


# Bank Churn Analysis

ĐÌNH THÁI HOÀNG

# Mục Lục

- 
- |          |                                 |
|----------|---------------------------------|
| <b>1</b> | <b>Tổng quan bộ dữ liệu</b>     |
| <b>2</b> | <b>Phân tích</b>                |
| <b>3</b> | <b>Thiết lập mô hình dự báo</b> |
| <b>4</b> | <b>Tổng kết</b>                 |



# Tổng quan dữ liệu



# Tổng quan dữ liệu

## Data dictionary

### BANK CUSTOMER DATA DICTIONARY

Column name	Description
customer_id	Account Number
credit_score	Credit Score
country	Country of Residence
gender	Sex
age	Age
tenure	From how many years he/she is having bank acc in ABC Bank
balance	Account Balance
products_number	Number of Product from bank
credit_card	Does this customer have a credit card ?
active_member	Is he/she an active Member of the bank ?
estimated_salary	Salary of Account holder
churn	Churn Status



# Tổng quan dữ liệu

```
>>> <class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   customer_id           10000 non-null  int64
 1   credit_score           10000 non-null  int64
 2   country               10000 non-null  object
 3   gender                10000 non-null  object
 4   age                   10000 non-null  int64
 5   tenure                10000 non-null  int64
 6   balance                10000 non-null  int64
 7   products_number       10000 non-null  int64
 8   credit_card           10000 non-null  int64
 9   active_member         10000 non-null  int64
10   estimated_salary      10000 non-null  int64
11   churn                 10000 non-null  int64
dtypes: int64(10), object(2)
memory usage: 937.6+ KB
```

Dữ liệu có:

