và nhiều thông tin.

2.4.2.4.2 Mục đích

Seaborn nhằm mục đích làm cho trực quan hóa và tạo điểm nhấn cho các dữ liệu khi khám phá dữ liệu. Các chức năng vẽ hoạt động trên các dataframe và mảng có chứa toàn bộ tập dữ liệu và thực hiện các phép aggregations cần thiết và mô hình thống kê phù hợp để tạo ra các đồ thị thông tin.

2.4.2.5 Scikit-learn

2.4.2.5.1. Định nghĩa

Scikit-learn (Sklearn) là thư viện mạnh mẽ nhất dành cho các thuật toán học máy được viết trên ngôn ngữ Python. Thư viện cung cấp một tập các công cụ xử lý các bài toán machine learning và statistical modeling gồm: classification, regression, clustering, và dimensionality reduction.

2.4.2.5.2 Mục đích

- Cung cấp các giải thuật về các task như là phân loại, hồi quy, clustering, v.v…, cung cấp các phương tiện tiện lợi trong việc đánh giá hyperparameter grid search, cross-validation, v.v…
- Cung cấp interface thống nhất cho các thuật toán khác nhau.

2.4.2.6 Flask

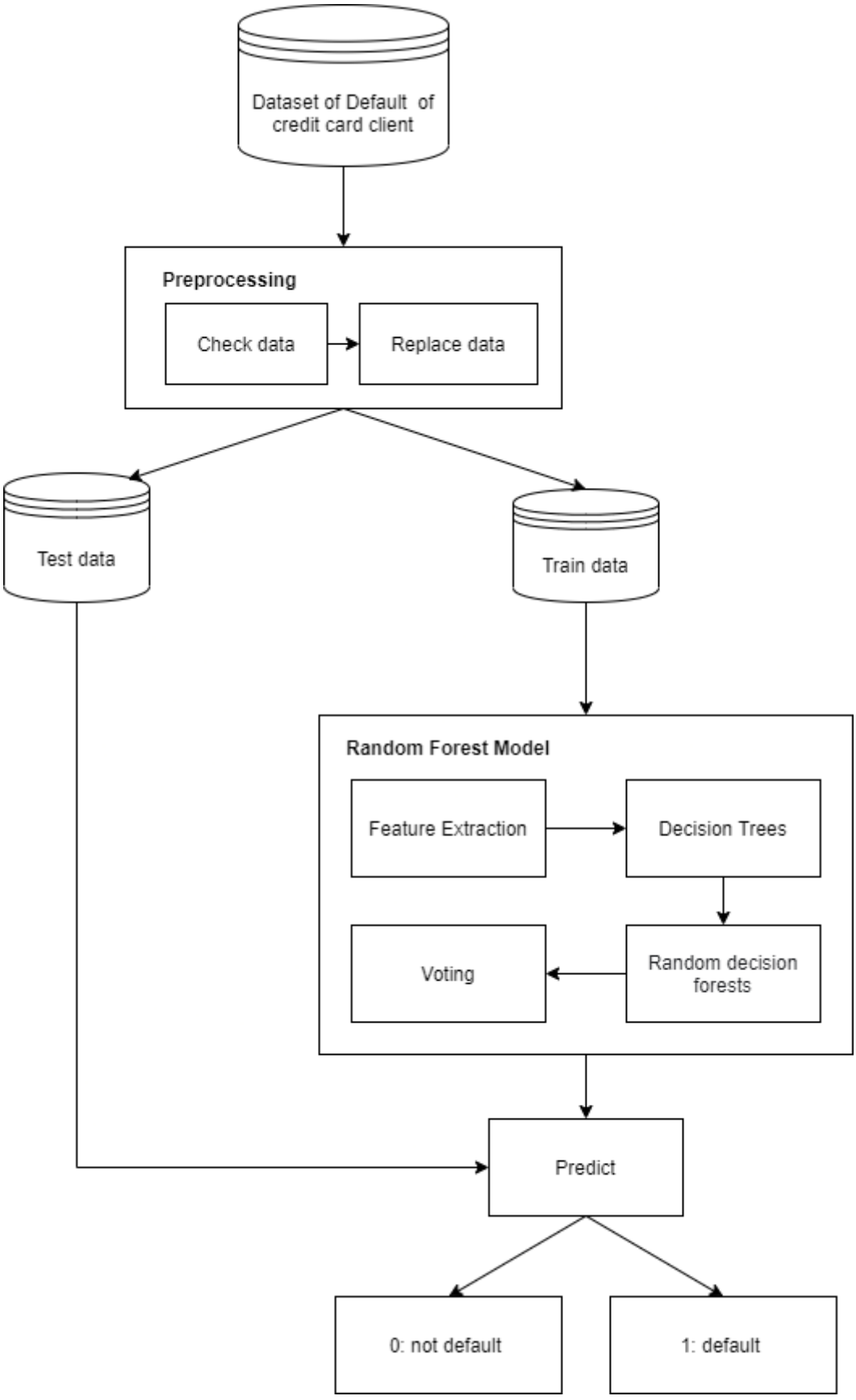
2.4.2.6.1 Định nghĩa

- Flask là một web frameworks, nó thuộc loại micro-framework được xây dựng bằng ngôn ngữ lập trình Python. Flask cho phép xây dựng các ứng dụng web từ đơn giản tới phức tạp. Flask có thể xây dựng các api nhỏ, ứng dụng web chẳng hạn như các trang web, blog, trang wiki hoặc một website dựa theo thời gian hay thậm chí là một trang web thương mại.

2.4.2.6.2 Mục đích

- Flask cung cấp cho các lập trình viên khả năng tùy biến khi phát triển ứng dụng web, nó cung cấp cho bạn các công cụ, thư viện và cơ chế cho phép bạn xây dựng một ứng dụng web nhưng nó sẽ không thực thi bất kỳ sự phụ thuộc nào hoặc cho bạn biết dự án sẽ như thế nào.
- Ứng dụng web có thể là blog, trang web thương mại hoặc một số trang web khác, nó vẫn cho phép các lập trình viên cơ hội sử dụng một số tiện ích mở rộng để thêm nhiều chức năng hơn cho ứng dụng web.

CHƯƠNG III: MÔ HÌNH GIẢI BÀI TOÁN (Framework)



Hình 5: Framework bài toán

Mô hình giải bài toán được thể hiện như hình trên, với dữ liệu ban đầu được tiền xử lý dữ liệu bằng cách kiểm tra dữ liệu sau đó thay thế những giá trị trùng gặp hoặc không có nghĩa. Sau đó dữ liệu được chia thành 2 tập dữ liệu là dữ liệu test và dữ liệu train - được dùng để cho máy học sau đó dùng dữ liệu test để kiểm tra độ chính xác của mô hình. Thuật toan được đề xuất là Random Forest, bằng cách trích xuất đặc trưng dữ liệu tạo ra nhiều cây quyết định ngẫu nhiên từ đó ta có một rừng cây và theo cơ chế bầu chọn (voting) để chọn ra mô hình tốt nhất. Dữ liệu kết quả cuối cùng (output) là 0 nếu khách hàng sẽ không vỡ nợ vào tháng tới hoặc 1 nếu khách hàng có nguy cơ vỡ nợ.

3.1 Tiền xử lý dữ liệu.

Là quá trình thu thập, làm sạch, tái cấu trúc và làm phong phú dữ liệu thô có sẵn thành một định dạng dễ sử dụng hơn. Điều này sẽ giúp nhanh chóng quá trình đưa ra quyết định và có được thông tin chi tiết tốt hơn trong thời gian ngắn hơn.

Các mục tiêu nhất định của bước này sẽ là xử lý các giá trị bị thiếu, kiểm tra các kiểu dữ liệu và các giá trị của dữ liệu đã chính xác và phù hợp hay chưa.

