# CREDIT RISK MODEL

## LOGISTIC REGRESSION

Người thực hiện: Trần Thị Thanh Hải

Oct 14, 2023

# CONTENTS

# 1 Introduction

# Abstract


- Việc đánh giá rủi ro tín dụng và dự đoán khả năng vỡ nợ dựa trên người dùng vay trực tuyến **đặc biệt quan trọng.**
- Trong tình huống kinh doanh vay trực tuyến, số tiền vay thường **thấp** và **khối lượng vay lớn**, và việc phê duyệt thủ công theo cách truyền thống không còn đáp ứng được nhu cầu của tình huống kinh doanh vay trực tuyến.

→ mô hình dự đoán rủi ro vỡ nợ cho khoản vay, sử dụng dữ liệu người dùng thực tế từ LendingClub.

# P2P ?


- Thuật ngữ "**P2P**" (hoặc "P2P lending") mô tả các hoạt động cho vay và vay mượn xảy ra trực tiếp giữa cá nhân [Wang, 2009].
- Các sàn giao dịch P2P lending là các nền tảng giúp tương tác giữa những người cho vay và người vay, nơi người vay đăng ký vay tiền trực tuyến và các nhà đầu tư cá nhân đấu giá để cung cấp khoản vay dưới dạng một quy trình, tương tự đấu giá [Klawït, 2008].


- **LendingClub** là một trong những nền tảng **P2P lending** lớn và nổi tiếng tại Hoa Kỳ.
- Nó là một trang web và dịch vụ trực tuyến cho vay trực tiếp giữa các cá nhân.
- **LendingClub** cung cấp một cơ hội cho người vay để xin vay tiền và cho phép các nhà đầu tư cá nhân đầu tư vào các khoản vay này thông qua một **quy trình trực tuyến**.

# 2 Dataset

# 2 Dataset

## Bộ dữ liệu từ LendingClub: tải tại đây gồm các thông tin sau:

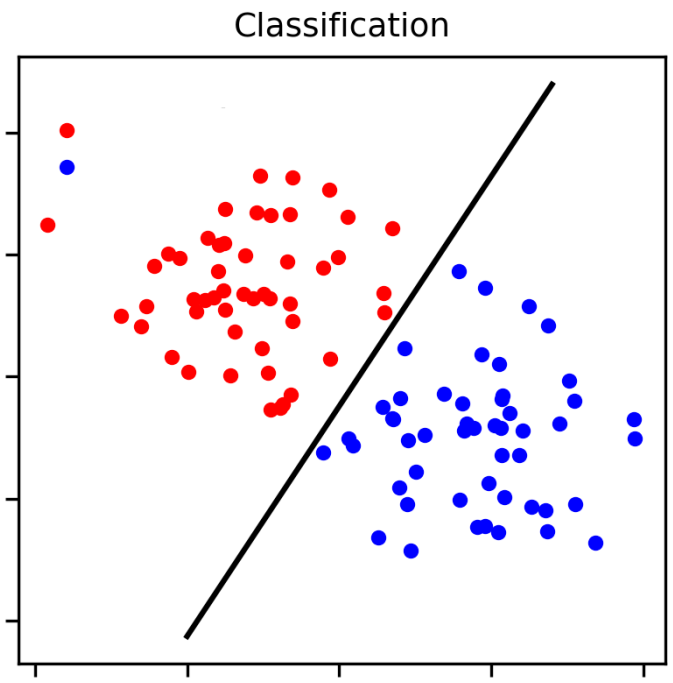
| | Các biến | Mô tả |
|---|---|---|
| 0 | loan_amnt | Số tiền được người vay đề nghị vay trong đơn xin vay. Nếu tại một thời điểm nào đó, bộ phận tín dụng giảm số tiền vay, thì giá trị này sẽ phản ánh số tiền vay mới. |
| 1 | term | Số lần thanh toán trên khoản vay, được tính bằng tháng và có thể là 36 hoặc 60 tháng (kỳ hạn). |
| 2 | int_rate | Lãi suất của khoản vay |
| 3 | installment | Số tiền hàng tháng mà người vay phải trả nếu khoản vay được giải ngân. |
| 4 | grade | Thứ hạng của khoản vay, được gán bởi LendingClub (LC), ví dụ: A, B, C.... |
| 5 | sub_grade | Thứ hạng con (subgrade) của khoản vay, mô tả chi tiết hơn về thứ hạng (grade) |
| 6 | emp_title | Chức vụ công việc mà người vay cung cấp khi nộp đơn vay.* |
| 7 | emp_length | Thời gian làm việc của người vay (đơn vị năm). |
| 8 | home_ownership | Tình trạng sở hữu nhà cửa, thông tin được cung cấp bởi người vay trong quá trình đăng ký hoặc được thu thập từ báo cáo tín dụng. Có các giá trị như RENT (thuê nhà), OWN (sở hữu nhà), MORTGAGE (đang trả nợ thế chấp), hoặc OTHER (khác). |
| 9 | annual_inc | Thu nhập hàng năm của người vay (người vay cung cấp trong quá trình đăng ký). |
| 10 | verification_status | Cho biết liệu thu nhập của người vay có được xác minh bởi LendingClub. |
| 11 | issue_d | Tháng mà khoản vay được giải ngân |
| 12 | loan_status | Trạng thái hiện tại của khoản vay |
| 13 | purpose | Mục đích sử dụng khoản vay, được cung cấp bởi người vay trong đơn xin vay. |
| 14 | title | Tiêu đề của khoản vay được cung cấp bởi người vay. |
| 15 | zip_code | Ba số đầu của mã bưu chính được cung cấp bởi người vay trong đơn xin vay. |

