

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KINH TẾ LUẬT

KHOA HỆ THỐNG THÔNG TIN



Báo cáo giữa kỳ môn Phân tích dữ liệu với R/python

**Áp dụng phương pháp Hồi quy Logistic vào
khai phá tập dữ liệu về Tiếp thị của ngân hàng**

Mã học phần:

212IS2901

Giảng viên hướng dẫn:

Ths. Nguyễn Phát Đạt

Thầy Trần Lê Tấn Thịnh (trợ giảng)

Sinh viên thực hiện: Nhóm Num

Võ Chí Giang - K194111533

Nguyễn Bá Thịnh An - K194111517

Phạm Minh Đạt - K194111530

Trần Thị Thảo Ly - K194111546

Trịnh Thị Tâm Oanh - K194111560

TP HCM, Ngày 24 tháng 05 năm 2022

LỜI CẢM ƠN

Lời đầu tiên, nhóm xin đặc biệt gửi lời cảm ơn đến Thầy Nguyễn Phát Đạt (giảng viên môn Phân tích dữ liệu với R/python) và thầy trợ giảng Trần Lê Tấn Thịnh. Các Thầy đã cung cấp kiến thức, chỉ bảo và đóng góp những ý kiến quý báu giúp nhóm hoàn thành được bài tập giữa kỳ môn học của mình.

Xuất phát từ mục đích học tập, tìm hiểu sâu hơn các kiến thức về Phân tích dữ liệu với R/python, nhóm đã quyết định chọn đề tài “ÁP DỤNG PHƯƠNG PHÁP HỒI QUY LOGISTIC VÀO KHAI PHÁ TẬP DỮ LIỆU VỀ TIẾP THỊ CỦA NGÂN HÀNG”. Trong quá trình thực hiện đồ án, dựa trên những kiến thức được các Thầy cung cấp trên giảng đường kết hợp với việc tự tìm hiểu những công cụ và kiến thức mới, nhóm đã cố gắng hoàn thiện báo cáo một cách tốt nhất.

Tuy nhiên, với năng lực của bản thân các thành viên còn nhiều hạn chế, bài tập ắt sẽ chưa hoàn thiện và còn nhiều sai sót nhưng nó là kết quả của sự nỗ lực của các thành viên trong nhóm, cũng như sự giúp đỡ của tất cả bạn bè và các Thầy.

Nhóm rất mong nhận sự góp ý từ phía các Thầy nhằm rút ra những kinh nghiệm quý báu và hoàn thiện vốn kiến thức để nhóm có thể tiến xa hơn trong học tập môn Phân tích dữ liệu với R/python nói riêng, và các môn học liên quan nói chung, cũng như làm nền tảng cho việc định hướng công việc trong tương lai.

Nhóm xin chân thành cảm ơn hai Thầy!

Nhóm thực hiện.

MỤC LỤC

LỜI CẢM ƠN	1
Danh mục hình ảnh	4
Danh mục bảng biểu.....	6
CHƯƠNG I: TỔNG QUAN ĐỀ TÀI	8
I.1. Yêu cầu và mục tiêu xây dựng bài toán.....	8
I.3. Đề xuất quy trình, sơ đồ thực hiện nghiên cứu.....	12
CHƯƠNG II: ĐỊNH NGHĨA CÁC METRICS/VARIABLE.....	12
II.1. Chi tiết câu hỏi nghiên cứu.....	13
II.2. Lựa chọn metrics/variables.....	13
II.3. Lựa chọn dữ liệu để trả lời câu hỏi:.....	15
III.3.1. Câu hỏi 1:	15
III.3.2. Câu hỏi 2:	21
CHƯƠNG III: PHÂN TÍCH KHÁM PHÁ DỮ LIỆU (EDA) BIÊN PHỤ THUỘC VỚI PHƯƠNG PHÁP DESCRIPTIVE ANALYTICS	24
III.1. Clean data.....	24
III.2. Phân tích khám phá dữ liệu (EDA)	26
III.2.1. Age	26
III.2.2. Job	28
III.2.3. Marital	30
III.2.4. Education:	31
III.2.5. Default.....	33
III.2.6. Balance:.....	34
III.2.7. Housing	35
III.2.8. Loan.....	36
III.2.9. Contact	37

III.2.10. Month	38
III.2.11. Duration.....	40
III.2.12. Campaign	41
III.2.13. Pdays:	42
III.2.14. Previous.....	43
III.2.15. Poutcome.....	44
III.2.16. Is_success:.....	45
Chương IV: Xây dựng mô hình hồi quy logistic dự báo biến phụ thuộc.....	45
Chương V: Đánh giá kết quả và lựa chọn mô hình.....	47

Danh mục hình ảnh

Hình 1: Quy trình thực hiện nghiên cứu	12
Hình 2: Mô tả dữ liệu tuổi của khách hàng	15
Hình 3: Biểu đồ thể tỉ lệ giữa nhóm tuổi và khả năng đăng ký sản phẩm	16
Hình 4: Tỷ lệ thành công và tổng số khách hàng liên lạc theo từng nhóm tuổi	16
Hình 5: Phân bố nhóm tuổi trong bộ dữ liệu.....	17
Hình 6: Thống kê mô tả của biến “balance”	17
Hình 7: Phân nhóm balance.	18
Hình 8: Biểu đồ thể hiện tỷ lệ thành công theo từng balance.	18
Hình 9: Tỷ lệ thành công theo từng nhóm balance	19
Hình 10: Mối liên hệ giữa tuổi và số dư tài khoản.....	19
Hình 11: Trung bình số dư theo từng nhóm tuổi	20
Hình 12: Tình trạng nợ mua nhà/ tình trạng nợ cá nhân	20
Hình 13: Biểu đồ thể hiện số dư trung bình theo từng nhóm nợ	21
Hình 14: Biểu đồ phân bố tỷ lệ thành công theo duration và campaign.....	22
Hình 15: Bảng thống kê tỷ lệ thành công theo tháng.....	23
Hình 16: Biểu đồ thể hiện tỷ lệ thành công theo từng tháng	23
Hình 18: Kiểm tra dữ liệu	24
Hình 19: Thống kê mô tả các biến numerical	25
Hình 20: Hàm xử lý outliers.....	25
Hình 21: Biểu đồ boxplot giữa tuổi và biến phụ thuộc	26
Hình 22: Thống kê mô tả biến tuổi	26
Hình 23: Mối liên hệ giữa nhóm tuổi và biến phụ thuộc	27
Hình 24: Phân bố nhóm tuổi trong bộ dữ liệu.....	27

Hình 25: Tỷ lệ thành công theo từng nhóm ngành việc làm.....	28
Hình 26: Biểu đồ thể hiện mối liên hệ giữa nhóm việc làm và biến phụ thuộc.....	28
Hình 27: Xử lý tỷ lệ thành công theo nhóm ngành.....	29
Hình 28: Bảng kết quả tỷ lệ thành công theo nhóm ngành.....	29
Hình 29: Tỷ lệ thành công theo tình trạng hôn nhân	30
Hình 30: Biểu đồ thể hiện mối liên hệ giữa tỷ lệ thành công và tình trạng hôn nhân .	30
Hình 31: Bảng tỷ lệ thành công theo tình trạng hôn nhân	31
Hình 32: Tỷ lệ thành công theo trình độ học vấn.....	31
Hình 33: Biểu đồ thể hiện mối liên hệ giữa tỷ lệ thành công và trình độ học vấn	32
Hình 34: Bảng thể hiện mối liên hệ giữa tỷ lệ thành công và trình độ học vấn.....	32
Hình 35: Tỷ lệ thành công theo tình trạng vợ nơ	33
Hình 36: Biểu đồ boxplot thể hiện tỷ lệ thành công theo số dư.....	34
Hình 37: Sửa giá trị ngoại lai của biến balance	34
Hình 38: Thống kê mô tả của biến balance.....	34
Hình 39: Biểu đồ thể hiện tỷ lệ thành công theo tình trạng nơ mua nhà	35
Hình 40: Bảng thể hiện tỷ lệ thành công theo tình trạng nơ mua nhà	36
Hình 41: Biểu đồ thể hiện tình trạng nơ cá nhân và tỷ lệ thành công.....	36
Hình 42: Bảng thể hiện tình trạng nơ cá nhân và tỷ lệ thành công	37
Hình 43: Bảng thể hiện tỷ lệ thành công theo các phương thức liên lạc	37
Hình 44: Bảng thể hiện tỷ lệ người dùng theo phương thức liên lạc.....	38
Hình 45: Phân bổ số lượng khách hàng được liên hệ theo từng tháng	38
Hình 46: Xử lý phân bổ số lượng khách hàng theo tháng	39
Hình 47: Tỷ lệ thành công của khách hàng theo từng tháng.....	39

Hình 48: Biểu đồ boxplot thể hiện tỷ lệ thành công theo thời lượng liên lạc với khách hàng.....	40
Hình 49: Thống kê mô tả của biến duration.....	40
Hình 50: Biểu đồ boxplot thể hiện giữa tỷ lệ thành công và số lần thực hiện cuộc gọi với khách hàng	41
Hình 51: Thống kê mô tả của biến campaign	41
Hình 52: Biểu đồ boxplot thể hiện tỷ lệ thành công và số ngày từ ngày cuối thực hiện chiến dịch.	42
Hình 53: Thống kê mô tả biến pdays	42
Hình 54: Biểu đồ boxplot thể hiện tỷ lệ thành công và số lần liên lạc trước chiến dịch	43
Hình 55: Thống kê mô tả biến previous.....	44
Hình 56: Bảng phân bố tỷ lệ thành công theo kết quả đầu ra của chiến dịch trước	44
Hình 57: Phân bố tỷ lệ biến phụ thuộc y.....	45
Hình 58: Xử lý các cột không cần thiết.....	45
Hình 59: Dummy các biến category	46
Hình 60: Phân chia tập dữ liệu	46
Hình 61: Trích chọn đặc trưng cho mô hình	47
Hình 62: Xây dựng mô hình Logistics Regression	47

Danh mục bảng biểu

Bảng 1: Bảng thông tin các biến.	11
Bảng 2: Bảng lựa chọn metrics và variables	15
Bảng 3: Bảng kết quả phân loại của mô hình 1	48
Bảng 4: Bảng kết quả phân loại của mô hình 2	49

