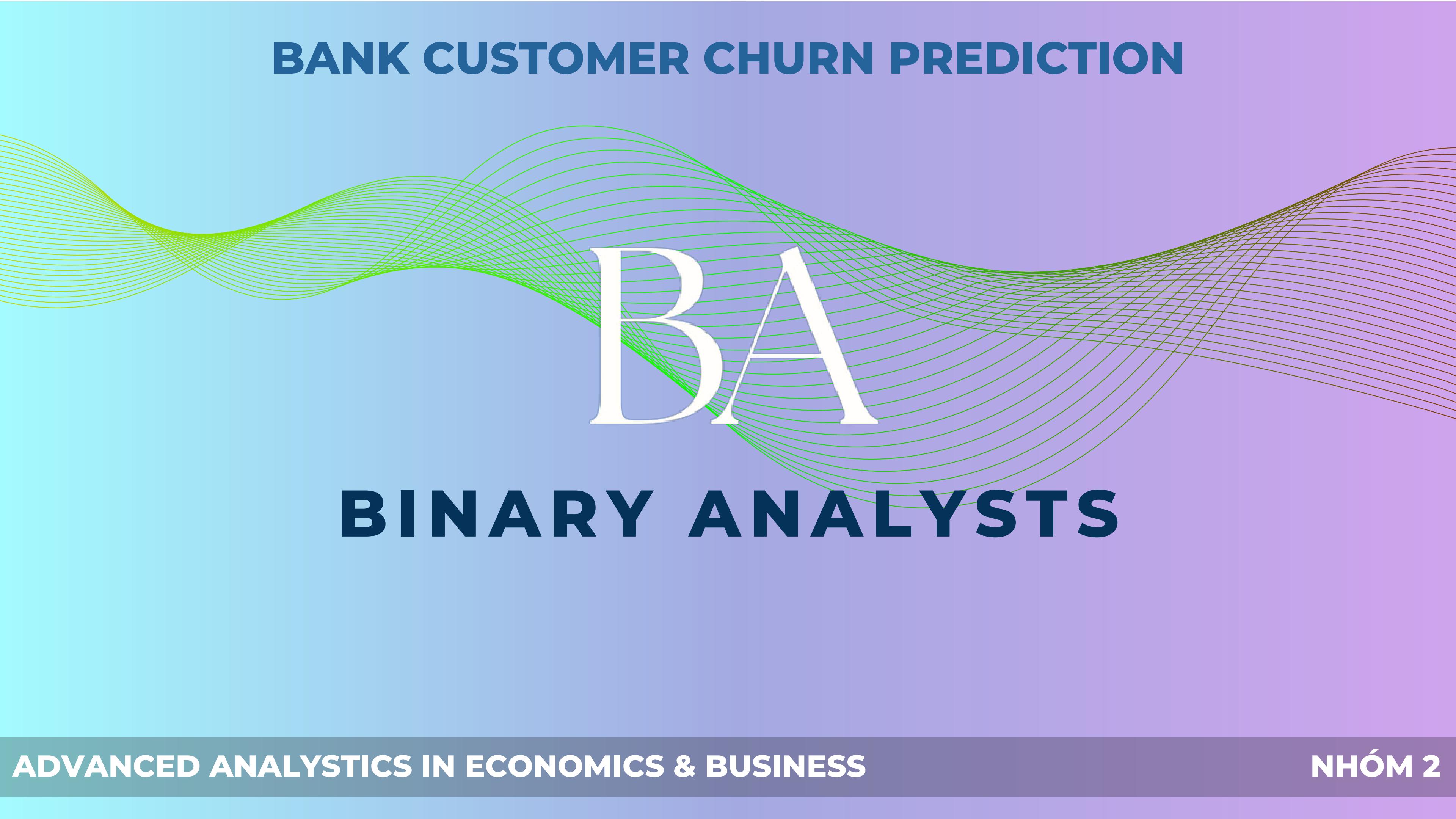


# BANK CUSTOMER CHURN PREDICTION



BA

## BINARY ANALYSTS

# Churn Customer



khách hàng đã sử dụng hoặc tương tác với một ngân hàng trong một khoảng thời gian nhất định, nhưng sau đó quyết định chấm dứt quan hệ hoặc dừng sử dụng dịch vụ của ngân hàng.

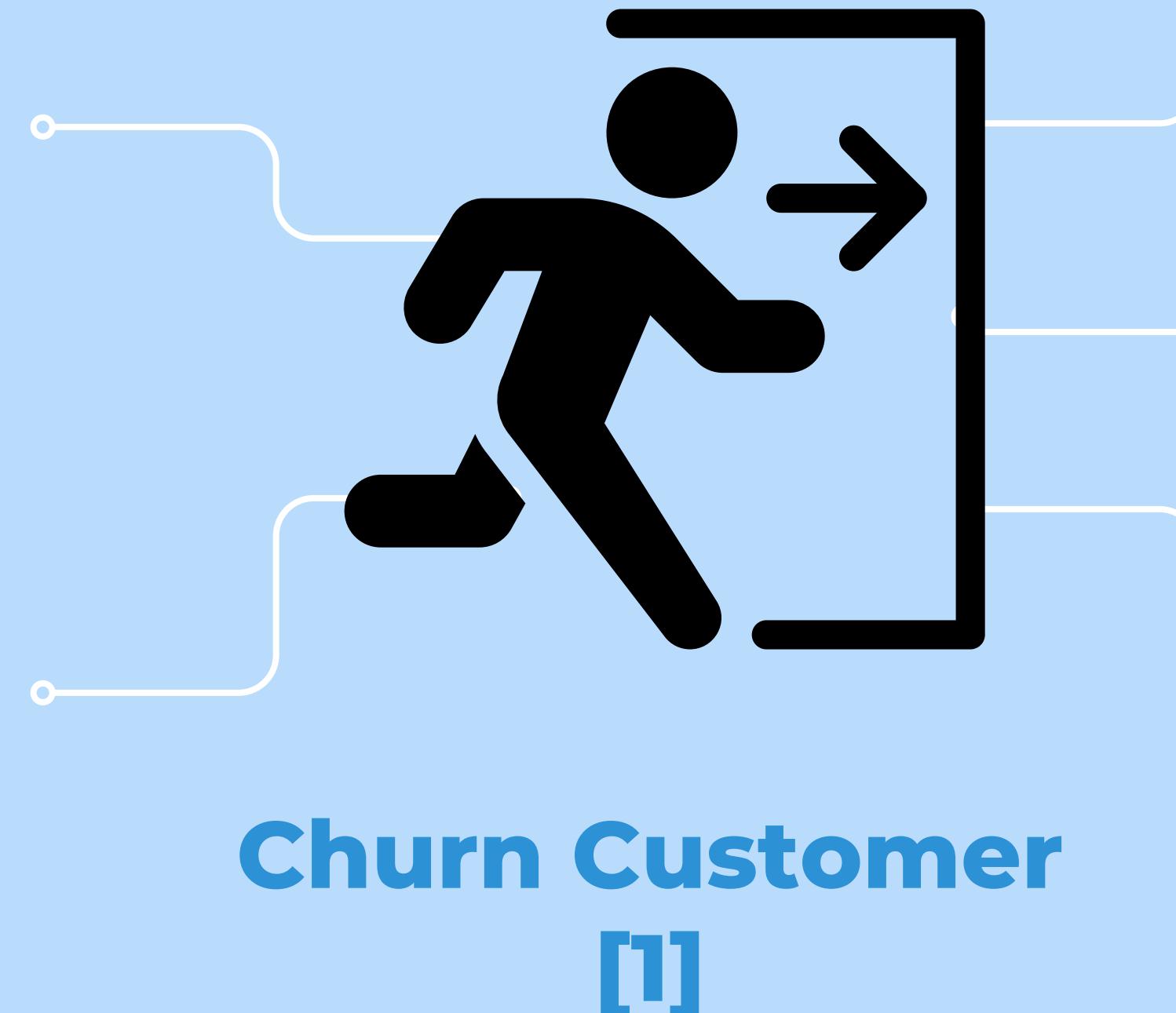
Khi khách hàng rời bỏ, doanh nghiệp mất đi doanh thu từ việc sử dụng dịch vụ, mất đi lòng trung thành và khả năng tạo ra các giao dịch hoặc tương tác lâu dài.

Dự đoán và quản lý khách hàng rời bỏ là một mối quan tâm lớn đối với các doanh nghiệp

# Xây dựng chỉ số "rời bỏ"

(1) Là khách hàng đã sử dụng dịch vụ trước năm 2021 (3 tháng cuối năm 2020)

(2) Là khách hàng không sử dụng dịch vụ vào tháng 6 năm 2021



(1) Là khách hàng đã sử dụng dịch vụ trước năm 2021 (3 tháng cuối năm 2020)