# 2 Datadet

| | | |
|---|---|---|
| **16** | addr_state | Tiểu bang được cung cấp bởi người vay trong đơn xin vay. |
| **17** | dti | Tỷ lệ nợ trên thu nhập hàng tháng của người vay, tính bằng cách chia tổng số tiền nợ hàng tháng (không bao gồm nợ thế chấp và khoản vay LC yêu cầu) cho thu nhập hàng tháng tự báo cáo của người vay. |
| **18** | earliest_cr_line | Tháng mà dòng tín dụng đầu tiên của người vay được mở. |
| **19** | open_acc | Số lượng tài khoản tín dụng mà người vay đang có và trong trạng thái "mở". |
| **20** | pub_rec | Số lượng bản ghi công khai về người vay có lịch sử tín dụng không tốt (có thể là do không tuân thủ hoặc vi phạm các cam kết tài chính, tín dụng). |
| **21** | revol_bal | Tổng số dư tín dụng quay vòng, cho biết mức độ nợ của người vay đối với các khoản tín dụng có hạn mức quay vòng, chẳng hạn như thẻ tín dụng. Số dư này có thể thay đổi theo thời gian tùy thuộc vào việc sử dụng và trả nợ. |
| **22** | revol_util | Tỷ lệ sử dụng hạn mức tín dụng quay vòng, hoặc số tiền mà người vay đang sử dụng so với tổng hạn mức tín dụng quay vòng có sẵn. Ví dụ, nếu bạn có hạn mức tín dụng quay vòng là 10.000 đô la và bạn đang nợ 2.000 đô la, thì tỷ lệ sử dụng hạn mức tín dụng quay vòng của bạn là 20% (2.000 / 10.000). |
| **23** | total_acc | Tổng số lượng tài khoản tín dụng trong hồ sơ tín dụng của người vay |
| **24** | initial_list_status | Tình trạng ban đầu của một khoản vay, có thể: "W" và "F" |
| **25** | application_type | Cho biết liệu khoản vay là cá nhân hay hai người đồng thời vay. |
| **26** | mort_acc | Số lượng tài khoản thế chấp. |
| **27** | pub_rec_bankruptcies | Số lượng thông tin công khai về phá sản trong hồ sơ của người vay. |

# 3 Method

## Classification



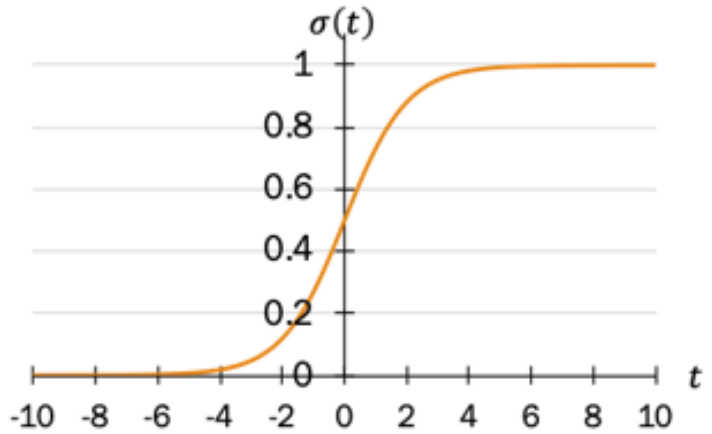

Bài toán phân loại trong Machine Learning (ML) là một loại bài toán mà mục tiêu là **phân tách** hoặc **gán nhãn** các **điểm dữ liệu** vào một trong nhiều nhóm hoặc lớp khác nhau dựa trên các **đặc điểm và thông tin của chúng.**

**Cụ thể, trong bài toán này**

- **"Charged Off" được gán nhãn 0**
- **"Fully Paid" được gán nhãn 1.**

## Logistic Regerssion | Hàm sigmoid

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$




Reflection/ Symmetry $1 - \sigma(t) = \frac{e^{-t}}{1 + e^{-t}} = \sigma(-t)$

Inverse $t = \sigma^{-1}(p) = \log\left(\frac{p}{1-p}\right)$

Domain $-\infty < t < \infty$

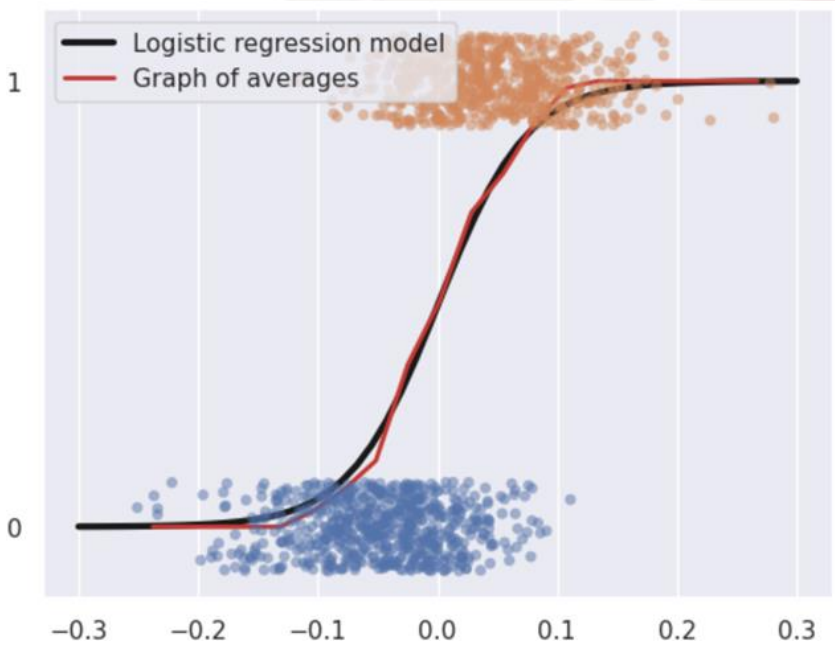
Range $0 < \sigma(t) < 1$

Derivative

$$\frac{d}{dt}\sigma(t) = \sigma(t)(1 - \sigma(t)) = \sigma(t)\sigma(-t)$$

## Logistic Regerssion

Chúng ta định nghĩa P là xác suất để 1 điểm dữ liệu thuộc về class 1, tức là xác suất để khách hàng đó có khả năng trả được nợ.