Bảng 5: Bảng kết quả phân loại của mô hình 3	49
--	----

CHƯƠNG I: TỔNG QUAN ĐỀ TÀI

I.1. Yêu cầu và mục tiêu xây dựng bài toán

- Yêu cầu:** Nhóm lựa chọn một tập dữ liệu về các thông tin tiếp thị của một ngân hàng. Bộ dữ liệu có liên quan đến các chiến dịch tiếp thị trực tiếp của một tổ chức ngân hàng Bò Đào Nha, cụ thể là các chiến dịch tiếp thị dựa trên các cuộc gọi điện thoại. Từ bộ dữ liệu này, nhóm tiến hành phân tích và xác định xem liệu khách hàng có tiến hành đăng ký một sản phẩm dịch vụ của ngân hàng hay không dựa trên những biến của bộ dữ liệu.
- Mục tiêu bài toán:**
 - Dự đoán dựa trên chiến dịch marketing trước và tìm ra yếu tố nào ảnh hưởng đến kết quả của chiến dịch nhằm thúc đẩy cho quá trình marketing về sau được hiệu quả hơn.
 - Tìm ra những phân khúc khách hàng dựa vào những thông tin có được. Việc này giúp cho nhận diện được profile của customer, xem những người có những đặc điểm như thế nào thì sẽ quan tâm đến việc subscribe và chú trọng những đối tượng này để phát triển các chiến dịch marketing trong tương lai.
- Bộ dữ liệu bao gồm 45211 quan sát với 17 biến, trong đó có 10 biến phân loại và 7 biến số (với những biến phân loại chứa những giá trị chỉ có hai giá trị là giá trị nhị phân).
- Tiếp theo, tiến hành mô tả đặc điểm của từng cột và chỉ ra sự phụ thuộc của chúng vào kết quả của bài toán. Sau này chúng ta sẽ tiến hành phân tích quan hệ nâng cao hơn giữa chúng.
- Các trường cụ thể bao gồm:

Tên biến	Loại biến	Mô tả	Giá trị
Độ Tuổi	Số	Độ tuổi của khách hàng	Giá trị từ 18 đến 95
Nghề nghiệp	Phân loại	Nghề nghiệp của khách hàng	‘management’, ‘technician’,

			‘entrepreneur’, ‘blue-collar’, ‘unknown’, ‘retired’, ‘admin.’, ‘services’, ‘self-employed’, ‘unemployed’, ‘housemaid’, ‘student’
Hôn nhân	Phân loại	Tình trạng hôn nhân của khách hàng	‘divorced’, ‘married’, ‘single’
Học vấn	Phân loại	Trình độ học vấn của khách hàng	‘primary’, ‘secondary’, ‘tertiary’, ‘unknown’
Tình trạng vỡ nợ	Nhị phân	Tín dụng có trong tình trạng vỡ nợ	‘no’, ‘yes’
Số dư	Số	Số dư tài khoản ngân hàng	Giá trị từ -8019 đến 102127
Khoản vay nhà ở	Nhị phân	Có vay tiền để mua nhà không	‘no’, ‘yes’
Khoản vay cá nhân	Nhị phân	Có khoản vay cá nhân nào không	‘no’, ‘yes’
Phương thức liên lạc	Phân loại	Loại phương thức liên lạc	‘cellular’, ‘telephone’, ‘unknown’

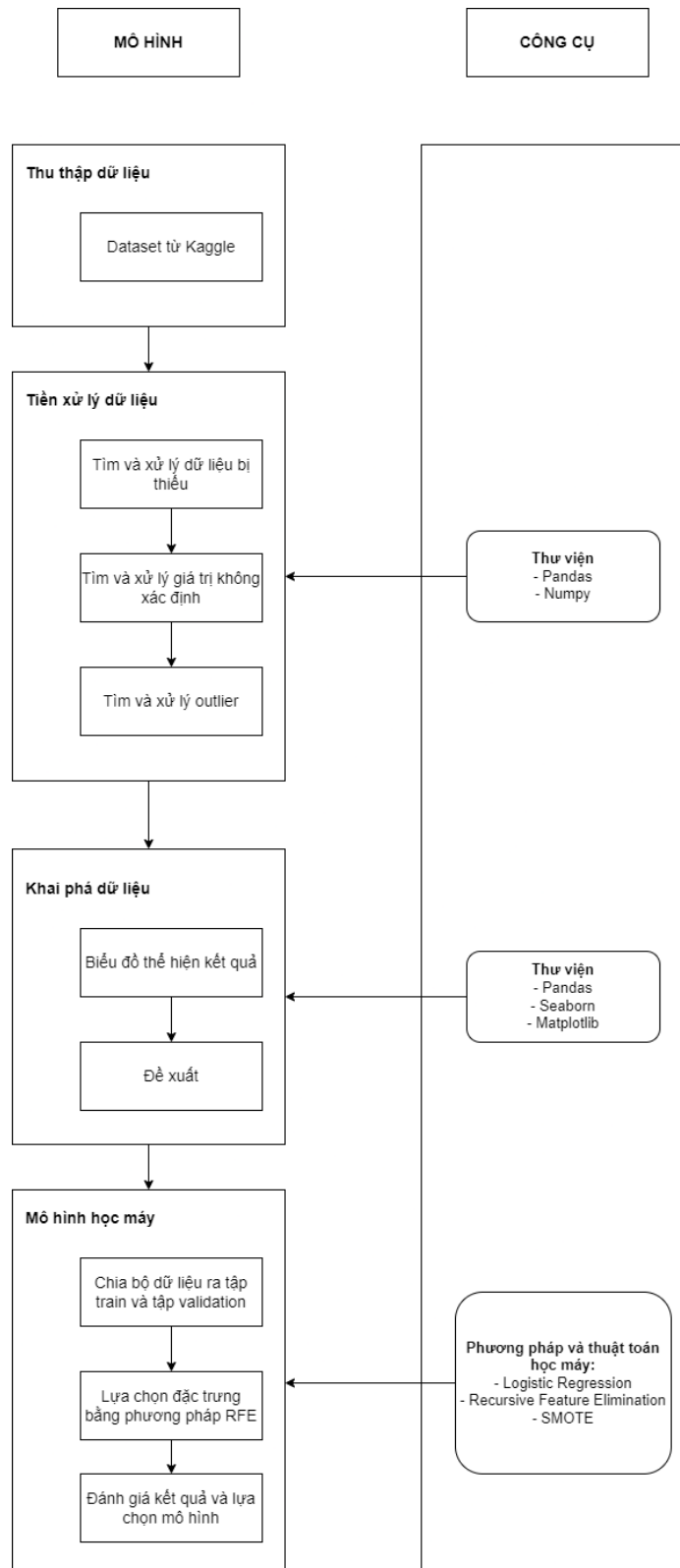
Ngày	Số	Ngày cuối cùng liên lạc trong tháng	Giá trị từ 1 đến 31
Tháng	Phân loại	Tháng cuối cùng liên lạc trong năm	'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'
Thời lượng	Số	Thời lượng cuộc liên lạc cuối (tính giây)	Giá trị từ 0 đến 4918
Chiến dịch hiện tại	Số	Số lượng liên hệ đã thực hiện trong chiến dịch hiện tại cho khách hàng.	Giá trị từ 1 đến 63
Số ngày trước đó	Số	Số ngày kể từ lần cuối liên lạc từ chiến dịch trước (kết quả 999 là khách hàng đã không được liên lạc trước đó)	Giá trị từ -1 đến 871
Chiến dịch trước	Số	Số lượng liên hệ đã thực hiện trong chiến dịch hiện trước cho khách hàng.	Giá trị từ 0 đến 275
Kết quả chiến dịch trước	Phân loại	Kết quả của chiến dịch tiếp thị trước đó	'failure', 'other', 'success', 'unknown'

Đăng ký (biến y)	Nhị phân	Khách hàng có đăng ký tiền gửi có kỳ hạn không?	‘no’, ‘yes’
---------------------	----------	--	-------------

Bảng 1: Bảng thông tin các biến.

I.2. Xây dựng câu hỏi nghiên cứu

- **Câu hỏi 1:** Đặc điểm chung của những nhóm khách hàng chấp nhận sản phẩm và doanh nghiệp nên tập trung nguồn lực vào nhóm khách hàng nào nhiều nhất?
- **Câu hỏi 2:** Những đặc điểm thành công của một chiến dịch tele-marketing khiến cho khách hàng đồng ý sử dụng sản phẩm. Từ đó đưa ra những giải pháp cho các chiến dịch trong tương lai?

I.3. Đề xuất quy trình, sơ đồ thực hiện nghiên cứu

Hình 1: Quy trình thực hiện nghiên cứu

CHƯƠNG II: ĐỊNH NGHĨA CÁC METRICS/VARIABLE

II.1. Chi tiết câu hỏi nghiên cứu

Câu hỏi 1: Doanh nghiệp nên đẩy mạnh tương tác và tập trung phát triển nhóm khách hàng có những đặc điểm cụ thể nào nào để gia tăng tỷ lệ chấp nhận sản phẩm ở các chiến dịch trong tương lai?

- **Câu hỏi nhỏ 1.1:** Ngân hàng đang tập trung tiếp thị vào nhóm khách hàng ở độ tuổi nào và nhóm khách hàng ở độ tuổi nào có xu hướng tham gia sản phẩm tiền gửi của ngân hàng nhiều nhất?
- **Câu hỏi nhỏ 1.2:** Độ tuổi và số dư tài khoản của các nhóm khách hàng khác nhau có mối tương quan gì đến tỷ lệ tham gia sản phẩm không?
- **Câu hỏi nhỏ 1.3:** Liệu rằng với một khách hàng trong tập dữ liệu thì housing, loan có mối liên hệ gì đến với balance của khách hàng đó hay không? Từ đây có thể suy ra điều gì ảnh hưởng đến việc chấp nhận sản phẩm hay không?

Câu hỏi 2: Những đặc điểm thành công của một chiến dịch telemarketing khiến cho khách hàng đồng ý sử dụng sản phẩm. Từ đó đưa ra những giải pháp cho các chiến dịch trong tương lai?

- **Câu hỏi 2.1:** Với những số liệu từ chiến dịch marketing đã thực hiện, thời lượng tối ưu của mỗi cuộc gọi với khách hàng và số lần gọi bao nhiêu mang lại khả năng chấp nhận sản phẩm cao nhất?
- **Câu hỏi 2.2:** Có mối liên hệ gì giữa thời gian thực hiện chiến dịch với kết quả chấp nhận sản phẩm của khách hàng hay không?