3.1.1 Kiểm tra dữ liệu

Dữ liệu ban đầu có 25 cột: gồm 24 thuộc tính và 1 nhãn *(label).*

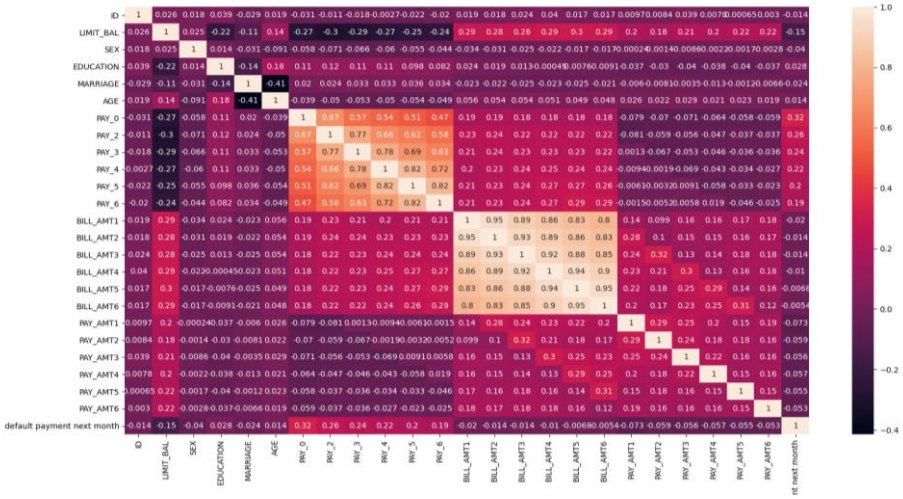
ID	Column	Non-Null Count	Dtype
0	ID	30000 non-null	Int64
1	LIMIT_BAL	30000 non-null	Int64
2	SEX	30000 non-null	Int64
3	EDUCATION	30000 non-null	Int64
4	MARRIAGE	30000 non-null	Int64
5	AGE	30000 non-null	Int64
6	PAY_0	30000 non-null	Int64
7	PAY_2	30000 non-null	Int64
8	PAY_3	30000 non-null	Int64
9	PA_4	30000 non-null	Int64
10	PAY_5	30000 non-null	Int64
11	PAY_6	30000 non-null	Int64
12	BILL_AMT1	30000 non-null	Int64
13	BILL_AMT2	30000 non-null	Int64
14	BILL_AMT3	30000 non-null	Int64
15	BILL_AMT4	30000 non-null	Int64
16	BILL_AMT5	30000 non-null	Int64
17	BILL_AMT6	30000 non-null	Int64
18	PAY_AMT1	30000 non-null	Int64
19	PAY_AMT2	30000 non-null	Int64
20	PAY_AMT3	30000 non-null	Int64
21	PAY_AMT4	30000 non-null	Int64
22	PAY_AMT5	30000 non-null	Int64
23	PAY_AMT6	30000 non-null	Int64
24	default	30000 non-null	Int64
Dtypes: int64(25)			

ID	0
LIMIT_BALL	0
SEX	0
EDUCATION	0
MARRIAGE	0
AGE	0
PAY_0	0
PAY_2	0
PAY_3	0
PA_4	0
PAY_5	0
PAY_6	0
BILL_AMT1	0
BILL_AMT2	0
BILL_AMT3	0
BILL_AMT4	0
BILL_AMT5	0
BILL_AMT6	0
PAY_AMT1	0
PAY_AMT2	0
PAY_AMT3	0
PAY_AMT4	0
PAY_AMT5	0
PAY_AMT6	0
Default payment next month	0
Dtype: int64	

Bảng 3: Thông tin bản dữ liệu

Dữ liệu được kiểm tra xem có bị thiếu giá trị hay không. Từ hình trên cho thấy dữ liệu đầy đủ, không bị mất mát các giá trị. Toàn bộ dữ liệu được lưu trữ dưới dạng số, không kèm chữ.

3.1.2 Làm mịn dữ liệu



Hình 6: Hình thể hiện mối tương quan Person giữa các biến

Thông qua hình 3 chúng ta có thể thấy rằng không có biến nào có mối tương quan chặt chẽ với vấn đề vỡ nợ của khách hàng ở tháng tiếp theo. Các biến "PAY_X" có một lượng nhỏ mối tương quan thuận, thể hiện rằng nếu khách hàng đã hoặc chưa vỡ nợ ở những tháng trước có thể quyết định xác suất vỡ nợ ở tháng tiếp theo.

Các biến “BILL_AMT” có mối tương quan thuận chặt chẽ với nhau. Điều này cũng có nghĩa là các mô hình chi tiêu của khách hàng có xu hướng không đổi.

Biến “LIMIT_BAL” có mối tương quan thuận với 'BILL_AMT' và mối tương quan nghịch với các biến 'PAY'.

Từ những điều trên cho thấy dữ liệu không có điều gì bất thường, tuy nhiên có một vài giá trị thuộc các biến vẫn còn khá mơ hồ, nên được xử lý để làm gọn dữ liệu, rút ngắn thời gian chạy thuật toán.

2	18,112
1	11,888
Name: SEX, dtype: int64	
2	15,964
1	13,659
3	323
0	54
Name: MARRIAGE , dtype : int64	

2	14,030
1	10,585
3	4,917
5	280
4	123
6	51
0	14
Name: EDUCATION , dtype : int64	

Hình 7: Giá trị và số lượng của biến giới tính, giáo dục và tình trạng hôn nhân

Biến 'EDUCATION' có giá trị “0” không được giải thích và biến “5”, “6” được giải thích rất mơ hồ. Vì ở biến “4” đã thể hiện giá trị khác.

Biến 'MARRIAGE' có một giá trị không rõ ràng khác là “0”.

Chúng ta có thể kết hợp các giá trị mơ hồ với nhau vì chúng cũng không mang lại nhiều ý nghĩa.

2	14,030
1	10,585
3	4,917
4	468
Name: EDUCATION , dtype : int64	

2	15,964
1	13,659
3	323
0	54
Name: MARRIAGE , dtype : int64	

Hình 8: Các giá trị mơ hồ đã được thay bằng giá trị mang ý nghĩa khác (Others)

Dữ liệu đã làm sạch một số biến nhất định, do đó mức độ tương quan giữa các biến đã được cải thiện một phần nhỏ.

“ Improvements in correlation: [0.00583615 - 0.00323549]”

	EDUCATION	MARRIAGE	default.payment.next.month
EDUCATION	1.000000	-0.136797	0.033842
MARRIAGE	-0.136797	1.000000	-0.027575
default.payment.next.month	0.033842	-0.027575	1.000000

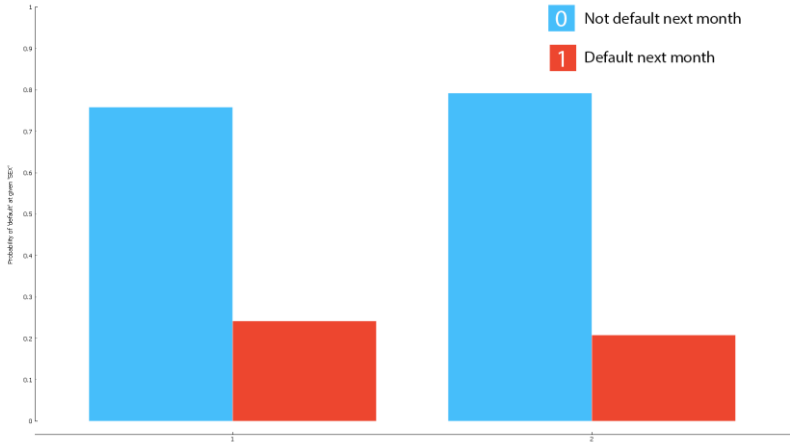
Hình 9: Giá trị tương quan giữa các biến sau khi thay thế một vài giá trị mơ hồ

Cách thay thế này chắc chắn sẽ có lợi cho các mô hình có thể giúp đạt được những cải tiến nhỏ và đi theo hướng tích cực.

Biến ID cũng cần được loại bỏ trong quá trình chạy thực nghiệm vì không có ý nghĩa trong việc dự đoán. Nên dữ liệu còn lại 23 biến và 1 nhãn.