$$P(Y=1\,|\,x) = \frac{1}{1+e^{-x^\top\theta}}$$

$$= \frac{1}{1+e^{-(\theta_0+\theta_1 x_1+\ldots+\theta_p x_p)}}$$

**Để dự đoán xác suất:**

- Tính $x^\top\theta$, với x là các features bộ dữ liệu
- Áp dụng hàm **sigmoid:** $\sigma(x^\top\theta)$

# 3 Method

## Logistic Regerssion | Thực nghiệm

## Part 1: Data Exploration

```
df = pd.read_csv('./dataset/lending_club_loan_two.csv')
df.head()
```

[2] ✓ 3.2s

| | loan_amnt | term | int_rate | installment | grade | sub_grade | emp_title | emp_length | home_ownership | annual_inc | verification_status | issue_d | loan_status |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 10000.0 | 36 months | 11.44 | 329.48 | B | B4 | Marketing | 10+ years | RENT | 117000.0 | Not Verified | Jan-2015 | Fully Paid |
| 1 | 8000.0 | 36 months | 11.99 | 265.68 | B | B5 | Credit analyst | 4 years | MORTGAGE | 65000.0 | Not Verified | Jan-2015 | Fully Paid |
| 2 | 15600.0 | 36 months | 10.49 | 506.97 | B | B3 | Statistician | < 1 year | RENT | 43057.0 | Source Verified | Jan-2015 | Fully Paid |
| 3 | 7200.0 | 36 months | 6.49 | 220.65 | A | A2 | Client Advocate | 6 years | RENT | 54000.0 | Not Verified | Nov-2014 | Fully Paid |
| 4 | 24375.0 | 60 months | 17.27 | 609.33 | C | C5 | Destiny Management Inc. | 9 years | MORTGAGE | 55000.0 | Verified | Apr-2013 | Charged Off |

## Logistic Regerssion | Thực nghiệm

### Part 2: Data Cleaning

Check null values

- Xóa các dòng chứa giá trị null, duplicates
- Xử lý các cột categorical

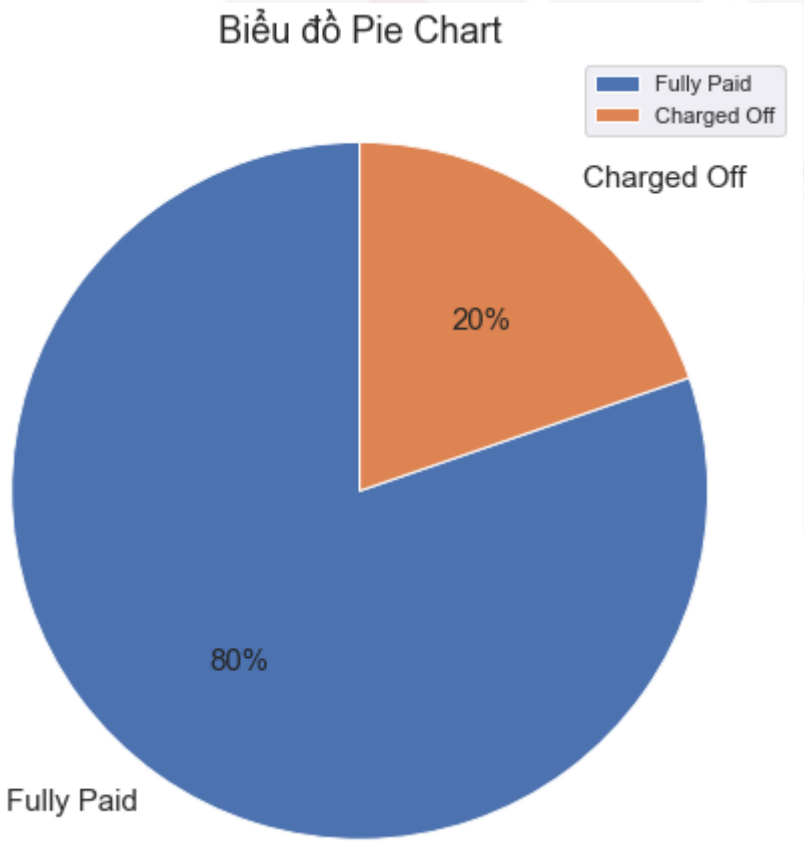
```
# Tính tỷ lệ % null mỗi cột
null_columns = df.columns[df.isnull().any()].tolist()
null_percentage = (df[null_columns].isnull().sum() / len(df)) * 100
null_percentage
```

✓ 1.8s

```
emp_title              5.789208
emp_length             4.621115
title                  0.443148
revol_util             0.069692
mort_acc               9.543469
pub_rec_bankruptcies   0.135091
dtype: float64
```

