TRƯỜNG ĐẠI HỌC NGOẠI THƯƠNG
DATAPOT DATA ANALYTICS GROUP

---------o0o---------

BÁO CÁO CUỐI KỲ

MÔN HỌC: PHÂN TÍCH DỮ LIỆU NÂNG CAO TRONG KINH TẾ VÀ KINH DOANH

ĐỀ TÀI:
XÂY DỰNG MÔ HÌNH DỰ ĐOÁN "KHÁCH HÀNG RỜI BỎ" THÔNG QUA DỮ LIỆU VỀ KHÁCH HÀNG SỬ DỤNG DỊCH VỤ NGÂN HÀNG

Nhóm thực hiện : **Nhóm 2**
Thành viên : **23000040 - Đoàn Mạnh Đức**
: **23000109 - Phạm Thu Thảo**
: **23000087 - Ngô Ngọc Minh**
: **23000120 - Đỗ Vy Anh**
: **23000091 - Lê Nhật Hoàng**
: **23000121 - Tô Nông Ngọc Ánh**

Hà Nội, tháng 5 năm 2023

THÔNG TIN THÀNH VIÊN

MÃ SINH VIÊN	HỌ VÀ TÊN	NHIỆM VỤ	ĐÓNG GÓP
23000040	Đoàn Mạnh Đức	- Trưởng nhóm - Phân công và theo dõi tiến độ công việc - Xử lý dữ liệu - Xây dựng mô hình - Tìm siêu tham số cho mô hình	100%
23000109	Phạm Thu Thảo	- Xây dựng chỉ số "rời bỏ" - Phân tích dữ liệu - Xây dựng mô hình	100%
23000087	Ngô Ngọc Minh	- Xây dựng chỉ số "rời bỏ" - Xử lý dữ liệu - Xây dựng mô hình	100%
23000120	Đỗ Vy Anh	- Phân tích dữ liệu - Thuyết trình	100%
23000091	Lê Nhật Hoàng	- Phân tích dữ liệu - Thuyết trình	100%
23000121	Tô Nông Ngọc Ánh	- Làm báo cáo và slides	100%

MỤC LỤC

1. LỜI MỞ ĐẦU

Khách hàng rời bỏ sử dụng dịch vụ là vấn đề mà không một doanh nghiệp nào mong muốn. Đặc biệt, đối với ngân hàng thương mại, loại hình doanh nghiệp có tỷ lệ cạnh tranh cao giữa các ngân hàng với nhau về số lượng khách sử dụng dịch vụ. Vì thế việc dự đoán được khả năng khách hàng rời bỏ là quan trọng và có tính cấp bách cao đối với ngân hàng. Sự khách hàng rời bỏ có thể gây thiệt hại nghiêm trọng cho các ngân hàng, từ việc mất doanh thu đến mất đi lòng trung thành và danh tiếng của họ.

Đề tài này đòi hỏi sự quan tâm đặc biệt bởi việc xây dựng một mô hình dự đoán chính xác có thể giúp ngân hàng nắm bắt được dấu hiệu sớm của khách hàng có ý định rời bỏ. Bằng cách phân tích dữ liệu về hành vi sử dụng dịch vụ ngân hàng, mô hình có thể đưa ra dự đoán và cung cấp thông tin quan trọng giúp ngân hàng thực hiện các biện pháp phòng ngừa và giữ chân khách hàng. Việc xây dựng mô hình dự đoán khách hàng rời bỏ cần đến sự kết hợp giữa các phương pháp phân tích dữ liệu, khai thác thông tin và thuật toán học máy. Điều này đòi hỏi sự nghiên cứu sâu và sự am hiểu về hành vi của khách hàng trong lĩnh vực ngân hàng. Đề tài cung cấp cơ hội phát triển công nghệ và ứng dụng thông tin để giúp ngân hàng tăng cường khả năng dự đoán và quản lý mối quan hệ với khách hàng.

Vì vậy, đề tài ***"Xây dựng mô hình dự đoán "khách hàng rời bỏ" thông qua dữ liệu về khách hàng sử dụng dịch vụ ngân hàng"*** không chỉ có tính cấp bách trong việc nghiên cứu và phát triển công nghệ, mà còn mang lại giá trị quan trọng cho lĩnh vực ngân hàng trong việc tăng cường sự cạnh tranh và đảm bảo sự bền vững của họ trong thị trường ngày càng cạnh tranh hiện nay.

Bài phân tích xây dựng mô hình dự đoán "Khách hàng rời bỏ" sử dụng dịch vụ ngân hàng với dữ liệu về thông tin cá nhân và việc sử dụng dịch vụ của khách hàng trong 03 tháng trước 31/12/2020 để dự đoán khả năng khách hàng rời bỏ sử dụng dịch vụ vào 06 tháng sau (06/2021).

Mô hình xây dựng chỉ số "rời bỏ" sử dụng bộ dữ liệu được kết hợp từ ba phần và nhắm đến tháng 6 năm 2021, cụ thể:

1. Dữ liệu giao dịch của khách hàng từ trước 31/12/2020 (Biến độc lập hay input của mô hình);
2. Dữ liệu giao dịch trong tháng 3 năm 2021 và tháng 6 năm 2021 (Biến phụ thuộc hay output của mô hình).

Lý do lựa chọn: Lựa chọn tháng 6 năm 2021 sẽ chắc chắn được khả năng khách hàng rời bỏ hơn vì:

Trường hợp 1. Sẽ có trường hợp khách hàng không sử dụng dịch vụ trong tháng 3 nhưng tháng 6 vẫn sử dụng lại dịch vụ.

Trường hợp 2. Sẽ có trường hợp khách hàng không sử dụng dịch vụ trong tháng 3 và tháng 6 cũng không sử dụng dịch vụ, lúc này càng có thể khẳng định rằng khách hàng này đã rời bỏ.

Đề tài sử dụng phương pháp định lượng từ Scikit Learn và sử dụng công cụ phân tích dữ liệu Python để phân tích - tổng hợp, thống kê, đi từ cái chung đến cái riêng và kết hợp giữa phân tích

thống kê và phân tích dự báo. Nhóm tác giả quyết định sử dụng 4 phương pháp xây dựng mô hình, bao gồm:

1. Logistics Rgression
2. Support Vector Classifier
3. Decision Tree Classifier
4. Random Forest Classifier

Nội dung bài báo cáo kết quả phân tích được trình bày như sau: Phần 2: Tổng quan về hành vi rời bỏ của khách hàng. Phần 3: Phương pháp và quy trình phân tích dữ liệu. Phần 4: Kết quả và đề xuất, kiến nghị. Phần 5: Kết luận.

Lời cảm ơn: Trong suốt quá trình hoàn thiện dự án, do trình độ hiểu biết còn hạn chế, thiếu kinh nghiệm thực tế nên bài nghiên cứu không thể tránh khỏi những sai sót, nhóm rất mong nhận được những ý kiến đóng góp nhận xét của các thầy/cô để báo cáo được hoàn thiện hơn. Chúng em xin chân thành cảm ơn các anh chị và Trung tâm Datapot đã luôn đồng hành và hỗ trợ chúng em trong suốt quá trình học tập.

2. TỔNG QUAN VỀ HÀNH VI RỜI BỎ CỦA KHÁCH HÀNG

2.1. Khái niệm "Khách hàng rời bỏ"

"Khách hàng rời bỏ" là thuật ngữ được sử dụng để chỉ khách hàng đã sử dụng hoặc tương tác với một công ty, tổ chức hoặc dịch vụ trong một khoảng thời gian nhất định, nhưng sau đó quyết định chấm dứt quan hệ hoặc dừng sử dụng sản phẩm, dịch vụ hoặc mua hàng từ công ty đó. Khách hàng rời bỏ thường xảy ra khi khách hàng không còn hài lòng với chất lượng, giá trị hoặc trải nghiệm mà công ty cung cấp.

2.2. Tác động của sự rời bỏ của khách hàng đến ngân hàng

Việc khách hàng rời bỏ có thể gây ảnh hưởng tiêu cực cho doanh nghiệp. Khi khách hàng rời bỏ, doanh nghiệp mất đi doanh thu từ việc mua hàng hoặc sử dụng dịch vụ, cũng như mất đi lòng trung thành và khả năng tạo ra các giao dịch hoặc tương tác lâu dài. Do đó, dự đoán và quản lý khách hàng rời bỏ là một mối quan tâm quan trọng đối với các doanh nghiệp, bao gồm cả ngành ngân hàng, để duy trì và phát triển mối quan hệ với khách hàng hiện tại.

3. PHƯƠNG PHÁP VÀ QUY TRÌNH PHÂN TÍCH DỮ LIỆU

3.1. Phương pháp xây dựng mô hình

3.1.1. Logistic Regression

Logistic Regression (Hồi quy Logistic) là một thuật toán học máy được sử dụng để mô hình hóa và dự đoán các biến phụ thuộc nhị phân hoặc đa trị dựa trên các biến độc lập.

Trong hồi quy Logistic, mục tiêu là xác định mối quan hệ giữa các biến độc lập (được đo bằng các giá trị liên tục hoặc rời rạc) và một biến phụ thuộc nhị phân (có hai giá trị đầu ra, chẳng hạn 0 và 1) hoặc đa trị (có nhiều hơn hai giá trị đầu ra). Điều này phù hợp khi chúng ta muốn dự đoán hoặc phân loại dữ liệu vào một trong các nhóm xác định.

Thuật toán Logistic Regression sử dụng hàm Logistic (hoặc hàm Sigmoid) để chuyển đổi đầu ra của một hàm tuyến tính thành một xác suất rơi vào một nhóm cụ thể. Hàm Sigmoid giới hạn đầu ra trong khoảng từ 0 đến 1, cho phép ước lượng xác suất dự đoán.

Quá trình huấn luyện mô hình Logistic Regression thường sử dụng phương pháp tối ưu hóa (chẳng hạn như Gradient Descent) để tìm các trọng số tối ưu cho mô hình, sao cho mô hình có khả năng dự đoán chính xác nhất các giá trị phụ thuộc.

Logistic Regression được sử dụng rộng rãi trong nhiều lĩnh vực trong việc dự đoán và phân loại dữ liệu.

3.1.2. Support Vector Classifier

Mô hình SVC (Support Vector Classifier) là một trong những mô hình phân loại phổ biến trong Machine Learning, thuộc họ các mô hình Support Vector Machines (SVM). Mô hình SVC được sử dụng cho các bài toán phân loại nhị phân và đa lớp, với khả năng tạo ra các siêu phẳng (hyperplane) trong không gian đặc trưng để tách các điểm dữ liệu thuộc các lớp khác nhau.

Đặc điểm của mô hình SVC:

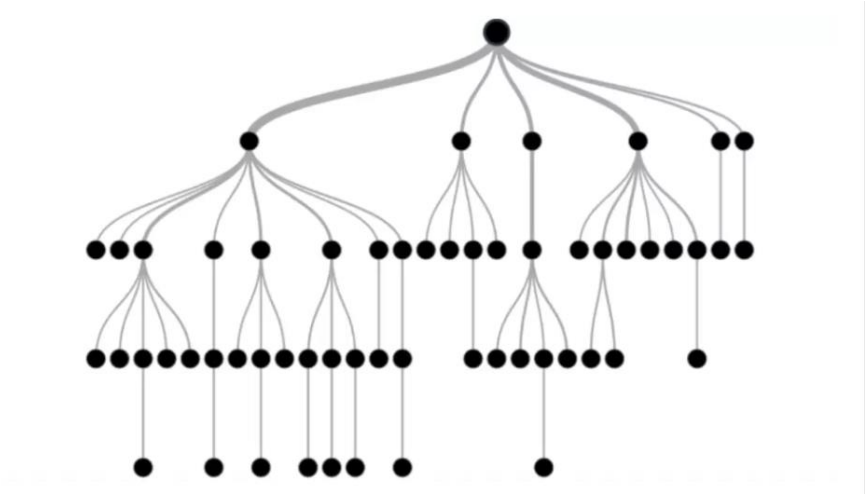
- Tạo ra siêu phẳng: Mô hình SVC tạo ra một hoặc nhiều siêu phẳng trong không gian đặc trưng, tối đa hóa khoảng cách giữa các điểm dữ liệu thuộc các lớp khác nhau.
- Hàm kernel: Mô hình SVC sử dụng các hàm kernel để chuyển đổi không gian đặc trưng ban đầu thành không gian cao hơn, giúp tạo ra siêu phẳng phân chia tốt hơn trong không gian mới.
- Hỗ trợ các vector: Các điểm dữ liệu quan trọng được gọi là các vector hỗ trợ (support vectors) và được sử dụng để xác định siêu phẳng phân chia.
- Điều chỉnh độ lề: Mô hình SVC có thể điều chỉnh độ lề (margin) để kiểm soát sự phù hợp của mô hình với dữ liệu. Độ lề là khoảng cách giữa siêu phẳng và các điểm dữ liệu gần nhất của các lớp.