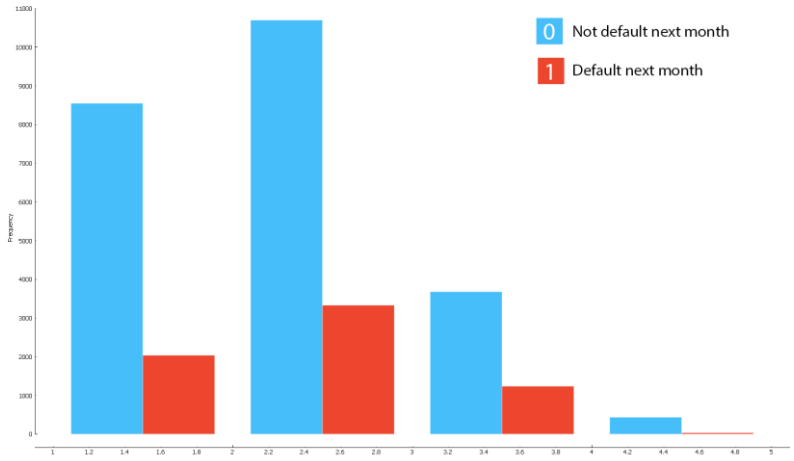
3.1.3 Trực quan hóa dữ liệu

Bước này sử dụng các biểu đồ trực quan hóa dữ liệu để biết xu hướng phân loại của các biến nhằm tìm ra xu hướng và thông tin trong dữ liệu.



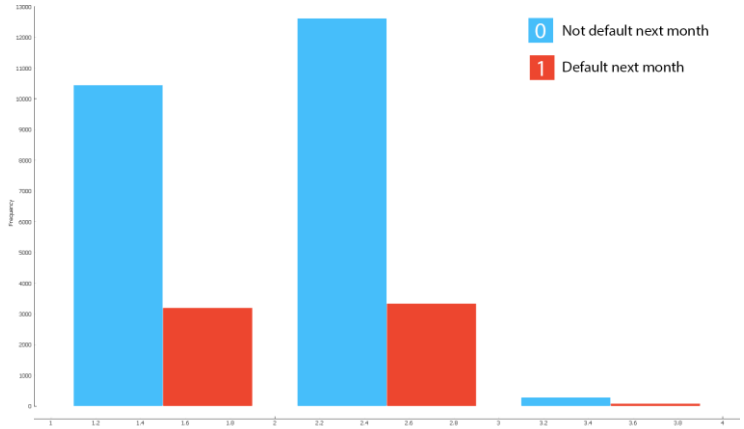
Biểu đồ 1: Biểu đồ thể hiện xác suất vỡ nợ giữa Nam và Nữ

Tỉ lệ Nam vỡ nợ cao hơn Nữ, với khoảng 24% đối với Nam và 20% đối với Nữ.



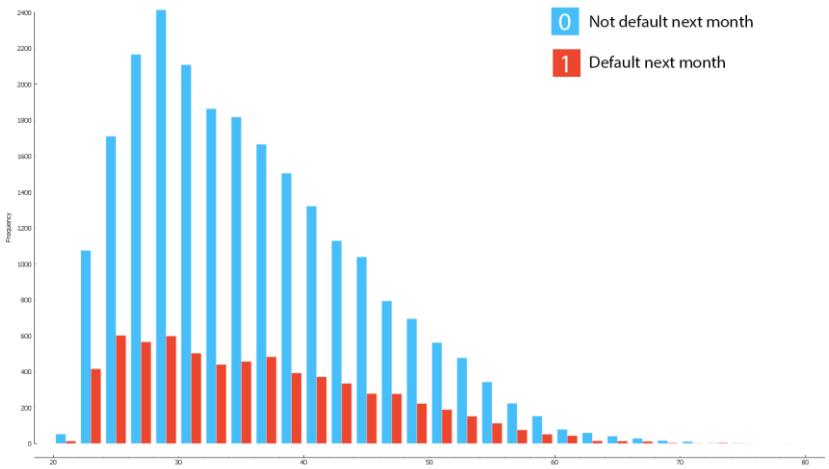
Biểu đồ 2: Biểu đồ trực quan dữ liệu theo thuộc tính giáo dục

Từ biểu đồ trên có thể thấy rằng trình độ học vấn càng cao thì có xu hướng làm giảm xác suất vỡ nợ. Điều này có ý nghĩa bởi vì người có ít giáo dục hơn thì có thể đồng nghĩa với mức lương và hạn mức tín dụng thấp hơn.



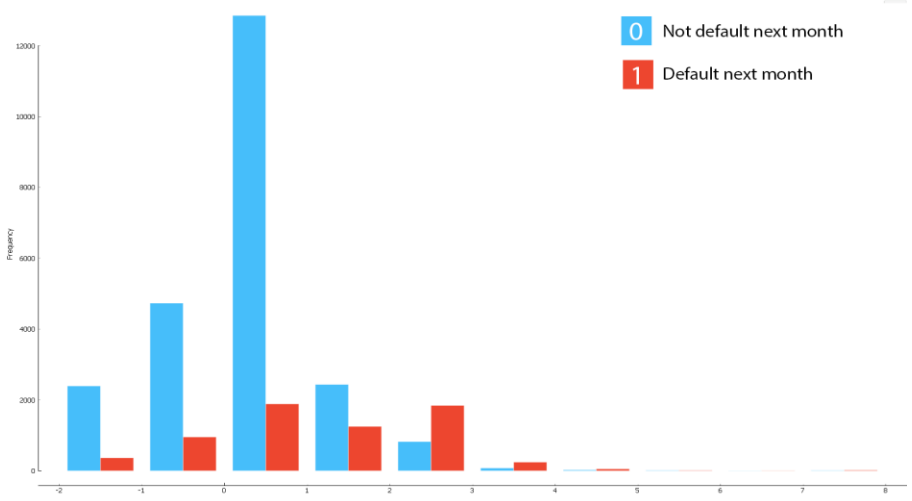
Biểu đồ 3: Biểu đồ so sánh tỉ lệ vỡ nợ dựa vào thuộc tính hôn nhân

Khách hàng kết hôn có tỉ lệ vỡ nợ cao hơn 23% so với 20%, tuy nhiên thuộc tính này có thể bị tác động do người có tuổi cao hơn thì thường lập gia đình và hạn mức tín dụng lớn nên có thể có xác suất vỡ nợ cao hơn.



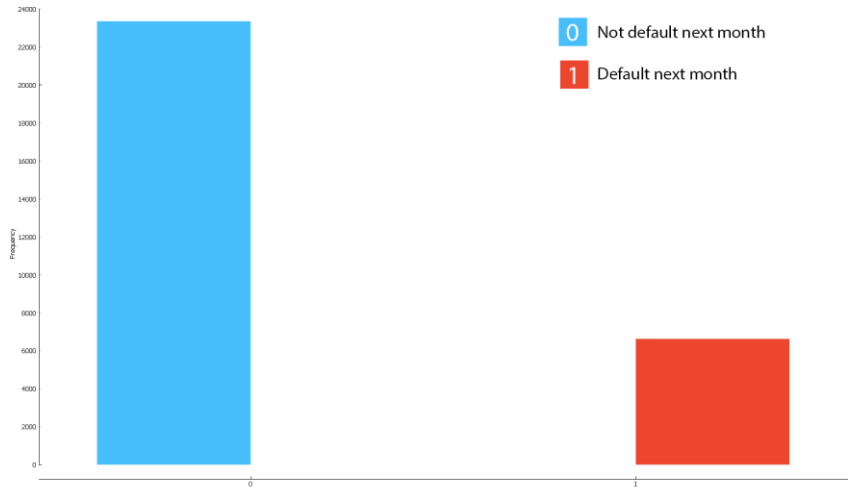
Biểu đồ 4: Biểu đồ thể hiện sự tương quan giữa tuổi tác với xác suất vỡ nợ

Khách hàng lớn tuổi thì có thể lệ vỡ nợ cao hơn khi họ cũng có hạn mức tín dụng cao nhất. Có thể điều này làm cho những khách hàng này có tỉ lệ vỡ nợ cao hơn so với các đối tượng khác.



Biểu đồ 5: Biểu đồ thể hiện tỉ lệ khách hàng thanh toán sớm hay muộn với tỉ lệ vỡ nợ

Biểu đồ trên của thuộc tính "PAY_0" đại diện cho thuộc tính PAY, nó thể hiện rằng khách hàng thanh toán tiền càng sớm thì tỉ lệ vỡ nợ càng thấp và ngược lại khách hàng thanh toán tiền càng chậm thì xác suất vỡ nợ càng cao. Điều này khá phù hợp về mặt logic với thực tế.