II.2. Lựa chọn metrics/variables

Câu hỏi lớn	Câu hỏi nhỏ	Variable	Metric
Đặc điểm chung của những nhóm khách hàng chấp nhận sản phẩm và doanh nghiệp nên tập	Ngân hàng đang tập trung tiếp thị vào nhóm khách hàng ở độ tuổi nào và nhóm khách hàng ở độ tuổi nào có xu hướng	Age	<ul style="list-style-type: none">• Min: 18• Max: 95

<p>trung nguồn lực vào nhóm khách hàng nào nhiều nhất?</p>	<p>tham gia sản phẩm tiền gửi của ngân hàng nhiều nhất?</p>		
	<p>Độ tuổi và số dư tài khoản của các nhóm khách hàng khác nhau có mối tương quan gì đến tỷ lệ tham gia sản phẩm không?</p>	<p>Balance</p>	<ul style="list-style-type: none"> • Min: -8019 • Max: 102127
	<p>Liệu rằng với một khách hàng trong tập dữ liệu thì housing, loan có mối liên hệ gì đến với balance của khách hàng đó hay không? Từ đây có thể suy ra điều gì ảnh hưởng đến việc chấp nhận sản phẩm hay không?</p>	<p>Housing</p>	<p>'yes', 'no', 'unknown'</p>
		<p>Loan</p>	<p>'yes', 'no', 'unknown'</p>
	<p>Trình độ giáo dục có ảnh hưởng đến quyết định lựa chọn sản phẩm tiền gửi của khách hàng hay không?</p>	<p>Education</p>	<p>'primary', 'secondary', 'tertiary' and 'unknown'</p>
<p>Những đặc điểm thành công của một chiến dịch tele-marketing khiến cho khách hàng đồng ý sử dụng sản phẩm. Từ đó</p>	<p>Với những số liệu từ chiến dịch marketing đã thực hiện, thời lượng tối ưu của mỗi cuộc gọi với khách hàng và số lần gọi bao nhiêu mang lại khả</p>	<p>Duration</p>	<ul style="list-style-type: none"> • Min: 0 • Max: 4918
		<p>Campaign</p>	<ul style="list-style-type: none"> • Min: 1 • Max: 63

đưa ra những giải pháp
cho các chiến dịch trong
tương lai?

năng chấp nhận sản phẩm cao
nhất?

Có mối liên hệ gì giữa thời
gian thực hiện chiến dịch với
kết quả chấp nhận sản phẩm
của khách hàng hay không?

Month 'jan', 'feb',
 'mar', ..., 'nov',
 'dec'

Bảng 2: Bảng lựa chọn metrics và variables

II.3. Lựa chọn dữ liệu để trả lời câu hỏi:

III.3.1. Câu hỏi 1:

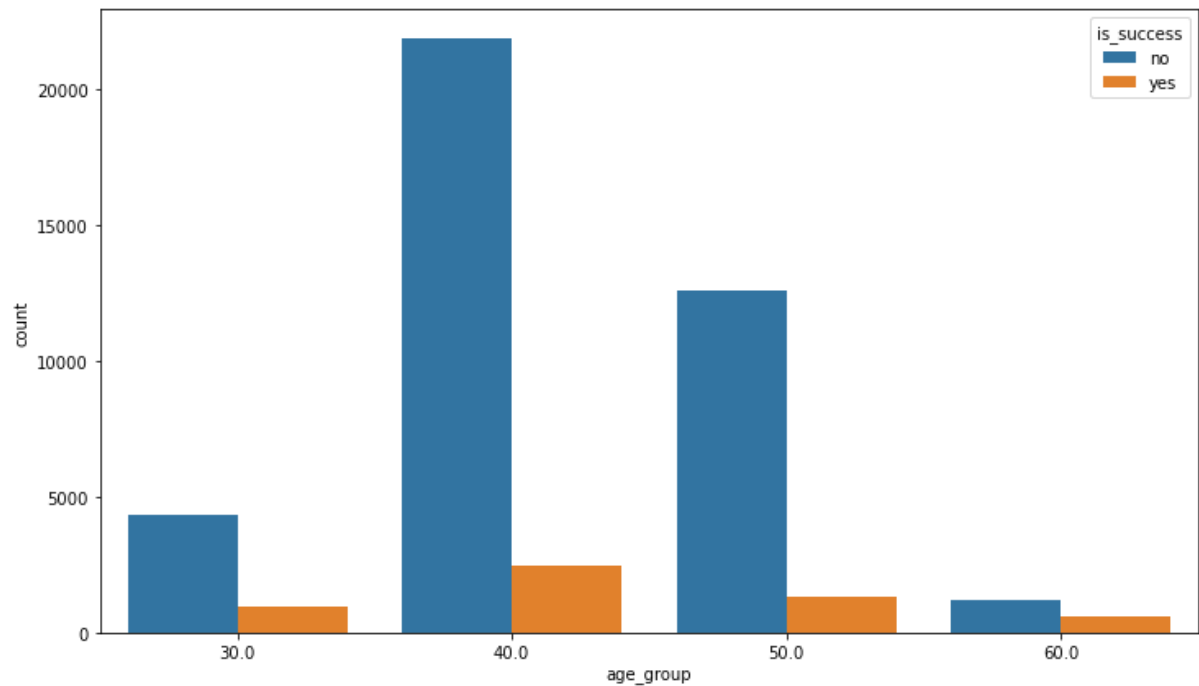
Doanh nghiệp nên đẩy mạnh tương tác và tập trung phát triển nhóm khách hàng có những đặc điểm cụ thể nào nào để gia tăng tỷ lệ chấp nhận sản phẩm ở các chiến dịch trong tương lai?

- **Câu hỏi nhỏ 1.1:** Ngân hàng đang tập trung tiếp thị vào nhóm khách hàng ở độ tuổi nào và nhóm khách hàng ở độ tuổi nào có xu hướng tham gia sản phẩm tiền gửi của ngân hàng nhiều nhất?

```
dataset.age.describe()  
[7] ✓ 0.9s Python  
... count 45211.000000  
mean 40.936210  
std 10.618762  
min 18.000000  
25% 33.000000  
50% 39.000000  
75% 48.000000  
max 95.000000  
Name: age, dtype: float64
```

Hình 2: Mô tả dữ liệu tuổi của khách hàng

Phân tích biến Age của dataset trên cho thấy độ tuổi trung bình của khách hàng được liên hệ trong chiến dịch xấp xỉ 41 tuổi.



Hình 3: Biểu đồ thể tỉ lệ giữa nhóm tuổi và khả năng đăng ký sản phẩm

Nhìn vào biểu đồ và số liệu trên ta có thể kết luận: Nhóm tuổi các khách hàng mà ngân hàng đang tập trung nguồn lực nhiều nhất nằm trong 2 khoản tuổi từ 30 đến dưới 45 tuổi và từ 45 tuổi đến dưới 60.

```
#Tính tỷ lệ thành công của từng nhóm tuổi
print('Success rate and total clients contacted for different age_groups:')
print('Clients age < 30 contacted: {}, Success rate: {}'.format(len(dataset[dataset['age_group'] == 30]),
dataset[dataset['age_group'] == 30].is_success.value_counts()[1]/len(dataset[dataset['age_group'] == 30])))

print('Clients of age 30-45 contacted: {}, Success rate: {}'.format(len(dataset[dataset['age_group'] == 40]),
dataset[dataset['age_group'] == 40].is_success.value_counts()[1]/len(dataset[dataset['age_group'] == 40])))

print('Clients of age 45-60 contacted: {}, Success rate: {}'.format(len(dataset[dataset['age_group'] == 50]),
dataset[dataset['age_group'] == 50].is_success.value_counts()[1]/len(dataset[dataset['age_group'] == 50])))

print('Clients of 60+ age contacted: {}, Success rate: {}'.format(len(dataset[dataset['age_group'] == 60]),
dataset[dataset['age_group'] == 60].is_success.value_counts()[1]/len(dataset[dataset['age_group'] == 60])))
```

[26] ✓ 0.1s Python

```
... Success rate and total clients contacted for different age_groups:
Clients age < 30 contacted: 5273, Success rate: 0.1759908970225678
Clients of age 30-45 contacted: 24274, Success rate: 0.10117821537447474
Clients of age 45-60 contacted: 13880, Success rate: 0.09402017291066282
Clients of 60+ age contacted: 1784, Success rate: 0.336322869955157
```

Hình 4: Tỷ lệ thành công và tổng số khách hàng liên lạc theo từng nhóm tuổi

```

for x in range(95, 101, 1):
    print("{}% of people having age are less than equal to {}".format(x, dataset.age.quantile(x/100)))
iqr = dataset.age.quantile(0.75) - dataset.age.quantile(0.25)
print('IQR {}'.format(iqr))

```

[8] ✓ 0.9s Python

```

... 95% of people having age are less than equal to 59.0
    96% of people having age are less than equal to 59.0
    97% of people having age are less than equal to 60.0
    98% of people having age are less than equal to 63.0
    99% of people having age are less than equal to 71.0
    100% of people having age are less than equal to 95.0
    IQR 15.0

```

Hình 5: Phân bố nhóm tuổi trong bộ dữ liệu

Tuy vậy, khi nhìn vào tỷ lệ thành công của từng nhóm tuổi (tức tỷ lệ người trong nhóm tuổi đó quyết định sử dụng sản phẩm) lại có một nghịch lý vô cùng lớn. Nhóm tuổi trên 60 tuổi chiếm trong tập dữ liệu vô cùng thấp (chỉ 3%) nhưng lại có tỷ lệ thành công vô cùng cao (33.6%) xấp xỉ gấp đôi so với nhóm tuổi dưới 30 tuổi (17.6%) và lớn hơn nhiều so với các nhóm tuổi còn lại (lần lượt 10,1% cho nhóm tuổi từ 30 - 45 và 9,4% cho nhóm tuổi từ 45 - 60).

Kết luận: Độ tuổi đăng ký nhiều nhất là từ 60 trở lên, được giải thích bởi lý do khi ngoài 60 ta có xu hướng tiết kiệm hơn sau khi nghỉ hưu so với trung niên, họ có xu hướng tích cực hơn với mục tiêu là tạo thu nhập đầu tư cao. Tiền gửi có kỳ hạn là đầu tư ít rủi ro nên được người lớn tuổi ưu tiên. Vì thế ngân hàng nên tập trung nguồn lực vào nhóm này hơn.

- **Câu hỏi nhỏ 1.2:** Độ tuổi và số dư tài khoản của các nhóm khách hàng khác nhau có mối tương quan gì đến tỷ lệ tham gia sản phẩm không?