Mô hình SVC thường được sử dụng trong các bài toán phân loại với dữ liệu tuyến tính và phi tuyến tính. Các tham số quan trọng trong mô hình SVC bao gồm loại hàm kernel, tham số điều chỉnh độ lề (C), và tham số điều chỉnh đặc trưng của các hàm kernel (như gamma trong kernel RBF).

3.1.3. Decision Tree Classifier

Decision Tree là một công cụ có các ứng dụng bao gồm nhiều lĩnh vực khác nhau. Decision Tree có thể được sử dụng để phân loại cũng như các bài toán hồi quy. Bản thân cái tên gợi ý rằng nó sử dụng một sơ đồ giống như cấu trúc cây để hiển thị các dự đoán là kết quả của một loạt các phân tách dựa trên tính năng. Nó bắt đầu với một Root Nodes và kết thúc bằng một quyết định của các lá.

Cây quyết định được sử dụng để xử lý các tập dữ liệu phi tuyến tính một cách hiệu quả. Công cụ cây quyết định được sử dụng trong cuộc sống thực trong nhiều lĩnh vực, chẳng hạn như kỹ thuật, quy hoạch dân dụng, luật và kinh doanh. Cây quyết định có thể được chia thành hai loại; cây quyết định biến phân loại và biến liên tục.



3.1.4. Random Forest Classifier

Random Forest Classifier là một thuật toán học máy được sử dụng cho bài toán phân loại. Nó là một dạng mở rộng của Decision Tree (Cây quyết định) và kết hợp nhiều cây quyết định thành một "rừng" để đưa ra dự đoán chính xác hơn.

Random Forest hoạt động bằng cách xây dựng một tập hợp các cây quyết định (decision trees) độc lập, mỗi cây được huấn luyện trên một tập dữ liệu con và một tập dữ liệu con khác nhau được lấy mẫu từ dữ liệu huấn luyện ban đầu. Khi cần dự đoán, mỗi cây trong rừng sẽ đưa ra một dự đoán riêng và sau đó dự đoán cuối cùng được xác định bằng cách lấy phiếu bầu từ tất cả các cây quyết định thành viên.

Cách thức hoạt động của Random Forest Classifier mang lại nhiều ưu điểm. Đầu tiên, nó có khả năng xử lý dữ liệu lớn và nhanh chóng. Thứ hai, nó có khả năng xử lý cả các biến rời rạc và biến liên tục. Thứ ba, Random Forest có khả năng ứng phó với nhiễu và tránh hiện tượng overfitting. Cuối cùng, nó cung cấp khả năng ước lượng độ quan trọng của các biến trong việc phân loại.

3.2. Quy trình phân tích dữ liệu

3.2.1. Sử dụng dữ liệu cần thiết

Đầu tiên, nhóm tác giả tiến hành loại bỏ một số trường mà nhóm cho rằng không có ý nghĩa trong việc xây dựng mô hình khách hàng rời bỏ.

```
# Drop unwanted columns (Heo So columns)
df.drop(['resid_province', 'resid_district', 'resid_wards',
        'savingValueMar2021_heoSo', 'savingValueJuin2021_heoSo',
        'totalLoginMar2021_heoSo', 'totalLoginJuin2021_heoSo', 'totalSavings2021_heoSo',
        'balanceJuin2021', 'nominal_interestJuin2021', 'real_interestJuin2021',
        'nhomno_xhtdJuin2021', 'categoryJuin2021', 'sub_productJuin2021',
        'loaikyhanJuin2021', 'sectorJuin2021', 'product_codeJuin2021'], axis=1, inplace=True)
df.head()
```

3.2.2. Xây dựng chỉ số "rời bỏ"

Để thuận tiện cho việc xây dựng mô hình dự đoán khách hàng rời bỏ phù hợp với bộ dữ liệu đã cho, nhóm tác giả quyết định định nghĩa "khách hàng rời bỏ" là khách hàng thỏa mãn các điều kiện như sau: (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. Hoặc: (1) Là khách hàng chưa 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.

Do đó, khách hàng không rời bỏ được xác định như sau: (1) Là khách hàng đã sử dụng dịch vụ trước năm 2021 (3 tháng cuối năm 2020); (2) Là khách hàng vẫn còn sử dụng dịch vụ vào tháng 6 năm 2021.

```
df['churn'] = -1

# Not Churn customer: Là khách hàng có sử dụng dịch vụ trong 3 tháng cuối năm 2020 (amount_3month không nan **và** khác 0);
#                                     **và** có sử dụng dịch vụ trong tháng 6 (total_act_juin2021 không nan **và** khác 0)
df.loc[(df['amount_3month'].notnull()) & (df['amount_3month']!=0)  &
       (df['total_act_juin2021'].notnull()) & (df['total_act_juin2021']!=0), 'churn'] = 0

# Churn1 customer: Là khách hàng có sử dụng dịch vụ trong 3 tháng cuối năm 2020 (amount_3month không nan **và** khác 0);
#                                     **và** không sử dụng dịch vụ trong tháng 6(total_act_juin2021 nan **hoặc** bằng 0)
df.loc[(df['amount_3month'].notnull()) & (df['amount_3month']!=0) &
        (df['total_act_juin2021'].isnull()) | (df['total_act_juin2021']==0), 'churn'] = 1

# Churn2 customer: Là khách hàng không sử dụng dịch vụ trong 3 tháng cuối năm 2020 (amount_3month nan **hoặc** khác 0);
#                                     **và** không sử dụng dịch vụ trong tháng 3 năm 2021 (total_act_mars2021 nan **hoặc** bằng 0);
#                                     **và** không sử dụng dịch vụ trong tháng 6 năm 2021 (total_act_juin2021 nan **hoặc** bằng 0)
df.loc[(df['amount_3month'].isnull()) | (df['amount_3month']==0)  &
       ((df['total_act_mar2021'].isnull()) | (df['total_act_mar2021']==0) &
        (df['total_act_juin2021'].isnull())) | (df['total_act_juin2021']==0), 'churn'] = 1
```

3.2.3. Phân tích khám phá dữ liệu (EDA)

3.2.3.1. Biến số

a) Tuổi khách hàng

Tính tuổi của từng khách hàng bằng cách lấy năm hiện tại là năm 2021 trừ đi năm sinh.

```
age=[]
for i in df['birth_incorp_date']:
    x=2021-i
    age.append(x)
df['age']=age
```

Tạo một danh sách chứa các giá trị biên của các khoảng tuổi để định nghĩa các nhóm tuổi 21 - 30, 31 - 40, 41 - 50, 51 - 60, 61 - 70, 71 - 80 là các tên tương ứng với các khoảng tuổi tương ứng. pd.cut() được sử dụng để chia các giá trị trong cột 'age' của DataFrame vào các khoảng tuổi đã

được xác định bởi bins và gán nhãn tương ứng từ names. Cột mới có tên 'AGE_BIN' được thêm vào DataFrame để lưu trữ kết quả.

Ví dụ: Nếu giá trị tuổi là 25, nó sẽ được gán nhãn '21-30' và lưu vào cột 'AGE_BIN' tương ứng với hàng tương ứng trong DataFrame.

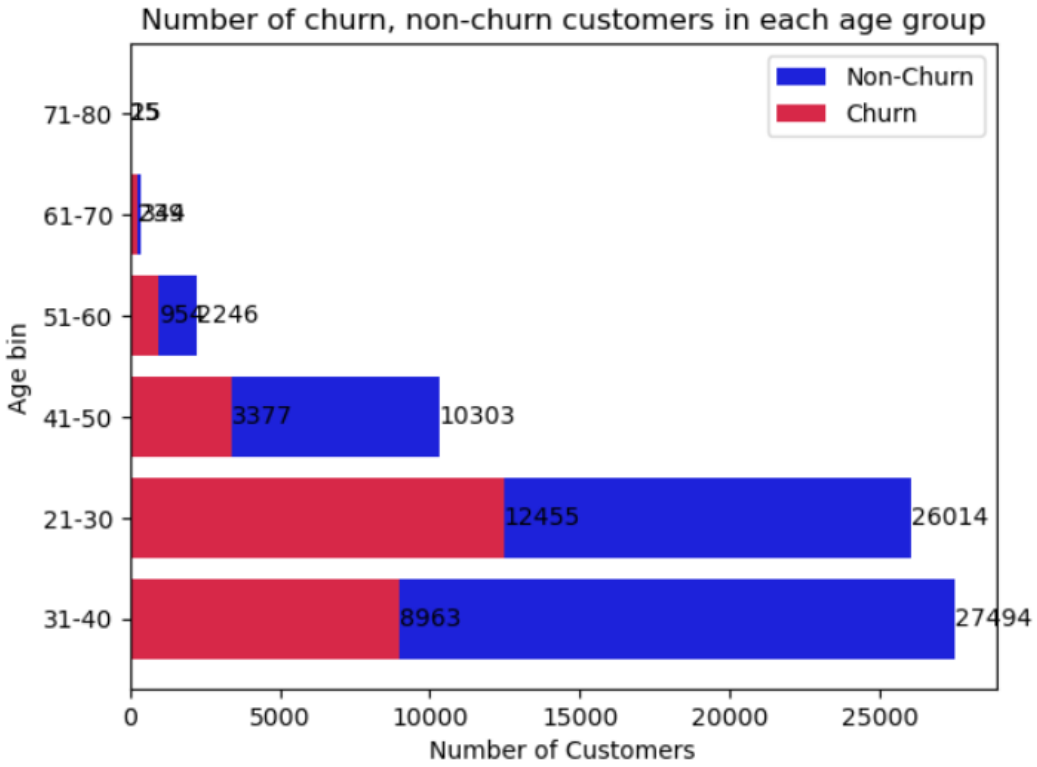
```
bins = [20,30,40,50,60,70,80]
names = ['21-30','31-40','41-50','51-60','61-70','71-80']
df['AGE_BIN'] = pd.cut(x=df.age, bins=bins, labels=names, right=True)
```

Đồ thị: Phân loại khách hàng rời bỏ theo độ tuổi

Đối tượng chủ yếu sử dụng dịch vụ của ngân hàng và được ghi lại trong dataset sẽ nằm trong nhóm tuổi 31-40 và 21-30.

- Đối với nhóm tuổi 31-40 (Số khách hàng: 27494): Số lượng khách hàng non-churn nhiều gấp 2 lần số lượng khách hàng churn. Điều này thể hiện độ trung thành của tập khách hàng này khi họ đã có thu nhập ổn định và nhu cầu sử dụng dịch vụ cao với nhiều mục đích như chuyển khoản, gửi tiết kiệm, v.v.

```
age_cnt = df.AGE_BIN.value_counts()
age_0 = (df.AGE_BIN[df['churn'] == 0].value_counts())
age_1 = (df.AGE_BIN[df['churn'] == 1].value_counts())

plt.subplots(figsize=(8,5))

plt.bar(age_0.index, age_0.values, label='Non-churn',color='#1d22da')
plt.bar(age_1.index, age_1.values, label='Churn',color='#d72848')
for x,y in zip(names,age_0):
    plt.text(x,y,y,fontsize=12)
for x,y in zip(names,age_1):
    plt.text(x,y,y,fontsize=12)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.title("Number of churn customers in each age group", fontsize=15)
plt.legend(loc='upper right', fontsize=10)
plt.show()
```



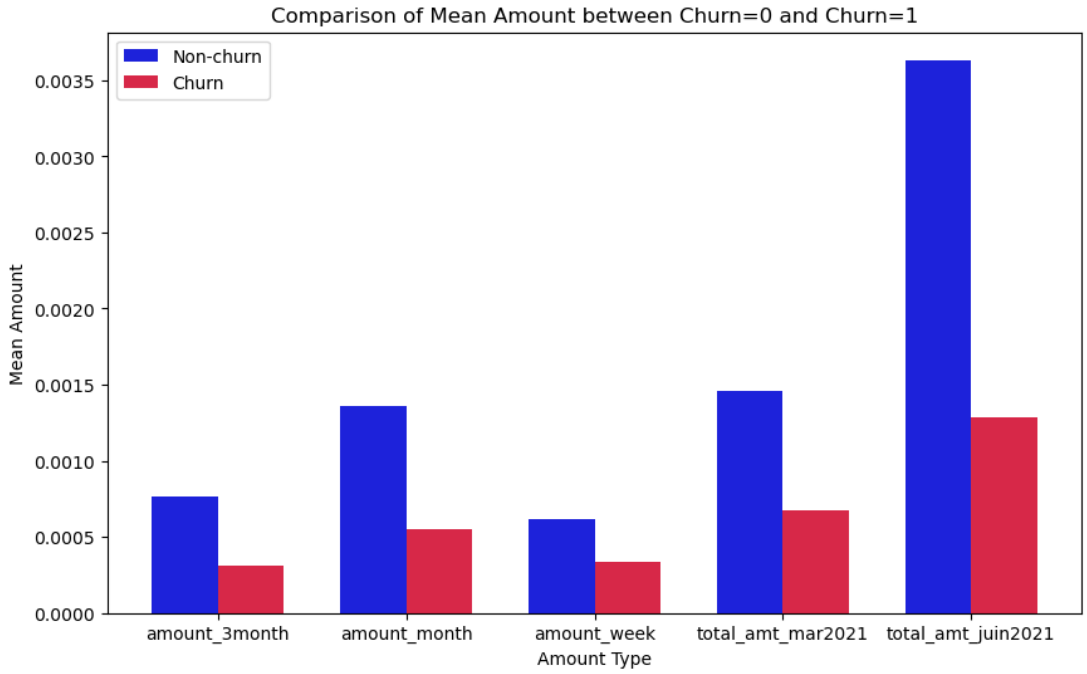
- Đối với nhóm tuổi 21-30 (Số khách hàng: 26014): Số lượng khách hàng non-churn và churn xấp xỉ bằng nhau. Tập khách hàng này còn trẻ nên chưa có thu nhập ổn định và chưa hiểu rõ tầm quan trọng của dịch vụ ngân hàng nói chung. Điều này dẫn tới hành vi sử dụng dịch vụ của tập khách hàng này chưa thực sự trung thành hay chưa sử dụng dịch vụ thường xuyên.
- Đối với nhóm tuổi 41-50 (Số khách hàng: 10303): Từ nhóm tuổi này trở đi, số lượng khách hàng được ghi lại trong dataset bắt đầu có xu hướng giảm dần do thiếu hiểu biết, sự quen thuộc và sự tin tưởng với tín dụng công nghệ và dịch vụ ngân hàng nói chung. Về độ trung thành của khách hàng, số lượng khách hàng non-churn gấp 3 lần số lượng khách hàng churn với insight giống với nhóm tuổi 31-40.