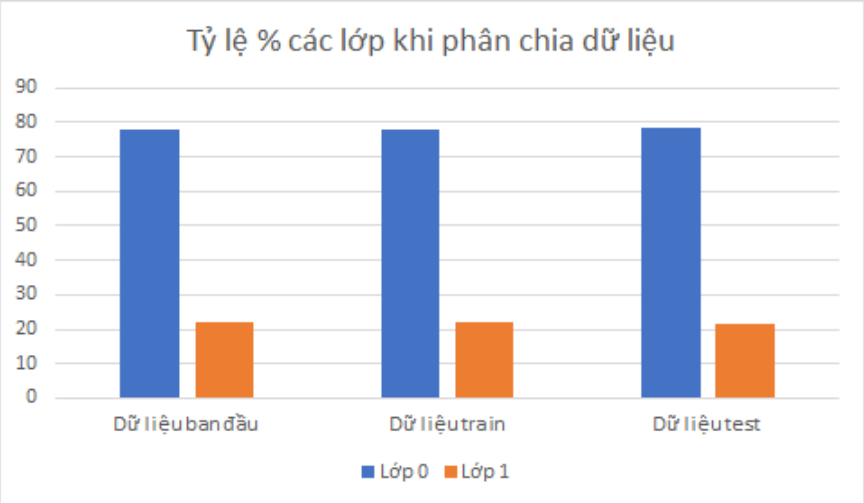


Biểu đồ 6: Biểu đồ so sánh tỉ lệ khách hàng vỡ nợ và không vỡ nợ tháng tiếp theo

Cuối cùng là biểu đồ thể hiện phần trăm khách hàng vỡ nợ là 22,12% so với tổng khách hàng.

3.2 Mô tả dữ liệu:

Dữ liệu được chia theo phương pháp Holdout với tỉ lệ 8:2, dữ liệu test chiếm 20%.



Biểu đồ 7: Tỷ lệ lớp 0 và 1 khi phân chia dữ liệu train và test (đơn vị %)

Tỷ lệ % từng lớp trong data:

- File dữ liệu gốc có 30000 dòng
- Có 77,88% lớp 0
- 22,12% lớp 1.

Tỷ lệ % từng lớp trong các file dữ liệu thực nghiệm đã chia.

- File train: Có 24000 dòng, trong đó có 77,75% lớp 0 và 22,25% lớp 1.
- File test: có 6000 dòng, gồm 78, 38% lớp 0 và 21,62% lớp 1.

STT	*Tên thuộc tính*	*Kiểu dữ liệu*	*Ý nghĩa*	*Trung bình hoặc giá trị phân biệt*
1	ID	Numeric	Số thứ tự	
2	LIMIT_BAL	Numeric	Số tiền tín dụng đã cho	167484.3227
3	SEX	Nominal	Giới tính	1 = nam; 2 = nữ
4	EDUCATION	Nominal	Giáo dục	1 = trường cao học; 2 = trường đại học; 3 = trường trung học; 4 = trường khác
5	MARRIAGE	Nominal	Tình trạng hôn nhân	1 = đã kết hôn; 2 = độc thân; 3 = người khác
6	AGE	Numeric	Tuổi	35
7	PAY_0	Nominal	Tình trạng trả nợ vào tháng 9 năm 2005	-1 = thanh toán hợp lệ; 1 = chậm thanh toán trong một tháng; 2 = chậm thanh toán trong hai tháng. . . 8 = chậm thanh toán trong tám tháng; 9 = chậm thanh toán từ chín tháng trở lên.
8	PAY_2	Nominal	Tình trạng hoàn trả vào tháng 8 năm 2005	
9	PAY_3	Nominal	Tình trạng hoàn trả vào tháng 7 năm 2005	
10	PAY_4	Nominal	Tình trạng hoàn trả vào tháng 6 năm 2005	
11	PAY_5	Nominal	Tình trạng hoàn trả vào tháng 5 năm 2005	
12	PAY_6	Nominal	Tình trạng hoàn trả vào tháng 4 năm 2005	
13	BILL_ATM1	Numeric	Số tiền hóa đơn sao kê trong tháng 9/2005	51223.3309
14	BILL_ATM2	Numeric	số tiền hóa đơn sao kê trong tháng 8/2005	49179.07517

15	BILL_ATM3	Numeric	Số tiền hóa đơn sao kê trong tháng 7/2005	47013.1548
16	BILL_ATM4	Numeric	Số tiền hóa đơn sao kê trong tháng 6/2005	43262.94897
17	BILL_ATM5	Numeric	Số tiền hóa đơn sao kê trong tháng 5/2005	40311.40097
18	BILL_ATM6	Numeric	Số tiền hóa đơn sao kê trong tháng 4/2005	38871.7604
19	PAY_ATM1	Numeric	Số tiền thanh toán trong tháng 9/2005	5663.5805
20	PAY_ATM2	Numeric	Số tiền thanh toán trong tháng 8/2005	5921.1635
21	PAY_ATM3	Numeric	Số tiền thanh toán trong tháng 7/2005	5225.6815
22	PAY_ATM4	Numeric	Số tiền thanh toán trong tháng 6/2005	4826.076867
23	PAY_ATM5	Numeric	Số tiền thanh toán trong tháng 5/2005	4799.387633
24	PAY_ATM6	Numeric	Số tiền thanh toán trong tháng 4/2005	5215.502567
25	default payment next month	Nominal	Vỡ nợ vào tháng tới	Có = 1, Không = 0

Bảng 4: Tỉ lệ từng lớp khi chia dữ liệu

3.3 Phương pháp đề xuất

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.

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 forests sử dụng tầm quan trọng của gini hoặc giảm tạp chất trung bình (MDI) để tính toán tầm quan trọng của từng tính năng. Gini tầm quan trọng còn được gọi là tổng giảm trong tạp chất nút. Đây là mức độ phù hợp hoặc độ chính xác của mô

hình giảm khi bạn thả biến. Độ lớn càng lớn thì biến số càng có ý nghĩa. Ở đây, giảm trung bình là một tham số quan trọng cho việc lựa chọn biến. Chỉ số Gini có thể mô tả sức mạnh giải thích tổng thể của các biến. Random Forests và cây quyết định Random Forests là một tập hợp của nhiều cây quyết định. Cây quyết định sâu có thể bị ảnh hưởng quá mức, nhưng Random forests ngăn cản việc lấp đầy bằng cách tạo cây trên các tập con ngẫu nhiên. Cây quyết định nhanh hơn tính toán. Random forests khó giải thích, trong khi cây quyết định có thể diễn giải dễ dàng và có thể chuyển đổi thành quy tắc.

Quá trình học của Random Forest bao gồm việc sử dụng ngẫu nhiên giá trịđầu vào, hoặc kết hợp các giá trị đó tại mỗi node trong quá trình dựng từng câyquyết định. Kết quả của Random Forest, qua thực nghiệm cho thấy, là tốt hơn khiso sánh với thuật toán Adaboost. Trong đó Random Forest có một số thuộc tính mạnh như:

(1) Độ chính xác của nó tương tự Adaboost, trong một số trường hợp còn tốt hơn.

(2) Thuật toán giải quyết tốt các bài toán có nhiều dữ liệu nhiễu.

(3) Thuật toán chạy nhanh hơn so với bagging hoặc boosting.

(4) Có những sự ước lượng nội tại như độ chính xác của mô hình phỏngđoán hoặc độ

mạnh và liên quan giữa các thuộc tính.

(5) Dễ dàng thực hiện song song.

(6) Tuy nhiên để đạt được các tính chất mạnh trên, thời gian thực thi củathuật toán khá

lâu và phải sử dụng nhiều tài nguyên của hệ thống.

Qua những tìm hiểu trên về giải thuật RF ta có nhận xét rằng RF là một phương pháp phân lớp tốt do: (1) Trong RF các sai số (variance) được giảm thiểu do kết quả của RF được tổng hợp thông qua nhiều người học (learner), (2) Việc chọn ngẫu nhiên tại mỗi bước trong RF sẽ làm giảm mối tương quan (correlation) giữa các người học trong việc tổng hợp các kết quả.