(2) Là khách hàng chưa sử dụng dịch vụ vào tháng 3 năm 2021

(3) Là khách hàng chưa sử dụng dịch vụ vào tháng 6 năm 2021

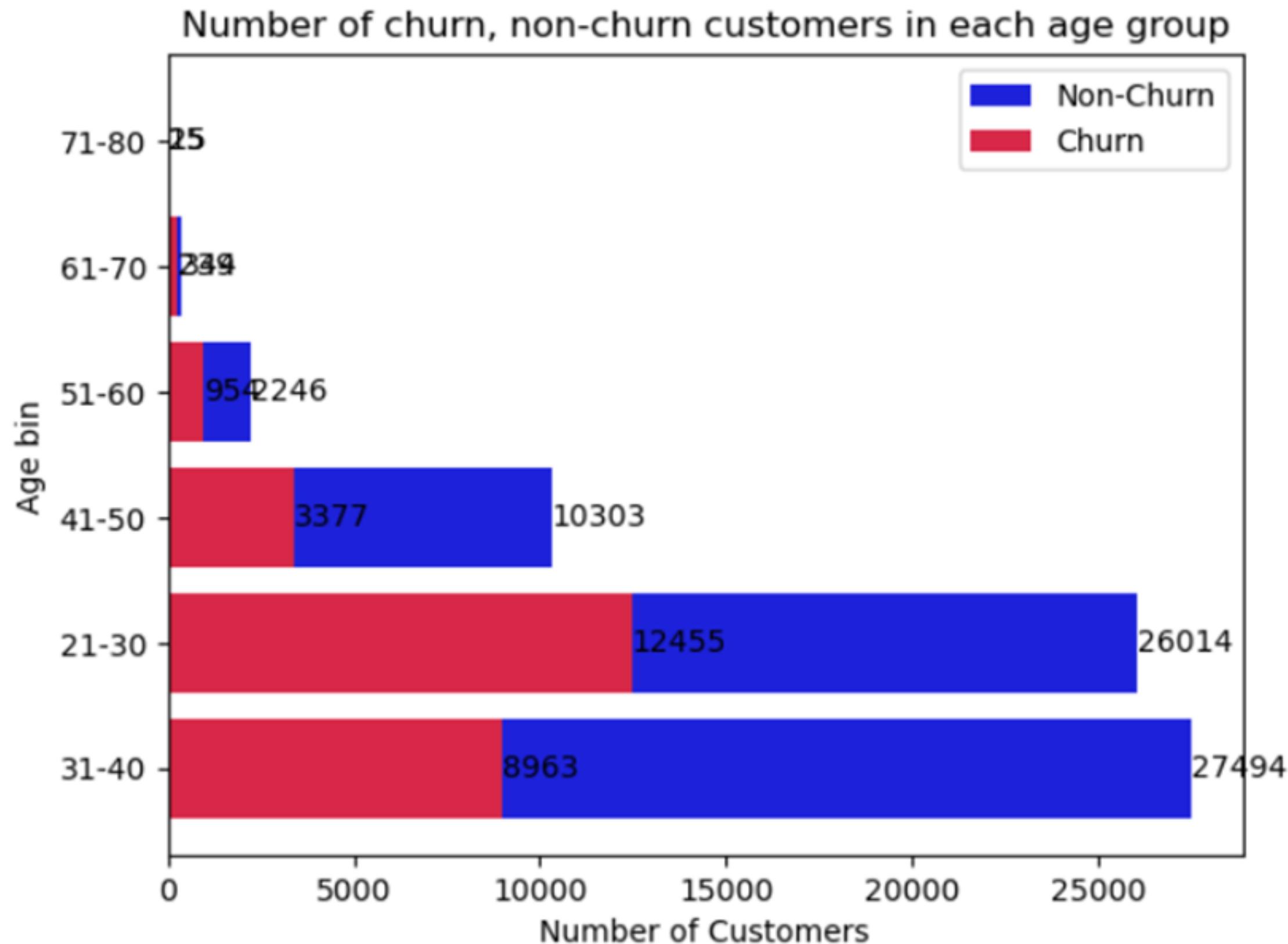
# Exploratory Data Analysis

Biến số  
Biến phân loại

## Phân loại khách hàng rời bỏ theo độ tuổi

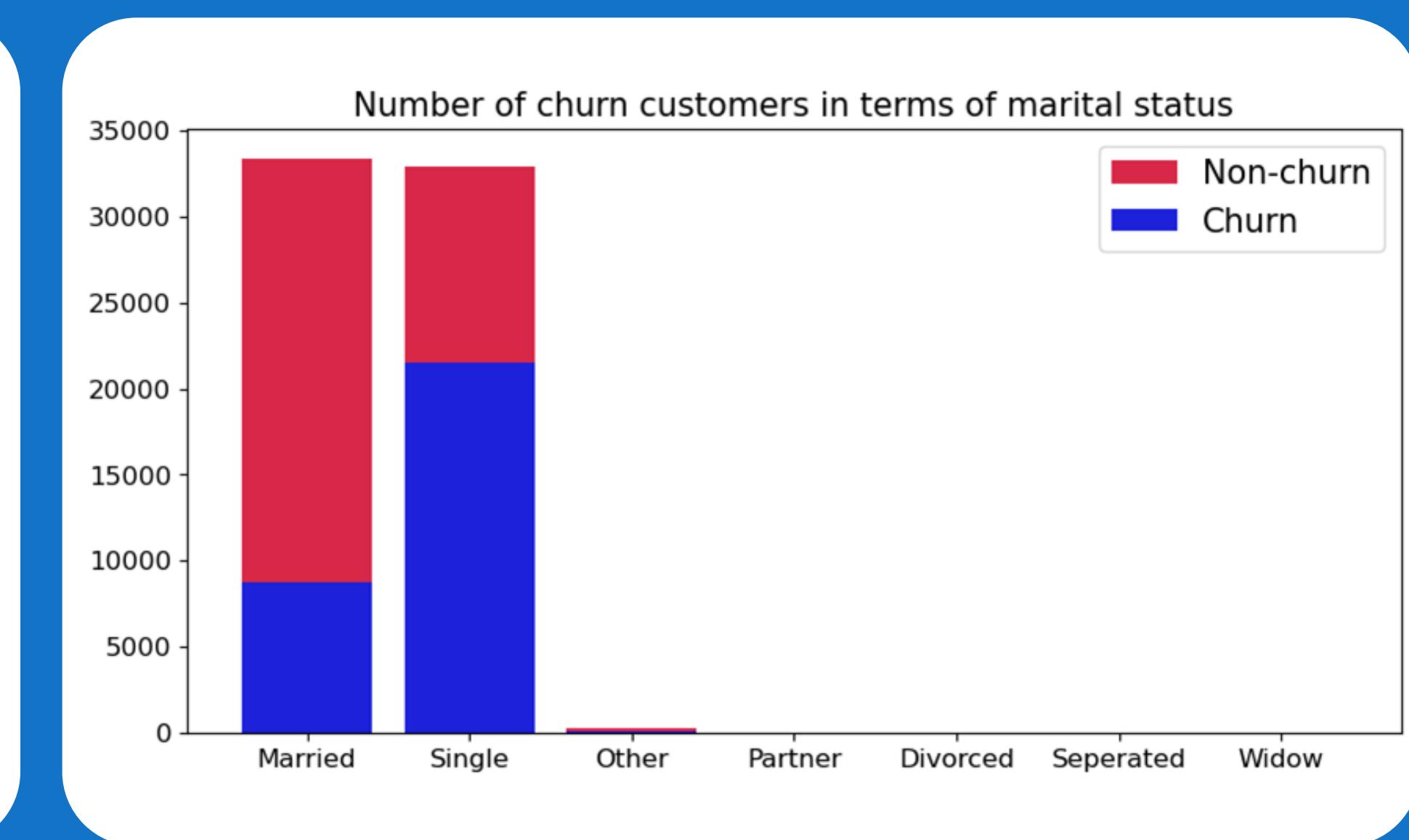
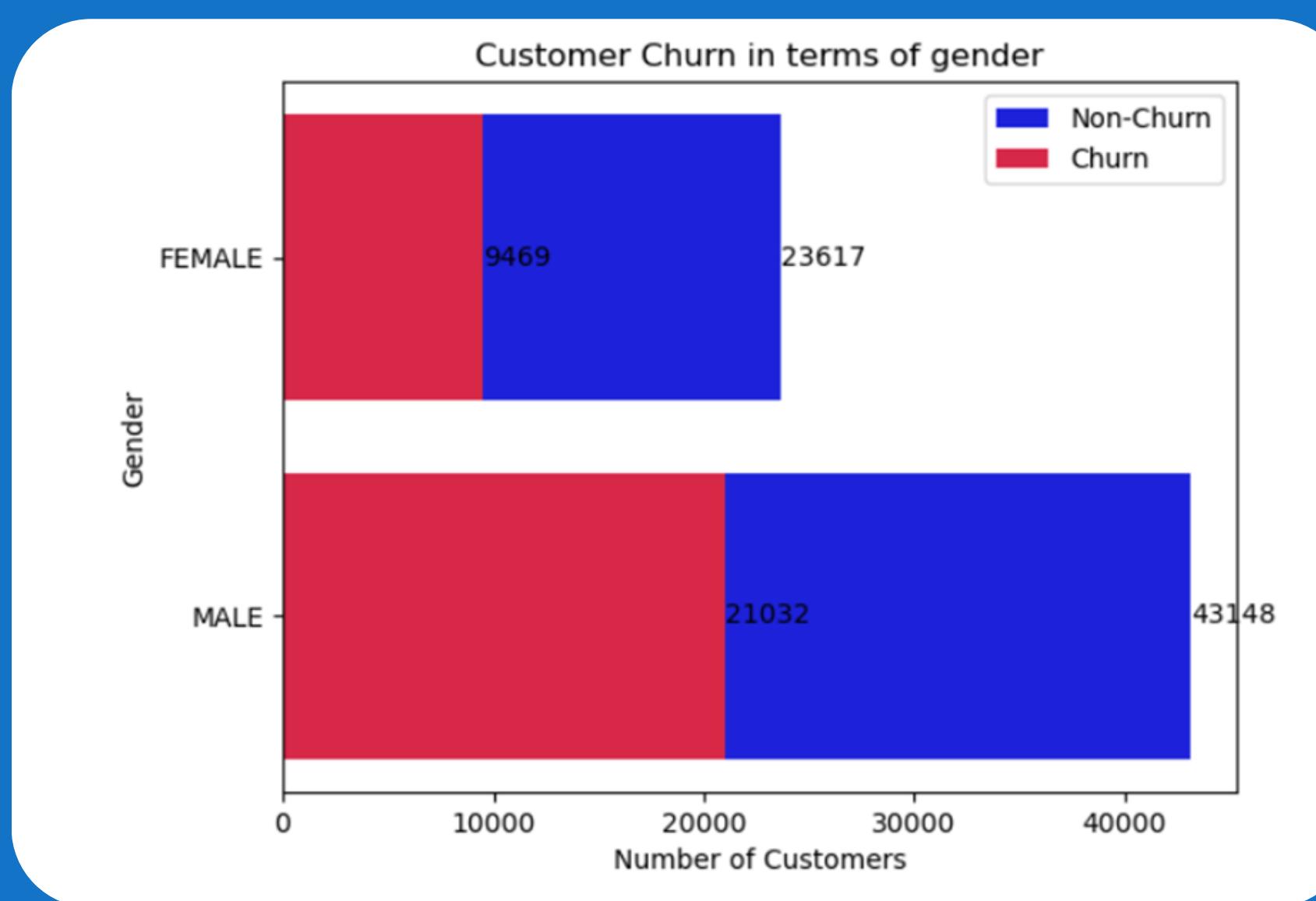
Chú thích:

- [ 0 ]
- [ 1 ]



Điểm chung của tất cả các tập khách hàng chia theo các nhóm tuổi khác nhau là số lượng khách hàng non-churn luôn lớn hơn khách hàng churn, thể hiện được mức độ uy tín và dịch vụ khách hàng tốt của ngân hàng.