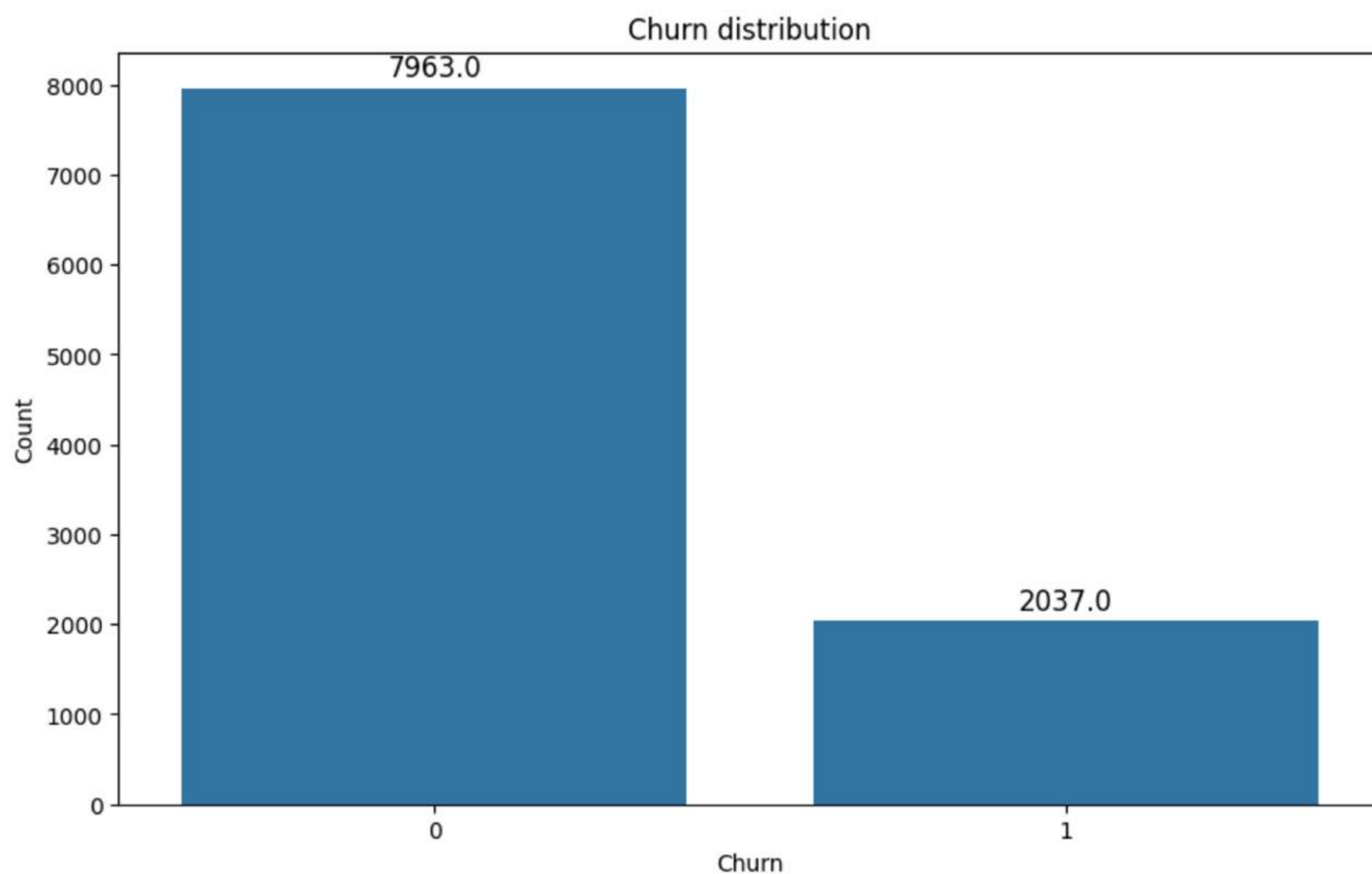
- 11 dòng
- 10000 cột

Trong đó không có dữ liệu null hoặc trống

# Phân tích

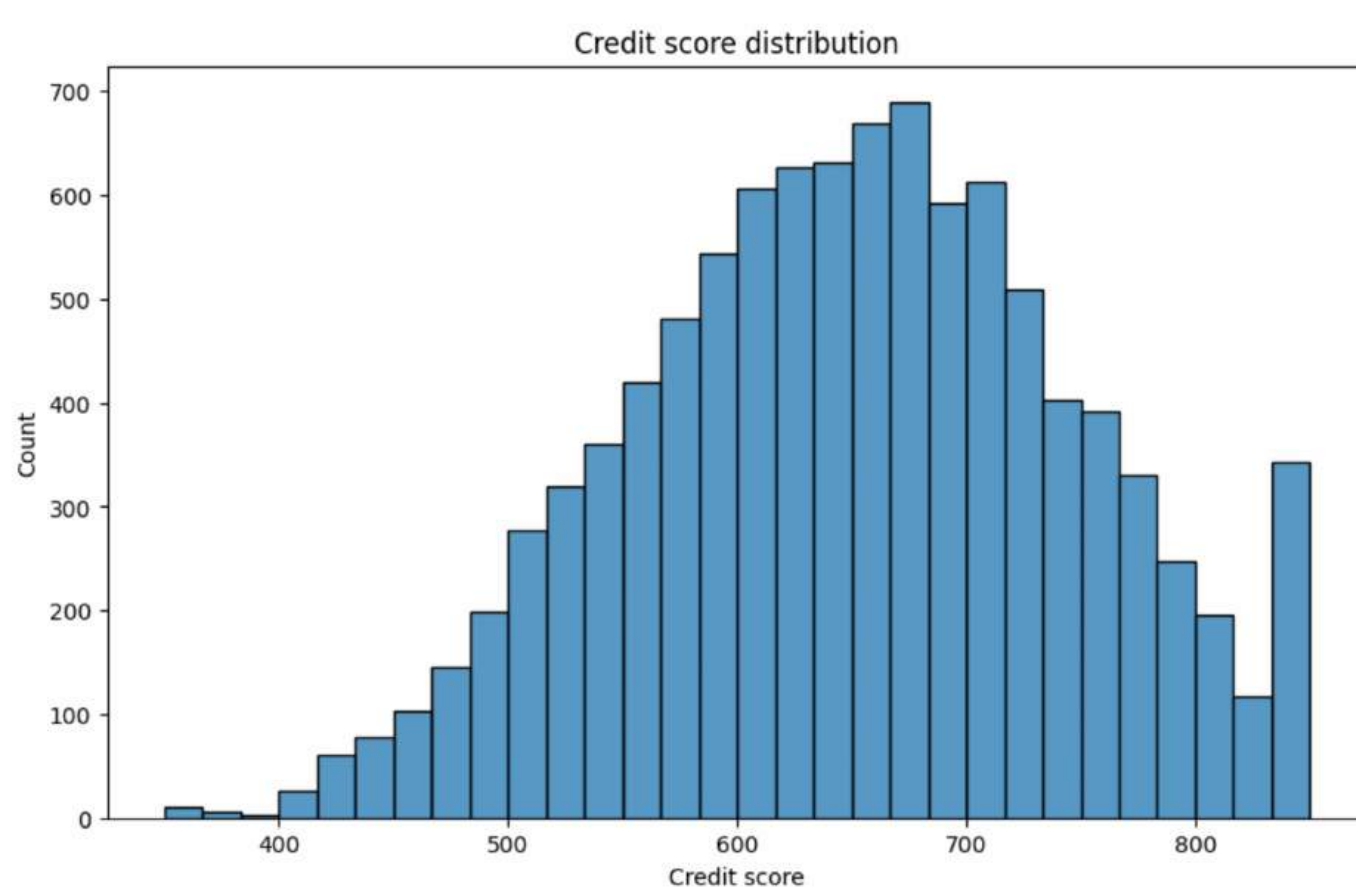


# Phân tích



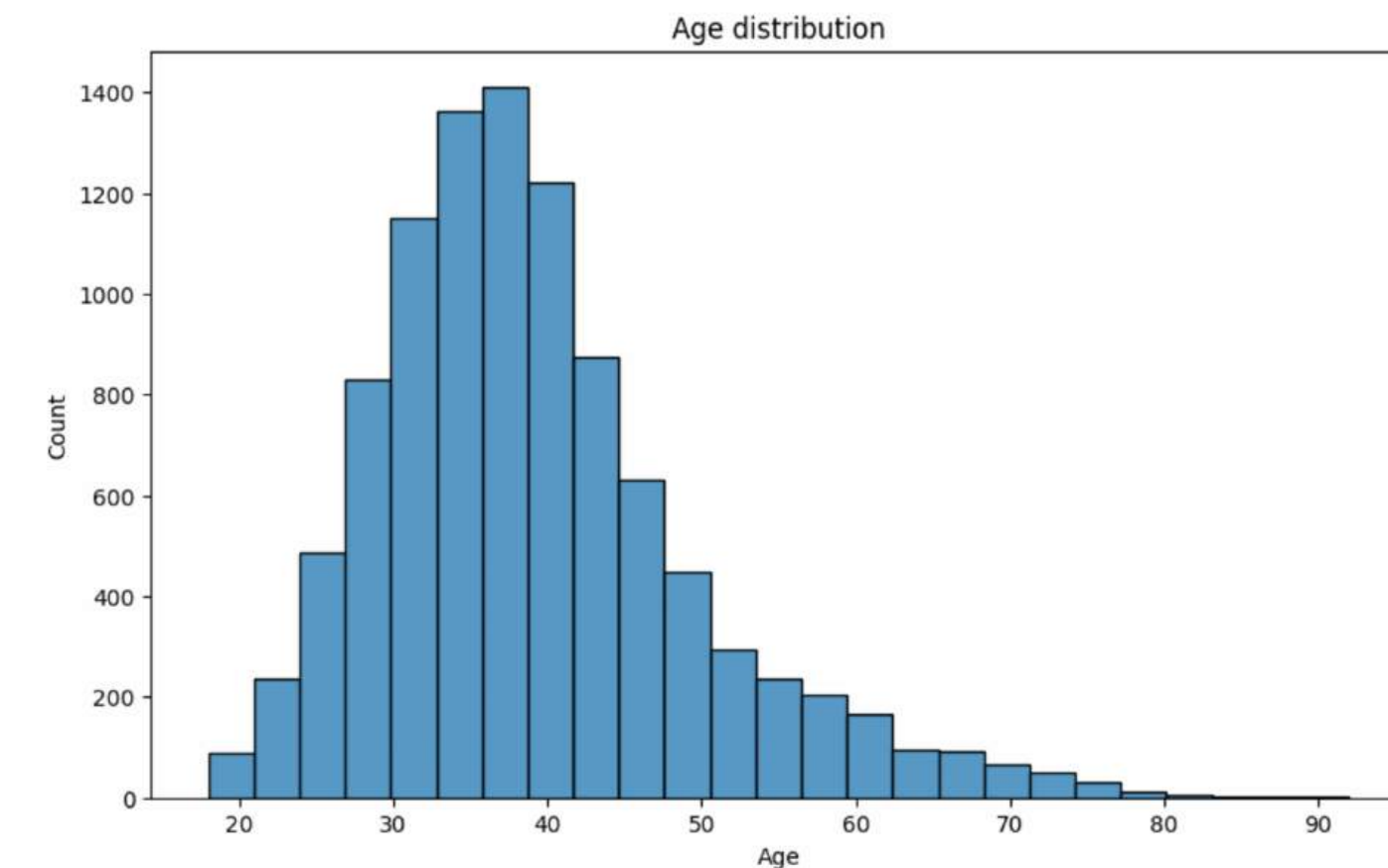
Dưới đây là bảng phân phối churn. Số lượng churn là xấp xỉ 2000 so với lượng khách hàng ở lại, chiếm 25% tổng số khách hàng. Đây là một tỉ lệ khá cao





Điểm tín dụng tập trung nhiều ở mức trung bình khá đến tốt, số người có điểm tín dụng dưới 500 khá ít.

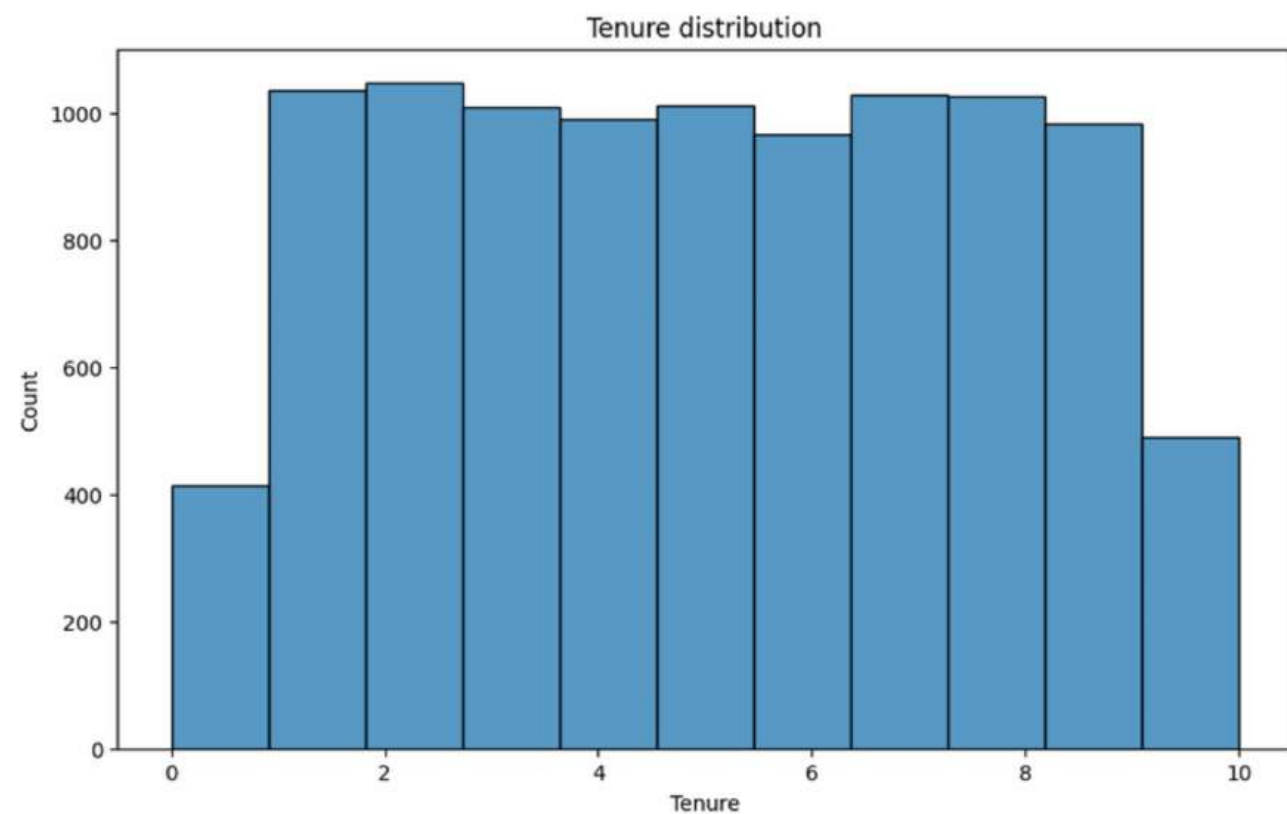
➤ Ít có khả năng khách hàng rời bỏ là do nợ xấu hoặc không có khả năng trả nợ