## Logistic Regerssion | Thực nghiệm

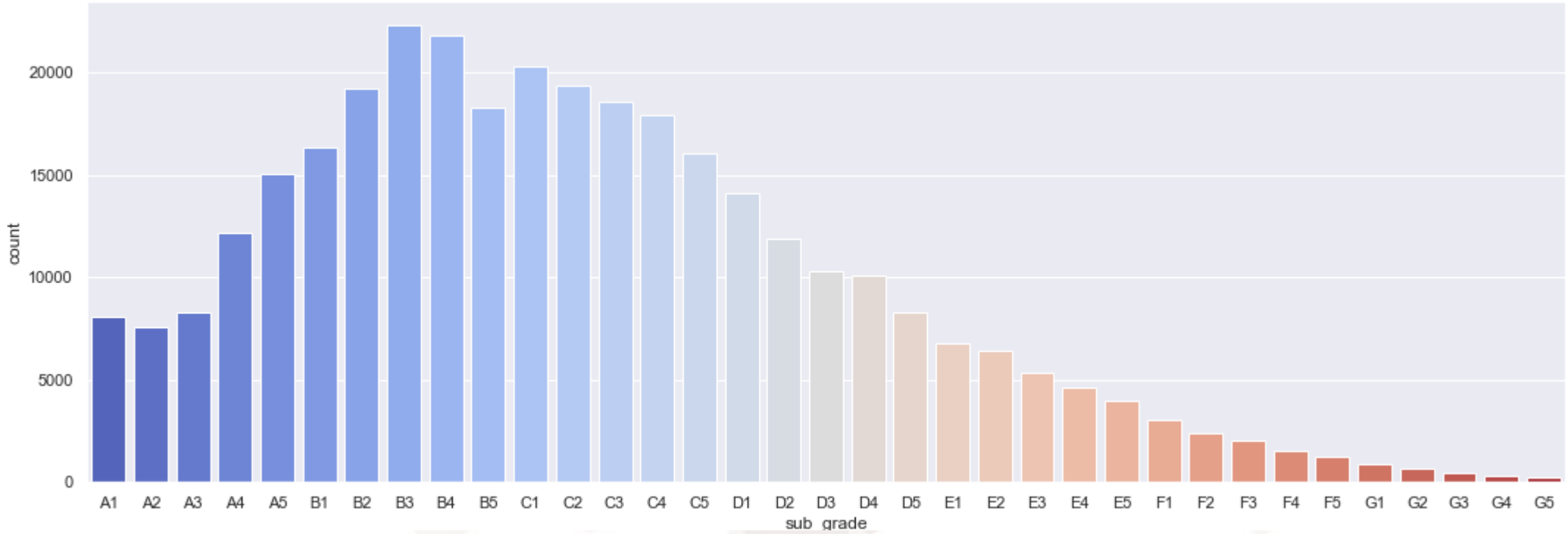
### Part 3: Data Visualization




Tỷ lệ các nhãn :

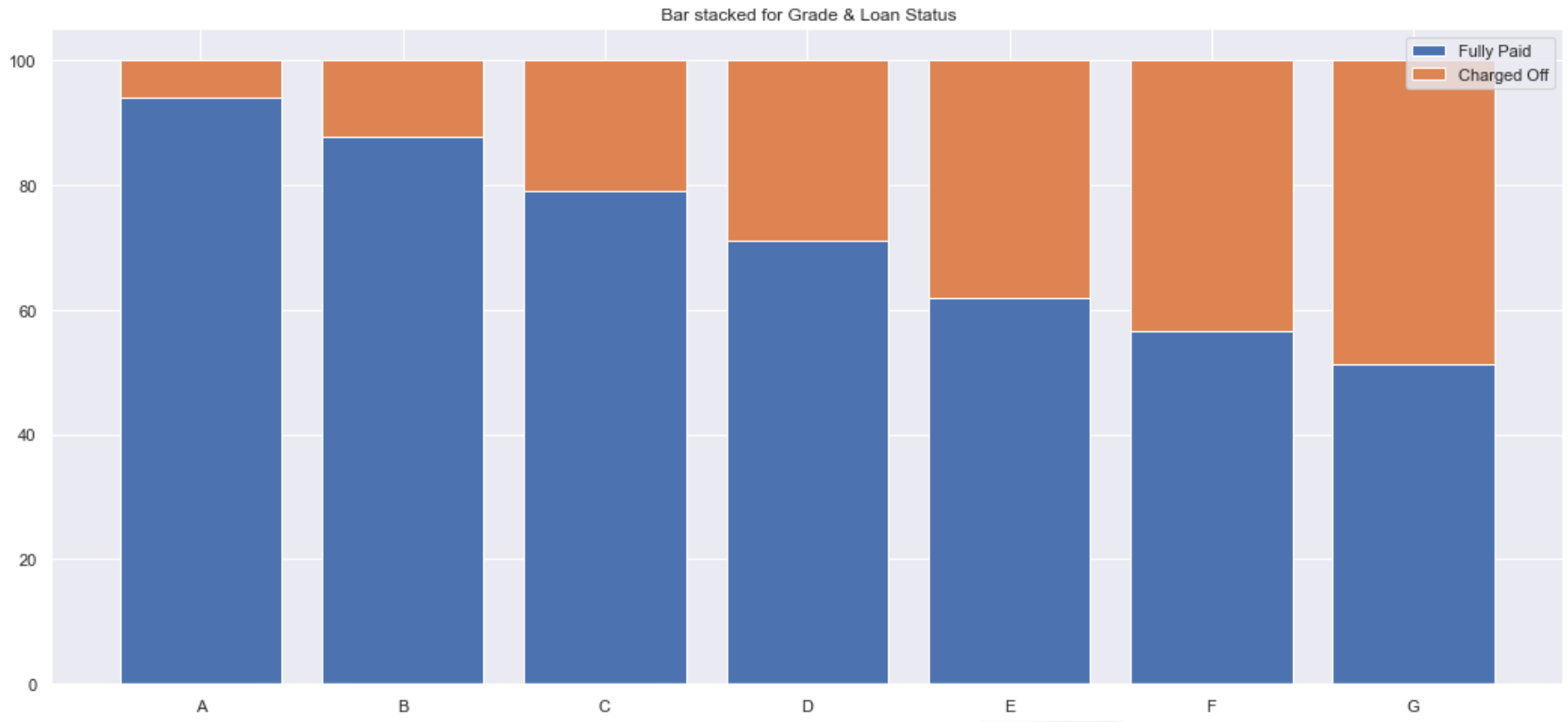
- 20% nhãn 0
- 80% nhãn 1

## Logistic Regerssion | Thực nghiệm




Số lượng các khách hàng theo các thứ hạng con (subgrade)