Phân tích dữ liệu về balance:

```

dataset_new.balance.describe()

```

[50] ✓ 0.1s Python

```

... count    45211.000000
    mean      1074.055318
    std       1708.752554
    min       -6847.000000
    25%         72.000000
    50%        448.000000
    75%       1322.000000
    max       10483.000000
    Name: balance, dtype: float64

```

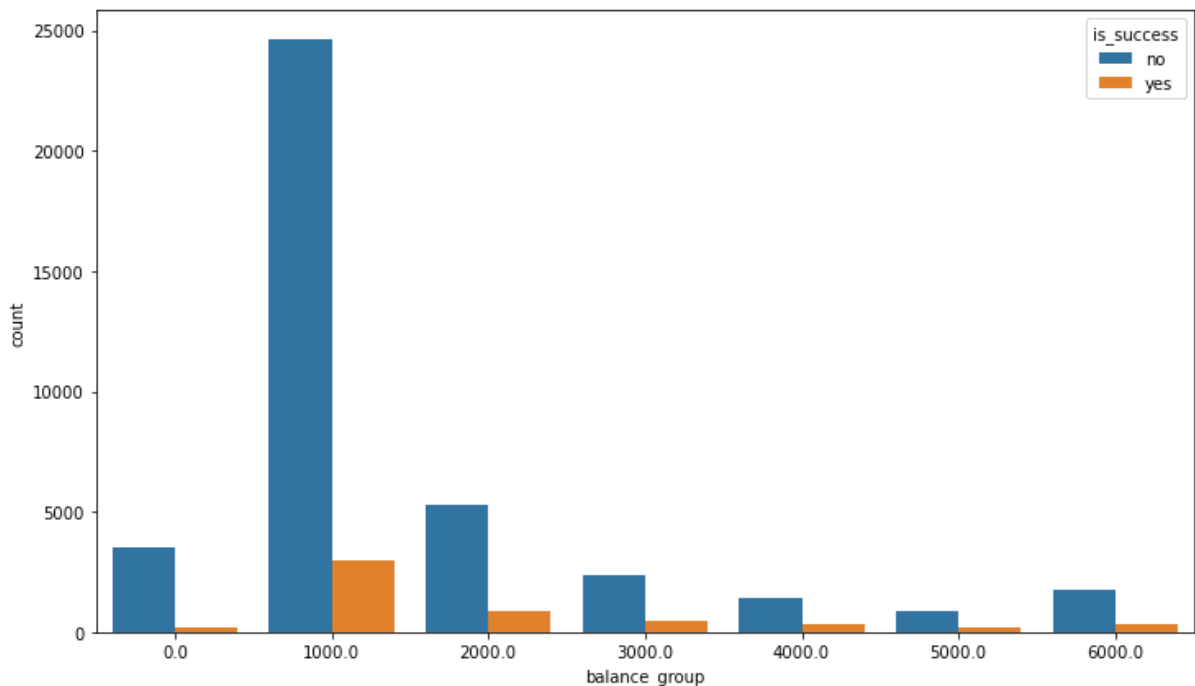
Hình 6: Thống kê mô tả của biến "balance"

Sau khi loại bỏ các outliers, ta có dữ liệu về balance hợp lý hơn, với trung bình balance rơi vào khoản 1074.

```
[76] lst = [dataset_new]
for column in lst:
    column.loc[column["balance"] < 0, 'balance_group'] = 0
    column.loc[(column["balance"] >= 0) & (column["balance"] < 1000), 'balance_group'] = 1000
    column.loc[(column["balance"] >= 1000) & (column["balance"] < 2000), 'balance_group'] = 2000
    column.loc[(column["balance"] >= 2000) & (column["balance"] < 3000), 'balance_group'] = 3000
    column.loc[(column["balance"] >= 3000) & (column["balance"] < 4000), 'balance_group'] = 4000
    column.loc[(column["balance"] >= 4000) & (column["balance"] < 5000), 'balance_group'] = 5000
    column.loc[column["balance"] >= 5000, 'balance_group'] = 6000
count_balance_response_pct = pd.crosstab(dataset_new['is_success'],
dataset_new['balance_group']).apply(lambda x: x/x.sum() * 100)
count_balance_response_pct = count_balance_response_pct.transpose()

[77] sns.countplot(x='balance_group', data=dataset_new, hue='is_success');
Python
```

Hình 7: Phân nhóm balance.



Hình 8: Biểu đồ thể hiện tỷ lệ thành công theo từng balance.

Biểu đồ trên cho thấy lượng khách hàng được liên hệ trong chiến dịch này đa số có số balance nằm trong khoản từ 0 - 1000. Tuy vậy khi xét đến tỷ lệ thành công, giá trị này có xu hướng ngày càng tăng khi nhóm khách hàng có số balance tăng lên (đạt mức tối đa ở 4000) và giảm nhẹ khi mức balance tăng thêm (biểu đồ hình dưới).

```

print('Success rate for different balance_group:')
print('Balance < 0: {}, Success rate: {}'.format(len(dataset_new[dataset_new['balance_group'] == 0]),
dataset_new[dataset_new['balance_group'] == 0].is_success.value_counts()[1]/len(dataset_new[dataset_new['balance_group'] == 0])))
print('Balance < 1000: {}, Success rate: {}'.format(len(dataset_new[dataset_new['balance_group'] == 1000]),
dataset_new[dataset_new['balance_group'] == 1000].is_success.value_counts()[1]/len(dataset_new[dataset_new['balance_group'] == 1000])))

print('Balance < 2000: {}, Success rate: {}'.format(len(dataset_new[dataset_new['balance_group'] == 2000]),
dataset_new[dataset_new['balance_group'] == 2000].is_success.value_counts()[1]/len(dataset_new[dataset_new['balance_group'] == 2000])))
print('Balance < 3000: {}, Success rate: {}'.format(len(dataset_new[dataset_new['balance_group'] == 3000]),
dataset_new[dataset_new['balance_group'] == 3000].is_success.value_counts()[1]/len(dataset_new[dataset_new['balance_group'] == 3000])))
print('Balance < 4000: {}, Success rate: {}'.format(len(dataset_new[dataset_new['balance_group'] == 4000]),
dataset_new[dataset_new['balance_group'] == 4000].is_success.value_counts()[1]/len(dataset_new[dataset_new['balance_group'] == 4000])))
print('Balance < 5000: {}, Success rate: {}'.format(len(dataset_new[dataset_new['balance_group'] == 5000]),
dataset_new[dataset_new['balance_group'] == 5000].is_success.value_counts()[1]/len(dataset_new[dataset_new['balance_group'] == 5000])))
print('Balance > 6000: {}, Success rate: {}'.format(len(dataset_new[dataset_new['balance_group'] == 6000]),
dataset_new[dataset_new['balance_group'] == 6000].is_success.value_counts()[1]/len(dataset_new[dataset_new['balance_group'] == 6000])))

[78] Python

... Success rate for different balance_group:
Balance < 0: 3765, Success rate: 0.055776892430278883
Balance < 1000: 27548, Success rate: 0.10697691302453899
Balance < 2000: 6136, Success rate: 0.13754889178617993
Balance < 3000: 2891, Success rate: 0.17018332756831547
Balance < 4000: 1716, Success rate: 0.17424242424242425
Balance < 5000: 1052, Success rate: 0.1682509505703422
Balance > 6000: 2103, Success rate: 0.15216357584403234

```

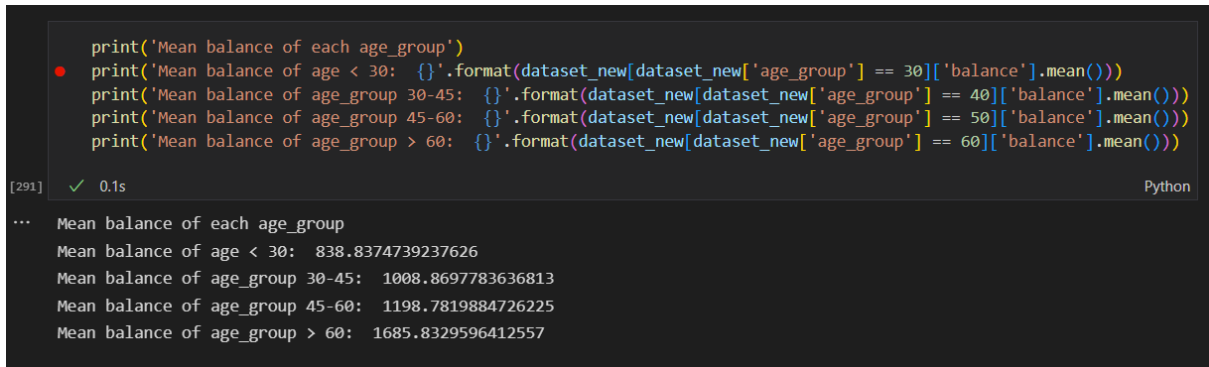
Hình 9: Tỷ lệ thành công theo từng nhóm balance

Phân tích mối liên hệ giữa tuổi và balance:



Hình 10: Mối liên hệ giữa tuổi và số dư tài khoản

Biểu đồ trên cho thấy một điều vô cùng rõ ràng rằng đa số khách hàng có độ tuổi trên 60 đều có số balance không âm.



```
print('Mean balance of each age_group')
print('Mean balance of age < 30: {}'.format(dataset_new[dataset_new['age_group'] == 30]['balance'].mean()))
print('Mean balance of age_group 30-45: {}'.format(dataset_new[dataset_new['age_group'] == 40]['balance'].mean()))
print('Mean balance of age_group 45-60: {}'.format(dataset_new[dataset_new['age_group'] == 50]['balance'].mean()))
print('Mean balance of age_group > 60: {}'.format(dataset_new[dataset_new['age_group'] == 60]['balance'].mean()))
```

[291] ✓ 0.1s Python

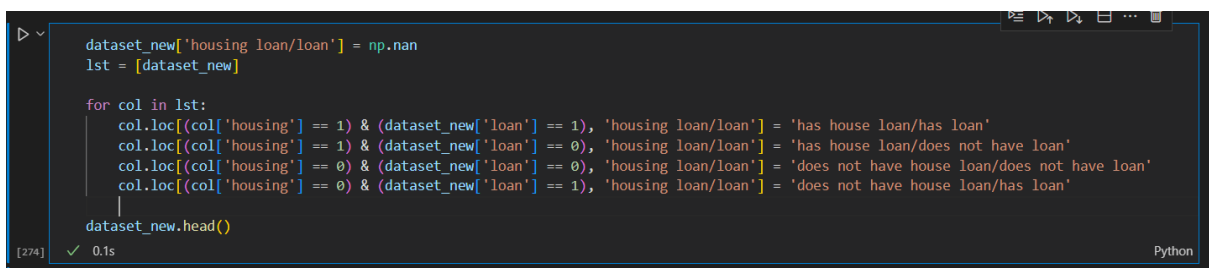
```
... Mean balance of each age_group
Mean balance of age < 30: 838.8374739237626
Mean balance of age_group 30-45: 1008.8697783636813
Mean balance of age_group 45-60: 1198.7819884726225
Mean balance of age_group > 60: 1685.8329596412557
```

Hình 11: Trung bình số dư theo từng nhóm tuổi

Tính trung bình balance của từng nhóm tuổi cũng cho thấy: Đa số khách hàng ở nhóm tuổi càng lớn thì có số balance càng cao.