Với tập khách hàng người cao tuổi từ 51-60 tuổi trở lên, họ không còn nằm trong tập khách hàng mục tiêu của ngân hàng với tổng khách hàng được ghi lại trong dataset dưới 2500.

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.

b) Tổng giá trị giao dịch của khách hàng

```python
#Tạo hai cột tách rời cho churn=0 và churn=1
df_churn_0 = df[df['churn'] == 0].drop(columns=['churn'])
df_churn_1 = df[df['churn'] == 1].drop(columns=['churn'])

# Tạo list tên cột
columns = ['amount_3month','amount_month','amount_week','total_amt_mar2021','total_amt_juin2021']

# Tạo list giá trị trung bình cho từng cột và từng trường hợp churn
churn_0_mean = [df_churn_0[col].mean() for col in columns]
churn_1_mean = [df_churn_1[col].mean() for col in columns]

# Tạo list chỉ số x cho các nhóm cột
x = np.arange(len(columns))

# Độ rộng mỗi cột
bar_width = 0.35
plt.figure(figsize=(10, 6))
# Vẽ biểu đồ grouped bar chart
plt.bar(x - bar_width/2, churn_0_mean, width=bar_width, label='Non-churn',color='#1d22da')
plt.bar(x + bar_width/2, churn_1_mean, width=bar_width, label='Churn',color='#d72848')
plt.xlabel('Amount Type')
plt.ylabel('Mean Amount')
plt.title('Comparison of Mean Amount between Churn=0 and Churn=1')
plt.xticks(x, [str(column) for column in columns])
plt.legend()
plt.show()
```



Đồ thị: Tổng giá trị giao dịch trong các giai đoạn

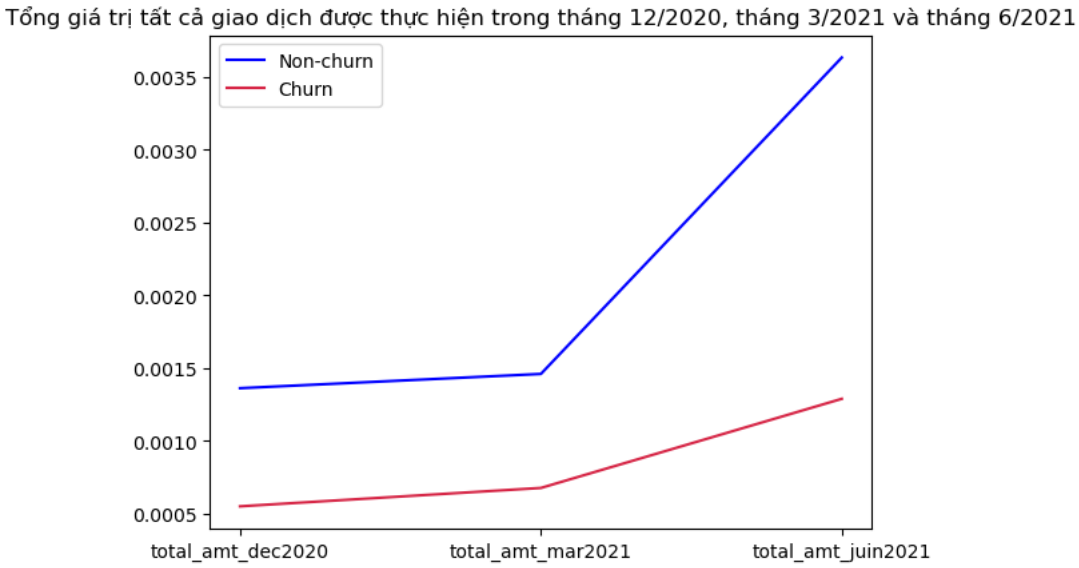
Trong biểu đồ đường thể hiện tổng giá trị tất cả giao dịch được thực hiện trong tháng, cụ thể là tháng 12/2020, tháng 3/2021 và tháng 6/2021, chúng ta có thể dễ dàng thấy được xu hướng tăng dần theo thời gian của cả 2 tập khách hàng churn và non-churn:

- Từ tháng 12/2020 đến tháng 3/2021: Biểu đồ đường của khách hàng churn và non-churn song song với nhau (hiệu số lượng khách hàng churn và non-churn gần như không thay đổi) thể hiện sự ổn định về lượng khách hàng đầu vào và đầu ra của MBBank.

- Từ tháng 3/2021 đến tháng 6/2021: Số lượng khách hàng non-churn tăng đột biến so với lượng khách hàng churn cho đến tháng 6/2021, lượng khách hàng non-churn gấp xấp xỉ 3.5 lần lượng khách hàng churn.

Kết quả tích cực này đã phần nào khách hàng được 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.

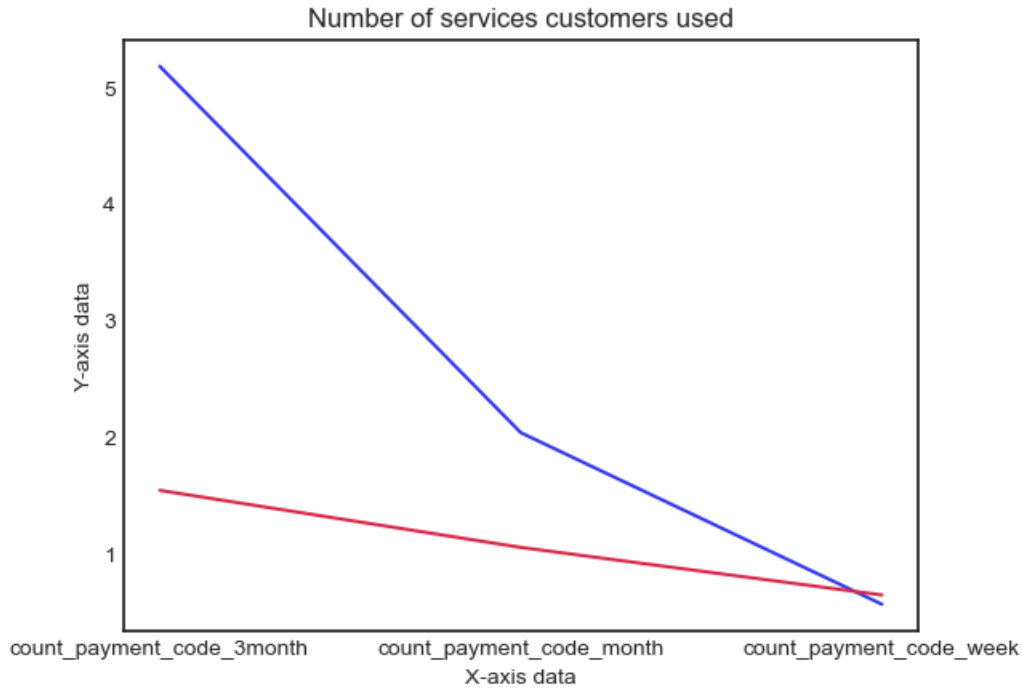
```python
#Tổng số tiền giao dịch của khách hàng trong tháng 11/2020, 3/2021 và 6/2021
df_churn_0 = df[df['churn'] == 0].drop(columns=['churn'])
df_churn_1 = df[df['churn'] == 1].drop(columns=['churn'])

columns = ['amount_month','total_amt_mar2021','total_amt_juin2021']#,'total_amt_mar2021','total_amt_juin2021']

# Tạo list giá trị trung bình cho từng cột và từng trường hợp churn
churn_0_mean = [df_churn_0[col].mean() for col in columns]
churn_1_mean = [df_churn_1[col].mean() for col in columns]

# Tạo list chỉ số x cho các nhóm cột
x = np.arange(len(columns))

plt.plot(columns,churn_0_mean,label='Non-churn',color='blue')
plt.plot(columns, churn_1_mean,label='Churn',color='#d72848')
plt.legend()
plt.title('Tổng giá trị tất cả giao dịch được thực hiện trong tháng 12/2020, tháng 3/2021 và tháng 6/2021')
custom_labels = ['total_amt_dec2020','total_amt_mar2021','total_amt_juin2021']
plt.xticks(x, custom_labels)
plt.show()
```



c) Tổng số lượng dịch vụ sử dụng

Đồ thị: Số lượng dịch vụ được sử dụng trong các giai đoạn của hai nhóm khách hàng

```python
df_churn_0 = df[df['churn'] == 0].drop(columns=['churn'])
df_churn_1 = df[df['churn'] == 1].drop(columns=['churn'])

# Tạo list tên cột
columns = ['count_payment_code_3month','count_payment_code_month','count_payment_code_week']#,'total_amt_mar2021','total_amt_juin2021']

# Tạo list giá trị trung bình cho từng cột và từng trường hợp churn
churn_0_mean = [df_churn_0[col].mean() for col in columns]
churn_1_mean = [df_churn_1[col].mean() for col in columns]

# Tạo list chỉ số x cho các nhóm cột
x = np.arange(len(columns))

plt.plot(columns,churn_0_mean,color='#3335ed' )
plt.plot(columns, churn_1_mean,color='#d72848')

plt.xlabel("X-axis data")
plt.ylabel("Y-axis data")
plt.title('Number of services customers used')
plt.show()
```



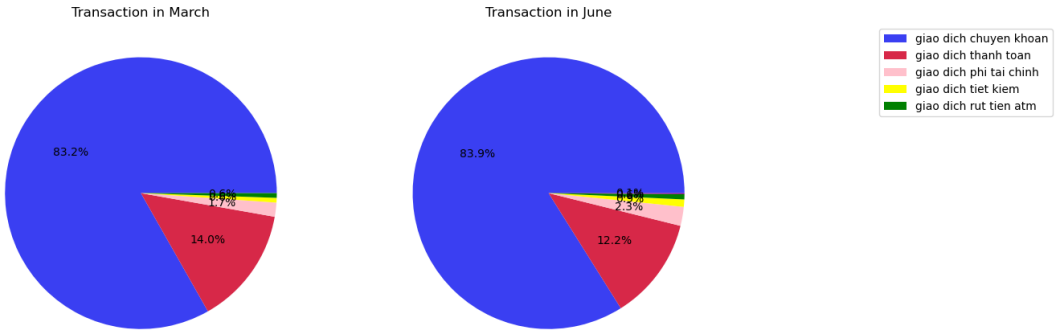
Đồ thị: Tỷ lệ dịch vụ được sử dụng trong tháng 3 và tháng 6 năm 2021:

Trong tất cả các loại giao dịch được thực hiện bởi khách hàng được ghi lại trong dataset, loại giao dịch được thực hiện nhiều nhất là Giao dịch chuyển khoản. Loại giao dịch này trong cả tháng 3/2021 và tháng 6/2021 đều chiếm phần lớn lần lượt với 83.2% và 83.9%.

Tiếp theo, loại dịch vụ được sử dụng nhiều thứ hai là Giao dịch thanh toán, nhưng cách xa về chỉ số so với Giao dịch chuyển khoản khi chỉ có 14% khách hàng sử dụng trong tháng 3/2021 và 12.2% trong tháng 6/2021.