CHƯƠNG IV: THỰC NGHIỆM

4.1 Môi trường thực nghiệm

4.1.1 Thông số thiết bị

- Hệ thống: Window 10 home phiên bản 64-bit
- Chip xử lý: Intel ® core (™) i5-8250U CPU @ 1.60GHz (8 CPUs), ~ 1.8 GHz
- Ram: 12GB
- Ổ cứng: SSD 256GB và HDD 500GB

4.1.2 Chương trình và thư viện

Sử dụng chương trình python version 3.8

Các thư viện trong python:

- Matplotlib version 3.3.3
- Numpy version 1.19.3
- Scikit-learn version 0.23.2
- Seaborn version 0.11.0
- Pandas version 1.1.5

4.2 Phương pháp đánh giá

Sử dụng 5 độ đo như đã giới thiệu ở phần cơ sở lý thuyết, bao gồm:

- Accuracy
- F1-score
- Recall
- Support
- Error rate

4.3 Tập dữ liệu

– Tập dữ liệu được sử dụng ở giai đoạn thực nghiệm là dữ liệu sau quá trình tiền xử lý

– Tên tập dữ liệu : Default of credit card clients (Sự vỡ nợ của khách hàng sử dụng thẻ tín dụng).

– Link download:

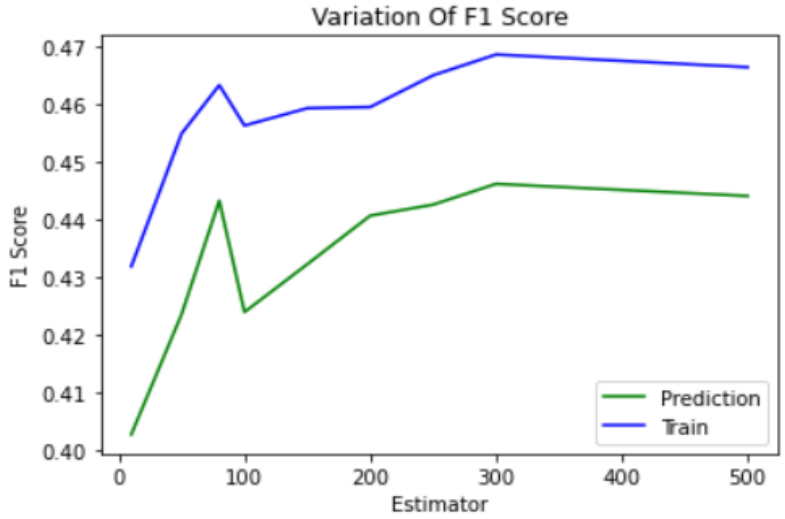
https://drive.google.com/file/d/1gi7v_AER4DWMT3B8BhdUfLQcSaCEzGuA/view?usp=sharing

4.4 Phương pháp thực nghiệm

4.4.1 Thuật toán Random Forest

Random Forest được coi là một tập hợp của một số cây quyết định. Những cây này kết hợp với nhau để đưa ra kết quả đầu ra. Giá trị trung bình của tất cả các cây được chọn làm đầu ra.

Độ sâu tối ưu cho cây được tìm thấy là 5 trong Mô hình Cây Quyết định, do đó giá trị độ sâu được sử dụng là 5.



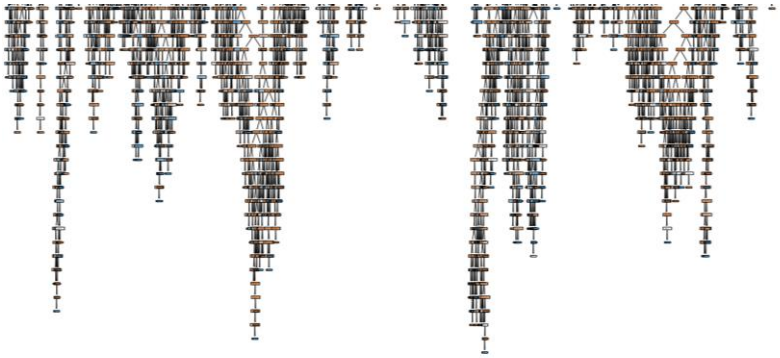
Biểu đồ 8: Biểu đồ thể hiện đo F1 Score với các giá trị độ sâu khác nhau

Confusion Matrix	
[[4441 249]	
[845 465]]	
True Positive (TP)	4,441
False Positive (FP)	249
True Negative (TN)	465
False Negative (FN)	845
Tỷ lệ dự đoán chính xác của thuật toán Random Forest là 82% Trong đó có 95% thực sự thanh toán tín dụng cho tháng tới.	

	Precision	**Recall**	**F1_Score**	**Support**
0	0.84	0.95	0.89	4690
1	0.65	0.35	0.46	1310
Accuracy			0.82	6000
Macro avg	0.75	0.65	0.67	6000
Weighted avg	0.80	0.82	0.8	6000

Hình 10: Kết quả chạy thực nghiệm bằng thuật toán Random Forest

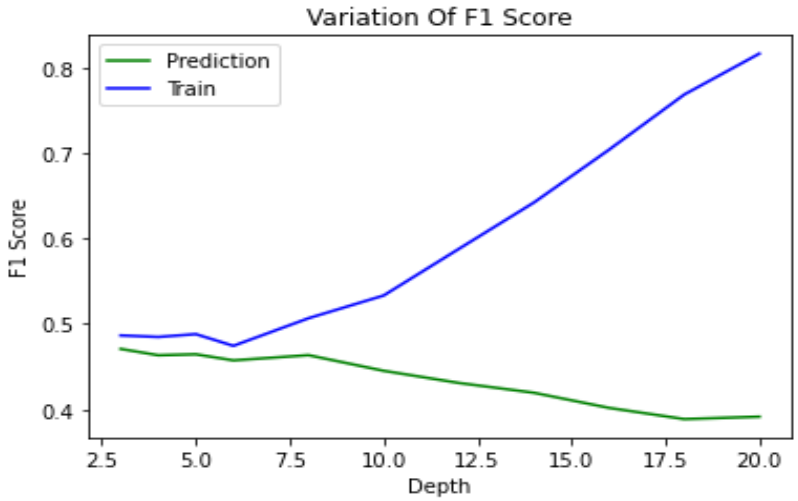
Độ chính xác được dự đoán là 82% trong đó 95% khách hàng thật sự thanh toán tín dụng cho tháng sau.



Hình 11: Hình ảnh mô tả thuật toán Random Forest

4.4.2 Thuật toán Decision tree

Giống với thuật toán Random Forest, trước hết chúng ta cần đi tìm độ sâu mà cây phải tạo ra các nút lá / nút dự đoán phù hợp nhất cho kết quả tốt nhất. Độ sâu thấp sẽ gây ra tình trạng underfitting và ngược lại cao thì gây ra overfitting.



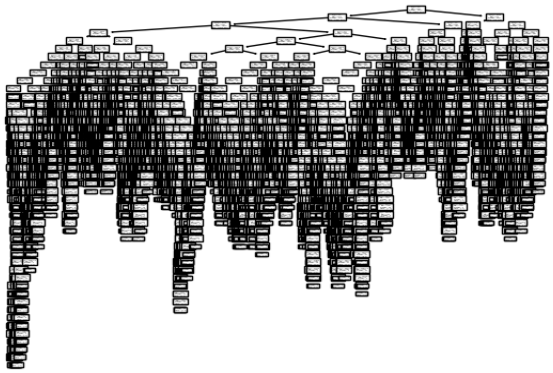
Biểu đồ 9: Biểu đồ thể hiện độ đo F1 score với từng giá trị độ sâu

Biểu đồ trên cho thấy rõ ràng một trường hợp overfitting khi độ sâu tăng lên. Độ sâu ở thuật toán decision tree cũng được chọn với giá trị bằng 5 để tránh overfitting.

Confusion Matrix	
[[3827 863]	
[772 538]]	
True Positive (TP)	3827
False Positive (FP)	863
True Negative (TN)	538
False Negative (FN)	772
Tỷ lệ dự đoán chính xác của thuật toán Decision tree là 73% - Trong đó có 82% thực sự thanh toán tín dụng cho tháng tới.	