# Exploratory Data Analysis > Biến số > Giới tính và tình trạng hôn nhân khách hàng

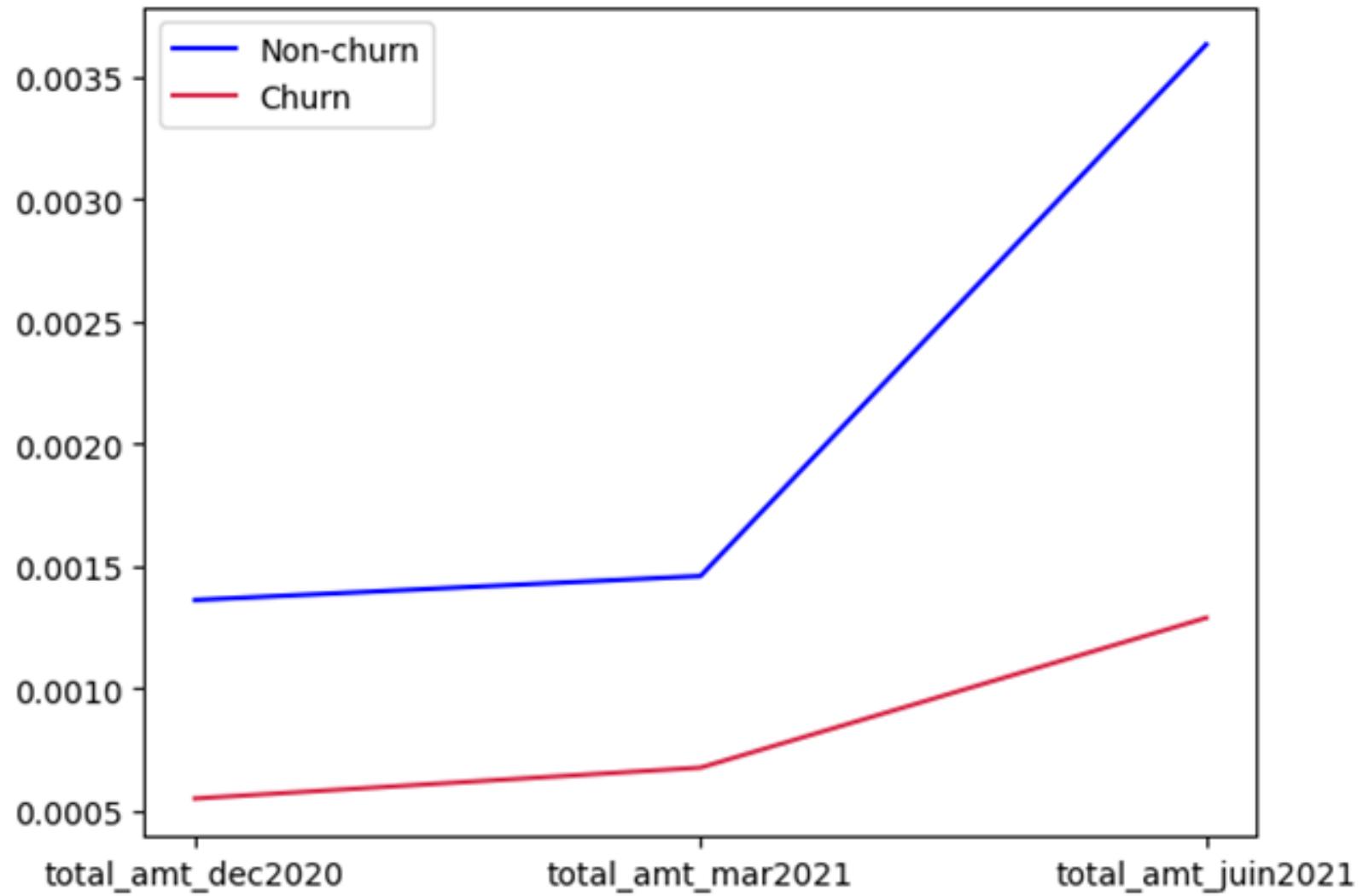


Số lượng nam giới sử dụng dịch vụ nhiều gấp đôi số lượng nữ giới. Tuy nhiên, số lượng khách hàng non-churn luôn lớn hơn churn ở cả 2 nhóm giới tính



- Tập Married: Số lượng khách hàng non-churn gấp 3 lần khách hàng churn
- Tập Single: Số lượng khách hàng churn gấp hơn 2 lần khách hàng non-churn

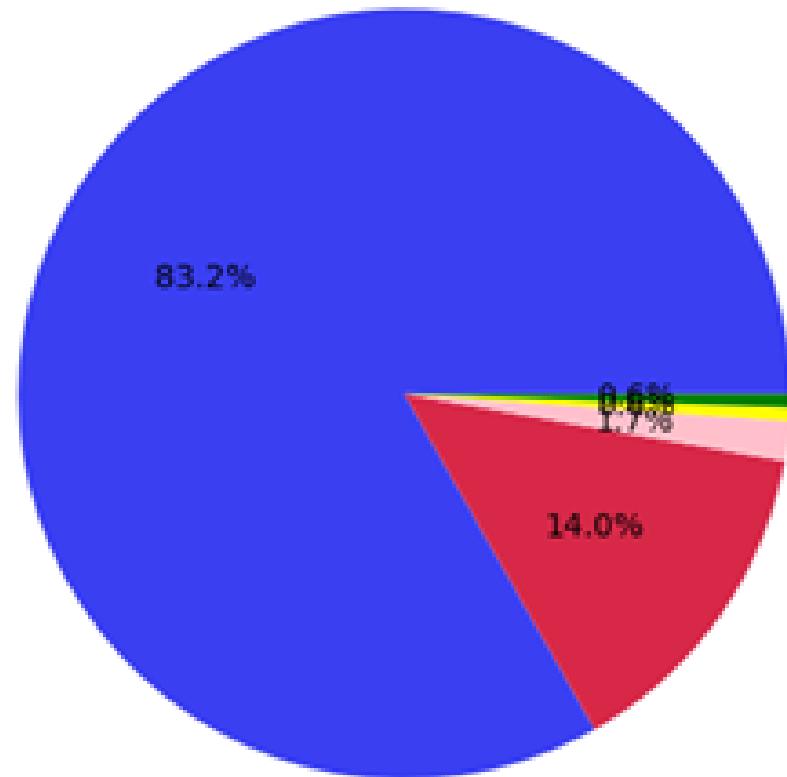
Tổng giá trị tất cả giao dịch được thực hiện trong tháng 12/2020, tháng 3/2021 và tháng 6/2021



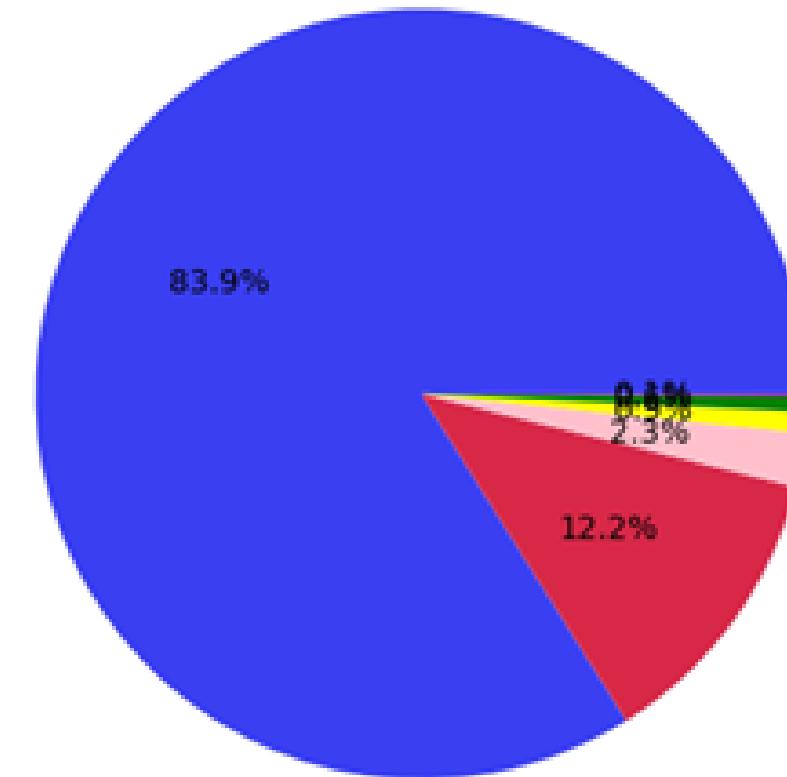
**Đồ thị tổng giá trị tất cả giao dịch trong các giai đoạn**

Đây là một kết quả tích cực chứng minh sự thành công của ngân hàng này trong việc thu hút khách hàng mới cũng như duy trì các khách hàng trung thành, và hạn chế tối đa khách hàng ít hoặc từ bỏ sử dụng dịch vụ của ngân hàng này.

Transaction in March



Transaction in June



giao dịch chuyển khoản  
giao dịch thanh toán  
giao dịch phi tài chính  
giao dịch tiết kiệm  
giao dịch rút tiền ATM

✓  
Giao dịch chuyển khoản trong cả tháng 3/2021 và tháng 6/2021 đều chiếm phần lớn lần lượt với 83.2% và 83.9%.