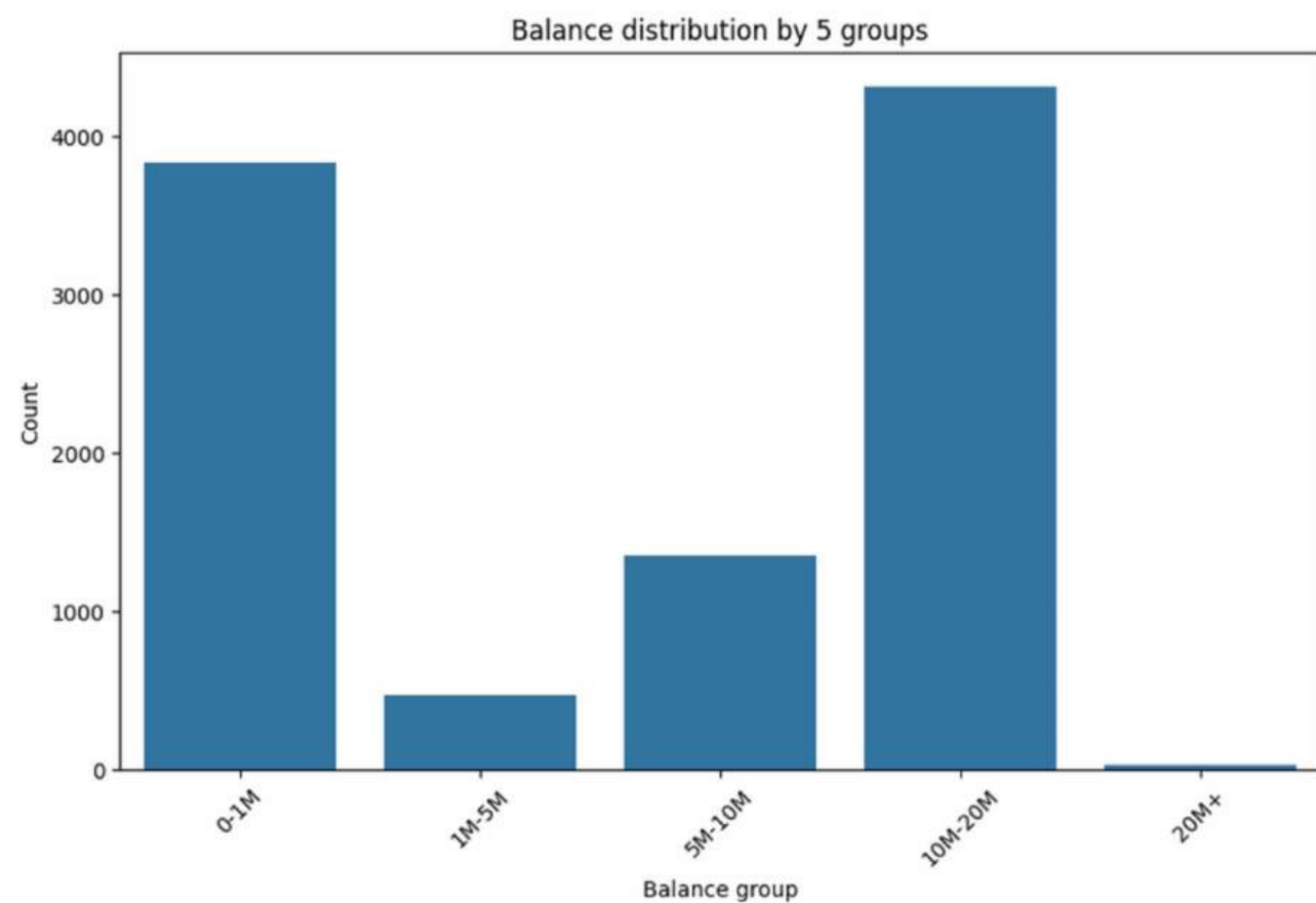
Độ tuổi khách hàng tập trung nhiều từ 28-45 tuổi. Tập khách chủ yếu của ngân hàng là những người trung niên và gần trung niên

➤ Khả năng nhóm khách hàng rời bỏ là nhóm những người cao tuổi nên không có nhu cầu sử dụng nhiều hoặc nhóm người trẻ mới được tạo tài khoản ngân hàng

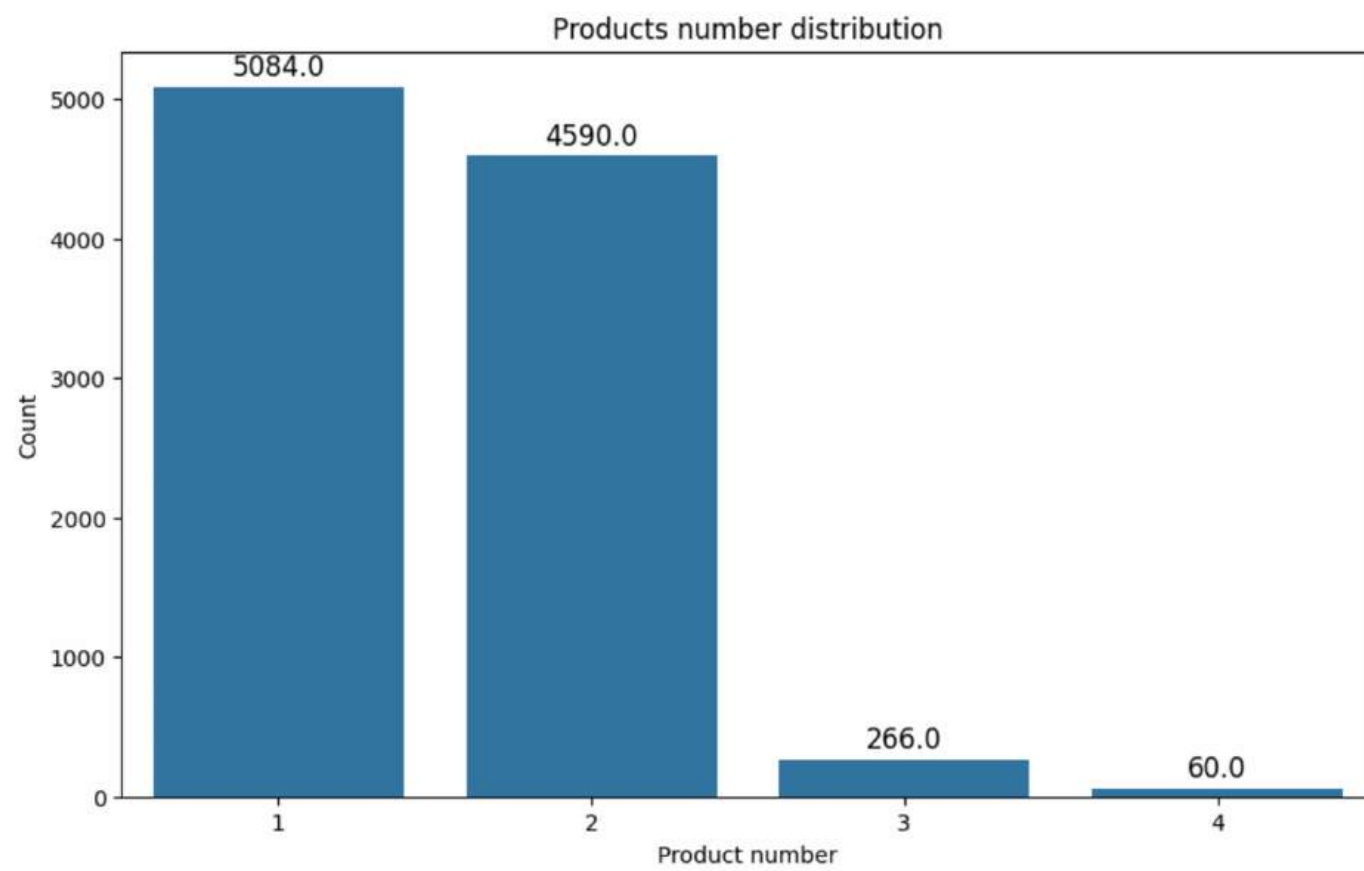




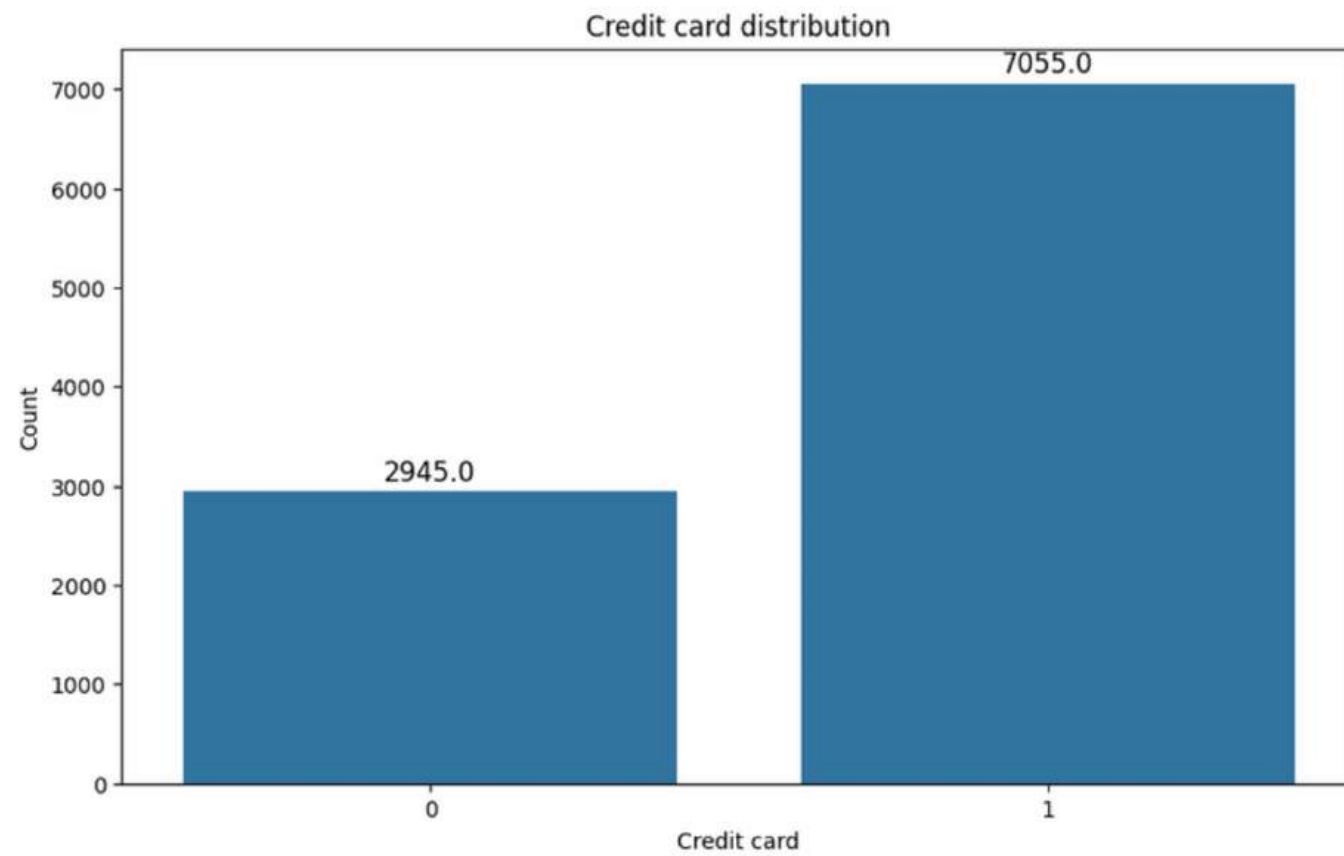
Thời gian khách hàng sở hữu tài khoản, tập trung nhiều từ 1-9 năm sau đó giảm mạnh ở năm thứ 10. Chưa có đủ dữ liệu để đưa ra kết luận nhưng có khả năng người dùng mới thì không thích dịch vụ của ngân hàng và rời đi trong 1 năm đầu, còn đến năm thứ 10 có thể là do data chưa thu thập đủ hoặc khách hàng tới năm thứ 10 thì không còn nhu cầu sử dụng nữa