	Precision	Recall	F1_Score	Support
0	0.83	0.82	0.82	4690
1	0.38	0.41	0.40	1310
Accuracy			0.73	6000
Macro avg	0.61	0.61	0.61	6000
Weighted avg	0.73	0.73	0.73	6000

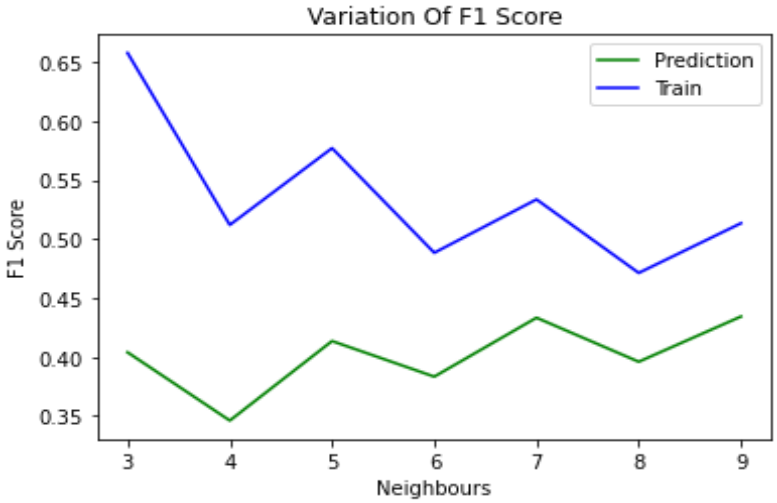
Hình 12: Kết quả thuật toán Decision Tree



Hình 13: Hình ảnh mô tả thuật toán Decision Tree

4.4.3 Thuật toán K-neighbors

Một trong những tham số của thuật toán K-Nearest Neighbor là giá trị 'n_neighbors'. Nó quyết định số lượng hàng xóm phải xem xét để phân loại một điểm. Cần tìm ra n để tối ưu hóa thuật toán này.



Biểu đồ 10: Thể hiện giá trị N_neighbor so với F1 score

Từ biểu đồ, khi chúng ta tăng số lượng hàng xóm, F1 score của dữ liệu train giảm và của dữ liệu test tăng lên.

Một xu hướng khác được quan sát là, F1 score cho các hàng xóm chẵn nhỏ hơn các số lẻ trước và sau chính nó. Điều này là do khi chúng ta đặt các hàng xóm là số chẵn, khả năng hòa sẽ xảy ra. Ví dụ: khi hàng xóm = 4, có khả năng 2 người trong số họ thuộc '0' và 2 người khác thuộc '1'. Trong tình huống này, thuật toán chọn nhãn xuất hiện đầu tiên trong dữ liệu huấn luyện.

Do đó, chọn n_neighbor = 5, cho các dự đoán tiếp theo. Mặc dù các giá trị cao hơn có điểm F1 lớn hơn, nhưng việc chọn các giá trị này sẽ khiến dữ liệu huấn luyện bị sai lệch.

Confusion Matrix	
[[4295 395]	
[1081 229]]	
True Positive (TP)	4,295
False Positive (FP)	395
True Negative (TN)	229
False Negative (FN)	1,081
Tỷ lệ dự đoán chính xác của thuật toán Kneibors là 75% Trong đó có 92% thực sự thanh toán tín dụng cho tháng tới.	

	Precision	**Recall**	**F1_Score**	**Support**
0	0.80	0.92	0.85	4690
1	0.37	0.17	0.24	1310
Accuracy			0.75	6000
Macro avg	0.58	0.55	0.55	6000
Weighted avg	0.70	0.75	0.72	6000

Hình 14: Kết quả dự đoán với thuật toán K-neighbors

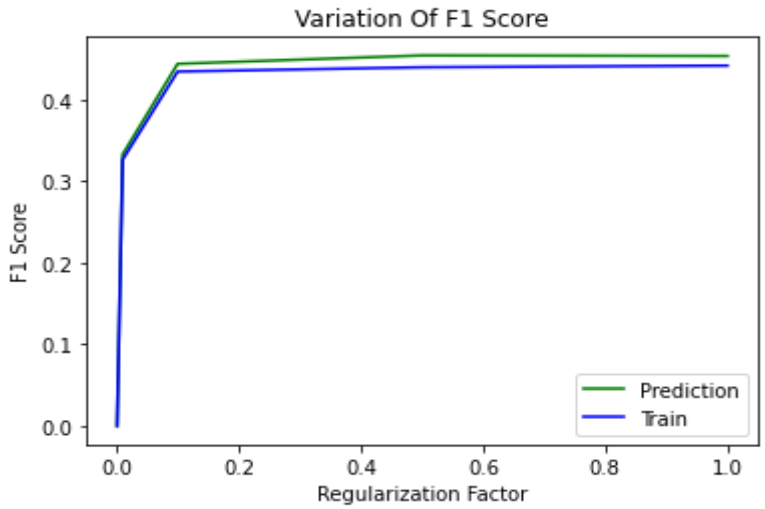
Tỷ lệ dự đoán chính xác là 75% trong đó có 92% khách hàng thực sự thanh toán cho tháng tới trong tổng số khách hàng được dự đoán sẽ thanh toán.

Độ chính xác và thu hồi đối với nhãn '0' là rất cao nhưng đối với nhãn '1' thì hoàn toàn ngược lại. Điều này có thể do tập dữ liệu không cân bằng.

4.4.4 Thuật toán Logistic Regression

Trong Logistic Regression, thay vì điều chỉnh một đường hồi quy, chúng ta dùng một hàm logistic hình chữ "S", hàm này dự đoán hai giá trị lớn nhất (0 hoặc 1). Đường cong từ hàm logistic cho biết khả năng xảy ra một điều gì đó, trong trường hợp này là liệu khách hàng sẽ mặc định (1) hay không (0).

Đối với thuật toán này, hệ số chính quy sẽ được thay đổi. Phương pháp điều chỉnh còn được gọi là phương pháp 'thu nhỏ', giảm hoặc thu nhỏ các hệ số.



Biểu đồ 11: Thể hiện tỉ lệ giữa F1 score và hệ số

Có thể thấy từ biểu đồ rằng F1 score đột ngột tăng trong phạm vi 0 đến 0,1 và không đổi đối với các giá trị còn lại. Để tránh overfitting, giá trị hệ số được chọn là 0,1

Confusion Matrix	
[[4689 1]	
[1310 0]]	
True Positive (TP)	4,689
False Positive (FP)	1
True Negative (TN)	0
False Negative (FN)	1,310
Tỷ lệ dự đoán chính xác của thuật toán Logistic Regression là 78% Trong đó có 100% thực sự thanh toán tín dụng cho tháng tới.	

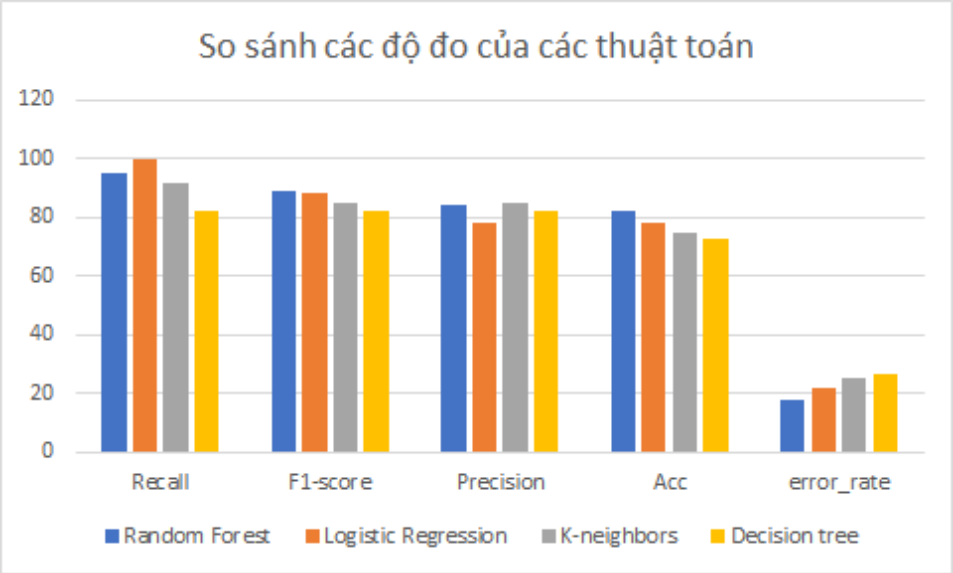
	Precision	Recall	F1_Score	Support
0	0.78	1.00	0.88	4690
1	0.00	0.00	0.00	1310
Accuracy			0.78	6000
Macro avg	0.39	0.50	0.44	6000
Weighted avg	0.61	0.78	0.69	6000

Hình 15: Kết quả dự đoán khi sử dụng thuật toán Logistic Regression

Tỷ lệ dự đoán chính xác là 78% trong đó 100% khách hàng được dự đoán sẽ thanh toán vào tháng tiếp theo.