✓  
Lớn thứ hai là giao dịch thanh toán với 14% khách hàng sử dụng trong tháng 3/2021 và 12.2% trong tháng 6/2021.

✓  
Các loại giao dịch con lại không đáng kể và xấp xỉ gần bằng nhau

# Xử lý và sắp xếp dữ liệu

Làm sạch dữ liệu

Xử lý giá trị thiếu

Xử lý giá trị ngoại lai

Mã hóa dữ liệu

# Checking Missing Value

Có thể thấy dữ liệu bị thiếu khá nhiều, cụ thể:

- Nhóm dữ liệu giao dịch theo tuần trước năm 2021 bị thiếu nhiều nhất (45.2%) (tuy nhiên có thể sử dụng dữ liệu giao dịch tháng và 3 tháng để thay thế)
- Nhóm dữ liệu giao dịch trong tháng 6 năm 2021 (29.336%) (tuy nhiên theo giả định từ ban đầu, dữ liệu bị nan ở tháng này hầu hết đều là khách hàng rời bỏ)

- **Tiến hành loại bỏ các giá trị bị thiếu (nan) của hai trường 'sex' và 'marital\_status'**



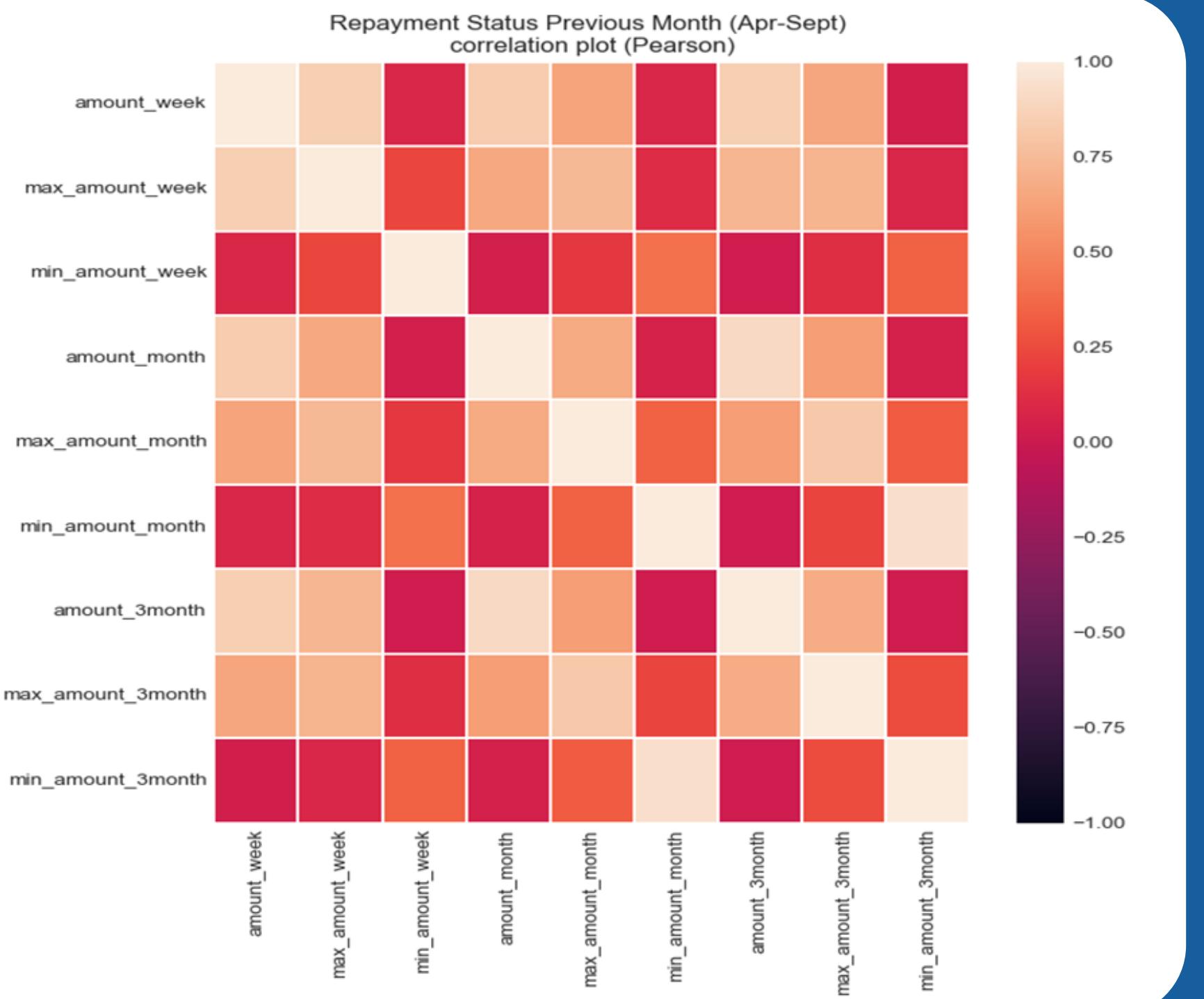
```
1 df = df.dropna(subset=['sex', 'marital_status'])
```

- **Nhận thấy feature "Loại giao dịch sử dụng nhiều nhất" không hiệu quả để xây dựng mô hình, nên nhóm quyết định loại bỏ.**

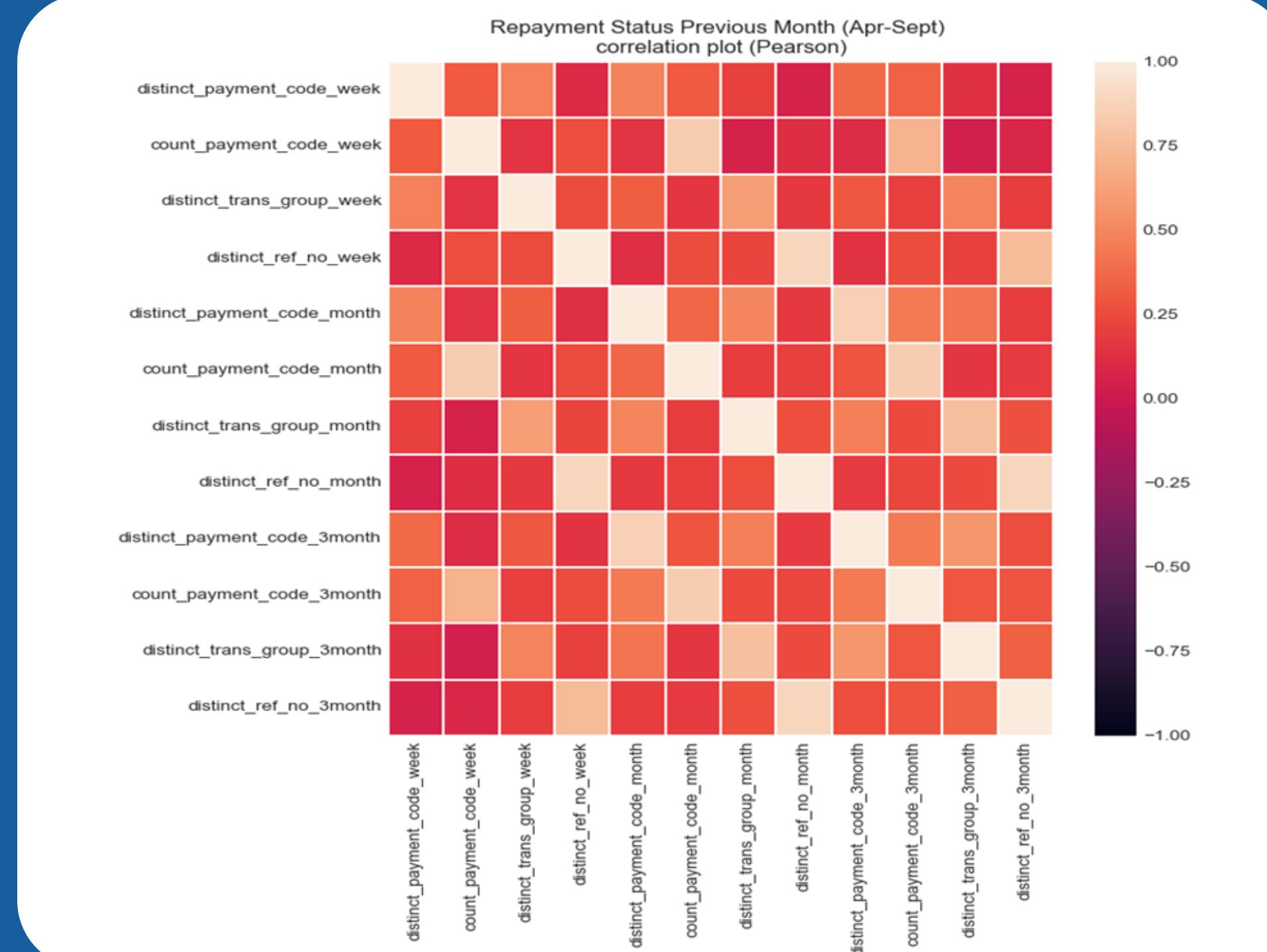


```
1 df.drop(['most_act_mar2021', 'most_act_juin2021'], axis=1, inplace=True)
```

# Làm sạch dữ liệu



Đồ thị tương quan các biến giá trị giao dịch



Đồ thị tương quan các biến dịch vụ ngân hàng

Nhóm bỏ các nhóm các chỉ số theo Tuần và theo Tháng, giữ lại nhóm chỉ số theo 3 Tháng để xây dựng mô hình.

# Xử lý giá trị thiêu

- Khách hàng rời bỏ: Điền các giá trị thiêu bằng 0 trong các cột 'amount\_3month', 'max\_amount\_3month', 'min\_amount\_3month' của các hàng có giá trị 'churn' bằng 1 trong DataFrame.

```
● ● ●  
1 mask = df['churn'] == 1  
2  
3 df.loc[mask, ['amount_3month', 'max_amount_3month', 'min_amount_3month']] = df.loc[mask, ['amount_3month', 'max_amount_3month', 'min_amount_3month']].fillna(0)
```

- Khách hàng không rời bỏ: Loại bỏ các hàng chứa giá trị thiêu (NaN) trong các cột 'amount\_3month', 'max\_amount\_3month', 'min\_amount\_3month' của các hàng có giá trị 'churn' bằng 0.