## Logistic Regerssion | Thực nghiệm

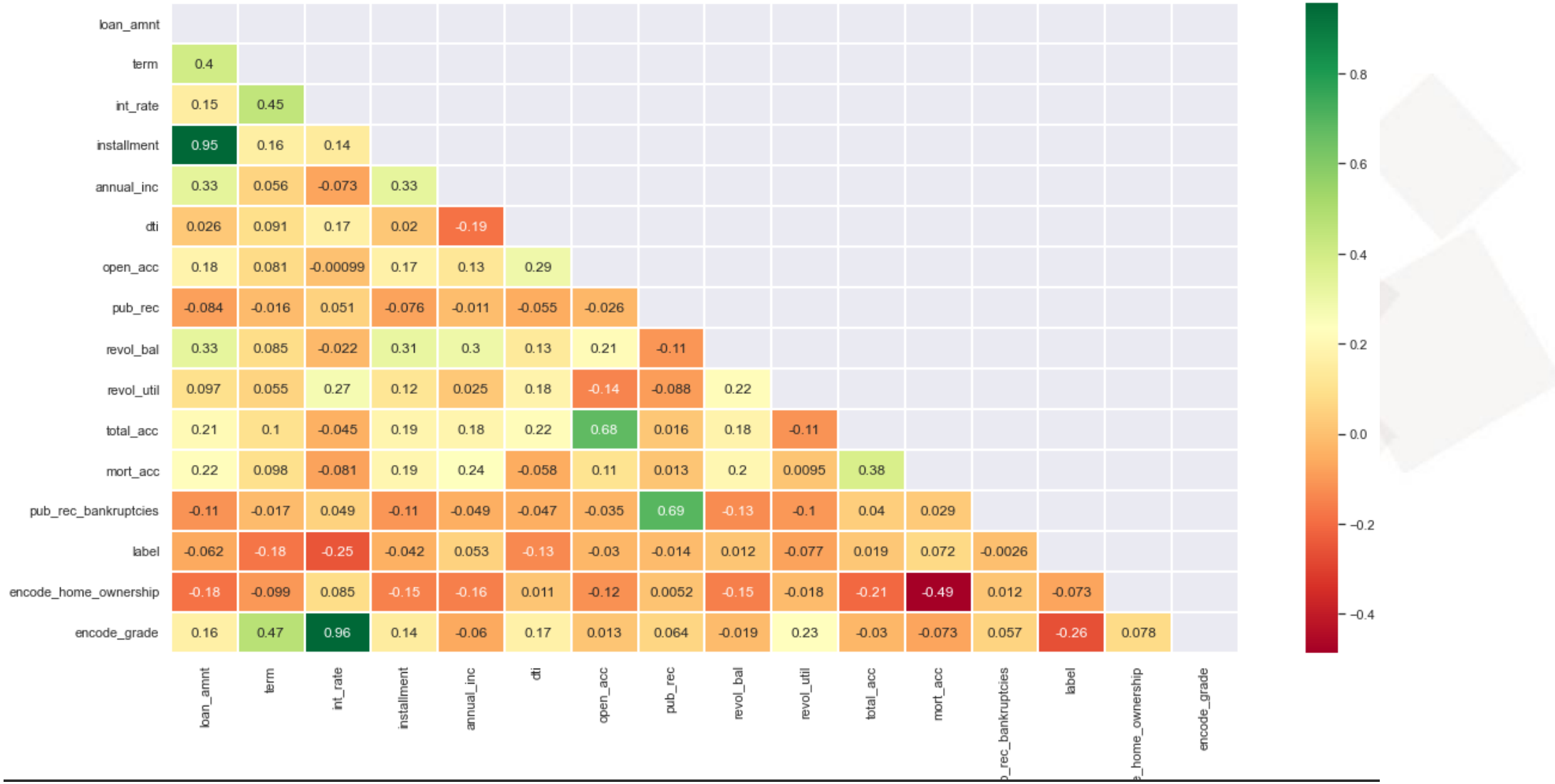



| grade | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| loan_status | | | | | | | |
| Charged Off | 5.966004 | 12.202065 | 21.08488 | 29.004392 | 38.078384 | 43.469709 | 48.746626 |
| Fully Paid | 94.033996 | 87.797935 | 78.91512 | 70.995608 | 61.921616 | 56.530291 | 51.253374 |

Dễ thấy, với các nhóm khác hàng xếp hạng càng thấp, tỷ lệ vỡ nợ càng cao.
Đặc biệt, nhóm **F** và **G** chiếm gần 50%.