Phân phối số dư tập trung nhiều ở mức 0-1M và 10-20M, số ít còn lại thì ở 1-5m và 5-10m, mốc 20m+ thì không đáng kể. Khả năng những người rời bỏ dịch vụ nằm nhiều ở mốc 0-1M

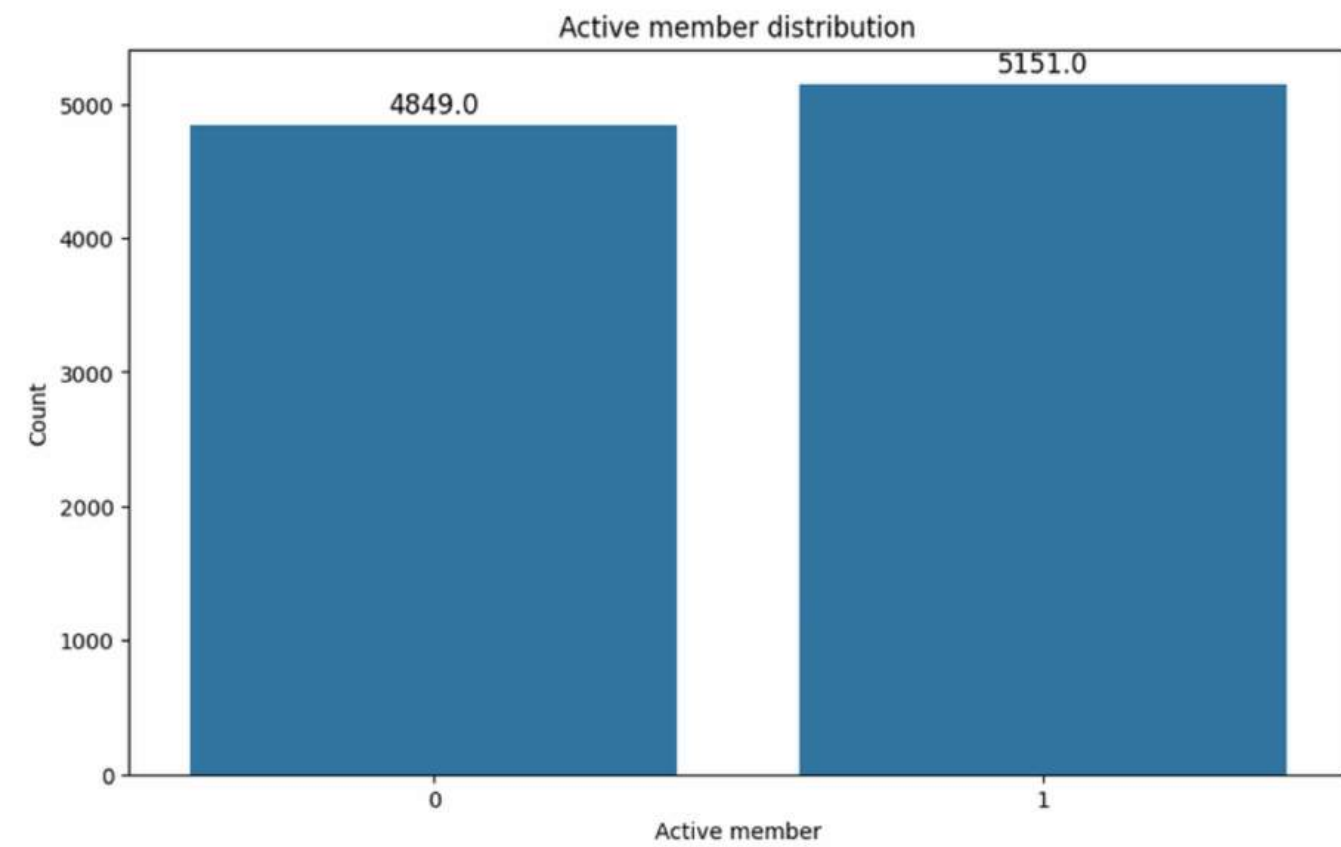


Hầu hết khách hàng chỉ có nhu cầu sử dụng từ 1-2 dịch vụ của ngân hàng. Các chỉ số này quyết định nhiều đến chỉ số churn

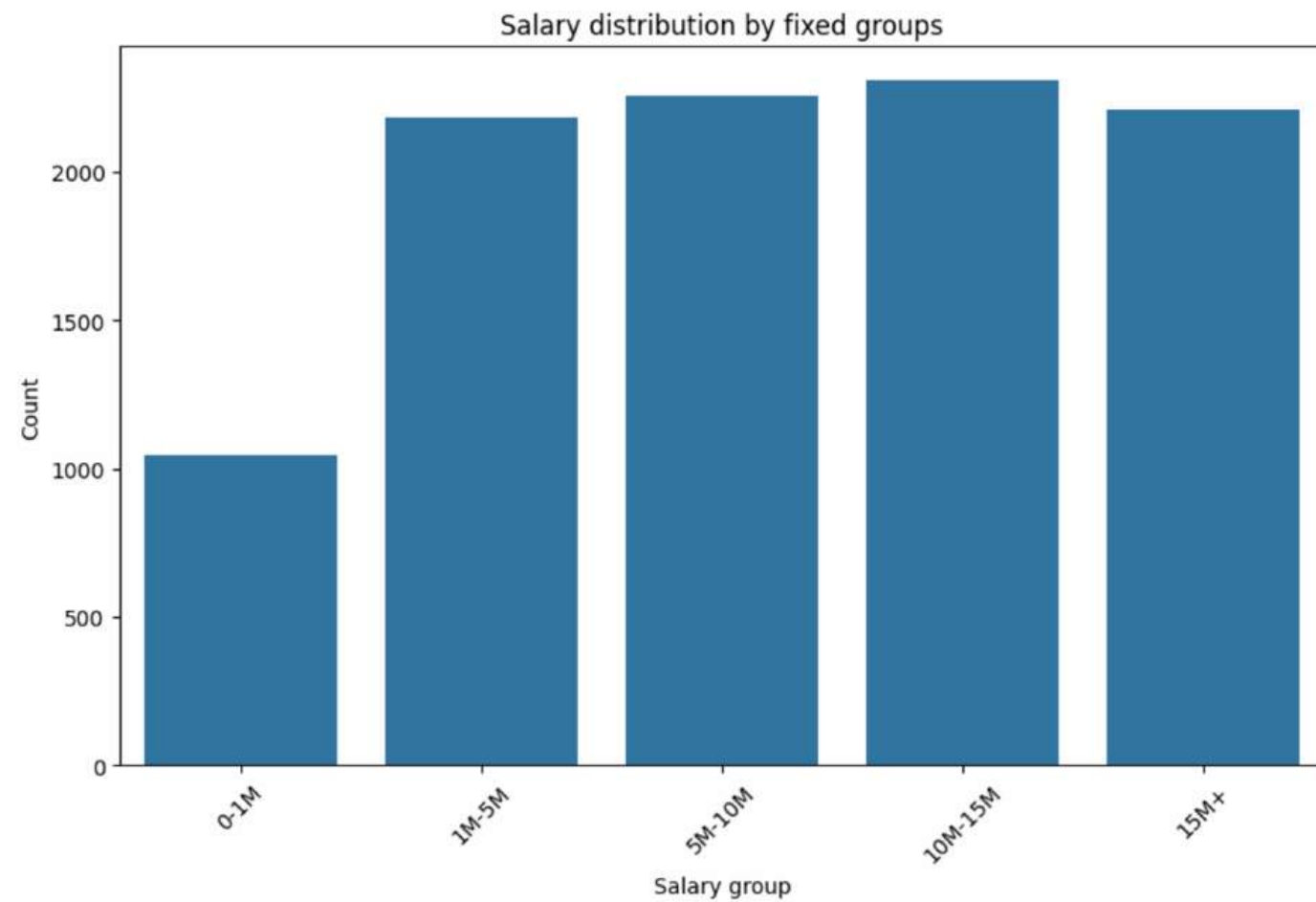


Phân phối của khách hàng sử dụng credit card khá giống với phân phối churn, khả năng khách hàng churn không sử dụng credit card?