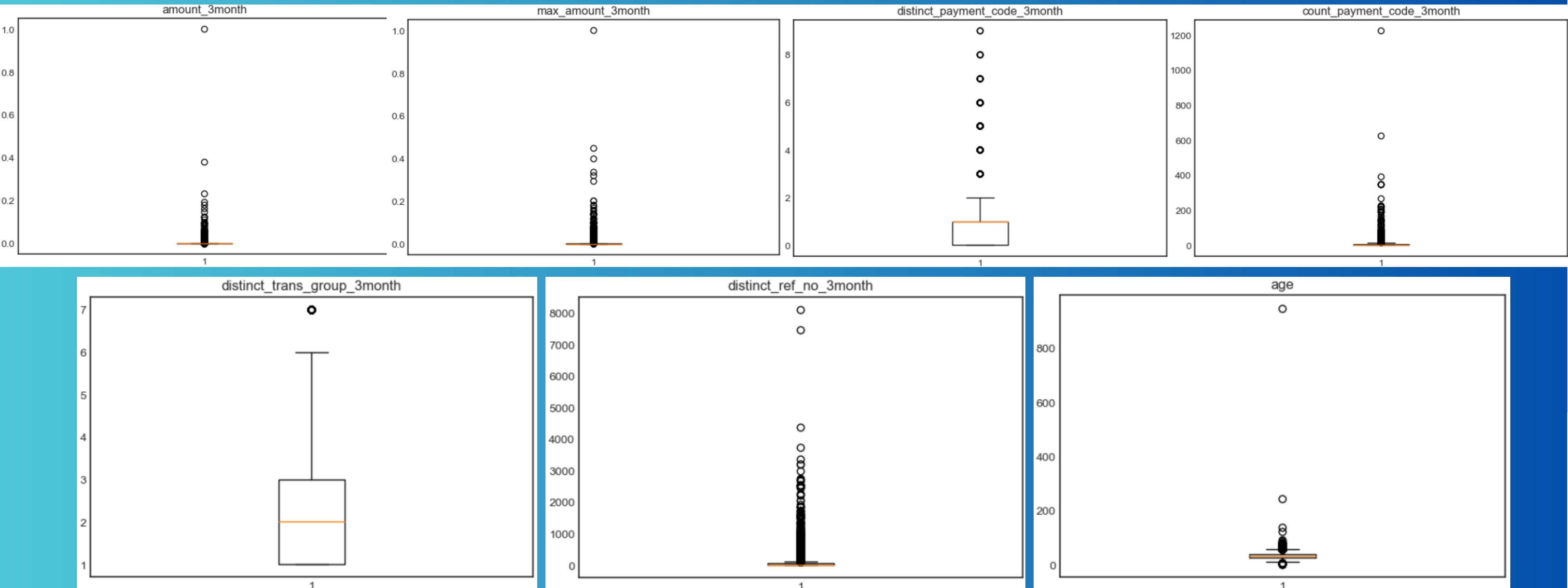
```
● ● ●  
1 mask = df['churn'] == 0  
2  
3 df.loc[mask, ['amount_3month', 'max_amount_3month', 'min_amount_3month']] = df.loc[mask, ['amount_3month', 'max_amount_3month', 'min_amount_3month']].dropna()
```

- Tính tuổi khách hàng bằng cách lấy năm 2021 - năm sinh. Sau đó, thực hiện loại bỏ hết giá trị thiêu ở cột này

```
● ● ●  
1 age=[]  
2 for i in df['birth_incorp_date']:  
3     x=2021-i  
4     age.append(x)  
5 df['age']=age
```

```
● ● ●  
1 df = df.dropna(subset=['age'])
```

# Giá trị ngoại lai



## Xử lý giá trị ngoại lai

**Thực hiện loại bỏ một số outlier với phương pháp interquantile range (IQR):**

- Xác định các khoảng phần tư thứ nhất Q1 và thứ ba Q3. IQR bằng Q3-Q1.
- Lấy các giá trị lớn hơn và bằng  $Q1 - 1.5 \text{IQR}$  và các giá trị nhỏ hơn và bằng  $Q3 + 1.5 \text{IQR}$ .

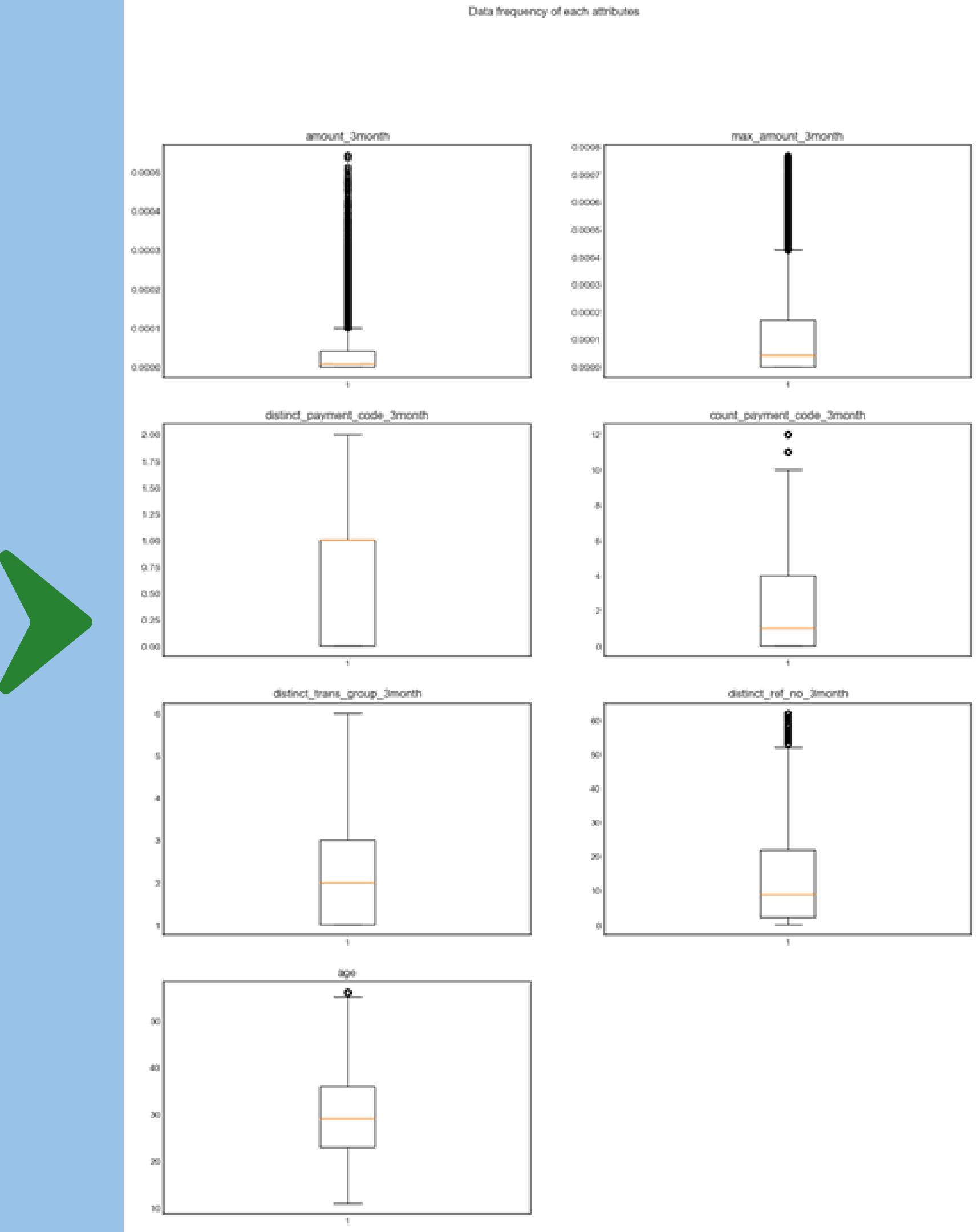
```

numerical_features=['age','amount_3month', 'max_amount_3month', 'min_amount_3month',
'distinct_payment_code_3month', 'count_payment_code_3month',
'distinct_trans_group_3month', 'distinct_ref_no_3month']
for cols in numerical_features:
    Q1 = df[cols].quantile(0.25)
    Q3 = df[cols].quantile(0.75)
    IQR = Q3 - Q1

    filter = (df[cols] >= Q1 - 1.5 * IQR) & (df[cols] <= Q3 + 1.5 *IQR)
    df = df.loc[filter]

    ✓ 0.3s
  
```

Python



# Encoding

**Với cột giới tính (gender), giá trị Female thay thế bằng số 0 và giá trị Male thay thế bằng số 1**

**Thực hiện lấy biến giả (get dummies) với các cột giới tính (sex) và cột tình trạng hôn nhân (marital\_status)**

```
1 encoders_nums = {  
2     "gender": {"FEMALE": 0, "MALE": 1}  
3 }  
4 df = df.replace(encoders_nums)
```

```
1 df = pd.get_dummies(df, columns=['sex',  
2                         'marital_status'])
```

# Checking Missing Value



```
1 # Checking missing value
2 total_missing_value = df.isnull().sum().sort_values(ascending = False)
3 percent_missing_value = (df.isnull().sum()/df.isnull().count()*100).sort_values(ascending = False)
4 pd.concat([total_missing_value, percent_missing_value], axis=1, keys=['Total', 'Percent']).transpose()
```

	sex	marital_status	amount_3month	max_amount_3month	min_amount_3month	distinct_payment_code_3month	count_payment_code_3month	distinct_trans_group_3month	distinct_ref_no_3month	id	churn	age
Total	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Percent	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0



```
1 df['churn'].value_counts()
```



```
0      29090
1      22490
```



```
1 df.shape
```



```
(51580, 12)
```