Kết luận: sau khi phân tích mối liên hệ giữa biến balance, age và kết quả cho thấy khách hàng có số balance càng cao có xu hướng đăng ký sản phẩm nhiều hơn, và nhóm có số balance trung bình cao nhất tập trung vào nhóm người trên 60 tuổi. Vì thế càng củng cố cho kết luận phía trên rằng ngân hàng nên tập trung nhiều hơn vào nhóm trên 60 tuổi.

- **Câu hỏi nhỏ 1.3:** Liệu rằng với một khách hàng trong tập dữ liệu thì housing, loan có mối liên hệ gì đến với balance của khách hàng đó hay không? Từ đây có thể suy ra điều gì ảnh hưởng đến việc chấp nhận sản phẩm hay không?



```
dataset_new['housing loan/loan'] = np.nan
lst = [dataset_new]

for col in lst:
    col.loc[(col['housing'] == 1) & (dataset_new['loan'] == 1), 'housing loan/loan'] = 'has house loan/has loan'
    col.loc[(col['housing'] == 1) & (dataset_new['loan'] == 0), 'housing loan/loan'] = 'has house loan/does not have loan'
    col.loc[(col['housing'] == 0) & (dataset_new['loan'] == 0), 'housing loan/loan'] = 'does not have house loan/does not have loan'
    col.loc[(col['housing'] == 0) & (dataset_new['loan'] == 1), 'housing loan/loan'] = 'does not have house loan/has loan'

dataset_new.head()
```

[274] ✓ 0.1s Python

Hình 12: Tình trạng nợ mua nhà/ tình trạng nợ cá nhân



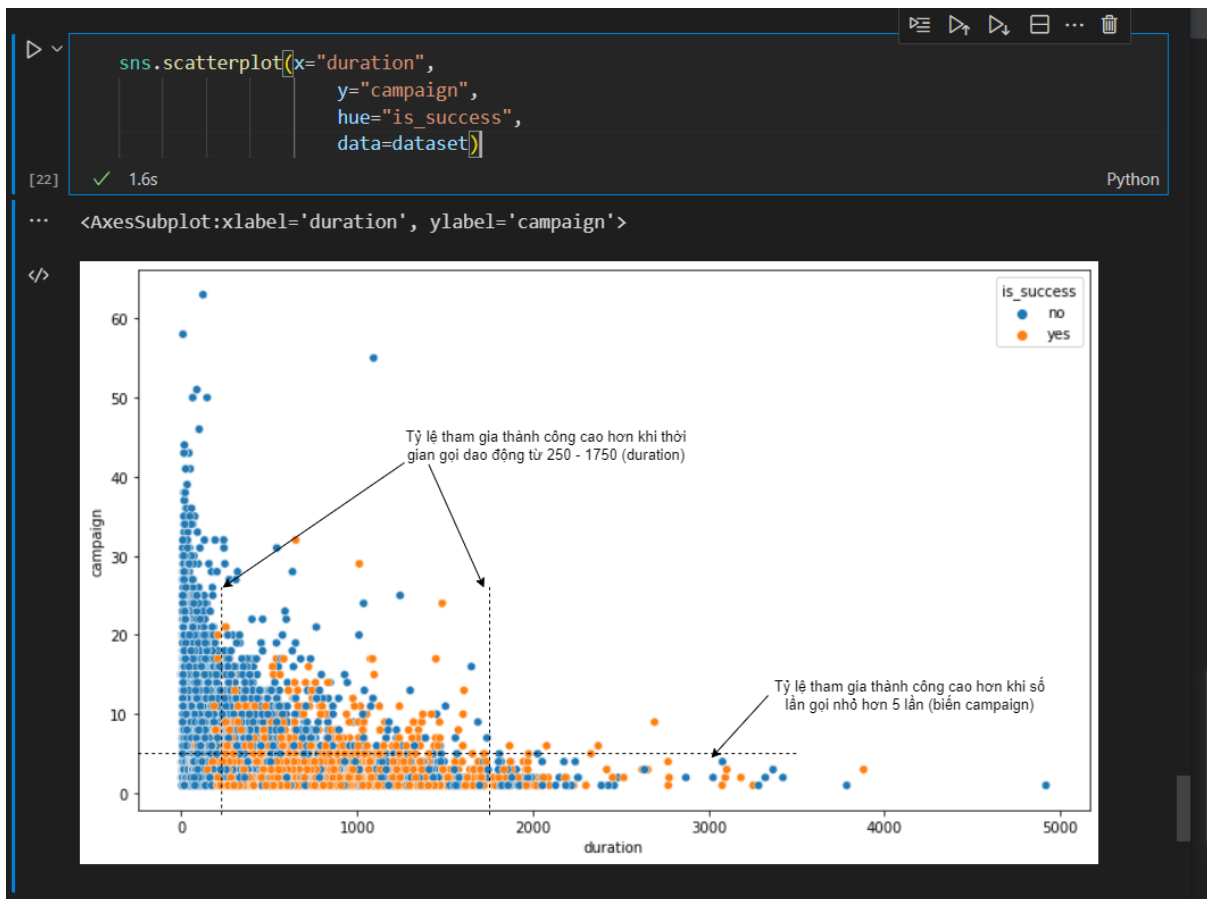
Hình 13: Biểu đồ thể hiện số dư trung bình theo từng nhóm nợ

Kết luận: nhóm khách hàng không có khoản loan hay house loan nào có số balance trung bình cao nhất, tiếp đến nhóm chỉ có khoản housing loan -> nên tập trung vào 2 nhóm này.

III.3.2. Câu hỏi 2:

Những đặc điểm thành công của một chiến dịch telemarketing khiến cho khách hàng đồng ý sử dụng sản phẩm. Từ đó đưa ra những giải pháp cho các chiến dịch trong tương lai?

Câu hỏi 2.1: Với những số liệu từ chiến dịch marketing đã thực hiện, thời lượng tối ưu của mỗi cuộc gọi với khách hàng và số lần gọi bao nhiêu mang lại khả năng chấp nhận sản phẩm cao nhất?



Hình 14: Biểu đồ phân bố tỷ lệ thành công theo duration và campaign.

Kết luận:

- Nhìn vào biểu đồ trên có thể thấy khi thời gian gọi từ 250 giây (4,1 phút) và 1750 giây (29,2 phút) và số lần gọi nhỏ hơn 5 lần thì hầu hết khách hàng sẽ đăng ký tham gia gửi tiền có kỳ hạn.
- Tuy nhiên khi số lần gọi càng nhiều thì khả năng khách hàng tham gia càng thấp => đồng nghĩa với việc khách hàng đang cảm thấy phiền hà khi bị ngân hàng làm phiền nhiều lần => không tham gia

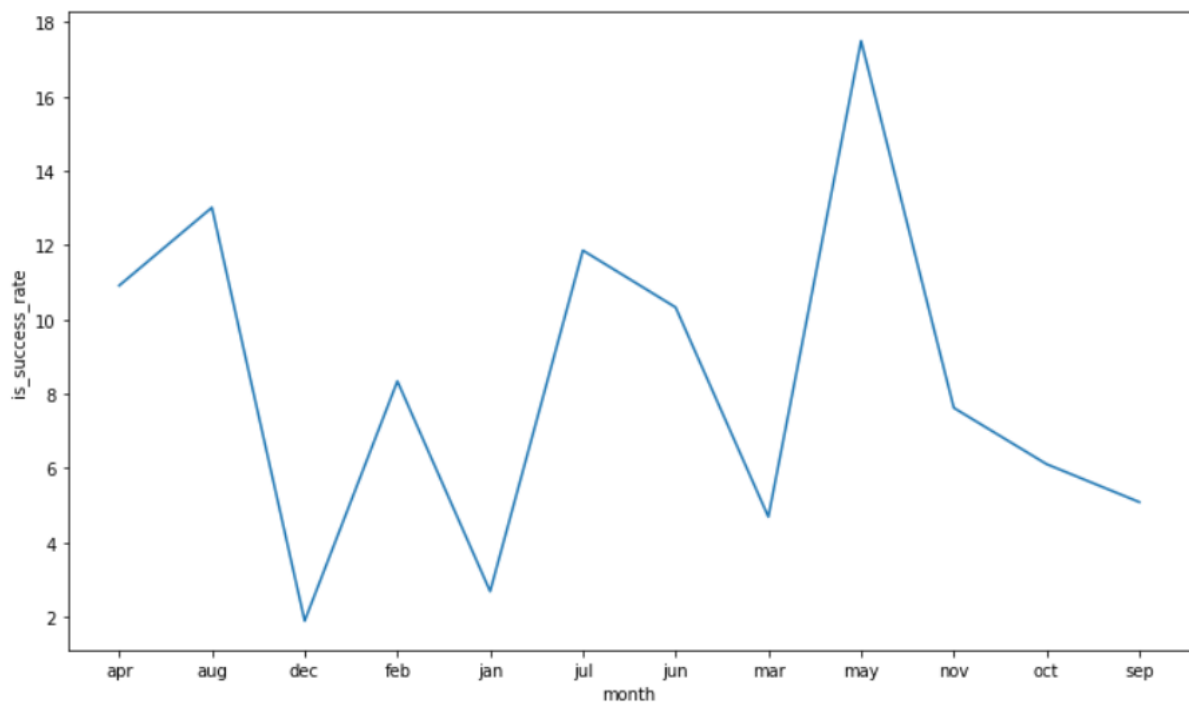
=> Ngân hàng cần điều chỉnh lại số lần gọi và thời gian của các cuộc gọi sao cho hợp lý, và lý tưởng nhất là: số lần gọi là từ 2->5 lần và thời gian cho mỗi lần gọi là từ 5->15 phút.