Tuy khả năng dự đoán khách hàng thanh toán tiếp tục cao tuy nhiên việc dự đoán sai khách hàng sẽ thanh toán thành vỡ nợ và đó không phải là một dấu hiệu tốt. Vì dự đoán một khoản vỡ nợ thực tế không phải là một khoản vỡ nợ, có thể gây tổn thất chi phí cơ hội cho ngân hàng.

4.5 Kết luận



Biểu đồ 12: Biểu đồ thể hiện so sánh các độ đo của các thuật toán

Từ biểu đồ, thuật toán Random forest có độ chính xác cao nhất là 82% với và với 95% khách hàng sẽ tiếp tục thanh toán vào tháng sau. Thuật toán Random Forest đã chiếm ưu thế hơn hẳn so với các thuật toán còn lại.

Thuật toán Logistic Regression cũng có độ chính xác không kém cạnh bao nhiêu, với 100% khách hàng được dự đoán thực sự thanh toán vào tháng sau. Tuy nhiên tỷ lệ khách hàng thanh toán thực sự bị dự đoán vỡ nợ khá nhiều. Điều này làm đánh mất chi phí cơ hội khá lớn đối với ngân hàng nếu mô hình này được áp dụng vào thực tế.

CHƯƠNG V: DEMO

5.1 Tổng quan

−Đây là mô hình phân loại cho một tập dữ liệu phổ biến,có chức năng dự đoán về tình trạng vỡ nợ thẻ tín dụng của tháng tiếp theo. Dự đoán dựa trên dữ liệu nhân khẩu học(giới tính, trình độ học vấn, tình trạng hôn nhân, tuổi tác số tiền tín dụng đã cho), tình trạng trả nợ, lịch sử thanh toán, bảng sao kê hóa đơn của các khách hàng sử dụng thẻ tín dụng ở Đài Loan từ tháng 4 năm 2005 đến tháng 9 năm 2005.

− Từ đó giúp người sử dụng khoanh vùng được khách hàng tiềm năng và đem lại lợi ích nhất định cho mình.

5.2 Công cụ và thư viện được sử dụng:

5.2.1 Công cụ khai thác dữ liệu:

−Ngôn ngữ lập trình: Python 3.8, HTML

−Công cụ khai thác dữ liệu: Jupyter Notebook

−Heroku: là nền tảng đám mây hỗ trợ hosting web miễn phí và có hỗ trợ backend là ngôn ngữ python phù hợp với đề tài.

−Github: là nền tảng để lưu trữ source code dùng để kết nối với Heroku nhằm hosting web một cách dễ dàng và mượt mà.

5.2.2 Thư viện kèm theo:

−Numpy (Numeric Python): là một thư viện toán học cho phép làm việc hiệu quả với ma trận và mảng, đặc biệt là dữ liệu ma trận nhiều chiều,dãy số và mảng lớn.

−Pandas: là thư viện mã nguồn mở với hiệu năng cao cho việc phân tích dữ liệu trong Python. Thư viện này có một số tính năng nổi bật sau:

- Import dữ liệu từ nguồn CSV.
- Xử lý, phân tích dữ liệu như thống kê, mô hình hóa.
- Xử lý phân tích dữ liệu dưới dạng bảng.
- Tinh chỉnh và làm sạch dữ liệu.

−Scikit-learn: là thư viện mạnh mẽ nhất dành cho các thuật toán máy học được viết trên ngôn ngữ Python. Thư viện này cung cấp một tập các công cụ xử lý các bài toán machine learning và statistical (ví dụ: classification). Nó tập trung vào

việc mô hình hóa dữ liệu. Scikit-Learn có một số thuật toán để thực hiện các nhiệm vụ khai thác dữ liệu và học máy, đáng chú ý là phân loại, phân cụm, lựa chọn mô hình, reduce và hồi quy.

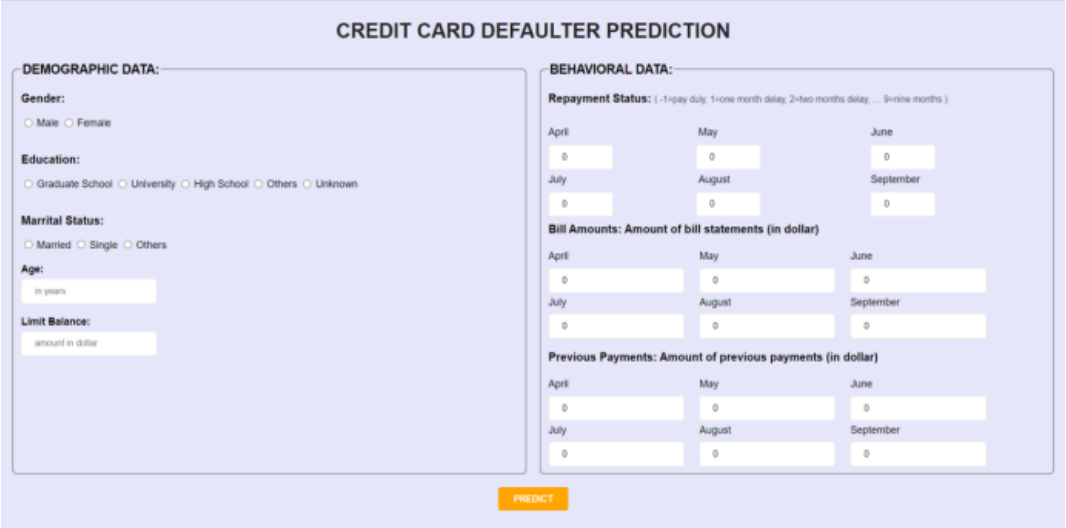
- Statsmodels: là một mô đun Python cung cấp các lớp và chức năng để ước tính nhiều mô hình thống kê khác nhau cũng như thực hiện các bài kiểm tra thống kê và thăm dò dữ liệu.
- Flask: Flask là một web frameworks, nó thuộc loại micro-framework được xây dựng bằng ngôn ngữ lập trình Python. Flask cung cấp công cụ, các thư viện và các công nghệ hỗ trợ để xây dựng các ứng dụng web.
- Gunicorn: là một trong những Python web server theo chuẩn WSGI (Web Server Gateway Interface). Nó hỗ trợ cho Web Framework Python Flask.
- Matplotlib: là một thư viện phổ biến trong Python dùng để trực quan hóa dữ liệu.
- Seaborn: là một thư viện trực quan hóa dữ liệu cho Python, nó được xây dựng trên thư viện Matplotlib. Seaborn thực hiện tất cả các ánh xạ ngữ nghĩa và tổng hợp thống kê quan trọng để tạo ra các lô thông tin.

5.3 Xây dựng demo

- Đào tạo mô hình phân loại sử dụng RandomForestClassifier để dự đoán về tình trạng vỡ nợ của khách hàng sử dụng thẻ tín dụng tháng tiếp theo.
- Làm sạch dữ liệu.
- Áp dụng mô hình phân loại máy học.
- Xây dựng web trên Heroku.
- Tải demo lên Github.
- Nhận thông tin của khách hàng từ ứng dụng Web.
- Hiển thị dự đoán.