## Logistic Regerssion | Thực nghiệm

Mối quan hệ tương quan giữa các biến

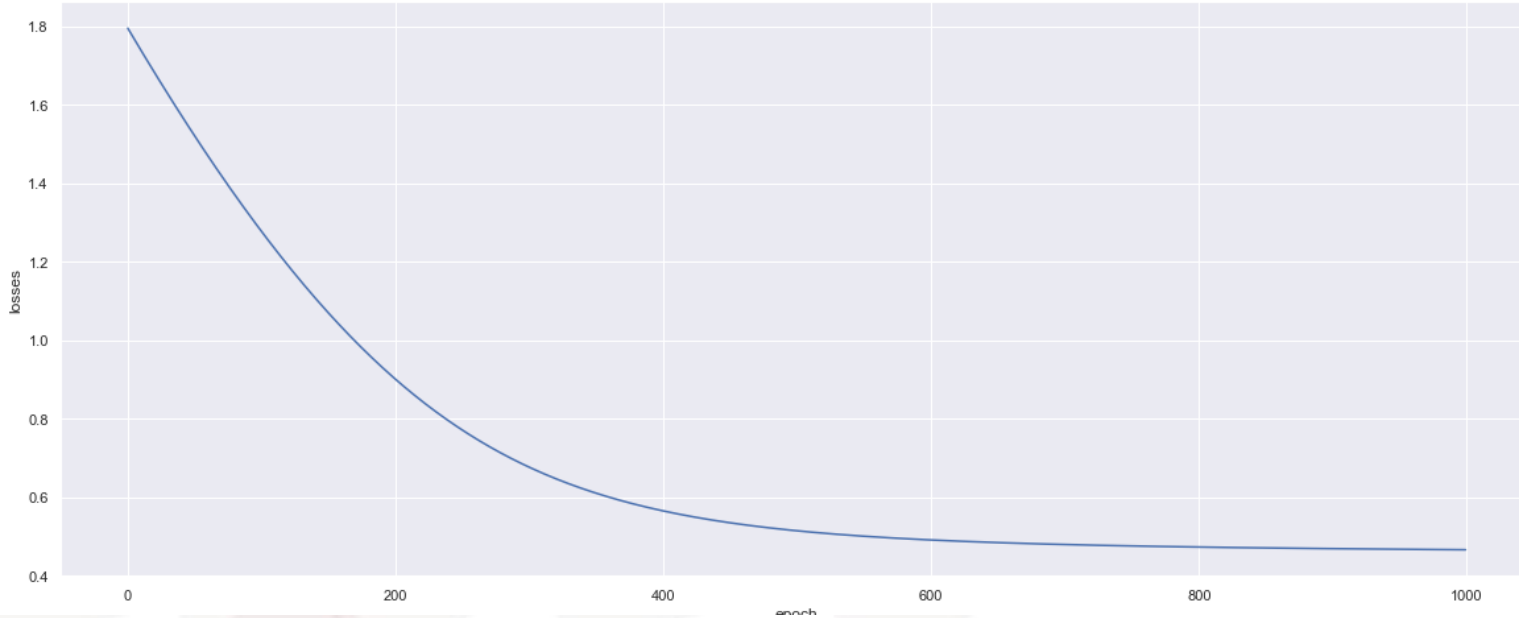


## Logistic Regerssion | Thực nghiệm

# Part 4: Model Building

```
# parameters

n_epochs = 1000
lr = 0.01

losses = []
for epoch in range(n_epochs):
  # get all the samples
  x = X_train
  y = y_train

  # predict y_hat
  y_pred = predict(x, theta)

  # compute loss
  loss = compute_cost(y_pred, y)
  losses.append(loss)

  # compute gradient
  gradient = compute_gradient(x, y, y_pred)

  # update weights
  theta = update_weight(theta, lr, gradient)

theta, losses
```