Với các loại giao dịch con lại như Giao dịch phi tài chính, Giao dịch tiết kiệm, Giao dịch rút tiền ATM, phần trăm khách hàng sử dụng không đáng kể và xấp xỉ gần bằng nhau.

```
plt.legend(labels=transaction_counts.index,bbox_to_anchor=(2, 1), loc='upper right')
plt.figure(figsize=(15, 10))
plt.show()
```

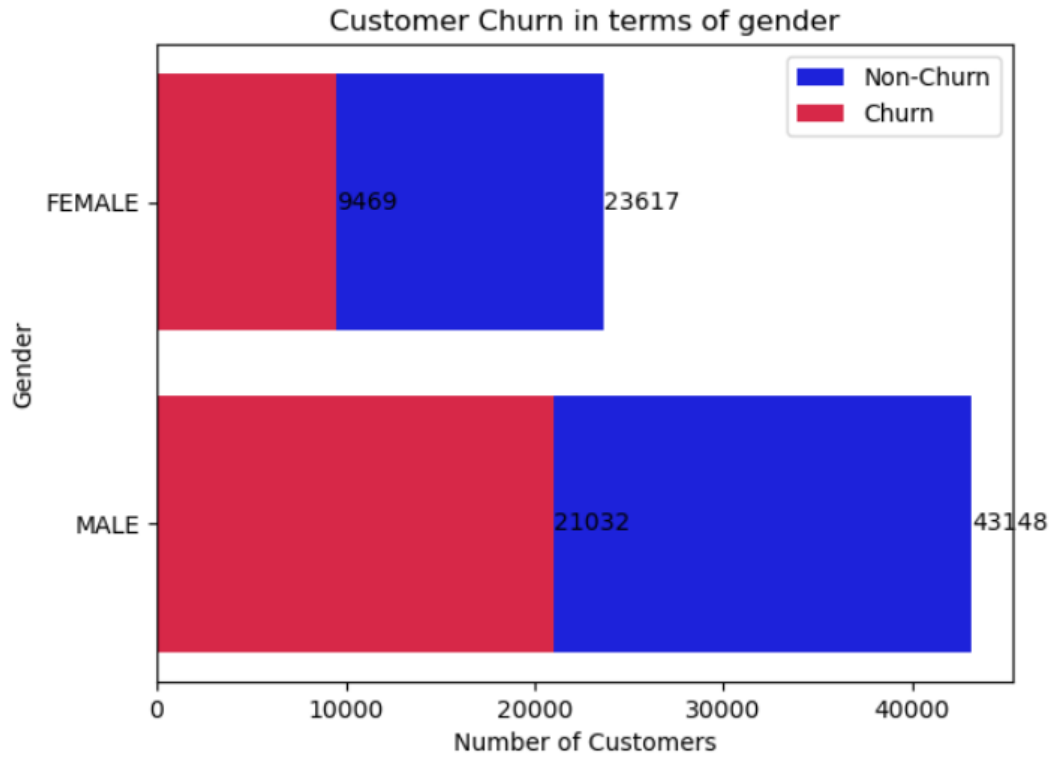


3.2.3.2. Biến phân loại

a) Giới tính khách hàng

Đồ thị: Giới tính và nhóm khách hàng

```
gender_cnt = df.local_ref_1.value_counts()
gender_0 = (df.local_ref_1[df['churn'] == 0].value_counts())
gender_1 = (df.local_ref_1[df['churn'] == 1].value_counts())

plt.subplots(figsize=(8,5))

plt.bar(gender_0.index, gender_0.values, label='Non-churn',color='#1d22da')
plt.bar(gender_1.index, gender_1.values, label='Churn',color='#d72848')
for x,y in zip(names,gender_0):
    plt.text(x,y,y,fontsize=12)
for x,y in zip(names,gender_1):
    plt.text(x,y,y,fontsize=12)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.title("Number of churn customers in each gender", fontsize=15)
plt.legend(loc='upper right', fontsize=15)
plt.show()
```



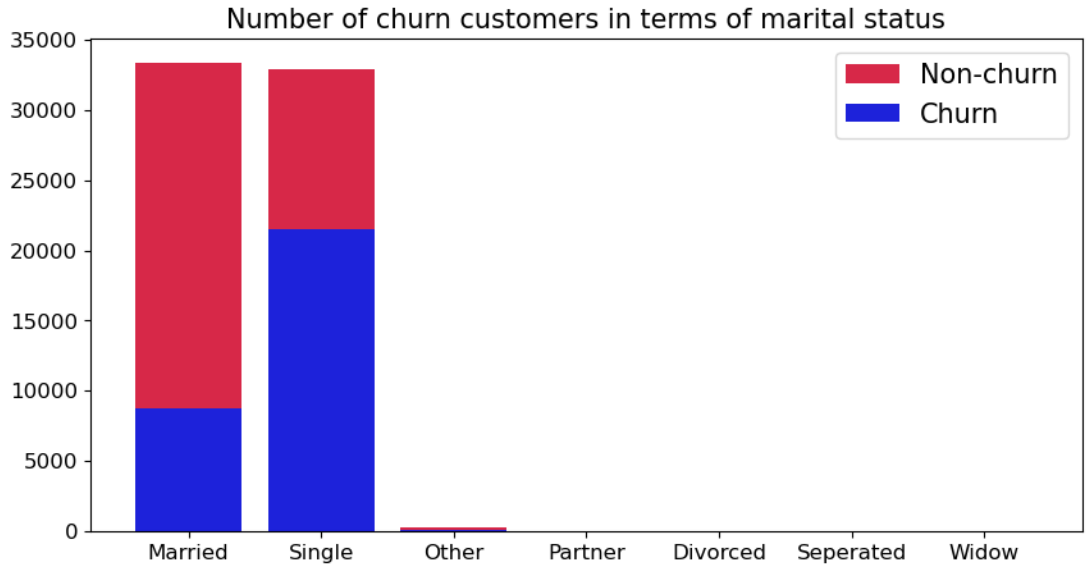
Xét về giới tính, số lượng nam giới sử dụng dịch vụ nhiều gấp đôi số lượng nữ giới. Tuy nhiên, số lượng khác hàng non-churn luôn lớn hơn churn ở cả 2 nhóm giới tính, cụ thể hơn:

- Đối với nam (Số khách hàng: 43148): Số lượng khách hàng churn và non-churn xấp xỉ bằng nhau.
- Đối với nữ (Số khách hàng: 23617): Số lượng khách hàng non-churn gấp 2.5 lần số lượng khách hàng churn.

b) Tình trạng hôn nhân

Đồ thị: Tình trạng hôn nhân và nhóm khách hàng

```
marital_status_cnt = df.vn_marital_status.value_counts()
marital_status_0 = (df.vn_marital_status[df['churn'] == 0].value_counts())
marital_status_1 = (df.vn_marital_status[df['churn'] == 1].value_counts())

plt.subplots(figsize=(10,5))

plt.bar(marital_status_0.index, marital_status_0.values, label='Non-churn',color='#d72848')
plt.bar(marital_status_1.index, marital_status_1.values,label='Churn',color='#1d22da')
#for x,y in zip(names,marital_status_0):
  #  plt.text(x,y,y,ha='center',fontsize=12)
#for x,y in zip(names,marital_status_1):
 #   plt.text(x,y,y,ha='center',fontsize=12)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.title("Number of churn customers in terms of marital status", fontsize=15)
plt.legend(loc='upper right', fontsize=15)
plt.show()
```



Xét về tình trạng hôn nhân, đa số khách hàng được ghi lại trong dataset thuộc tập khách hàng Married (Đã kết hôn) và Single (Độc thân):

Tập khách hàng Married: Số lượng khách hàng non-churn gấp 3 lần khách hàng churn do người có gia đình thường có trách nhiệm tài chính chung và mong muốn tăng cường sự ổn định tài chính với vợ/chồng trong việc quản lý các khoản chi tiêu hằng ngày, tiết kiệm và đầu tư; đồng thời do công việc ổn định hơn và tổng thu nhập của người đã kết hôn và vợ/chồng của họ khả năng cao sẽ lớn hơn người có tình trạng hôn nhân khác.

Tập khách hàng Single: Số lượng khách hàng churn gấp hơn 2 lần khách hàng non-churn do tập khách hàng này đa số sẽ nằm trong độ tuổi dưới 30 (độ tuổi đang trong quá trình học chương trình giáo dục cơ bản và thu nhập chưa nhiều để hình thành nhu cầu sử dịch vụ ngân hàng) và chưa có bạn đồng hành để có trách nhiệm cao trong việc quản lí tài chính.

3.2.4. Xử lý và sắp xếp dữ liệu

3.2.4.1. Làm sạch dữ liệu

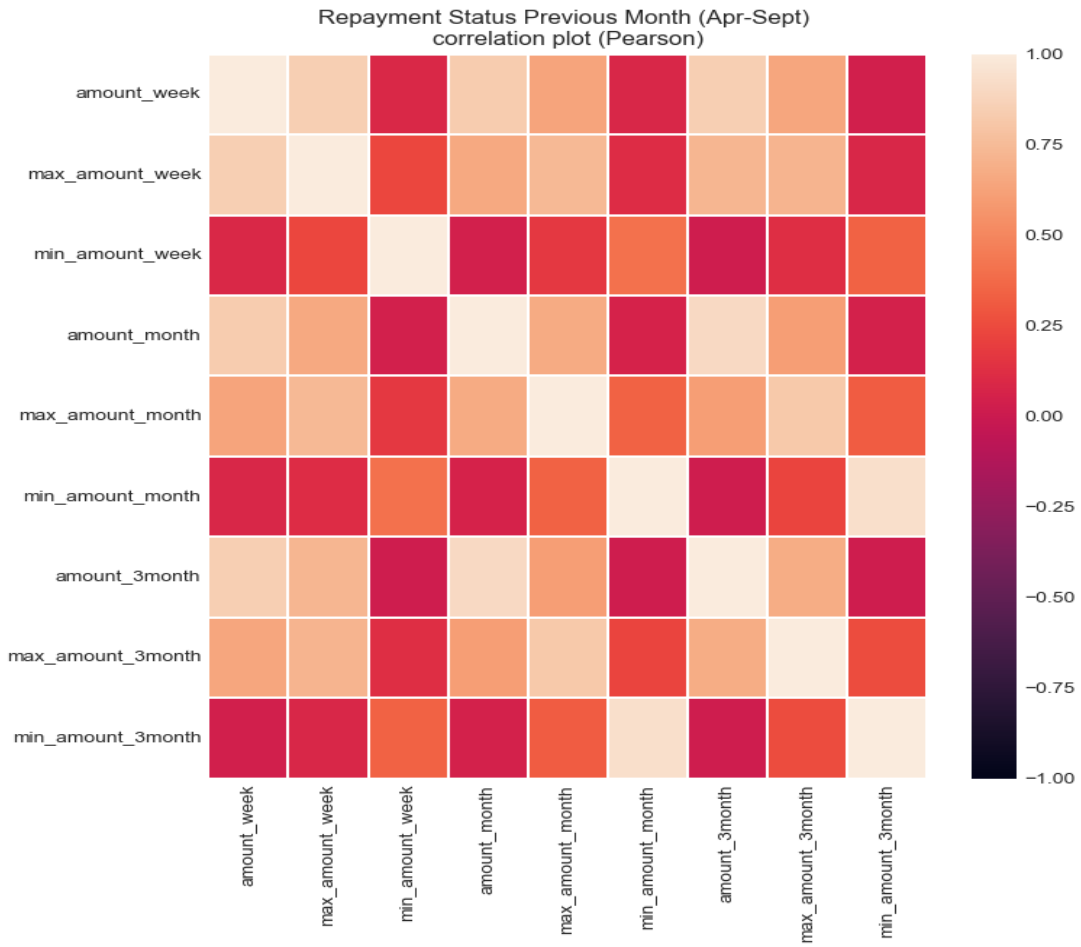
Đồ thị: Tương quan các biến giá trị giao dịch

Xét mối tương quan giữa các biến:

+ 'amount_week',
+ 'amount_month',
+ 'amount_3month',
+ 'max_amount_week',
+ 'min_amount_week',
+ 'max_amount_month',
+ 'min_amount_month',
+ 'max_amount_3month',
+ 'min_amount_3month'

Sau đó, vẽ heatmap thể hiện ma trận tương quan của các biến.

```
var = ['amount_week',
       'max_amount_week', 'min_amount_week',
       'amount_month', 'max_amount_month',
       'min_amount_month',
       'amount_3month', 'max_amount_3month',
       'min_amount_3month'
       ]

plt.figure(figsize = (8,8))
plt.title('Repayment Status Previous Month (Apr-Sept) \ncorrelation plot (Pearson)')
corr = df[var].corr()
sns.heatmap(corr,xticklabels=corr.columns,yticklabels=corr.columns,linewidths=.1,vmin=-1, vmax=1)
plt.show()
```