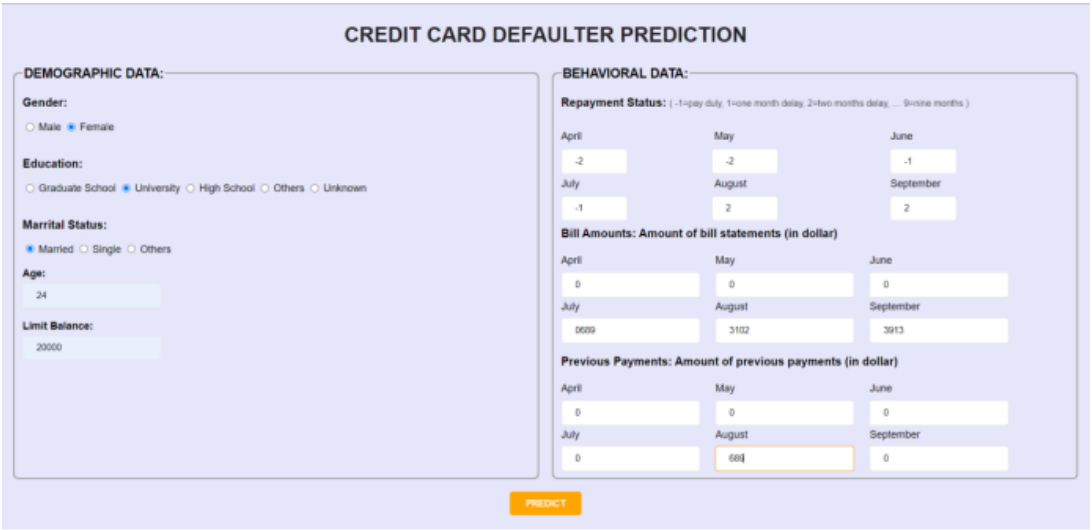
5.4 Demo Web

- Link Demo Web: https://data-mining-g3.herokuapp.com/
- Giao diện Web Demo “Credit Card Defaulter Prediction”



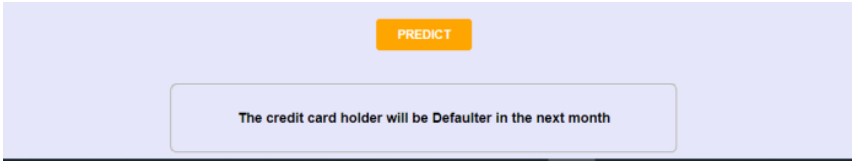
Hình 16: Giao diện Web demo

- Điền đầy đủ thông tin theo yêu cầu. Sau đó click vào button "PREDICT".



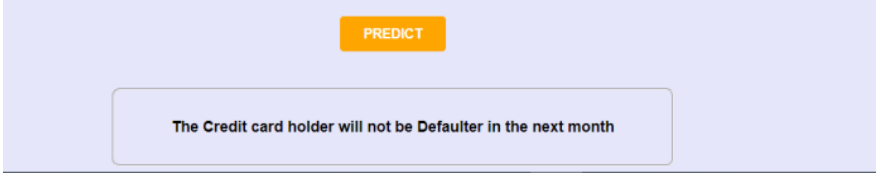
Hình 17: Giao diện web khi nhập liệu

- Dự đoán "Tháng tới khách hàng này có thể rơi vào tình trạng vỡ nợ."



Hình 18: Giao diện dự đoán "khách hàng vỡ nợ"

- Dự đoán "Tháng tới khách hàng này có thể không rơi vào tình trạng vỡ nợ."



Hình 19: Giao diện dự đoán "khách hàng không vỡ nợ"

CHƯƠNG VI: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết quả

Với kết quả thu được là chương trình dự đoán vỡ nợ của khách hàng khi sử dụng thẻ tín dụng dựa trên nhiều yếu tố từ cơ bản như thông tin cá nhân cho tới các yếu tố liên quan tới tín dụng như thanh toán định kỳ hàng tháng và chi thu hàng tháng..., từ đó ngân hàng có thể theo dõi được tình trạng vay nợ của khách hàng một cách an toàn, đảm bảo được quyền lợi cho cả khách hàng và ngân hàng.

6.2 Ưu điểm - nhược điểm

6.2.1 Ưu điểm

- Có thể dựa trên các yếu tố khách quan và chủ quan để dự đoán khả năng vỡ nợ của khách hàng.
- Cho kết quả tương đối chính xác.
- Nguồn thông tin của khách hàng mà ngân hàng cung cấp là bảo mật tuyệt đối.
- Cung cấp cho ngân hàng nguồn tham khảo từ đó đem lại lợi ích cho ngân hàng.
- Tránh khả năng bị mất vốn quá nhiều.

6.2.2 Nhược điểm

- Các yếu tố để dự đoán còn bị hạn chế.
- Còn phải dựa trên nhiều biến, tỉ lệ chính xác chưa đạt được trên 90%

6.3 Hướng phát triển

- Xây dựng chương trinh dự đoan với một file gồm nhiều dữ liệu
- Có thể phát triển nhiều hướng cho chương trình này, có thể phát triển cả về mặt ứng dụng và mặt thuật toán.
- Xây dựng hệ thống lưu trữ thông tin khách hàng để ứng dụng một cách nhanh chóng, đưa ra kết quả kịp thời.
- Lưu trữ toàn hệ thống về thực trạng vay/cho vay/từ chối cho vay của toàn bộ khách hàng.
- Định kỳ kiểm tra lại tính chính xác của mô hình bằng cách kiểm tra chỉ số GINI, thời gian 6 tháng/lần hoặc có thể thay đổi theo thực trạng về tỷ lệ nợ xấu cũng như tình hình kinh tế từng thời điểm.

TÀI LIỆU THAM KHẢO

[1] Yeh, I-Cheng. (2016). Default of credit card clients Data Set. Truy xuất từ http://archive.ics.uci.edu/ml/machine-learning-databases/00350, 14/10/2020.

[2] ThS. Nguyễn Duy Nhất, ThS. Hồ Trung Thành, và ThS.Lê Thị Kim Hiền. (2015). Khai Phá Dữ liệu Trong Kinh Doanh. Nhà xuất bản Đại Học Quốc Gia TP. Hồ Chí Minh.

[3] Nguyễn Văn Hoàng (2019a). Giới thiệu về Numpy (một thư viện chủ yếu phục vụ cho khoa học máy tính của Python). Truy xuất từ: https://viblo.asia/p/gioi-thieu-ve-numpy-mot-thu-vien-chu-yeu-phuc-vu-cho-khoa-hoc-may-tinh-cua-python-maGK7kz9Kj2, 10/12/2020.

[4] Nguyễn Văn Hoàng (2019b). Giới thiệu về Pandas (một thư viện phổ biến của Python cho việc phân tích dữ liệu). Truy xuất từ https://viblo.asia/p/gioi-thieu-ve-pandas-mot-thu-vien-pho-bien-cua-python-cho-viec-phan-tich-du-lieu-aWj53Nnel6m, 10/12/2020.

[5] Nguyễn Văn Hoàng (2019c). Một số hàm hữu ích để xử lý nhanh dữ liệu trong một danh sách(list) trong Python. Truy xuất từ https://viblo.asia/p/mot-so-ham-huu-ich-de-xu-ly-nhanh-du-lieu-trong-mot-danh-sachlist-trong-python-L4x5xdJ15BM, 11/12/2020.

[6] Harry (2017). 7 library/tool nên biết khi bắt đầu Machine Learning/Deep Learning trên Python. Truy xuất từ https://studylinux.wordpress.com/2017/11/03/7-librarytool-nen-biet-khi-bat-dau-machine-learningdeep-learning-tren-python/, 12/12/2020.

[7] Thái Doãn Hùng (2020). Thư Viện Scikit-learn Trong Python Là Gì?. Truy xuất từ https://codelearn.io/sharing/scikit-learn-trong-python-la-gi, 12/12/2020.

[8] FireBirD (2019). Vẽ đồ thị trong Python với thư viện Matplotlib. Truy xuất từ https://allaravel.com/blog/ve-do-thi-trong-python-voi-thu-vien-matplotlib, 13/12/2020.

[9] Johnson, Justin (2019). Python Numpy Tutorial (with Jupyter and Colab). Truy xuất từ https://cs231n.github.io/python-numpy-tutorial/#numpy, 13/12/2020.