## 4.1 Logistic Regression from scratch



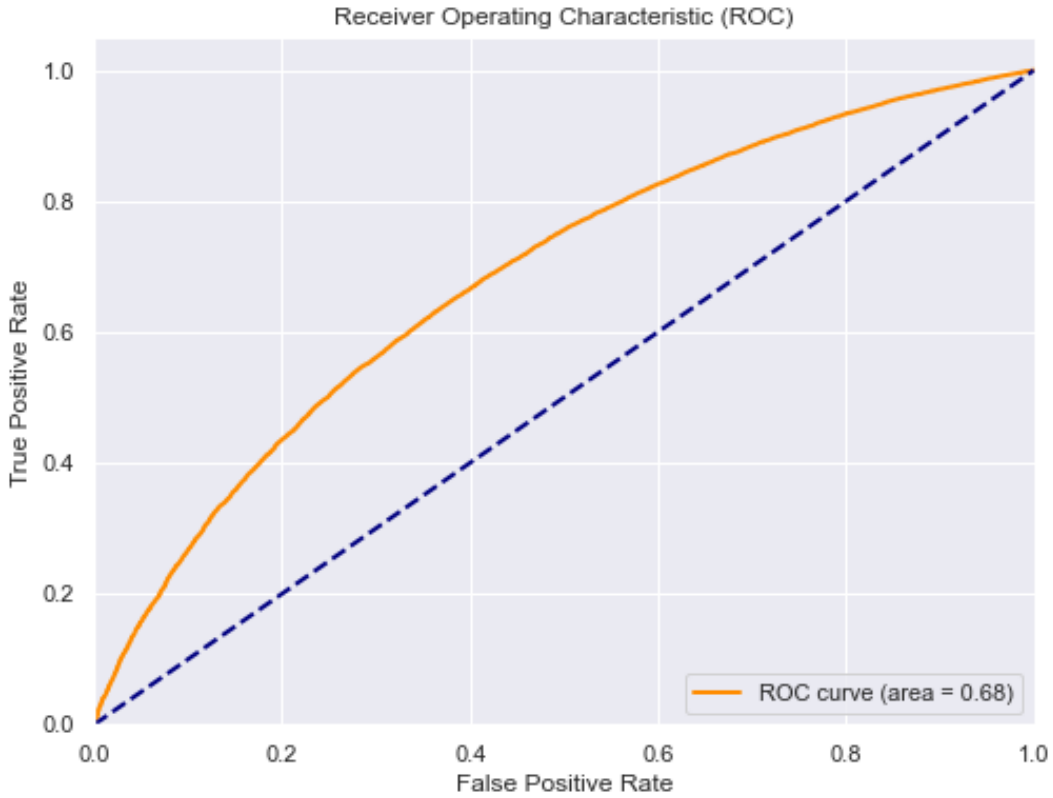

Hàm loss giảm và bắt đầu hội tụ từ epoch 800

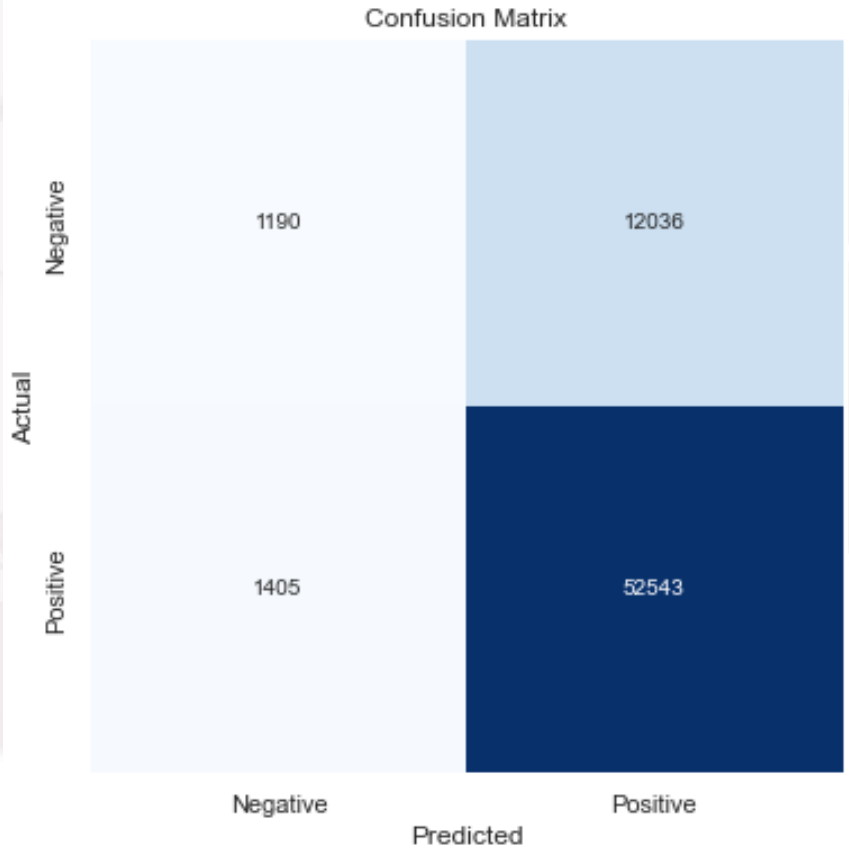
| accuracy | precision | recall | f1-score |
|---|---|---|---|
| 0.799908 | 0.813624 | 0.973956 | 0.8866 |

## Logistic Regerssion | Thực nghiệm

# Part 4: Model Building

## Logistic Regression from scratch



## Logistic Regerssion | Thực nghiệm

### Logistic Regression from Scikit-learn librabry

```
from sklearn.linear_model import LogisticRegression
log_reg_model = LogisticRegression()
log_reg_model.fit(X_train, y_train)
```

✓ 0.9s

```
▾ LogisticRegression
LogisticRegression()
```

```
from sklearn.tree import DecisionTreeClassifier
decision_tree_model = DecisionTreeClassifier()
decision_tree_model.fit(X_train, y_train)
```

✓ 7.5s

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
from sklearn.ensemble import RandomForestClassifier
random_forest_model = RandomForestClassifier()
random_forest_model.fit(X_train, y_train)
```

✓ 4m 12.8s