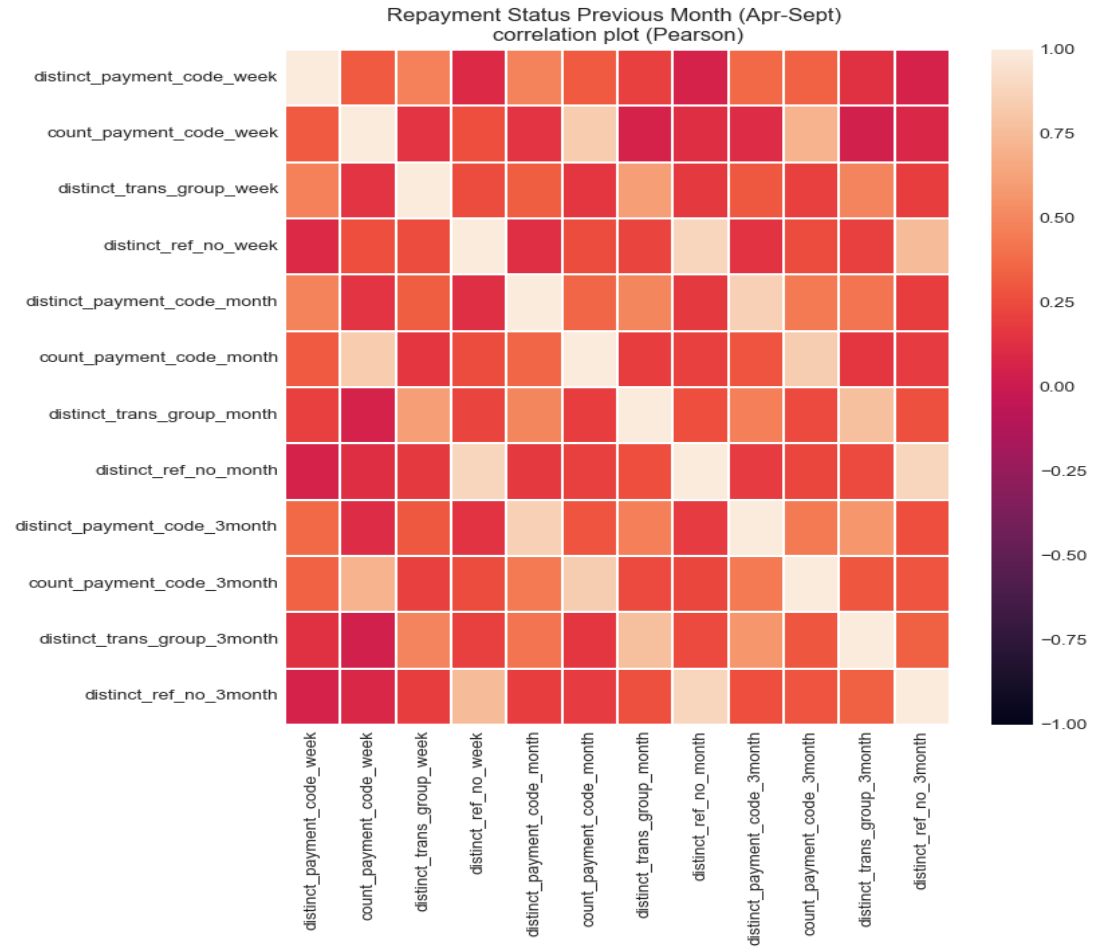
Đồ thị: Tương quan các biến dịch vụ ngân hàng

+ 'distinct_payment_code_week',
+ 'count_payment_code_week',
+ 'distinct_trans_group_week',
+ 'distinct_ref_no_week',
+ 'distinct_payment_code_month',
+ 'count_payment_code_month',

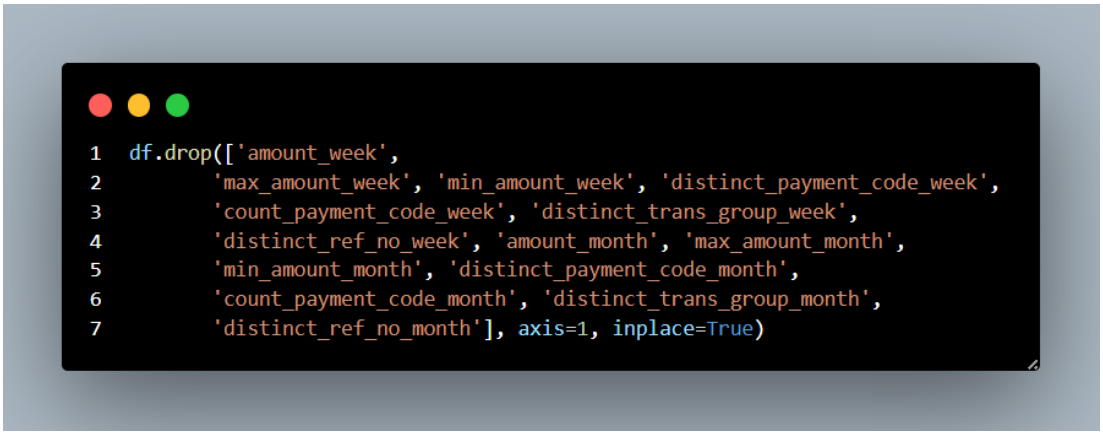
+ 'distinct_trans_group_month',
+ 'distinct_ref_no_month',
+ 'distinct_payment_code_3month',
+ 'count_payment_code_3month',
+ 'distinct_trans_group_3month',
+ 'distinct_ref_no_3month'

Sau đó, vẽ heatmap thể hiện ma trận tương quan của các biến.

```
var = ['distinct_payment_code_week',
       'count_payment_code_week', 'distinct_trans_group_week',
       'distinct_ref_no_week', 'distinct_payment_code_month',
       'count_payment_code_month', 'distinct_trans_group_month',
       'distinct_ref_no_month', 'distinct_payment_code_3month',
       'count_payment_code_3month', 'distinct_trans_group_3month',
       'distinct_ref_no_3month'
       ]

plt.figure(figsize = (8,8))
plt.title('Repayment Status Previous Month (Apr-Sept) \ncorrelation plot (Pearson)')
corr = df[var].corr()
sns.heatmap(corr,xticklabels=corr.columns,yticklabels=corr.columns,linewidths=.1,vmin=-1, vmax=1)
plt.show()
```



Có thể thấy rằng các nhóm chỉ số theo Tuần, theo Tháng, theo 3 Tháng có tương quan với nhau rất cao. Tuy nhiên, tỷ lệ missing value ở nhóm các chỉ số theo 3 Tháng là ít nhất.
Vậy, 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.



```
df.drop(['amount_week',
         'max_amount_week', 'min_amount_week', 'distinct_payment_code_week',
         'count_payment_code_week', 'distinct_trans_group_week',
         'distinct_ref_no_week', 'amount_month', 'max_amount_month',
         'min_amount_month', 'distinct_payment_code_month',
         'count_payment_code_month', 'distinct_trans_group_month',
         'distinct_ref_no_month'], axis=1, inplace=True)
```

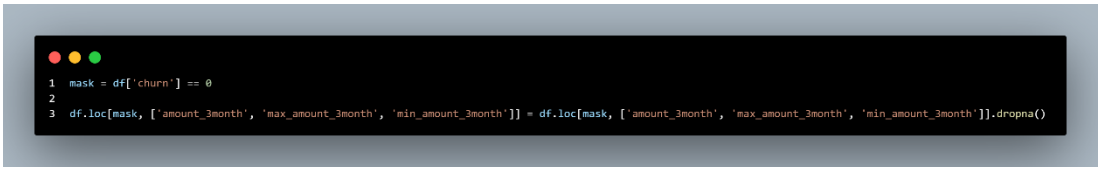
3.2.4.2. Xử lý giá trị thiếu

Điền các giá trị thiếu bằng 0 trong các cột 'amount_3month', 'max_amount_3month', 'min_amount_3month' của các hàng có giá trị 'churn' bằng 1 trong DataFrame.
+ Đầu tiên mask = df['churn'] == 1 tạo ra một mặt nạ (mask) có tên là 'mask' để chỉ định các hàng trong DataFrame df có giá trị 'churn' bằng 1.
+ Dựa trên mặt nạ 'mask' và danh sách các cột 'amount_3month', 'max_amount_3month', 'min_amount_3month'. Điều này sẽ trả về một DataFrame con gồm các hàng có giá trị 'churn' bằng 1 và chỉ bao gồm các cột được liệt kê.
+ Dòng cuối cùng .fillna(0) áp dụng phương thức fillna() để điền các giá trị thiếu trong DataFrame con trên bằng 0. Điều này sẽ thay thế các giá trị NaN (thiếu) trong các cột 'amount_3month', 'max_amount_3month', 'min_amount_3month' của các hàng có giá trị 'churn' bằng 1 bằng 0.

```
mask = df['churn'] == 1

df.loc[mask, ['amount_3month', 'max_amount_3month', 'min_amount_3month']] = df.loc[mask, ['amount_3month', 'max_amount_3month', 'min_amount_3month']].fillna(0)
```

Tương tự, loại bỏ các hàng chứa giá trị thiếu (NaN) trong các cột 'amount_3month', 'max_amount_3month', 'min_amount_3month' của các hàng có giá trị 'churn' bằng 0.
+ Đầu tiên, mask = df['churn'] == 0 tạo ra một mask có tên là 'mask' để chỉ định các hàng trong DataFrame df có giá trị 'churn' bằng 0.
+ Tiếp theo, truy cập vào DataFrame dựa trên mặt nạ 'mask' và danh sách các cột 'amount_3month', 'max_amount_3month', 'min_amount_3month'. Điều này sẽ trả về một DataFrame con gồm các hàng có giá trị 'churn' bằng 0 và chỉ bao gồm các cột được liệt kê.

+ Cuối cùng, loại bỏ các hàng chứa giá trị thiếu (NaN) trong DataFrame con. Điều này sẽ loại bỏ các hàng có giá trị thiếu trong các cột 'amount_3month', 'max_amount_3month', 'min_amount_3month' của các hàng có giá trị 'churn' bằng 0.



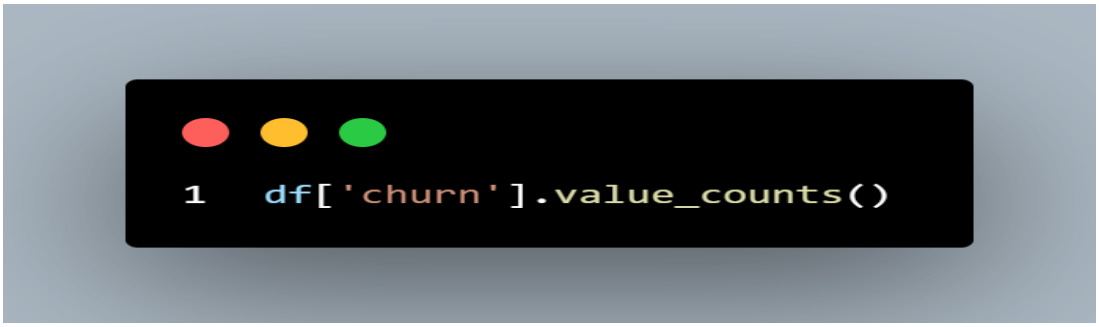
```
mask = df['churn'] == 0

df.loc[mask, ['amount_3month', 'max_amount_3month', 'min_amount_3month']] = df.loc[mask, ['amount_3month', 'max_amount_3month', 'min_amount_3month']].dropna()
```

Sau khi đã thực hiện các thay đổi, DataFrame còn 96918 hàng và 28 cột.

Sau đó, các cột không sử dụng cho xây dựng mô hình được loại bỏ.

```
df.drop(['most_act_mar2021_count', 'total_act_mar2021', 'total_amt_mar2021',
         'most_act_juin2021_count', 'total_act_juin2021', 'total_amt_juin2021'], axis=1, inplace=True)
```

Đếm số lượng các giá trị trong cột 'churn' trong DataFrame và hiển thị kết quả theo số lần xuất hiện của từng giá trị.



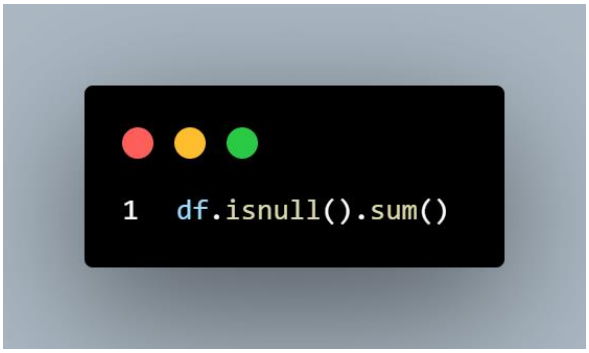
```
df['churn'].value_counts()
```

Kết quả co thấy:

0	66501
1	30417
Name: churn	dtype: int64

Điều này nghĩa là, hiện trong DataFrame có 30417 khách hàng rời bỏ và 66501 khách hàng không rời bỏ.