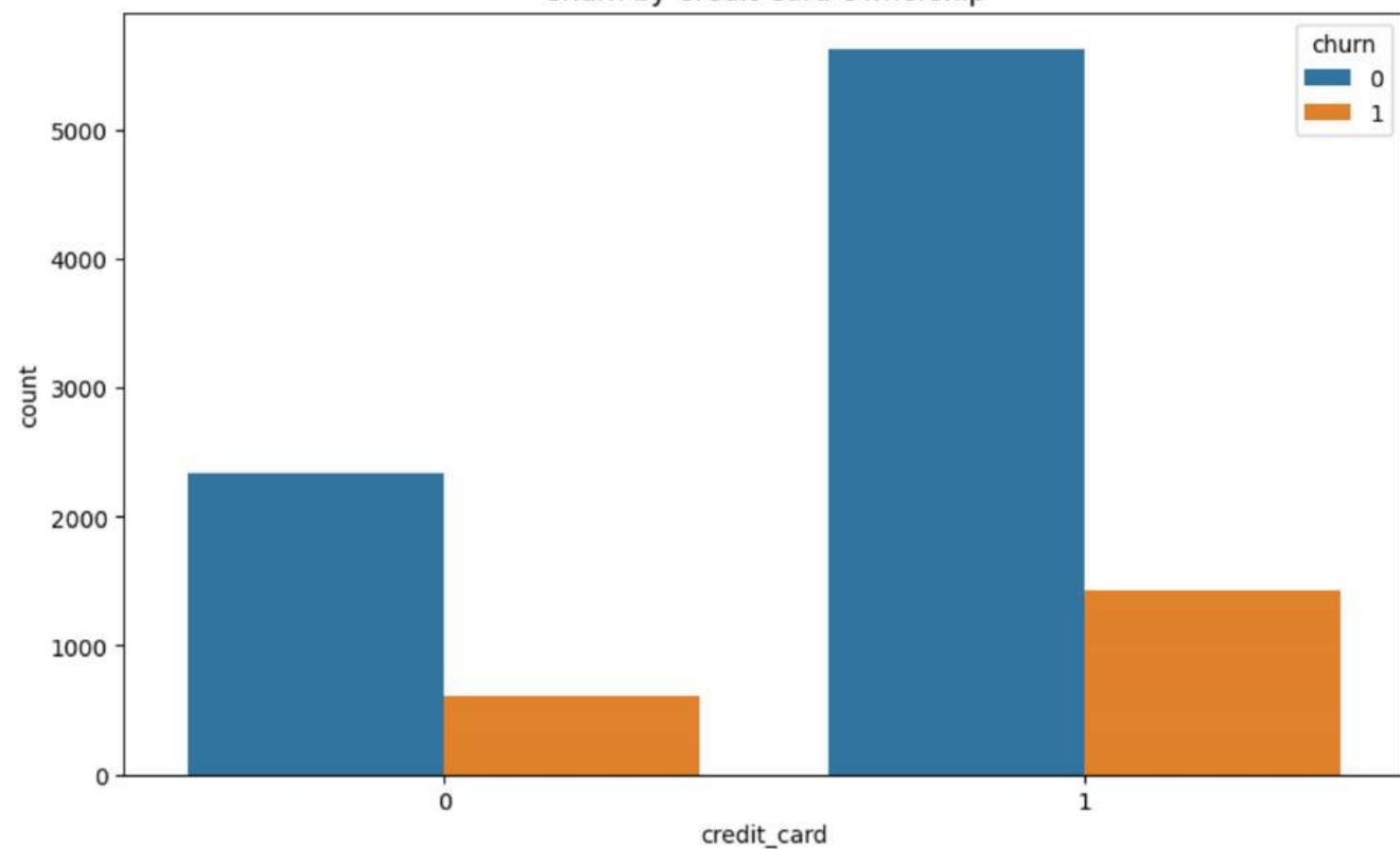


Phân phối của các thành viên hoạt động tích cực là xấp xỉ nhau. Sẽ có nhóm trung thành sử dụng dịch vụ ngân hàng và nhóm còn lại thì có thể thỉnh thoảng mới dùng



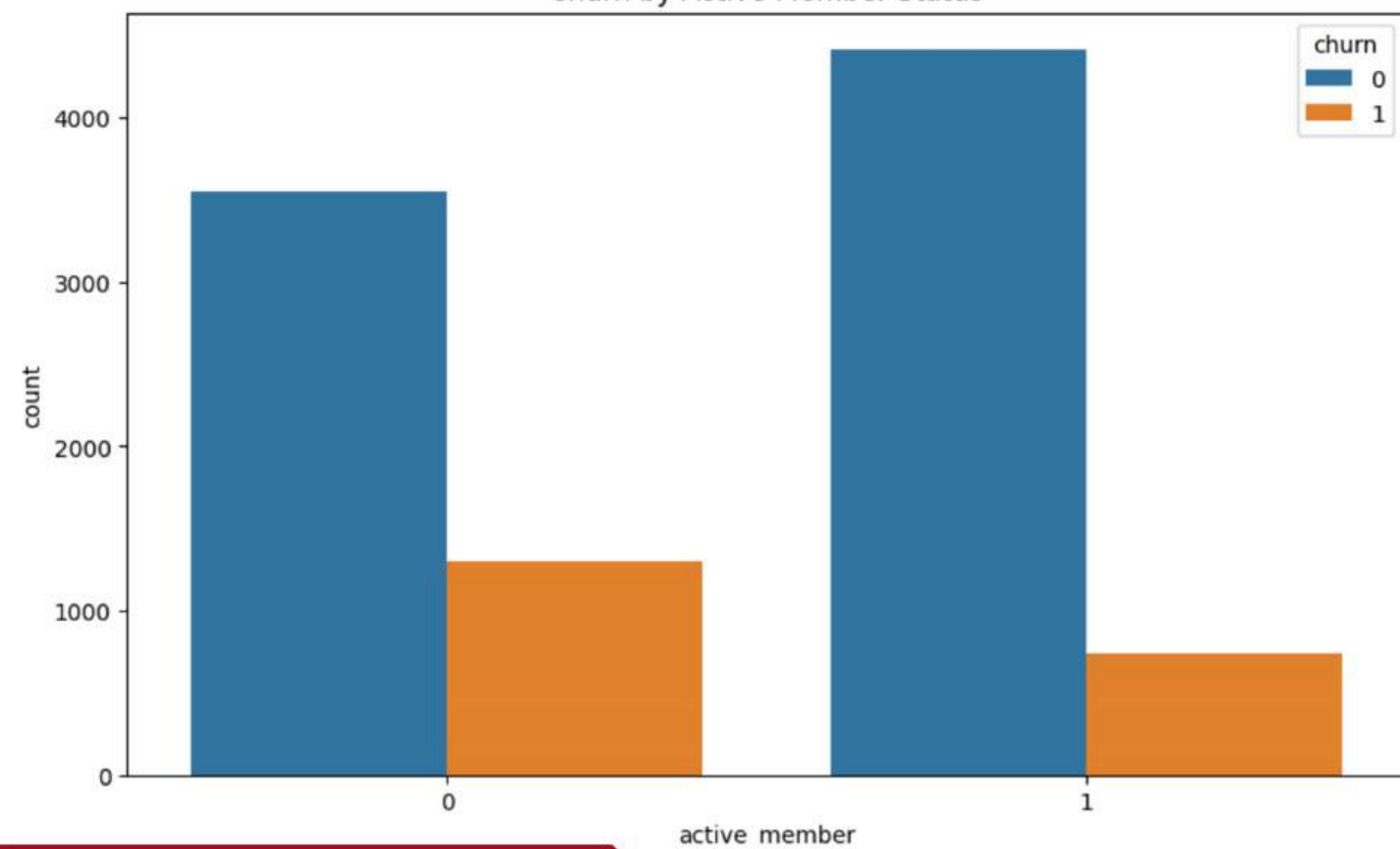
Phân phối tiền lương thấp nhất ở nhóm 0-1M, các nhóm còn lại xấp xỉ nhau