```
▾ RandomForestClassifier
RandomForestClassifier()
```

| Model | Precision | Recall | F1-score | Accuracy |
|---|---|---|---|---|
| Logistic Regression | 0.814561 | 0.980302 | 0.889779 | 0.804954 |
| Decision Tree | 0.824426 | 0.802583 | 0.813358 | 0.704191 |
| Random Forest | 0.815229 | 0.976421 | 0.888574 | 0.803337 |

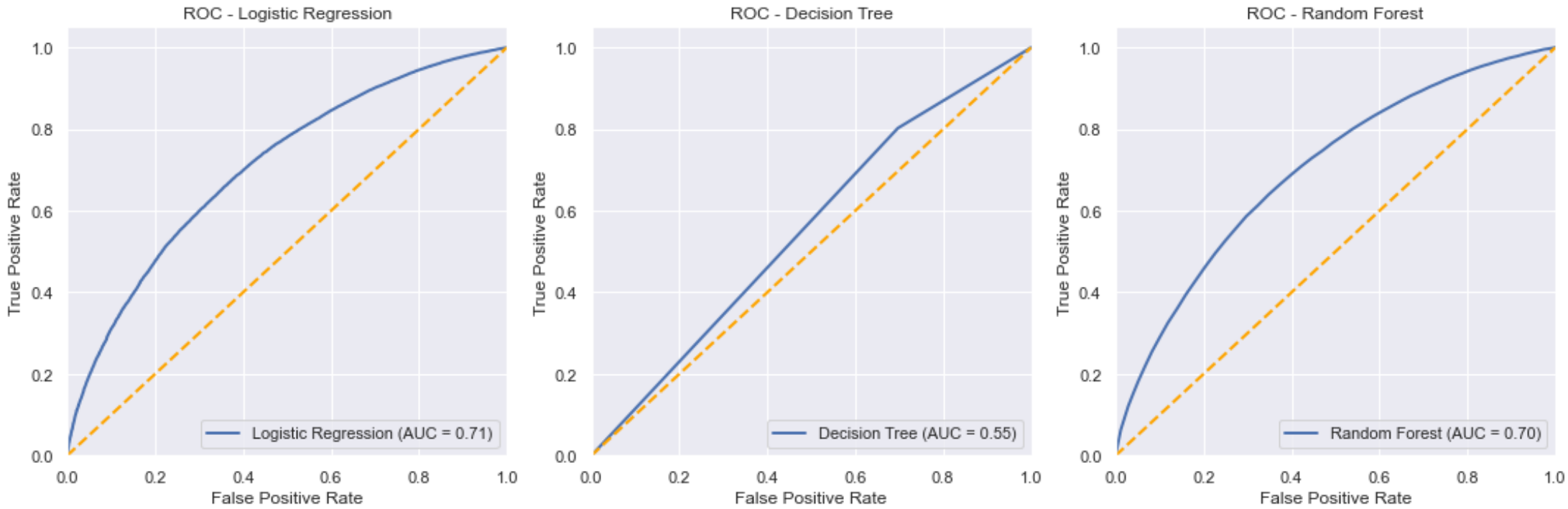
Với 3 model từ thư viện scikit-learn:

- LogisticRegression
- DecisionTreeClassifier
- RandomForestClassifier

→ model **Logistic Regression** cho hiệu quả có phần nhỉnh hơn các model khác

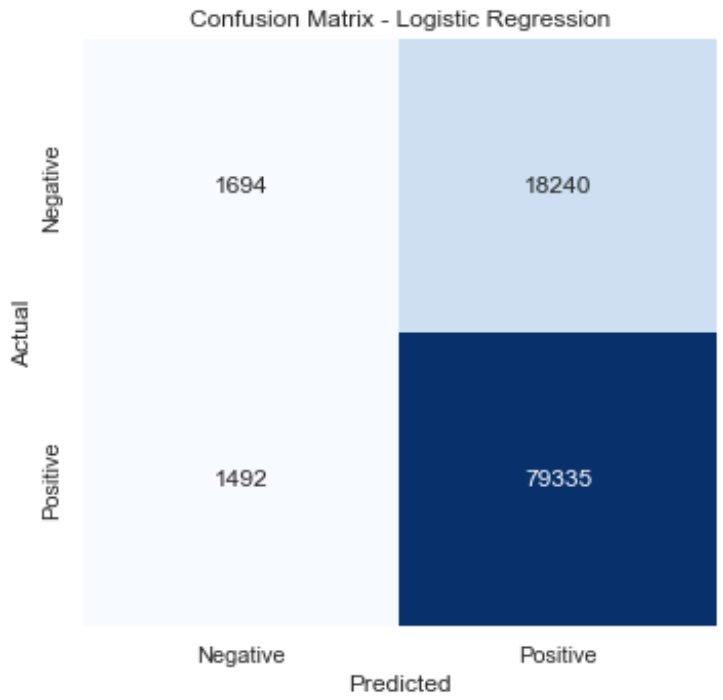
## Logistic Regerssion | Thực nghiệm

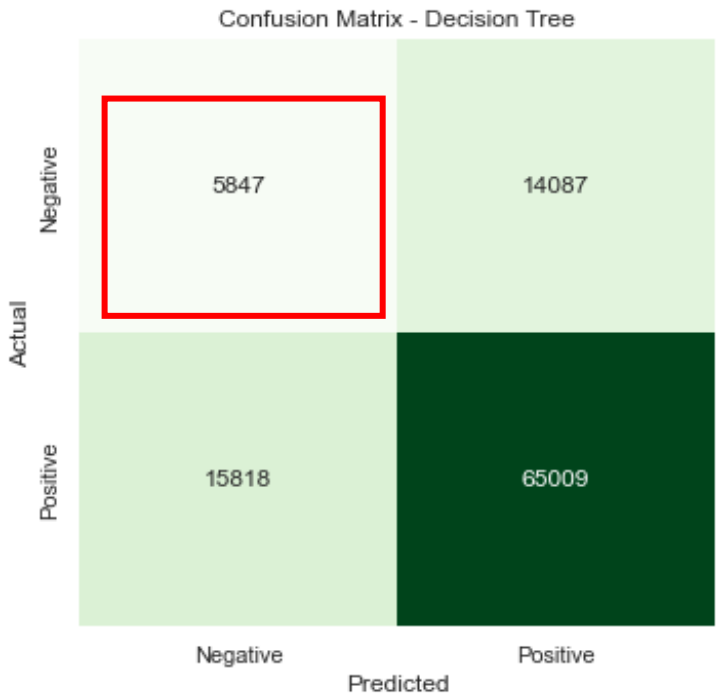
### Logistic Regression from Scikit-learn librabry

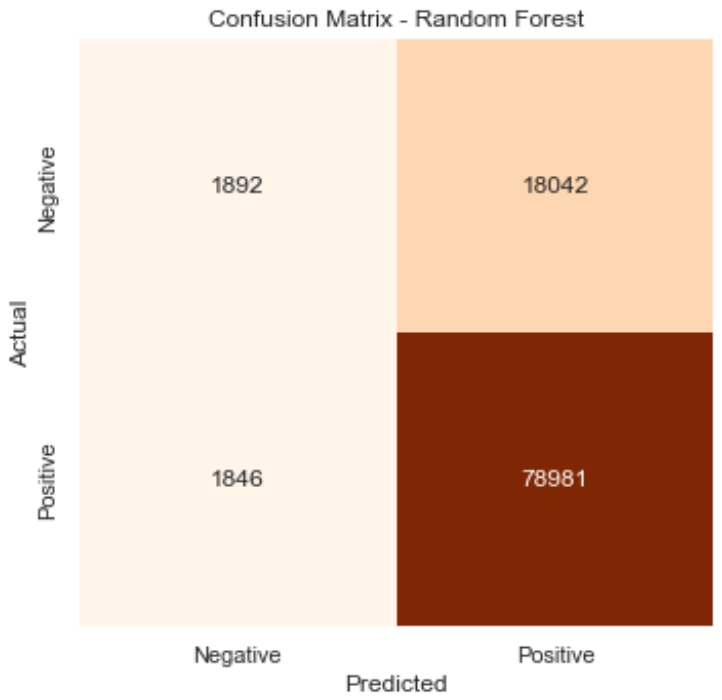


# 4 Result

## Logistic Regerssion | Thực nghiệm  Logistic Regression from Scikit-learn librabry




| Model | Precision | Recall | F1-score | Specificity | Accuracy |
|---|---|---|---|---|---|
| Logistic Regression | 0.814561 | 0.980302 | 0.889779 | 0.089814 | 0.804954 |
| Decision Tree | 0.824426 | 0.802583 | 0.813358 | 0.302908 | 0.704191 |
| Random Forest | 0.815229 | 0.976421 | 0.888574 | 0.097425 | 0.803337 |

Mặc dù **Accuraccy** của model **DecisionTreeClassifier thấp** nhất trong số các model. Tuy nhiên, **tỷ lệ dữ đoán đúng mẫu âm tính** (Charged Off) trên tổng số mẫu âm tính lại cao nhất. Với **5847** mẫu âm tính được phân loại đúng, cao hơn rất nhiều so với 1684, và 1892 của 2 model còn lại.

→ Tùy yêu cầu của bài toán, ta có thể linh động chọn các model để áp dụng

# THANK YOU