Đếm lại số lượng giá trị thiếu (NaN) trong từng cột của DataFrame:

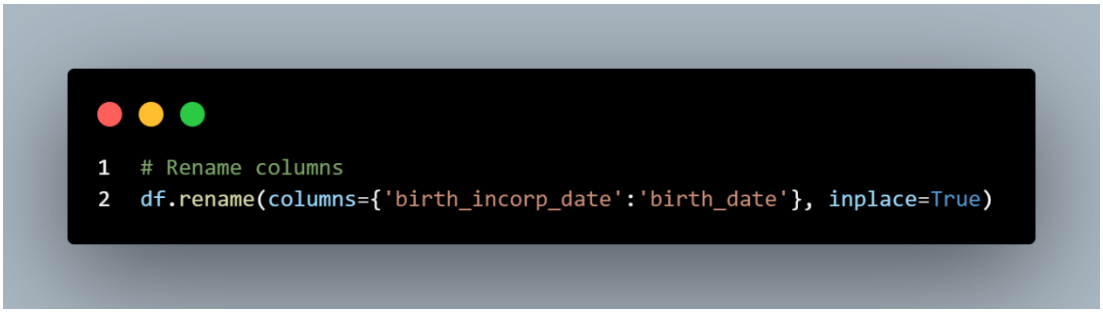


```
df.isnull().sum()
```

```
sex                              0
marital_status                   0
birth_incorp_date                4
amount_3month                    0
max_amount_3month                0
min_amount_3month                0
distinct_payment_code_3month     0
count_payment_code_3month        0
distinct_trans_group_3month      0
distinct_ref_no_3month           0
id                               0
churn                            0
dtype: int64
```

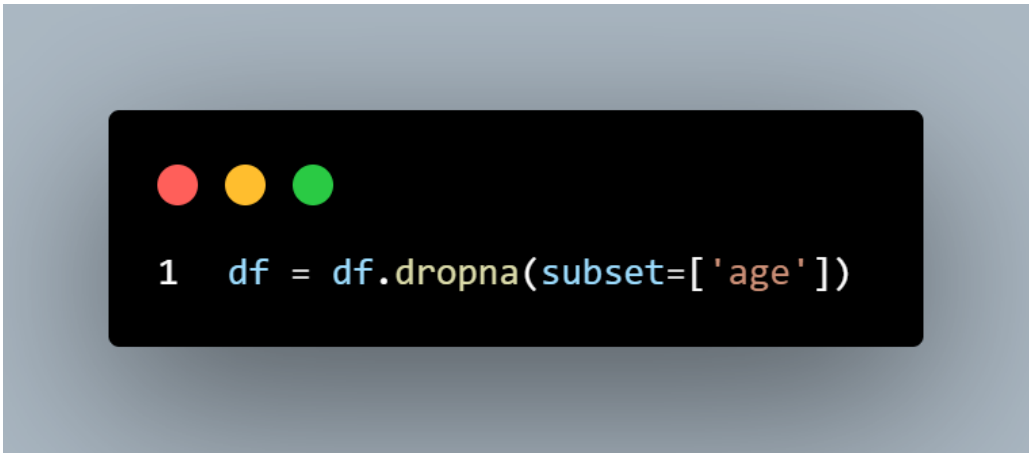
Sau khi kiểm tra, ta thấy các cột cần dùng để xây dựng mô hình về cơ bản đã không có missing values, chỉ còn cột 'birth_incorp_date' có 4 missing values, ta tiếp tục xử lý.

Đổi tên cột 'birth_incorp_date' thành 'birth_date'.



```
# Rename columns
df.rename(columns={'birth_incorp_date':'birth_date'}, inplace=True)
```

Tính tuổi khách hàng và ghép vào Data Frame. Cuối cùng, thực hiện bỏ hết giá trị thiếu ở cột này.



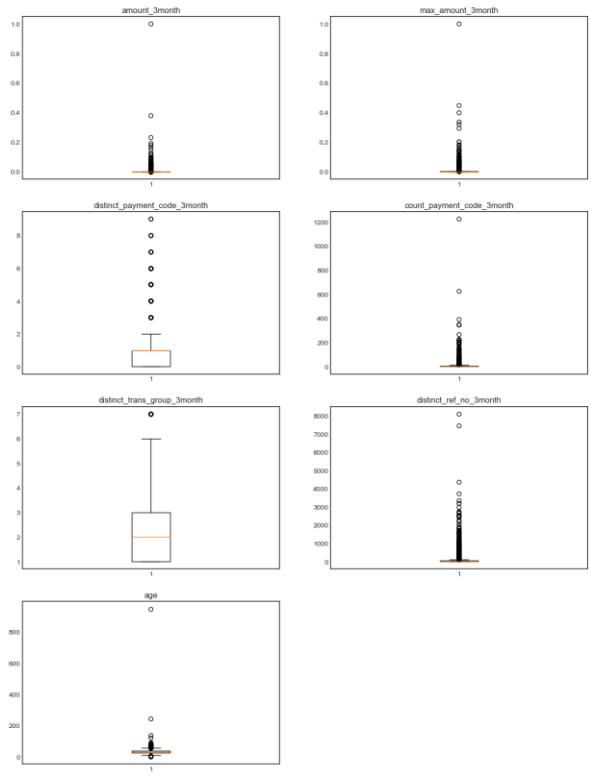
```
df = df.dropna(subset=['age'])
```

Đến đây, cơ bản đã hoàn thành việc xử lý việc thiếu dữ liệu trong các biến phục vụ cho mô hình.

3.2.4.3. Xử lý giá trị ngoại lai

Đồ thị thể hiện phân phối, giá trị trung vị và giá trị ngoại lai của dữ liệu.

```python
plt.figure(figsize = (15, 20))
plt.style.use('seaborn-white')

ax=plt.subplot(421)
plt.boxplot(df['amount_3month'])
ax.set_title('amount_3month')

ax=plt.subplot(422)
plt.boxplot(df['max_amount_3month'])
ax.set_title('max_amount_3month')

ax=plt.subplot(423)
plt.boxplot(df['distinct_payment_code_3month'])
ax.set_title('distinct_payment_code_3month')

ax=plt.subplot(424)
plt.boxplot(df['count_payment_code_3month'])
ax.set_title('count_payment_code_3month')

ax=plt.subplot(425)
plt.boxplot(df['distinct_trans_group_3month'])
ax.set_title('distinct_trans_group_3month')

ax=plt.subplot(426)
plt.boxplot(df['distinct_ref_no_3month'])
ax.set_title('distinct_ref_no_3month')

ax=plt.subplot(427)
plt.boxplot(df['age'])
ax.set_title('age')

plt.suptitle('Data frequency of each attributes')
```

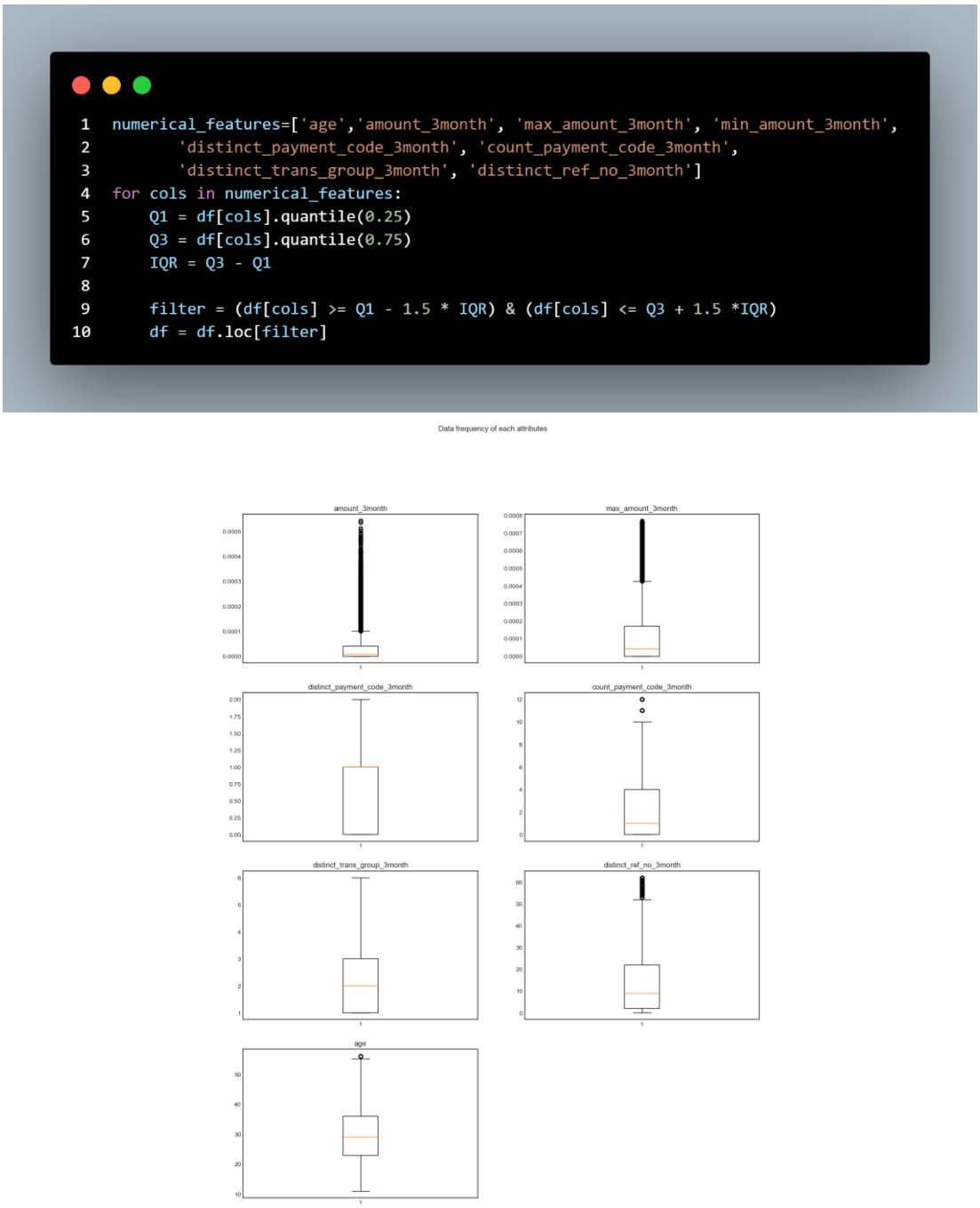


Đồ thị cho thấy xuất hiện các giá trị ngoại lai trong các trường của DataFrame. Các giá trị ngoại lai này có thể là kết quả của lỗi trong quá trình thu thập hoặc đo lường dữ liệu hoặc do một sự kiện đặc biệt nào đó tạo nên. Nhận thấy các giá trị ngoại lai có thể ảnh hưởng đến phân phối dữ liệu và dẫn đến thiếu sót trong mô hình. Do đó, nhóm tác giả quyết định xóa bỏ các giá trị ngoại lai dựa trên phương pháp IQR (Interquartile Range) để đảm bảo tính đáng tin cậy của mô hình.

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

- Trước hết, ta xác định các khoảng phần tư thứ nhất Q1 và thứ ba Q3. IQR bằng Q3-Q1.
- Ta thực hiện lấy các giá trị lớn hơn và bằng Q1 - 1.5IQR và các giá trị nhỏ hơn và bằng Q3 + 1.5IQR.

```python
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]
```

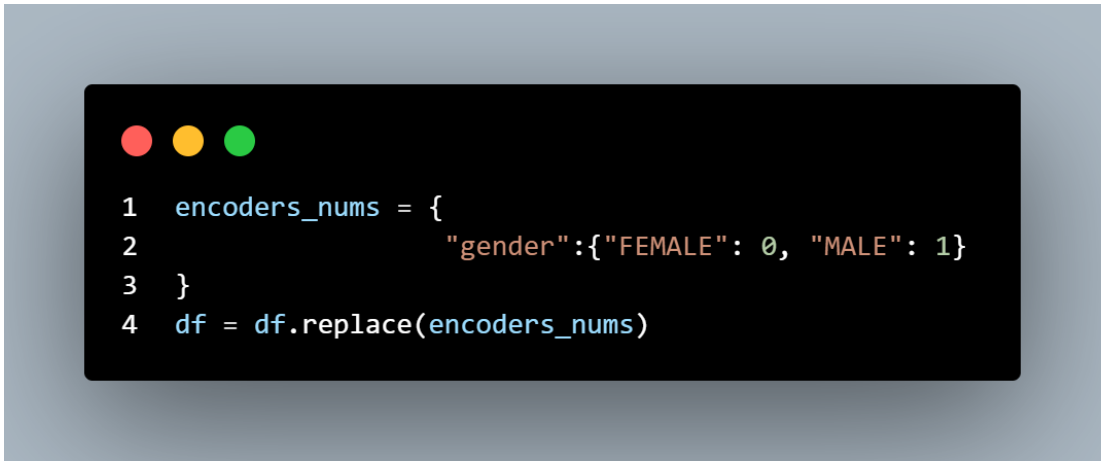


Sau đó, kiểm tra lại dữ liệu sau khi lọc, kết quả cho thấy rằng dữ liệu không còn có giá trị ngoại lai và đã cân bằng hơn giữa hai nhóm khách hàng rời bỏ và không rời bỏ.