Churn by Credit Card Ownership



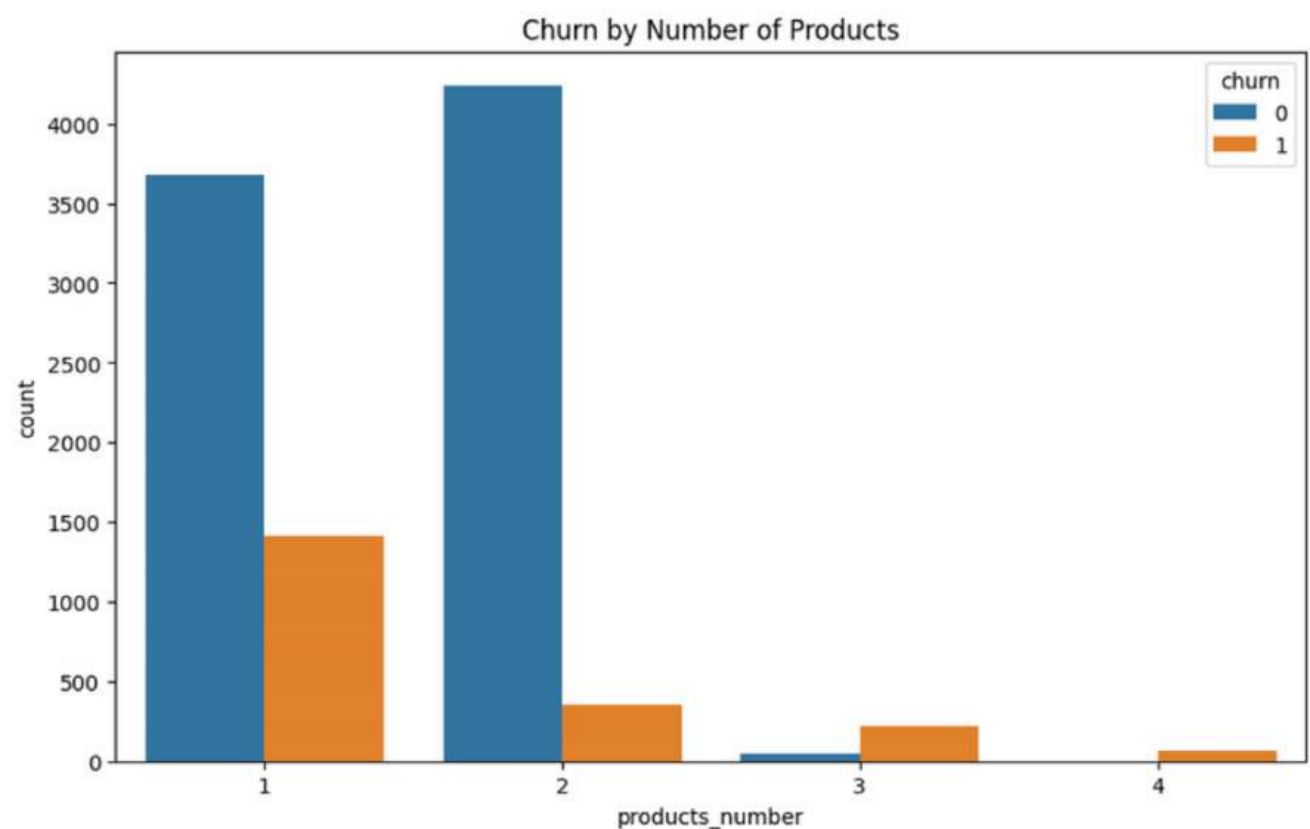
Nhìn vào bảng này thì tỉ lệ churn của nhóm sử dụng credit card và nhóm không sử dụng có tỉ lệ churn ngang nhau