# Model Development

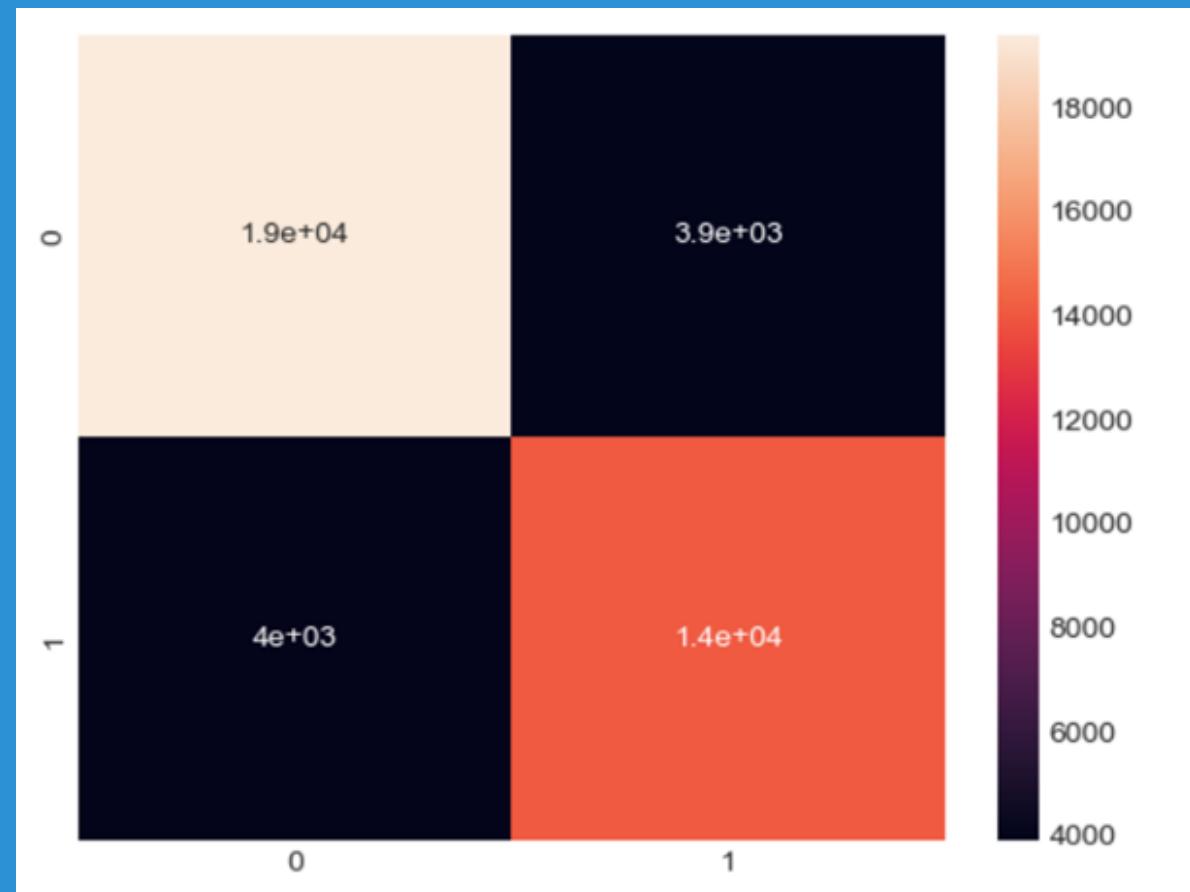
Logistic Regression

Support Vector Classifier

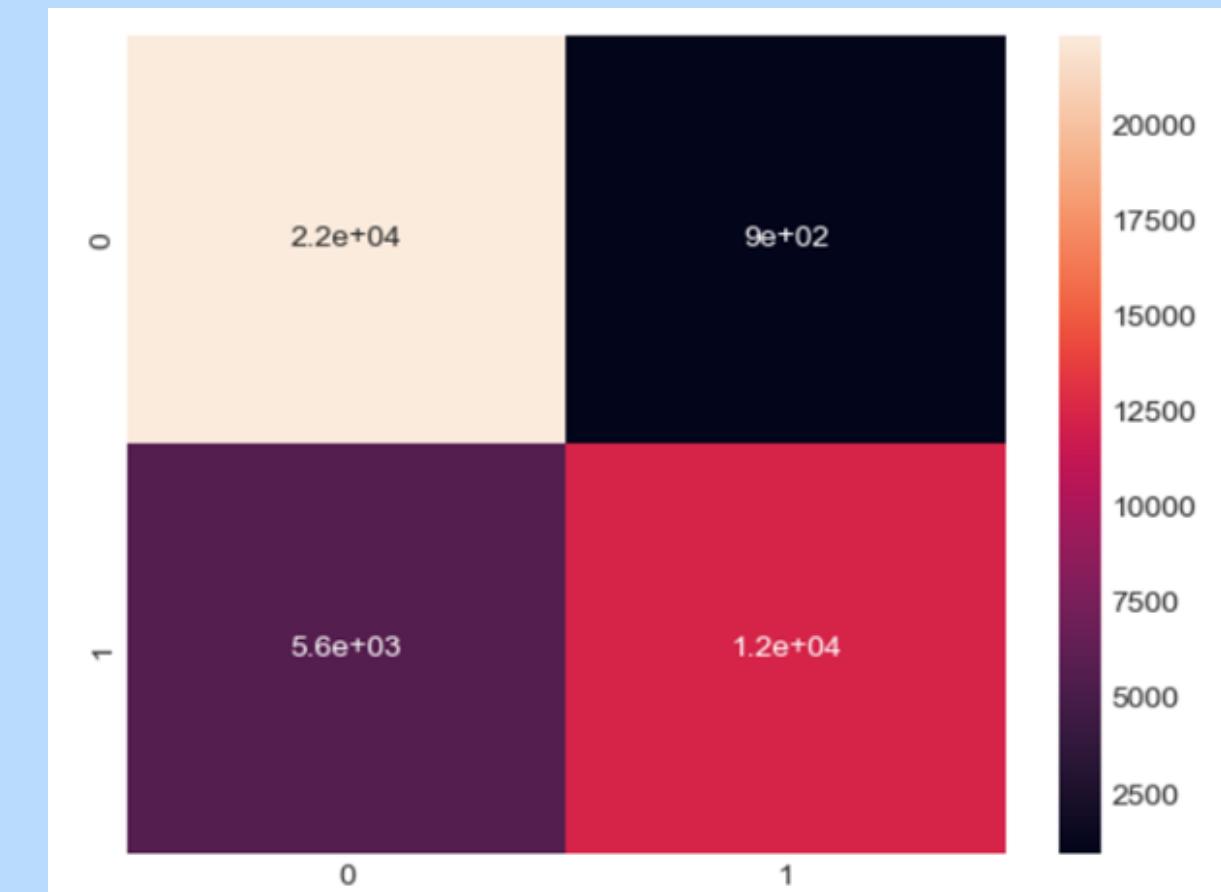
Decision Tree Classifier

Random Forest Classifier

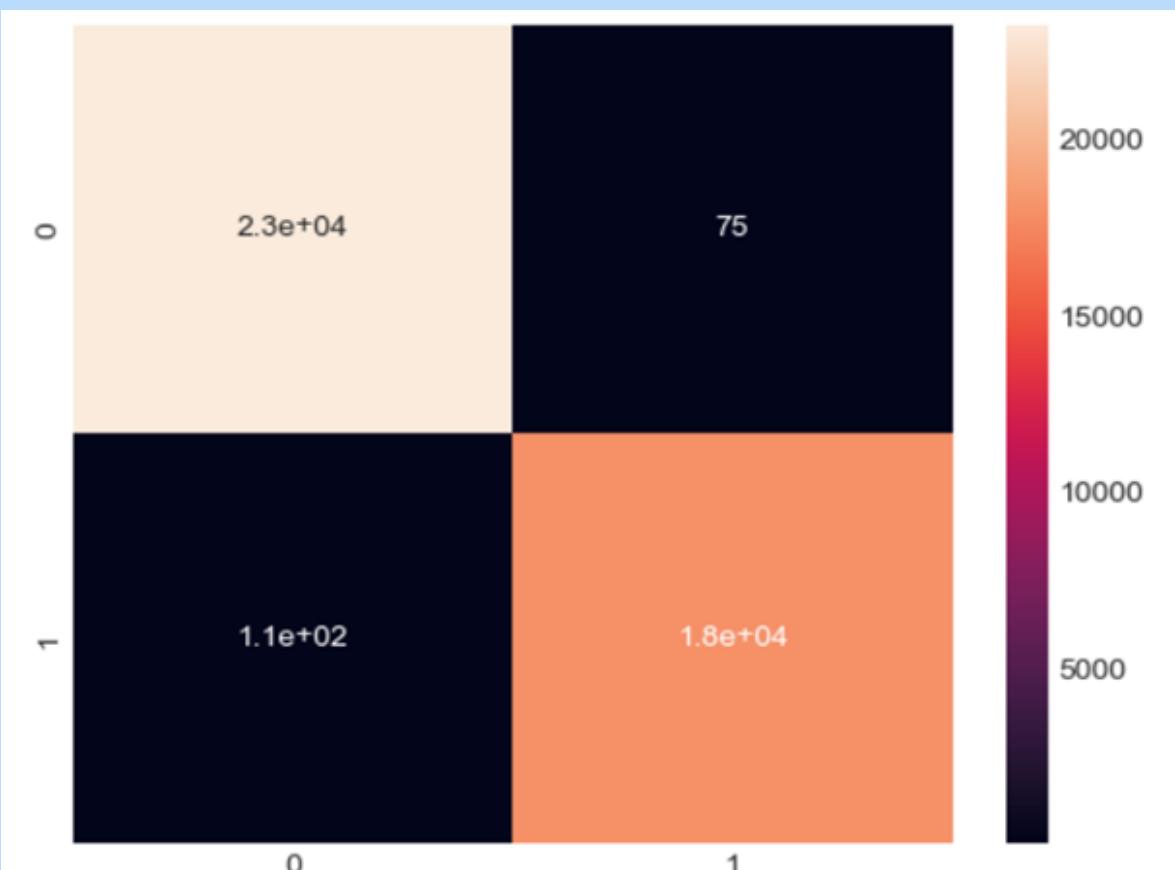
## Logistic Regression



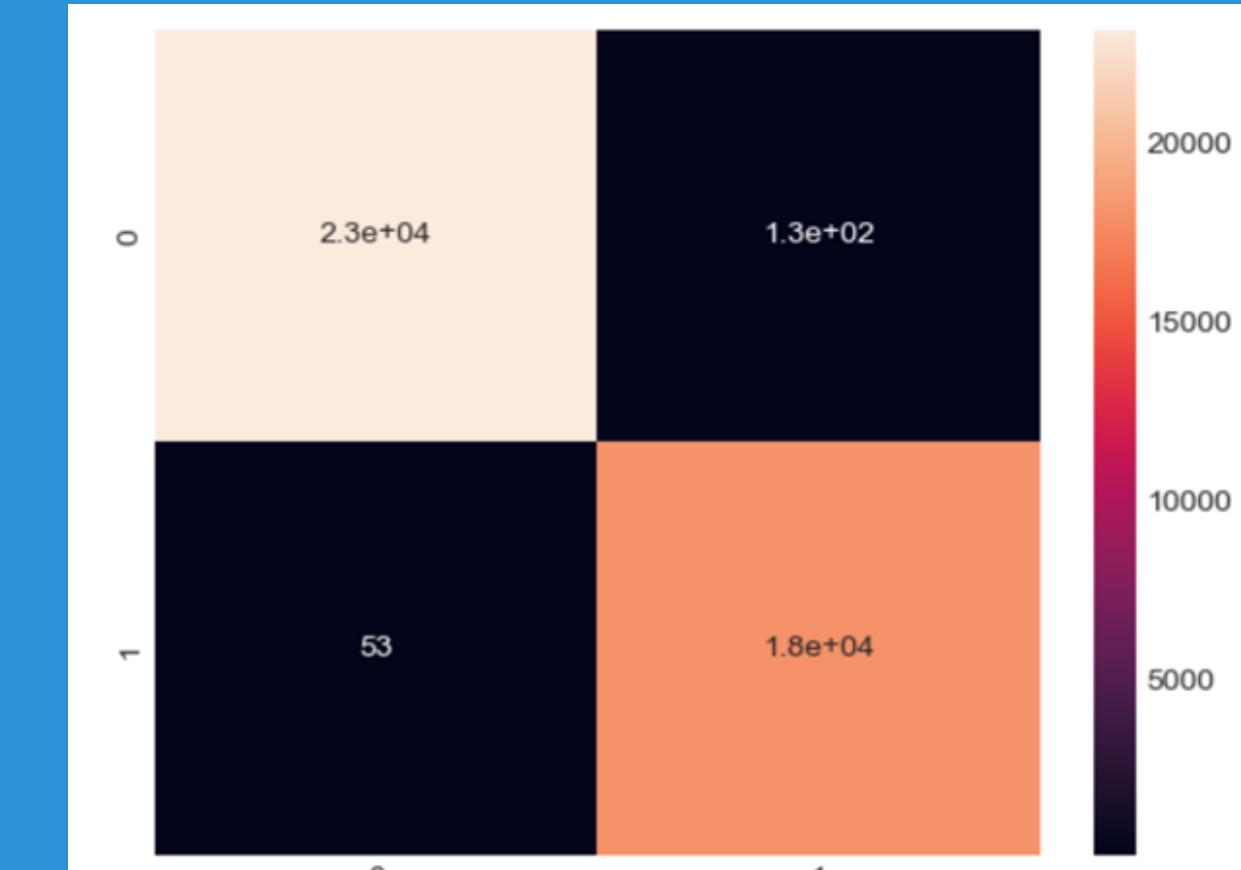
## Support Vector Classifier



## Decision Tree Classifier



## Random Forest Classifier



# Đánh giá và lựa chọn mô hình

Đánh giá các mô hình dựa trên các metric: Accuracy, Precision, Recall, F1 Score

	Classifier	Accuracy score	Precision Score	Recall Score	F1 Score
0	Logistics Regression	0.820061	0.818581	0.754558	0.785256
1	SVC	0.841508	0.930954	0.687528	0.790911
2	Decision Tree Classifier	0.806029	0.768966	0.796910	0.790911
3	Random Forest Classifier	0.868505	0.923725	0.762394	0.835415

Để lựa chọn mô hình tốt nhất, sử dụng metric ‘Recall’.

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

Mô hình Decision Tree có recall score cao nhất

Lựa chọn mô hình Decision Tree để tinh chỉnh.

# Tìm siêu tham số

## Tìm hyper-parameter cho những parameter:

min\_samples\_split  
2, 5, 10

max\_depth  
None, 5, 10, 15

criterion  
gini và entropy

min\_samples\_leaf  
1, 2, 4



```
1 # Get the accuracy scores
2 train_accuracy_DTC = accuracy_score(y_train, train_class_preds_best_tree)
3 test_accuracy_DTC = accuracy_score(y_test, test_class_preds_best_tree)
4
5 print("The accuracy on train data after tuning is ", train_accuracy_DTC)
6 print("The accuracy on test data after tuning is ", test_accuracy_DTC)
```

The accuracy on train data after tuning is 0.845070763861962  
The accuracy on test data after tuning is 0.8422838309422257