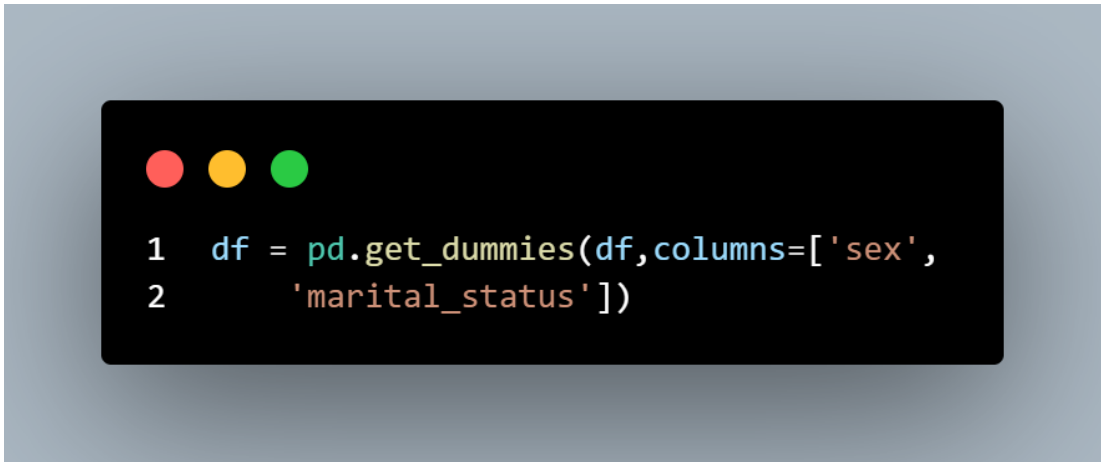
0	29090
1	22490
Name: churn	dtype: int64

3.2.4.4. Mã hóa dữ liệu

Mã hóa nhãn cho cột ‘sex’ bằng cách sử dụng một từ điển có tên là "encoders_nums", các giá trị chuỗi "FEMALE" được quy thành 0 và "MALE" được quy thành 1. Tạo ra các biến giả cho cột ‘sex’ và ‘marital_status’ với giá trị 0 và 1.



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



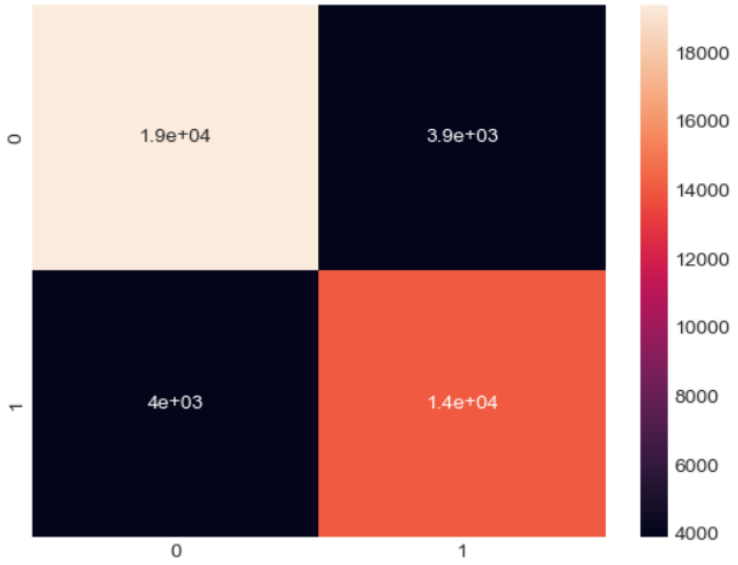
```
df = pd.get_dummies(df,columns=['sex',
     'marital_status'])
```

3.2.5. Phát triển và xây dựng mô hình (Model Development)

Logistic Regression

Confusion Matri

Hình vẽ cho thấy: Số lượng khách hàng churn và non-churn được dự báo đúng nhiều hơn hẳn số lượng các dự báo có kết quả sai



Cross-Validation Score

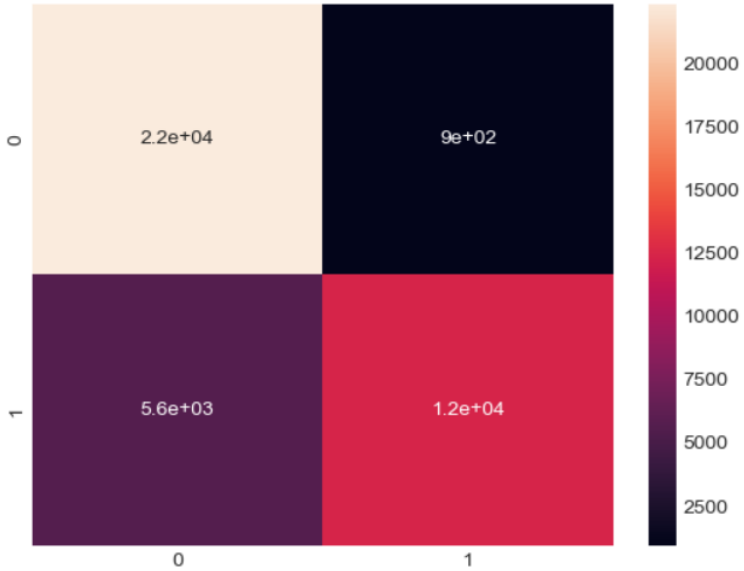
The mean accuracy of the folds	0.82 +- 0.003
The mean precision of the folds	0.82 +- 0.004
The mean recall of the folds	0.75 +- 0.005
The mean f1 of the folds	0.79 +- 0.004
The mean roc_auc of the folds	0.88 +- 0.003

Kết quả đánh giá cho thấy mô hình đang học khá tốt, đưa ra hiệu quả cao

Support Vector Classifier

Confusion Matrix

Hình vẽ cho thấy: Số lượng khách hàng churn và non-churn được dự báo đúng nhiều hơn hẳn số lượng các dự báo có kết quả sai.



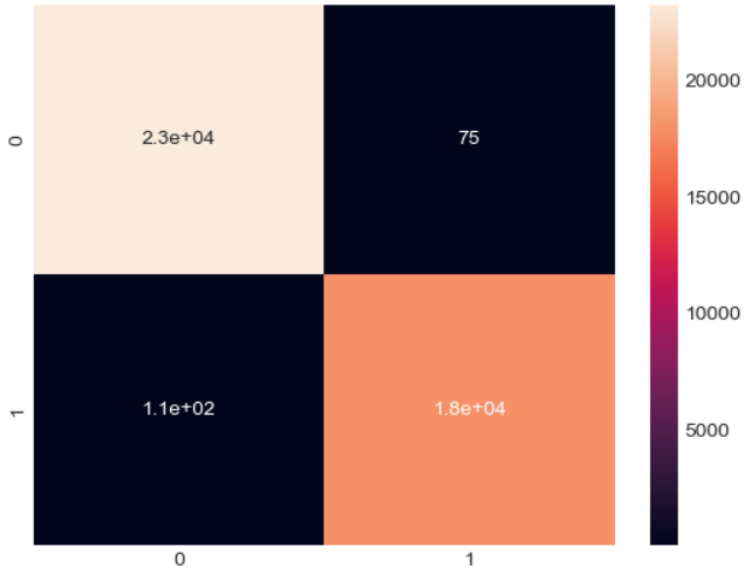
Cross-Validation Score

The mean accuracy of the folds	0.84 +- 0.003
The mean precision of the folds	0.93 +- 0.005
The mean recall of the folds	0.69 +- 0.006
The mean f1 of the folds	0.79 +- 0.004
The mean roc_auc of the folds	0.87 +- 0.003

Decision Tree Classifier

Confusion Matrix

Hình vẽ cho thấy: Số lượng khách hàng churn và non-churn được dự báo đúng nhiều hơn hẳn số lượng các dự báo có kết quả sai



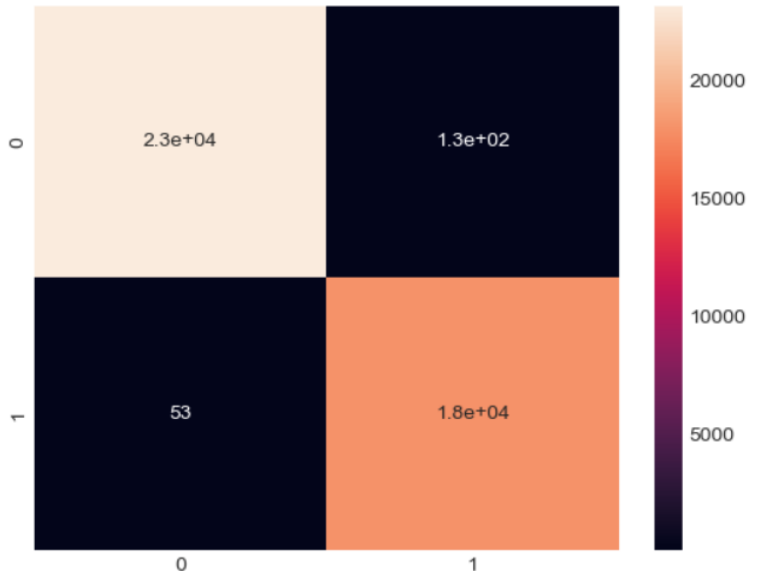
Cross-Validation Score

The mean accuracy of the folds	0.81 +- 0.001
The mean precision of the folds	0.77 +- 0.0009
The mean recall of the folds	0.79 +- 0.002
The mean f1 of the folds	0.78 +- 0.0005
The mean roc_auc of the folds	0.81 +- 0.0003

Random Forest Classifier

Confusion Matrix

Hình vẽ cho thấy: Số lượng khách hàng churn và non-churn được dự báo đúng nhiều hơn hẳn số lượng các dự báo có kết quả sai



Cross-Validation Score

The mean accuracy of the folds	0.86+- 0.001
The mean precision of the folds	0.92 +- 0.004
The mean recall of the folds	0.76 +- 0.001
The mean f1 of the folds	0.83 +- 0.001
The mean roc_auc of the folds	0.91 +- 0.0007

3.2.6. Đánh giá và lựa chọn mô hình (Model Evaluation and Selection)

Đánh giá các mô hình dựa trên các metric: accuracy, precision, recall, F1 và Roc_auc

	Classifier	Accuracy score	Precision Score	Recall Score	F1 Score	ROC_AUC score
0	Logistics Regression	0.820061	0.818581	0.754558	0.785256	0.880793
1	SVC	0.841508	0.930954	0.687528	0.790911	0.872182
2	Decision Tree	0.808162	0.768098	0.796521	0.790911	0.808823
3	Random Forest	0.869596	0.921789	0.762283	0.835723	0.915574

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

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

Vì một mô hình hiệu quả là mô hình không bỏ sót lượt 'churn' nào. Việc phân loại những khách hàng 'non-churn' là những khách hàng 'churn', hơn là một mô hình không phân loại những khách hàng non-churn là những người churn, và bỏ sót rất nhiều khách hàng churn. Nói cách khác, việc phân loại sai khách hàng không rời bỏ tốt hơn là khi phân loại khách hàng rời bỏ.

Mô hình Decision Tree có recall score cao nhất => Lựa chọn mô hình Decision Tree để tinh chỉnh.