Churn by Active Member Status

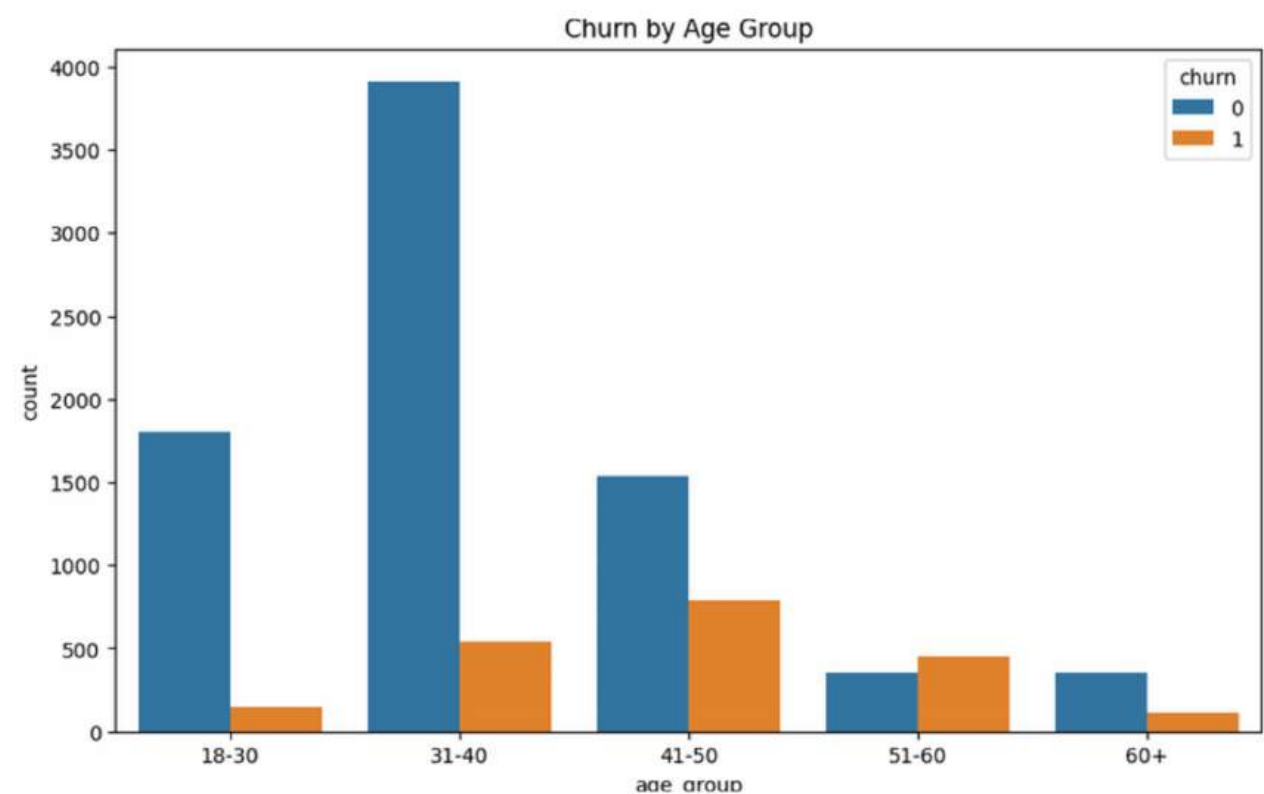


Tương tự với member status

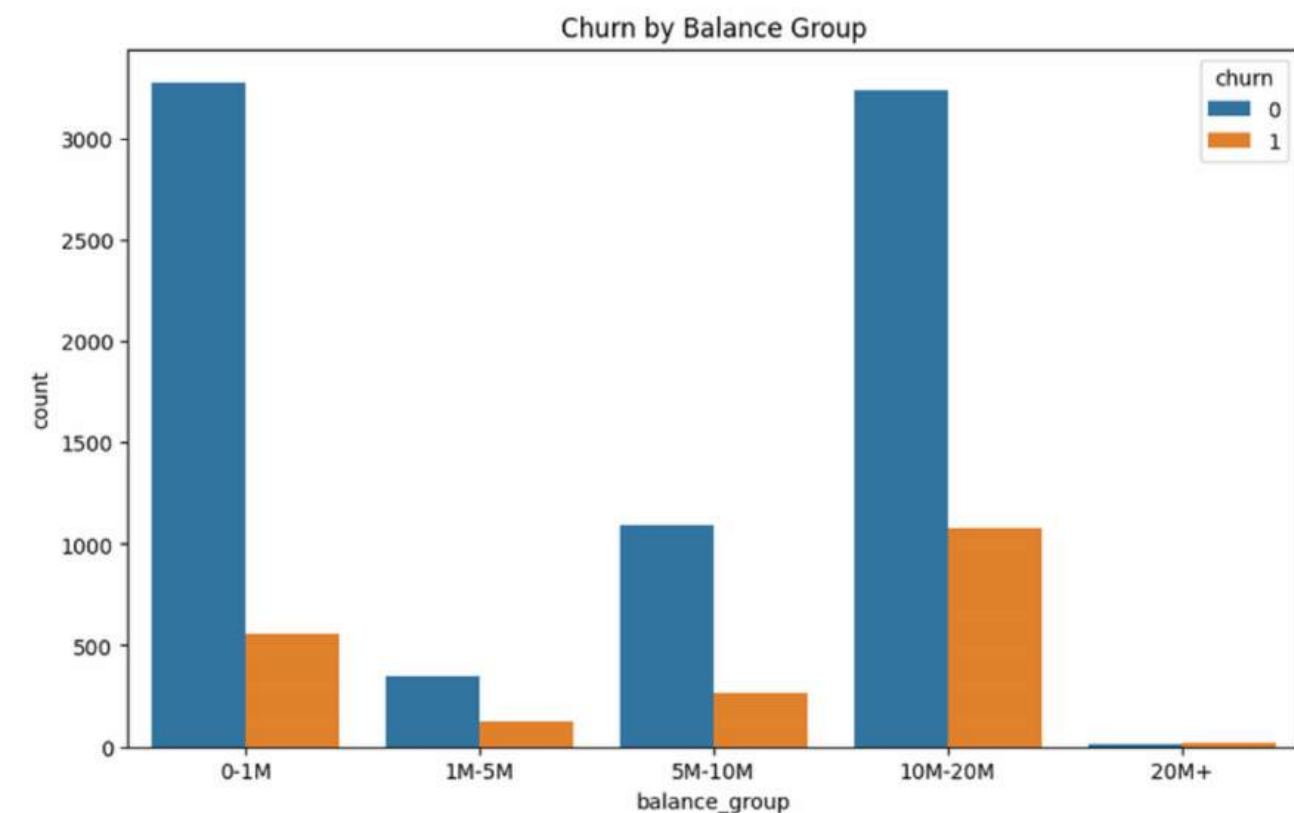




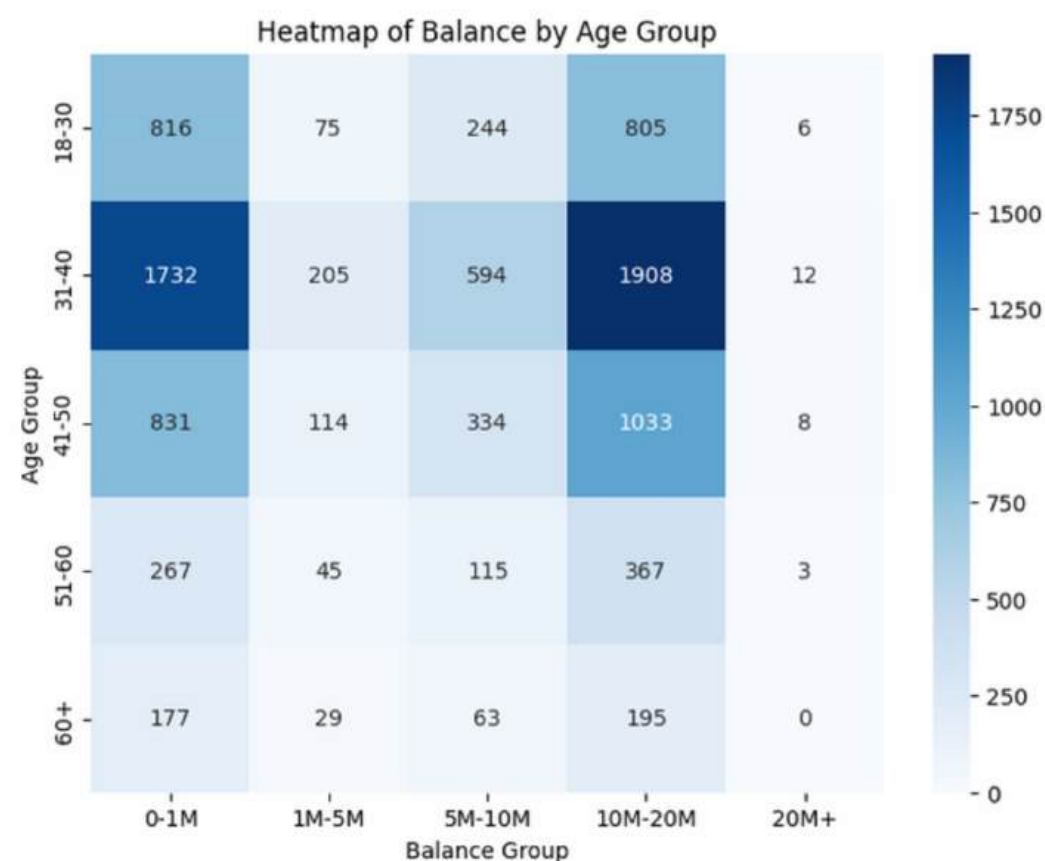
Tỉ lệ churn ở nhóm sử dụng 1 dịch vụ là cao nhất, đây có thể là nhóm chỉ sử dụng thẻ ghi nợ hoặc thẻ tín dụng và đã tìm thấy dịch vụ ở 2 mảng này tốt hơn. Tỉ lệ churn ở nhóm sử dụng 3 và 4 dịch vụ là cao nhất, nhóm này có thể sau khi trải nghiệm tất cả các dịch vụ thì quyết định sử dụng dịch vụ của ngân hàng mới



Đối với các nhóm tuổi, tỉ lệ churn cao nhất nằm ở nhóm người cao tuổi và trung niên. Cho thấy những người từ cuối trung niên có xu hướng từ bỏ dịch vụ, có thể do không còn nhu cầu hoặc không còn cần sử dụng tiền nhiều như trước



Tỉ lệ churn giữa các nhóm balance tương đối đồng đều, tuy nhiên thì số lượng churn ở nhóm 0-1M và nhóm 10-20M chiếm khá cao so với hai nhóm còn lại. Với việc khách hàng đa số là người cao tuổi thì khả năng số dư 10-20M cũng thuộc về nhóm đấy, ta sẽ đào sâu hơn ở chart dưới

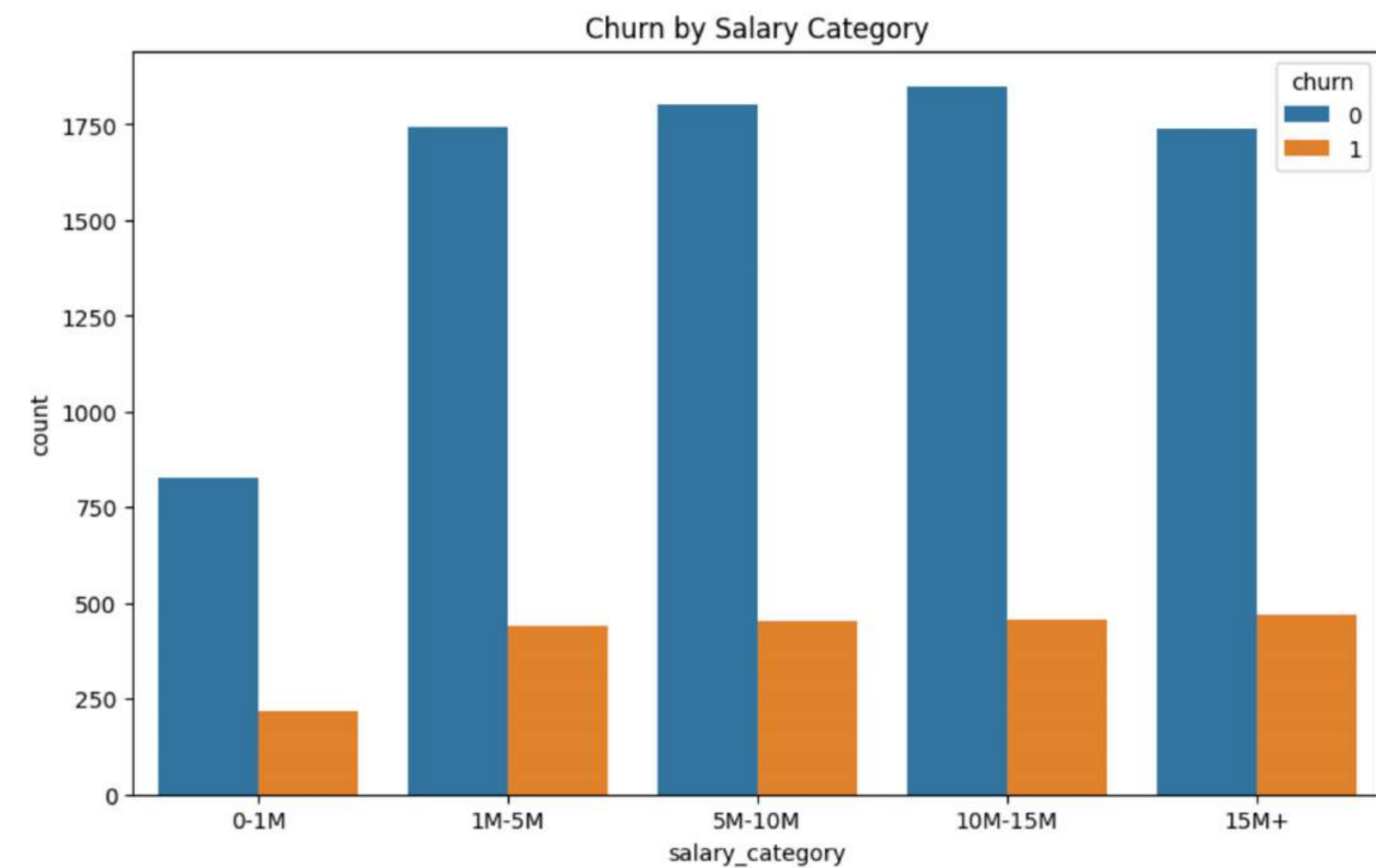


Bất ngờ là số dư 0-1M và 10-20M đều tập trung nhiều ở nhóm trung niên và cuối trung niên. Đối với nhóm 0-1M thì có thể là họ đã rời bỏ dịch vụ từ lâu và còn sót lại số dư. Còn đối với nhóm 10-20M có thể do họ quá giàu và ngâm số dư đó trong tài khoản mà không gửi tiết kiệm