Câu hỏi 2.2: Có mối liên hệ gì giữa các tháng thực hiện chiến dịch với kết quả chấp nhận sản phẩm của khách hàng hay không?

```
[153] df_success
```

	is_success count	is_success_rate
month		
apr	577	10.909435
aug	688	13.008130
dec	100	1.890717
feb	441	8.338060
jan	142	2.684818
jul	627	11.854793
jun	546	10.323313
mar	248	4.688977
may	925	17.489128
nov	403	7.619588
oct	323	6.107015
sep	269	5.086028

Hình 15: Bảng thống kê tỷ lệ thành công theo tháng.



Hình 16: Biểu đồ thể hiện tỷ lệ thành công theo từng tháng

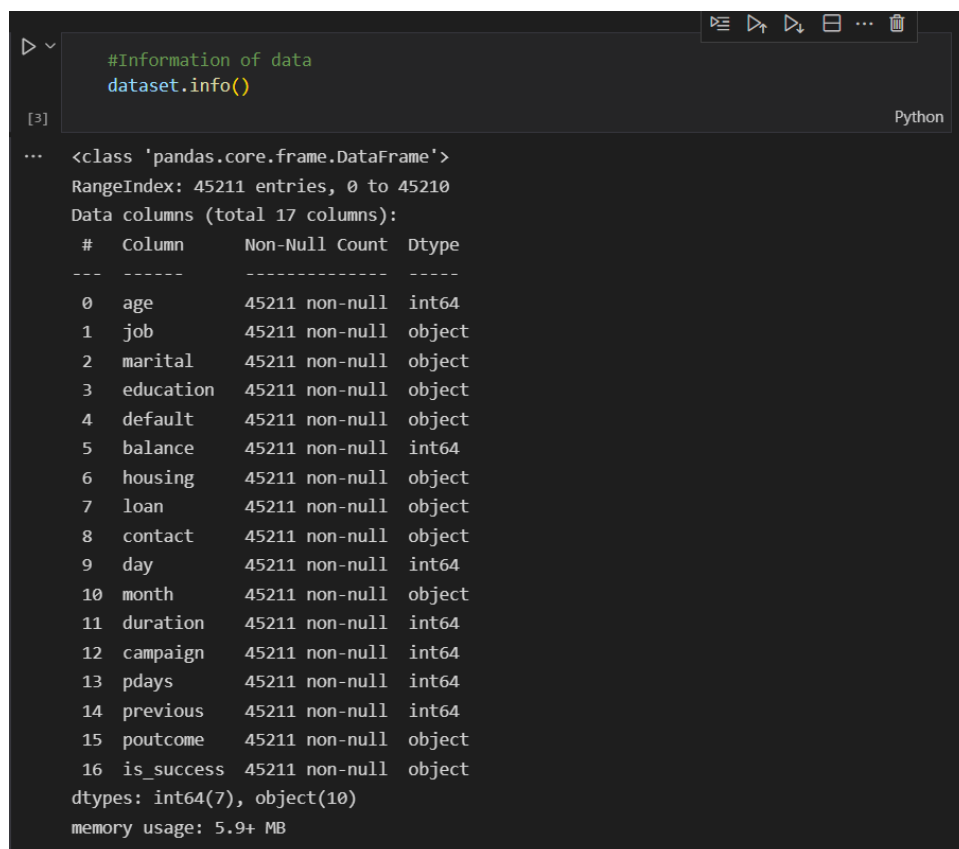
Kết luận:

- Ngân hàng liên hệ với khách hàng hầu hết là tháng tháng 5, 8, đồng nghĩa với tỷ lệ đăng ký cao.
- Tỷ lệ đăng ký ở tháng 1,12,3 thấp hơn => cho thấy ngân hàng đang đi sai đường, ngân hàng nên chuyển thời gian tiếp thị vào mùa xuân Tuy nhiên ngân hàng nên cân trọng vì yếu tố này mang tính chất thời gian nên có thể thay đổi theo từng năm

CHƯƠNG III: PHÂN TÍCH KHÁM PHÁ DỮ LIỆU (EDA) BIẾN PHỤ THUỘC VỚI PHƯƠNG PHÁP DESCRIPTIVE ANALYTICS

III.1. Clean data

- Trước khi tiến hành EDA, chúng ta kiểm tra tập dữ liệu có giá trị nào bị miss hay không?.



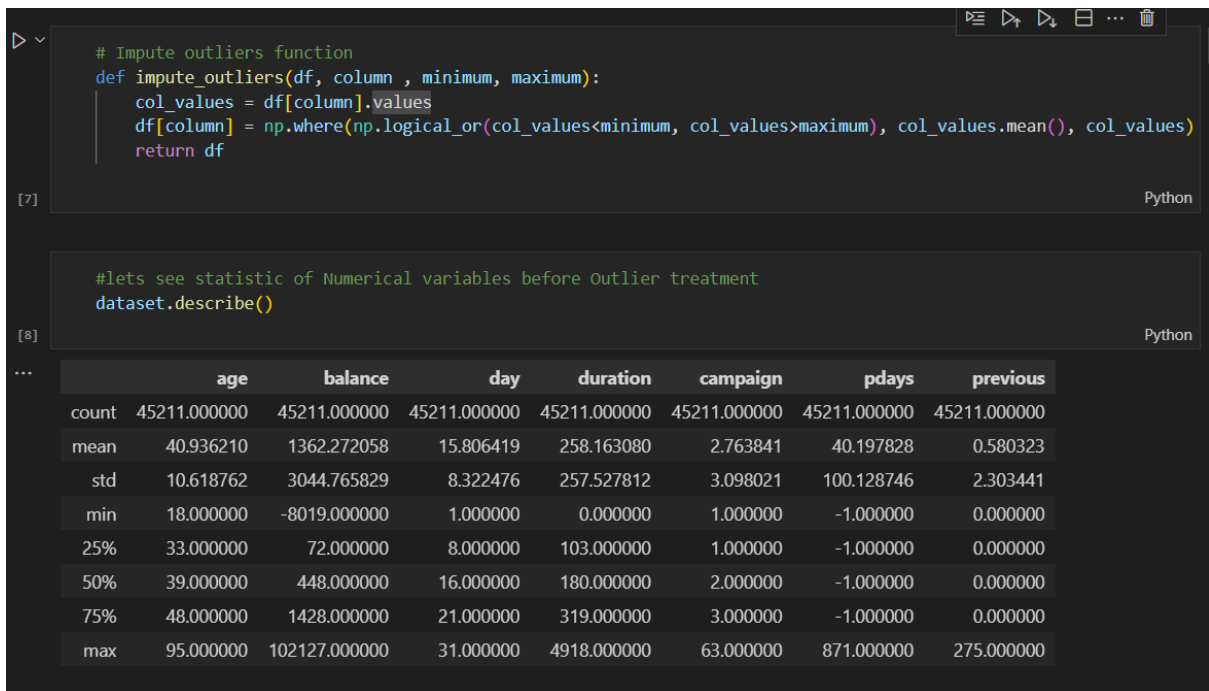
```
#Information of data
dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 17 columns):
 #   Column          Non-Null Count  Dtype  
---  --
 0   age             45211 non-null  int64  
 1   job             45211 non-null  object  
 2   marital         45211 non-null  object  
 3   education       45211 non-null  object  
 4   default         45211 non-null  object  
 5   balance         45211 non-null  int64  
 6   housing         45211 non-null  object  
 7   loan            45211 non-null  object  
 8   contact         45211 non-null  object  
 9   day             45211 non-null  int64  
10  month           45211 non-null  object  
11  duration        45211 non-null  int64  
12  campaign        45211 non-null  int64  
13  pdays          45211 non-null  int64  
14  previous        45211 non-null  int64  
15  poutcome       45211 non-null  object  
16  is_success      45211 non-null  object  
dtypes: int64(7), object(10)
memory usage: 5.9+ MB
```

Hình 17: Kiểm tra dữ liệu

=> Kết quả trên cho thấy dữ liệu không bị mất.

- Kiểm tra outliers:



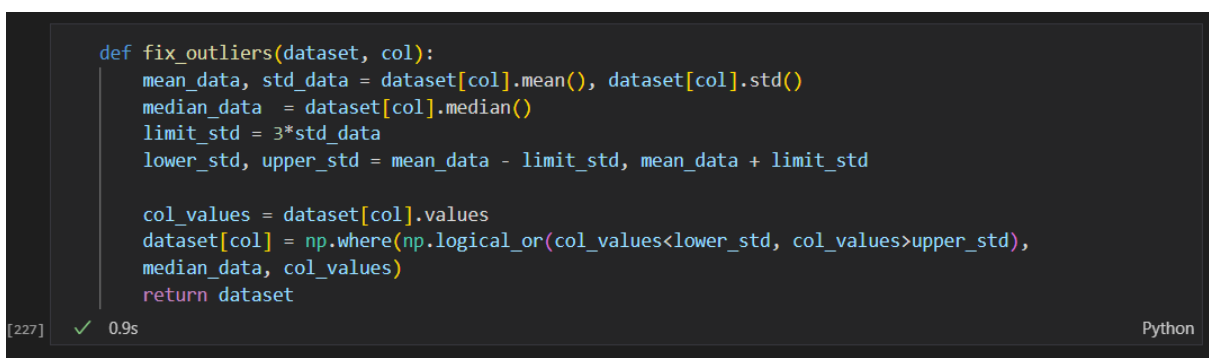
```
# Impute outliers function
def impute_outliers(df, column , minimum, maximum):
    col_values = df[column].values
    df[column] = np.where(np.logical_or(col_values<minimum, col_values>maximum), col_values.mean(), col_values)
    return df
```

```
#lets see statistic of Numerical variables before Outlier treatment
dataset.describe()
```

	age	balance	day	duration	campaign	pdays	previous
count	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000
mean	40.936210	1362.272058	15.806419	258.163080	2.763841	40.197828	0.580323
std	10.618762	3044.765829	8.322476	257.527812	3.098021	100.128746	2.303441
min	18.000000	-8019.000000	1.000000	0.000000	1.000000	-1.000000	0.000000
25%	33.000000	72.000000	8.000000	103.000000	1.000000	-1.000000	0.000000
50%	39.000000	448.000000	16.000000	180.000000	2.000000	-1.000000	0.000000
75%	48.000000	1428.000000	21.000000	319.000000	3.000000	-1.000000	0.000000
max	95.000000	102127.000000	31.000000	4918.000000	63.000000	871.000000	275.000000