Accuracy ở tập train ↑ 0.04%

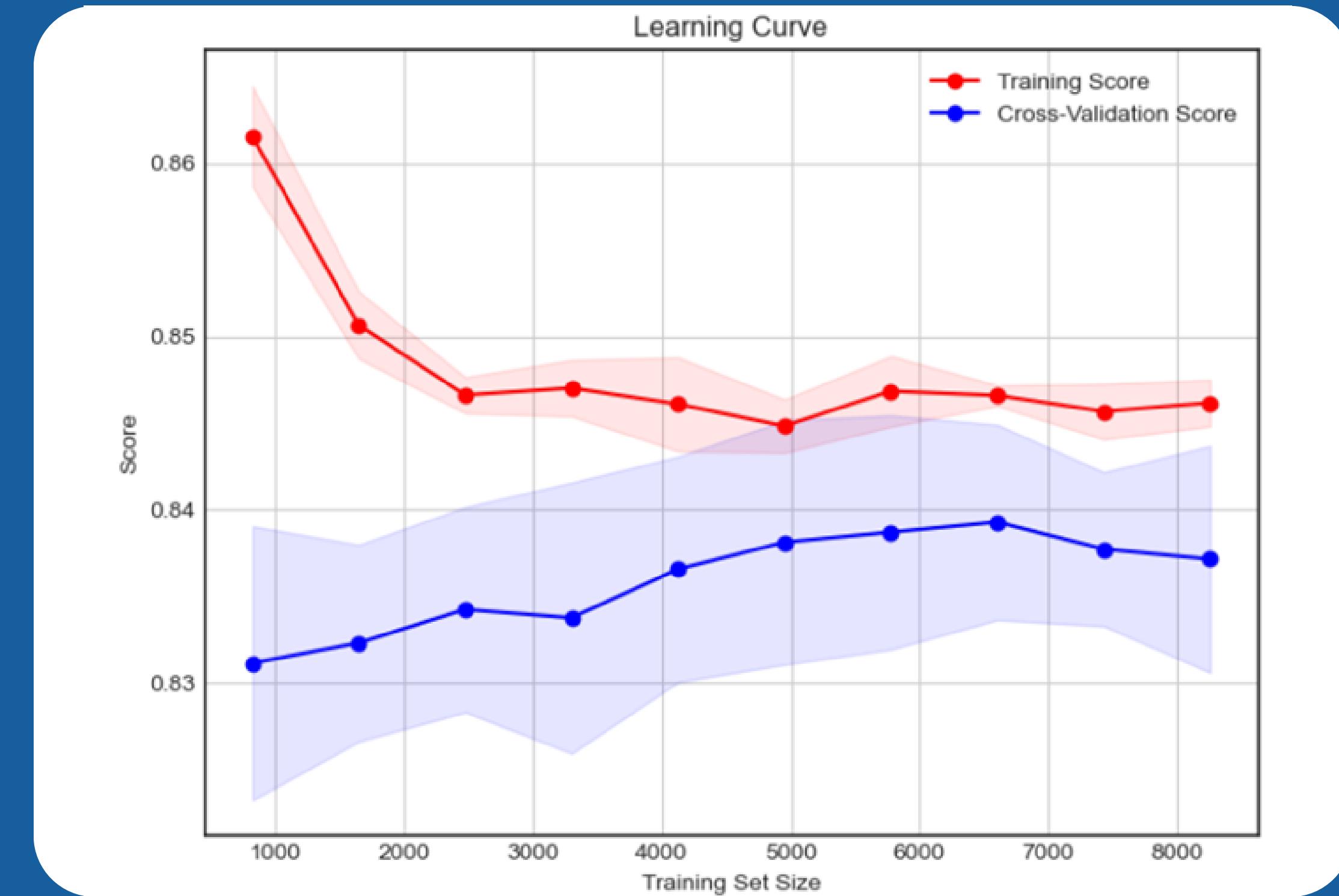


Tunning có hiệu quả



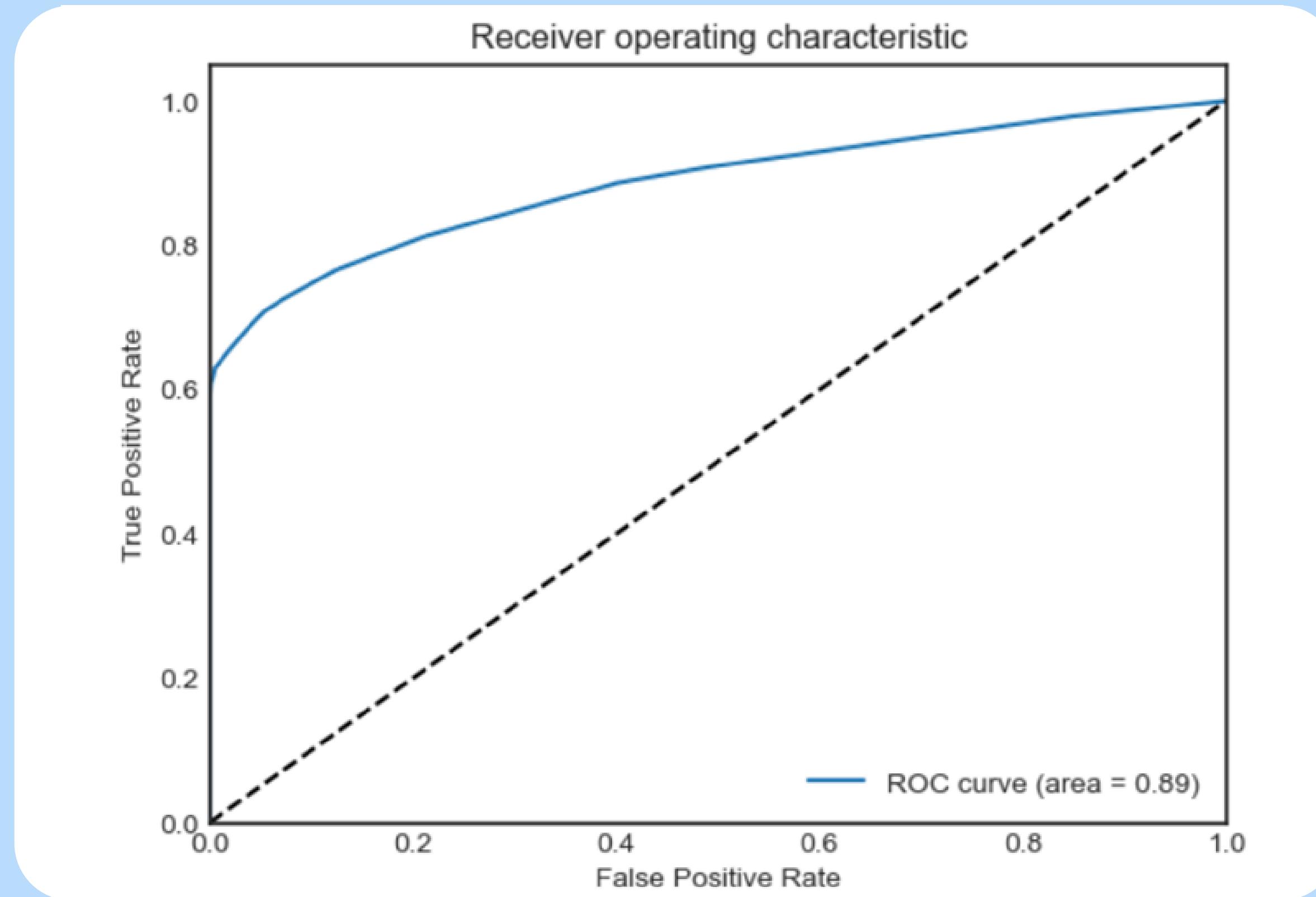
Chọn siêu tham số: entropy và  
max\_depth = 5

## Learning Curve



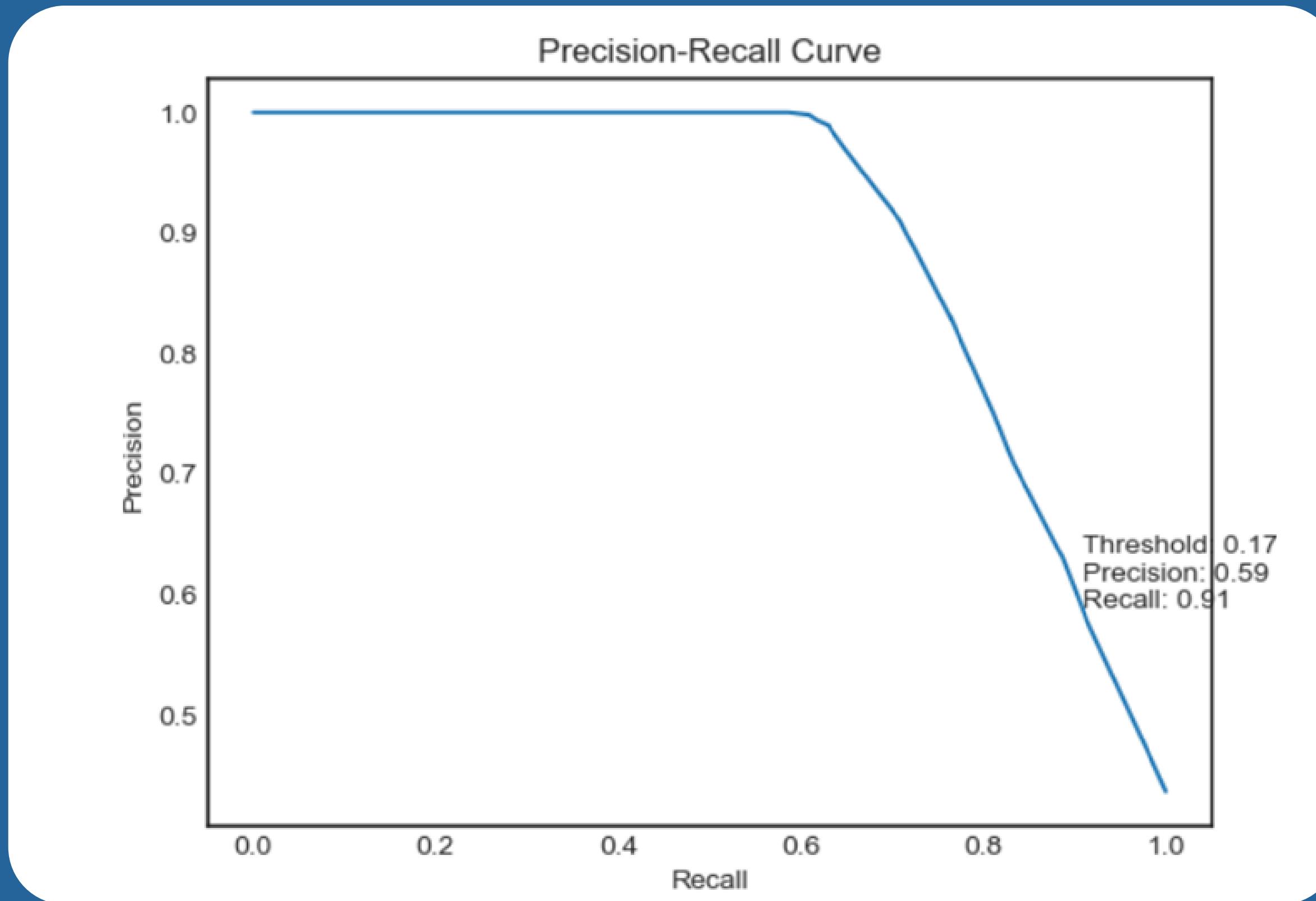
- Training Score và Cross-Validation Score khá tốt, không chênh lệch nhau quá nhiều (nằm trong khoảng 0.86 đến 0.83)
- Mô hình có xu hướng học tốt khi dữ liệu nằm trong khoảng 1000 đến 5000 Training Set Size

## ROC AUC Curve



ROC AUC của mô hình đạt 89% (lớn hơn mức thấp nhất cho phép là 50%)

## Precision Recall Curve



➤ Nhóm chọn điểm mà Recall đạt cao nhất là 0.91 và Precision đạt 0.59 với 'threshold\_index' bằng 3

# Kết luận

Tóm tắt: Dự đoán khả năng rời bỏ của mỗi khách hàng sau 6 tháng sử dụng.

Xây dựng chỉ số rời bỏ: Với giả định rằng khách hàng rời bỏ là khách hàng đã từng sử dụng dịch vụ trước đó hoặc khách hàng chưa từng sử dụng dịch vụ nhưng đều không có giao dịch vào tháng 'target', hay tháng 6 năm 2021.

Phân tích dữ liệu:

- Nhóm tuổi ổn định (từ 30 đến 40 tuổi) có số lượng khách hàng rời bỏ ít hơn so với ở Nhóm tuổi chưa ổn định (từ 21 đến 30);
- Giá trị thanh toán trung bình của cả hai nhóm rời bỏ hay không đều có xu hướng tăng theo thời gian;
- Giao dịch chuyển khoản là giao dịch được thực hiện nhiều nhất trong tháng 3 và tháng 6 năm 2021;
- Lượng khách hàng nam lớn hơn gấp hai lần lượng khách hàng nữ;
- Chủ yếu khách hàng rời bỏ đều trong tình trạng độc thân.

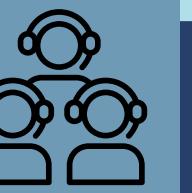
Xây dựng mô hình:

- Bốn mô hình: Logistic Regression, SVC, Decision Tree Classifier và Random Forest Classifier.
- Chỉ số \*\*Recall Score\*\* được lựa chọn để đánh giá và lựa chọn mô hình. Kết quả cho thấy mô hình Decision Tree Classifier có hiệu năng tốt nhất vì có chỉ số Recall Score cao nhất trong bốn mô hình.
- Sau đó, tìm siêu tham số cho mô hình này cho kết quả Accuracy cao hơn mô hình ban đầu (0.04%).
- Khi lựa chọn Threshold Index là 3, Recall Score tốt nhất (91%), nhưng Precision Score trên 50% (59%).

# REFERENCES



TRUSTTHEDATA. (2019). Bank Customer Churn Prediction



SHUBHAM KUMAR (2020). Churn Modelling



SHRUTI PANDIT. (2019). Logistic Regression-Customer Churn for Telecom Domain



ATINDRABANDI. (2018). Telecom Churn Prediction



SAGNIK. (2020). Kaggle

# THANK YOU

*for listening*

[www.datapot.vn](http://www.datapot.vn)