Cần vận động nhóm này gửi tiết kiệm hoặc sử dụng các dịch vụ khác





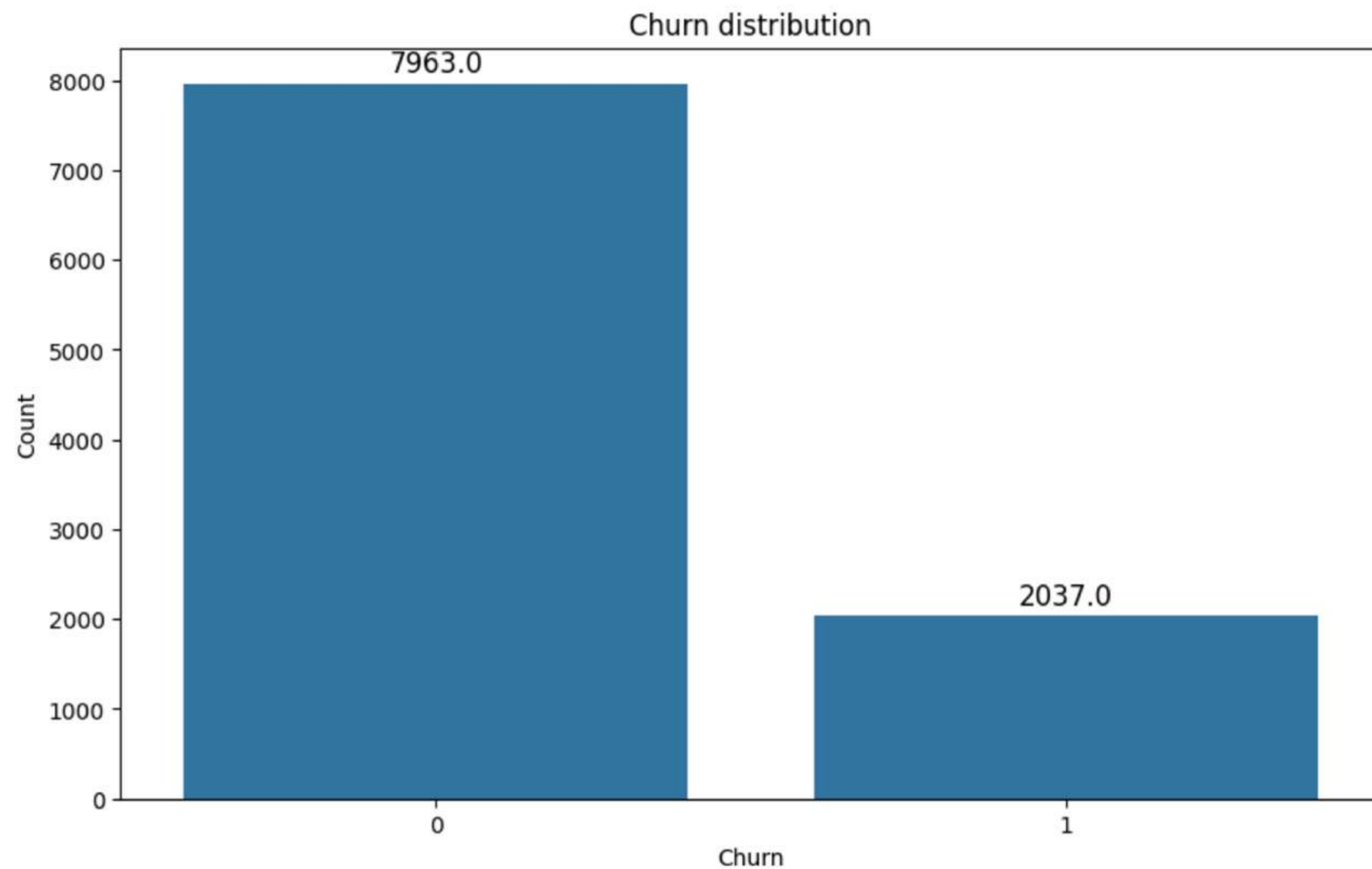
Nhóm tiền lương phân bố tỉ lệ đồng đều, tuy nhiên số lượng vẫn rất cao ở mọi mốc. Tập trung vận động dịch vụ với các nhóm có lương cao và mời chào mở tài khoản đối với nhóm lương thấp

# Thiết lập mô hình dự báo



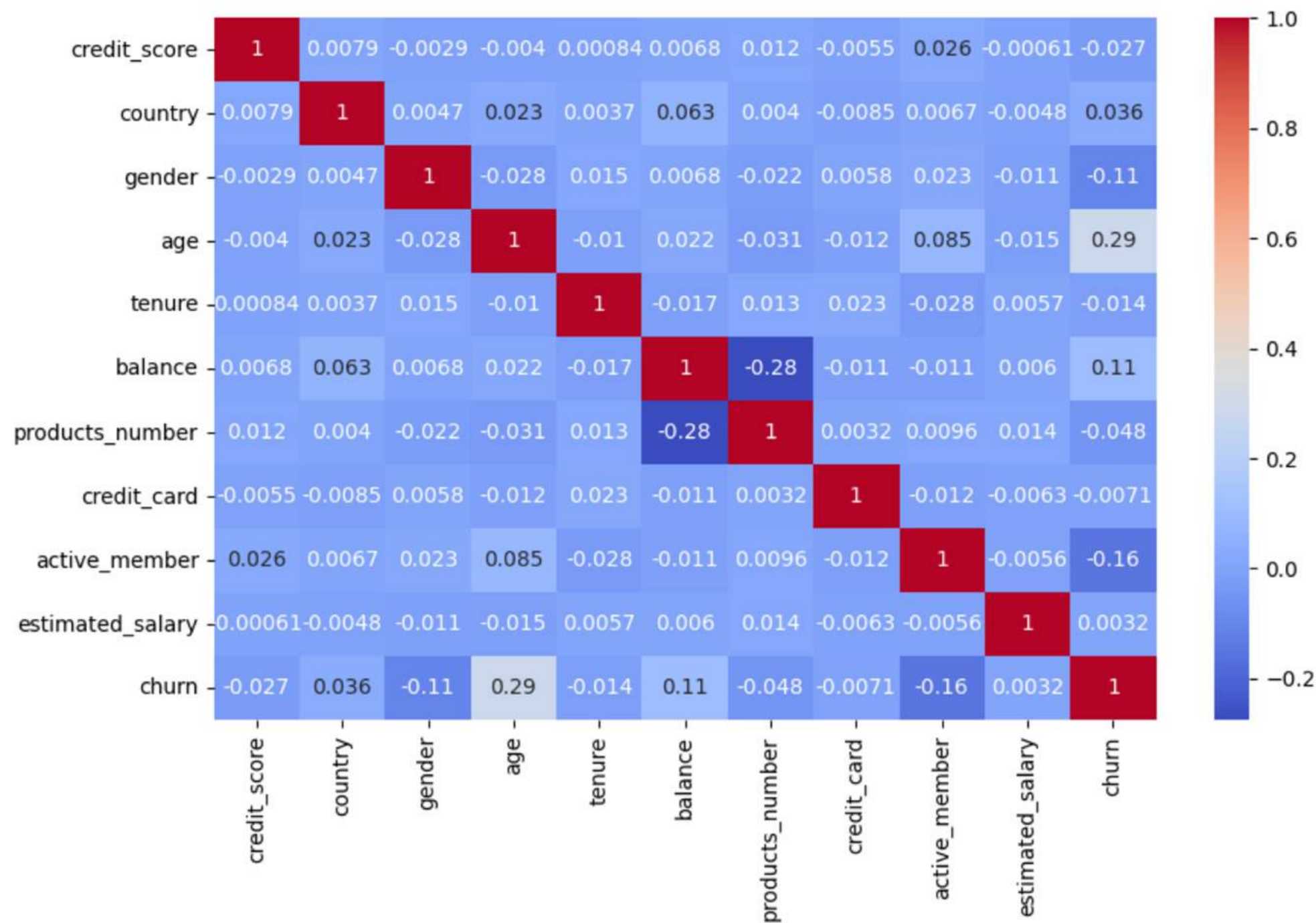


# Thiết lập mô hình dự báo



Mô hình cũng đang bị mất cân bằng dữ liệu với cột churn ít hơn cột không churn

# Thiết lập mô hình dự báo



Tương quan giữa các biến khá là yếu, nên mô hình có thể sẽ có nhiều sai sót. Có thể bổ sung thêm dữ liệu nhằm khắc phục

# Thiết lập mô hình dự báo

```
Suggested code may be subject to a license | leehj01/Bigdata_test
## Random Forest
X = data_scaled.drop("churn", axis=1)
y = data_scaled["churn"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
model = RandomForestClassifier()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

Sau khi chạy hàng loạt mô hình thì mô hình Random Forest là có accuracy cao nhất vì vậy đây sẽ là mô hình đầu tiên tôi chạy

	precision	recall	f1-score	support
0	0.88	0.97	0.92	2416
1	0.77	0.44	0.56	584
accuracy			0.87	3000
macro avg	0.82	0.70	0.74	3000
weighted avg	0.86	0.87	0.85	3000

Có thể thấy mô hình đang hiệu quả hơn trong việc dự đoán các khách hàng không churn, điều này là do việc mất cân bằng dữ liệu



# Thiết lập mô hình dự báo

```
## Mô hình sau khi oversampling|
from imblearn.over_sampling import SMOTE
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X_train, y_train)
model = RandomForestClassifier()
model.fit(X_resampled, y_resampled)
y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

Mô hình mới được oversampling

	precision	recall	f1-score	support
0	0.90	0.91	0.90	2416
1	0.60	0.59	0.59	584
accuracy			0.84	3000
macro avg	0.75	0.75	0.75	3000
weighted avg	0.84	0.84	0.84	3000

Sau khi oversampling xong thì precision lại giảm, vì vậy tôi sẽ giữ nguyên mô hình đầu tiên



# Tổng kết

## **Đối với nguyên nhân churn và cách khắc phục**

- Qua phân tích trên thì ta có thể thấy ngân hàng gặp hai vấn đề là người dùng mới thì bỏ đi ngay nhiều và người dùng lâu năm thì ngân tiền trong tài khoản
- Ngân hàng có thể áp dụng các khuyến mãi hay ưu đãi cho người mới sử dụng dịch vụ ngân hàng
- Đối với nhóm đang có nhiều tiền nhàn rỗi thì ngân hàng nên chủ động liên hệ với họ để remind đồng thời cũng giới thiệu các sản phẩm của ngân hàng

## **Đối với mô hình dự báo**

- Ngân hàng có thể bổ sung thêm dữ liệu để tăng độ chính xác của mô hình
- có thể thêm các biến khác như Home ownership hoặc Marital status để chỉ ra được nhiều sự tương quan hơn
- Mô hình cũng có thể tinh chỉnh lại các parameter sao cho phù hợp để khắc phục oversampling



# Thank you