Hình 18: Thống kê mô tả các biến numerical

=> Từ bảng trên ta có thể thấy, các biến ('balance', 'duration', 'campaign', 'pdays', 'previous') có những giá trị outliers. Vì thế ta cần xây dựng hàm loại bỏ outliers như hình bên dưới:



```
def fix_outliers(dataset, col):
    mean_data, std_data = dataset[col].mean(), dataset[col].std()
    median_data = dataset[col].median()
    limit_std = 3*std_data
    lower_std, upper_std = mean_data - limit_std, mean_data + limit_std

    col_values = dataset[col].values
    dataset[col] = np.where(np.logical_or(col_values<lower_std, col_values>upper_std),
                             median_data, col_values)
    return dataset
```

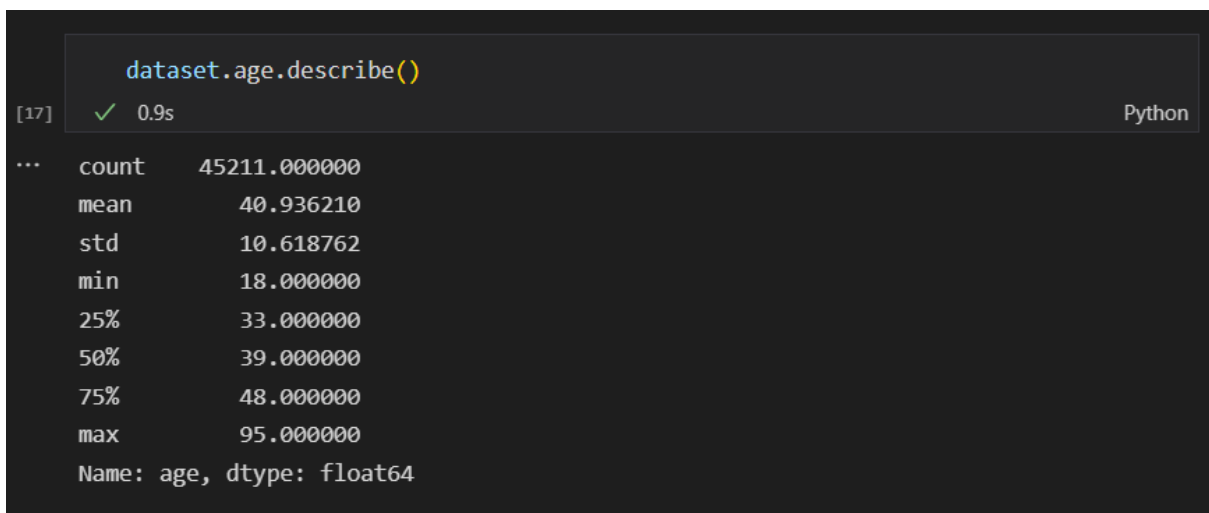
Hình 19: Hàm xử lý outliers

III.2. Phân tích khám phá dữ liệu (EDA)

III.2.1. Age

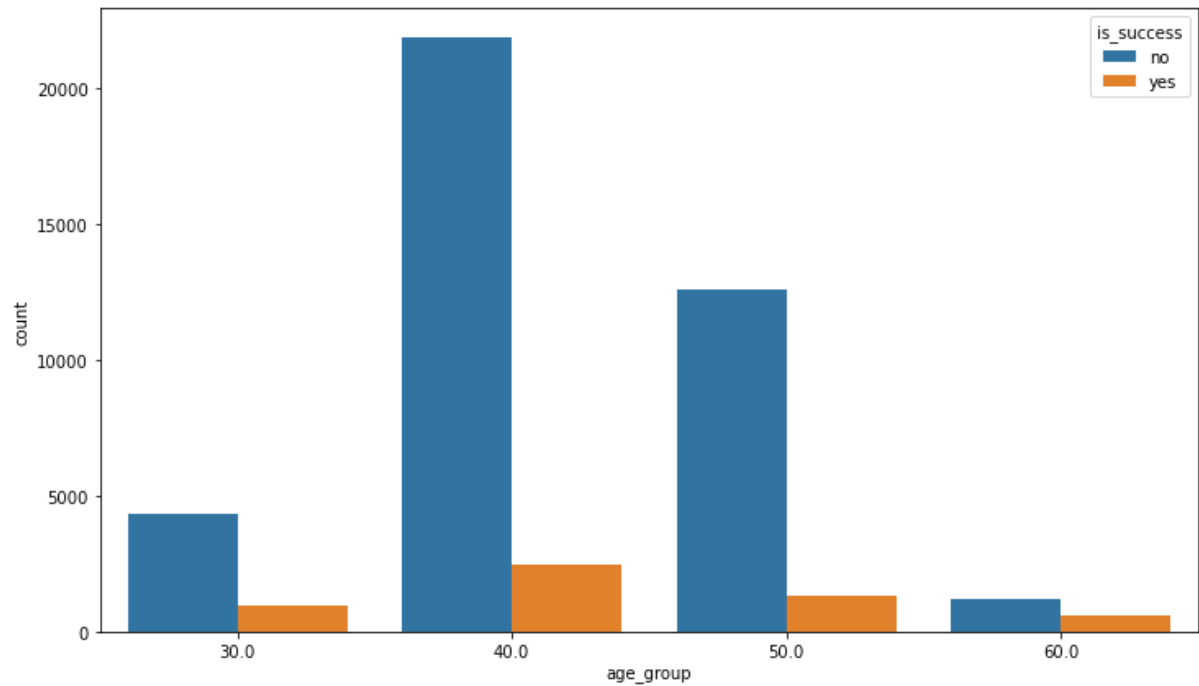


Hình 20: Biểu đồ boxplot giữa tuổi và biến phụ thuộc



Hình 21: Thống kê mô tả biến tuổi

Độ tuổi trung bình của các khách hàng được liên hệ xấp xỉ 41 tuổi, khách hàng lớn tuổi nhất là 95 tuổi và nhỏ nhất là 18 tuổi.



Hình 22: Mối liên hệ giữa nhóm tuổi và biến phụ thuộc

```

for x in range(95, 101, 1):
    print("{}% of people having age are less than equal to {}".format(x, dataset.age.quantile(x/100)))
iqr = dataset.age.quantile(0.75) - dataset.age.quantile(0.25)
print('IQR {}'.format(iqr))

```

[8] Python

```

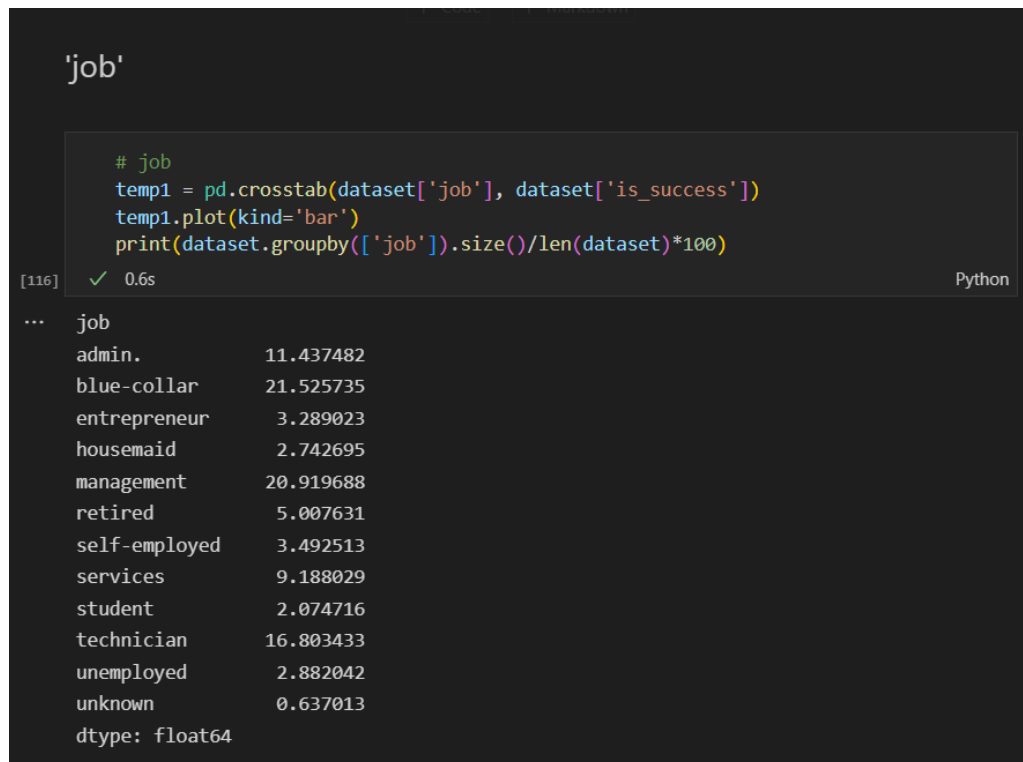
... 95% of people having age are less than equal to 59.0
... 96% of people having age are less than equal to 59.0
... 97% of people having age are less than equal to 60.0
... 98% of people having age are less than equal to 63.0
... 99% of people having age are less than equal to 71.0
... 100% of people having age are less than equal to 95.0
... IQR 15.0

```

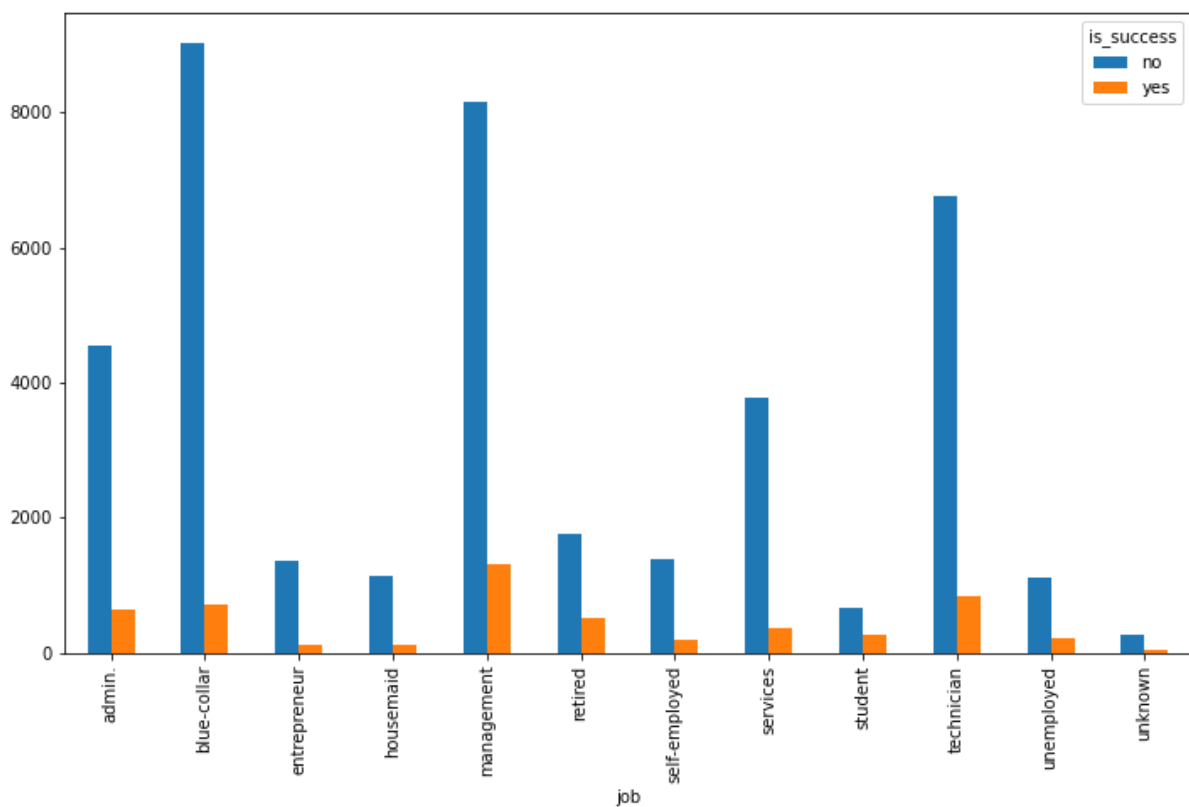
Hình 23: Phân bố nhóm tuổi trong bộ dữ liệu