3.2.7. Tìm siêu tham số (Model Tuning)

Nhóm lựa chọn mô hình có kết quả tốt nhất là **Decision Tree** để tinh chỉnh

Các tìm hyper-parameter cho những parameter sau:

+ criterion: gini và entropy

+ max_depth: None, 5, 10, 15

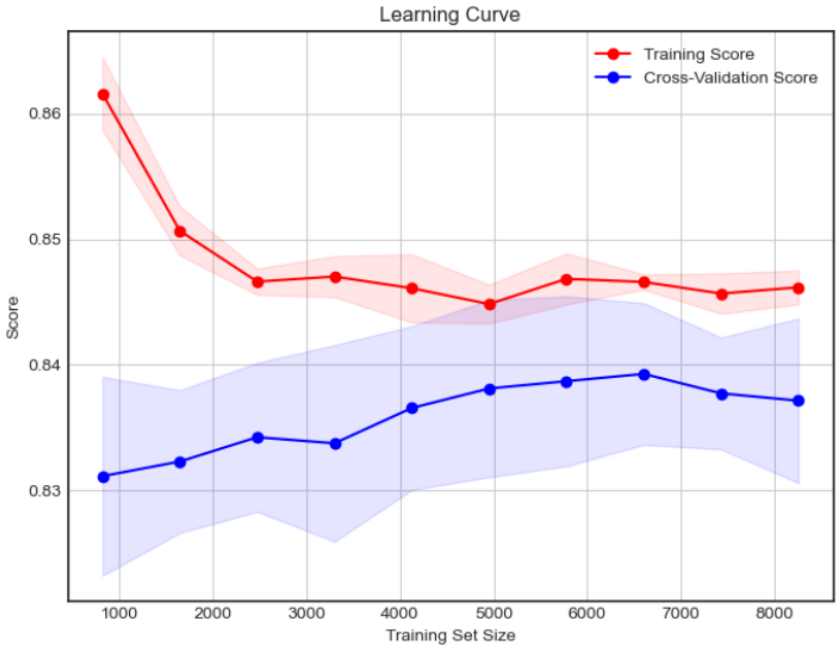
+min_samples_split: 2, 5, 10

+ min_samples_leaf: 1, 2, 4

Áp dụng GridSearchCV để tìm hyperparameter. Kết quả thu được khi áp dụng lại mô hình Decision Tree với hyper-parameter có accuracy tăng lên 0.03% => mô hình có hiệu quả hơn

The accuracy on train data after tunning	0.845070763861962
The accuracy on test data after tunning	0.8422838309422257

Learning Curve



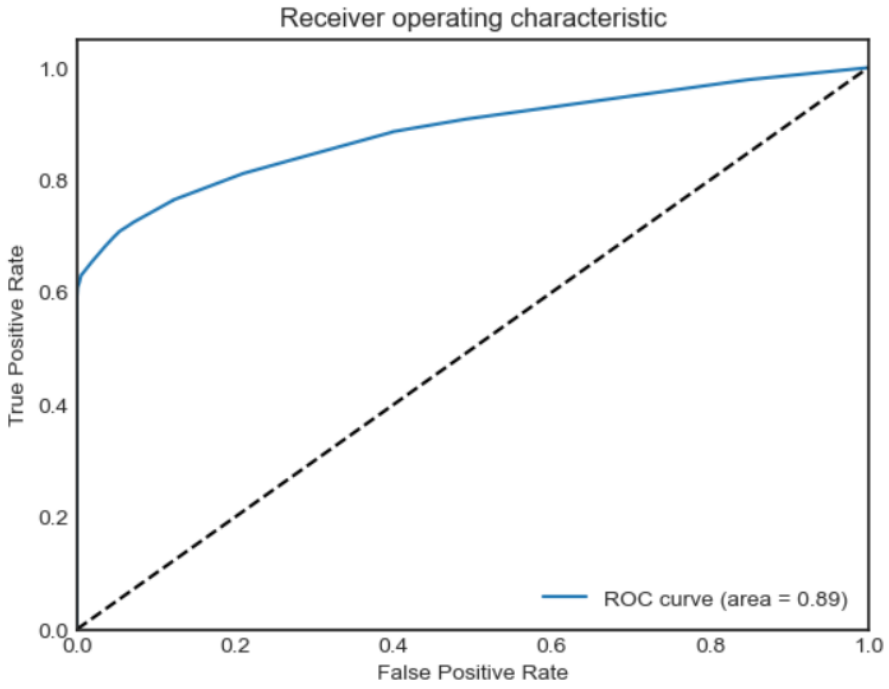
Nhìn chung, cả Trainning Score và Cross-Validation Score không quá cao (nằm trong khoảng 0.86 đến 0.83) nhưng cũng không có khác biệt quả lớn nữa hai chỉ số này.

- Mô hình có xu hướng học tốt khi dữ liệu nằm trong khoảng 1000 đến 5000 Trainning Set Size.

- Tuy nhiên, từ điểm 5000 Trainning Set Size trở đi, mô hình bắt đầu không hội tụ, thậm chí có xu hướng tách nhau ở gần điểm 6000 Trainning Set Size. Nhưng sau đấy đã có xu hướng hội tụ hơn.

ROC AUC Curve

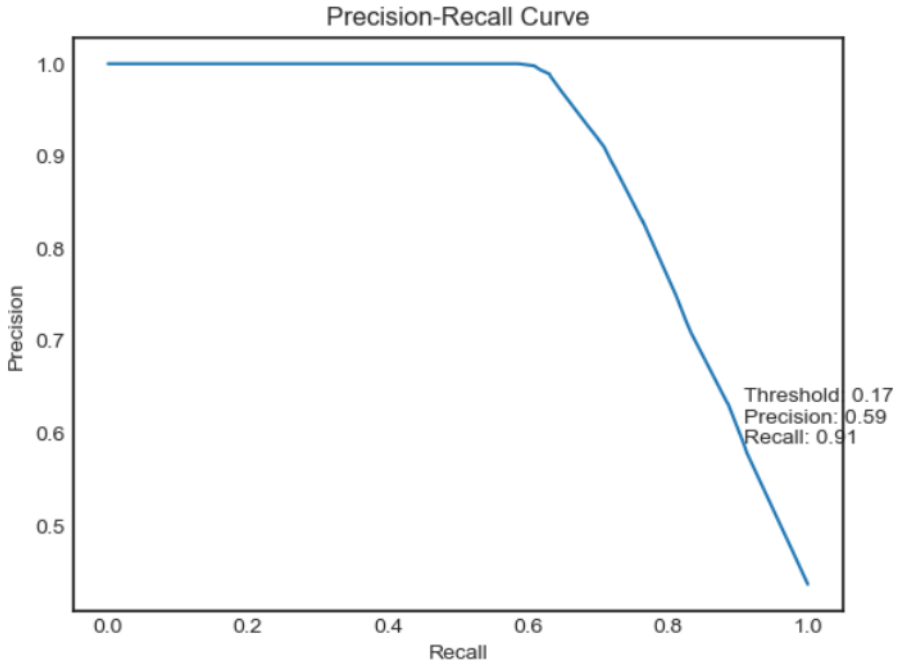
ROC AUC của mô hình đạt **89%** (lớn hơn mức thấp nhất cho phép là 50%)



Precision Recall Curve

Với **'threshold_index'** bằng 3, nói cách khác, nhóm đang chọn điểm mà Recall đạt 0.91 và Precision đạt 0.59.

Nhóm chọn điểm Recall đạt cao nhất (vì lựa chọn metric recall để đánh giá mô hình) và precision không dưới 50%.



4. KIẾN NGHỊ VÀ ĐỀ XUẤT

4.1. Kiến về ứng dụng trong hoạt động kinh doanh của ngân hàng

Việc xác định khách hàng rời bỏ hay không rời bỏ là bước quan trọng trong việc nâng cao hiệu quả hoạt động của ngân hàng. Dự đoán được khách hàng rời bỏ mang lại những lợi ích như sau:

Xác định khách hàng rời bỏ giúp các doanh nghiệp hiểu được nguyên nhân và lý do tại sao khách hàng quyết định chấm dứt mối quan hệ với công ty. Điều này cho phép tổ chức nắm bắt thông tin quan trọng về sự hài lòng của khách hàng, vấn đề về sản phẩm hoặc dịch vụ, hoặc các yếu tố khác có thể gây ra việc rời bỏ.

Xác định khách hàng rời bỏ sớm giúp tổ chức có thời gian và cơ hội để thực hiện các biện pháp hợp lý để giữ chân khách hàng. Điều này có thể bao gồm việc cung cấp các ưu đãi đặc biệt, cải thiện dịch vụ hoặc tương tác khách hàng, hoặc thực hiện các biện pháp khác để tăng sự hài lòng và trung thành của khách hàng.

Xác định và giữ chân khách hàng hiện tại có thể hiệu quả hơn so với việc tiếp tục tìm kiếm và thu hút khách hàng mới. Việc tiếp thị lại với khách hàng hiện có thường ít tốn kém hơn và có khả năng tạo ra doanh thu cao hơn. Bằng cách xác định khách hàng rời bỏ sớm, tổ chức có thể triển khai các chiến lược tiếp thị lại mục tiêu để giữ chân và tái kích hoạt khách hàng.

Xác định khách hàng rời bỏ cung cấp thông tin quan trọng để đánh giá hiệu quả của các chiến dịch tiếp thị, chất lượng sản phẩm hoặc dịch vụ, và chiến lược kinh doanh tổng thể. Nếu tỷ lệ rời bỏ cao, có thể cần xem xét và điều chỉnh chiến lược để cải thiện mối quan hệ và giữ chân khách hàng.

Với mô hình mà nhóm đề ra, hệ thống có thể dựa vào dữ liệu của 3 tháng cuối năm và dự báo khách hàng liệu có rời bỏ hay không vào thời gian là tháng 6 năm sau. Trên cơ sở kết quả đó có chiến lược giữ chân khách hàng, hoặc chăm sóc tốt hơn với các khách hàng tiềm năng.

4.2. Kiến nghị về mô hình dữ liệu và thu nhập dữ liệu

Trong quá trình thực hiện EDA, nhóm nhận thấy một số trường có tác động tới việc phân loại khách hàng có đáng tin cậy hay không như tình trạng hôn nhân của khách hàng, tuy nhiên chưa được khảo sát đầy đủ. Dữ liệu về năm sinh còn thiếu chính xác dẫn đến việc xuất hiện các outliers. Như vậy cần thực hiện khảo sát nhân khẩu học của khách hàng cẩn thận và chu đáo hơn.

Các quan sát về giao dịch, dịch vụ trong mô hình nên được quan sát tại nhiều thời điểm hơn và dài hạn hơn để có thể xác định khách hàng rời bỏ hay không một cách chính xác nhất. Thay vì chỉ có 2 thời điểm quan sát trong khoảng thời gian trong vòng 6 tháng để xác định khách hàng có rời bỏ hay không.

5. KẾT LUẬN

Tóm lại, bài phân tích sử dụng dữ liệu sử dụng dịch vụ ngân hàng của khách hàng tại một ngân hàng thương mại tại Việt Nam nhằm phân tích dữ liệu và xây dựng mô hình học máy để dự đoán khả năng rời bỏ (không còn sử dụng dịch vụ) của mỗi khách hàng sau 6 tháng sử dụng.

Bài phân tích xây dựng chỉ số rời bỏ bằng dữ liệu sử dụng dịch vụ của khách hàng vào các giai đoạn (3 tháng trước năm 2021, tháng 3 năm 2021 và tháng 6 năm 2021). 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. Nhóm đã xây dựng được chỉ số rời bỏ hoặc không 'Churn' khá chính xác.

Sau đó, phân tích dữ liệu cho thấy một số kết quả đáng chú ý, bao gồm:

– 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.

Cuối cùng, xây dựng mô hình, bốn mô hình được lựa chọn, bao gồm: 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 là 0.04%. TTuy nhiên, Recall Score tốt nhất đạt được lên tới 91%, với Precision Score đạt 59% khi lưa chọn Threshold Index là 3.

Do sự hạn chế về mặt kiến thức và thời gian nên bài nghiên cứu không thể tránh khỏi những thiếu sót nhất định. Nhóm nghiên cứu mong nhận được sự đóng góp ý kiến của các thầy cô và các chuyên gia để có thể có được những cải thiện tốt hơn

PHỤ LỤC

Nhóm chúng em trong thời gian qua đã thực hiện bài tập nhóm cuối khóa xây dựng mô hình dự đoán "khách hàng rời bỏ" thông qua dữ liệu về khách hàng sử dụng dịch vụ ngân hàng. Khi thực hiện bài tập, nhóm đã gặp các khó khăn trong công việc thực hiện các nhiệm vụ đề ra. Trước hết, nhóm đã gặp khó khăn trong việc xử lý dữ liệu. Dữ liệu được cung cấp có một số trở ngại như chất lượng của dữ liệu, một số nhóm dữ liệu đã bị mã hóa vì lý do bảo mật nên rất khó để nhóm có thể hiểu tường tận về bộ dữ liệu. Đồng thời, việc xây dựng định nghĩa về "Khách hàng rời bỏ" của nhóm gặp khó khăn khi phải xác định mục tiêu cần thiết, các dữ liệu phục vụ cho quá trình tính toán phù hợp và đảm bảo tính logic cho thiết lập định nghĩa của bài. Nhóm đã tốn nhiều thời gian và công sức để giải quyết các khó khăn trên. Nhờ vậy, bài tập của nhóm đã có một số kết quả nhất định.

Nhìn chung, nhóm đã có một khoảng thời gian trải nghiệm quý báu và bổ ích khi được thực hành với một bộ dữ liệu có tính thực tiễn cao. Bài tập nhóm này đã giúp các thành viên trong nhóm phát triển không chỉ các kĩ năng kĩ thuật mà còn phát triển các kĩ năng mềm như làm việc nhóm, giao tiếp và nâng cao kĩ năng cá nhân khác. Nhóm chúng em có thể nâng cao hơn các kĩ năng này thông qua các buổi họp chi tiết và có kế hoạch rõ ràng hơn trong các bài tập hoặc dự án tương lai.

Trong tương lai, các thành viên trong nhóm sẽ có thể áp dụng được các kĩ năng học trong thời gian thực hiện bài tập nhóm để có thể nâng cao hiệu suất làm việc trong các dự án tương lai. Việc áp dụng các kĩ thuật trong việc lên kế hoạch và hoạch định các dự án sẽ giúp ích cho các thành viên của nhóm trong tương lai