Như kết luận ở Chương 2, nhóm khách hàng có tỷ lệ thành công cao nhất là nhóm có độ tuổi lớn hơn 60 tuổi nhưng lại có tỷ lệ liên lạc thấp nhất (chỉ 3%)

III.2.2. Job



Hình 24: Tỷ lệ thành công theo từng nhóm ngành việc làm.



Hình 25: Biểu đồ thể hiện mối liên hệ giữa nhóm việc làm và biến phụ thuộc.

Như chúng ta có thể nhìn thấy trên biểu đồ trên, những khách hàng liên lạc cao nhất làm các nghề: 'blue-collar', 'management' & 'technician'.

```

table = PrettyTable(['Job', 'Total Clients', 'Success rate'])
table.add_row(['Blue-collar', len(dataset[dataset['job'] == 'blue-collar']), dataset[dataset['job'] == 'blue-collar'].is_success.value_counts()[1]/len(dataset[dataset['job'] == 'blue-collar'])])
table.add_row(['Management', len(dataset[dataset['job'] == 'management']), dataset[dataset['job'] == 'management'].is_success.value_counts()[1]/len(dataset[dataset['job'] == 'management'])])
table.add_row(['Technician', len(dataset[dataset['job'] == 'technician']), dataset[dataset['job'] == 'technician'].is_success.value_counts()[1]/len(dataset[dataset['job'] == 'technician'])])
table.add_row(['Admin', len(dataset[dataset['job'] == 'admin.']), dataset[dataset['job'] == 'admin.'].is_success.value_counts()[1]/len(dataset[dataset['job'] == 'admin.'])])
table.add_row(['Services', len(dataset[dataset['job'] == 'services']), dataset[dataset['job'] == 'services'].is_success.value_counts()[1]/len(dataset[dataset['job'] == 'services'])])
table.add_row(['Retired', len(dataset[dataset['job'] == 'retired']), dataset[dataset['job'] == 'retired'].is_success.value_counts()[1]/len(dataset[dataset['job'] == 'retired'])])
table.add_row(['Self-employed', len(dataset[dataset['job'] == 'self-employed']), dataset[dataset['job'] == 'self-employed'].is_success.value_counts()[1]/len(dataset[dataset['job'] == 'self-employed'])])
table.add_row(['Entrepreneur', len(dataset[dataset['job'] == 'entrepreneur']), dataset[dataset['job'] == 'entrepreneur'].is_success.value_counts()[1]/len(dataset[dataset['job'] == 'entrepreneur'])])
table.add_row(['Unemployed', len(dataset[dataset['job'] == 'unemployed']), dataset[dataset['job'] == 'unemployed'].is_success.value_counts()[1]/len(dataset[dataset['job'] == 'unemployed'])])
table.add_row(['Housemaid', len(dataset[dataset['job'] == 'housemaid']), dataset[dataset['job'] == 'housemaid'].is_success.value_counts()[1]/len(dataset[dataset['job'] == 'housemaid'])])
table.add_row(['Student', len(dataset[dataset['job'] == 'student']), dataset[dataset['job'] == 'student'].is_success.value_counts()[1]/len(dataset[dataset['job'] == 'student'])])
table.add_row(['Unknown', len(dataset[dataset['job'] == 'unknown']), dataset[dataset['job'] == 'unknown'].is_success.value_counts()[1]/len(dataset[dataset['job'] == 'unknown'])])
print(table)

```

[117] ✓ 0.4s Python

Hình 26: Xử lý tỷ lệ thành công theo nhóm ngành

```

'''
+-----+-----+-----+
|      Job      | Total Clients | Success rate |
+-----+-----+-----+
| Blue-collar   | 9732         | 0.07274969173859433 |
| Management    | 9458         | 0.13755550856417847 |
| Technician    | 7597         | 0.11056996182703699 |
| Admin         | 5171         | 0.12202668729452718 |
| Services      | 4154         | 0.08883004333172845 |
| Retired       | 2264         | 0.22791519434628976 |
| Self-employed | 1579         | 0.11842938568714376 |
| Entrepreneur  | 1487         | 0.08271687962340282 |
| Unemployed    | 1303         | 0.15502686108979277 |
| Housemaid     | 1240         | 0.08790322580645162 |
| Student       | 938          | 0.2867803837953092  |
| Unknown       | 288          | 0.11805555555555555 |
+-----+-----+-----+
'''

```

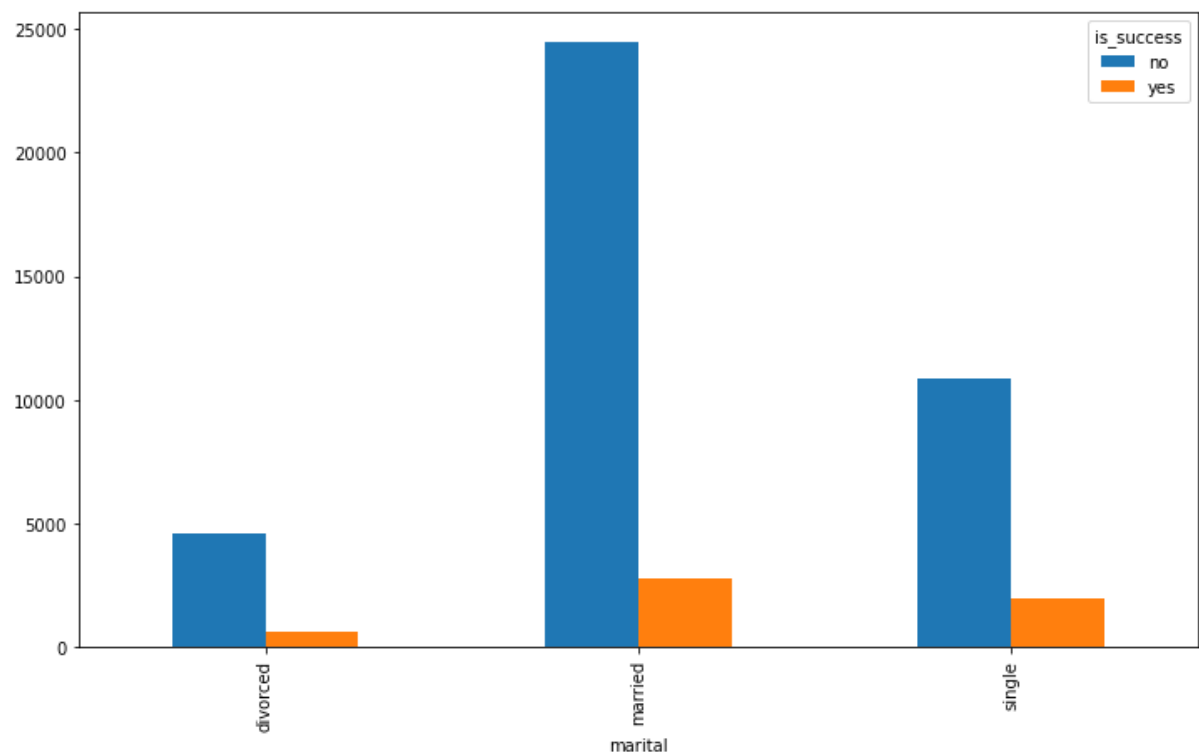
Hình 27: Bảng kết quả tỷ lệ thành công theo nhóm ngành.

Có thể thấy, tỷ lệ thành công cao nhất nằm ở nhóm 'Student'.

III.2.3. Marital



Hình 28: Tỷ lệ thành công theo tình trạng hôn nhân



Hình 29: Biểu đồ thể hiện mối liên hệ giữa tỷ lệ thành công và tình trạng hôn nhân

```

table = PrettyTable(['Marital', 'Total Clients', 'Success rate'])
table.add_row(['Married', len(dataset[dataset['marital'] == 'married']), dataset[dataset['marital'] == 'married'].is_success.value_counts()[1]/len(dataset[dataset['marital'] == 'married'])])
table.add_row(['Single', len(dataset[dataset['marital'] == 'single']), dataset[dataset['marital'] == 'single'].is_success.value_counts()[1]/len(dataset[dataset['marital'] == 'single'])])
table.add_row(['Divorced', len(dataset[dataset['marital'] == 'divorced']), dataset[dataset['marital'] == 'divorced'].is_success.value_counts()[1]/len(dataset[dataset['marital'] == 'divorced'])])
print(table)

```

[104] ✓ 0.2s Python

Marital	Total Clients	Success rate
Married	27214	0.10123465863158668
Single	12790	0.1494917904612979
Divorced	5207	0.11945458037257538

Hình 30: Bảng tỷ lệ thành công theo tình trạng hôn nhân

Những khách hàng được liên hệ nhiều nhất nằm ở nhóm đã kết hôn với hơn 60% nhưng tỷ lệ thành công là cao nhất lại thuộc về nhóm những khách hàng còn độc thân (xấp xỉ 15%).

III.2.4. Education:

```

temp3 = pd.crosstab(dataset['education'], dataset['is_success'])
temp3.plot(kind='bar')
print(dataset.groupby(['education']).size()/len(dataset)*100)

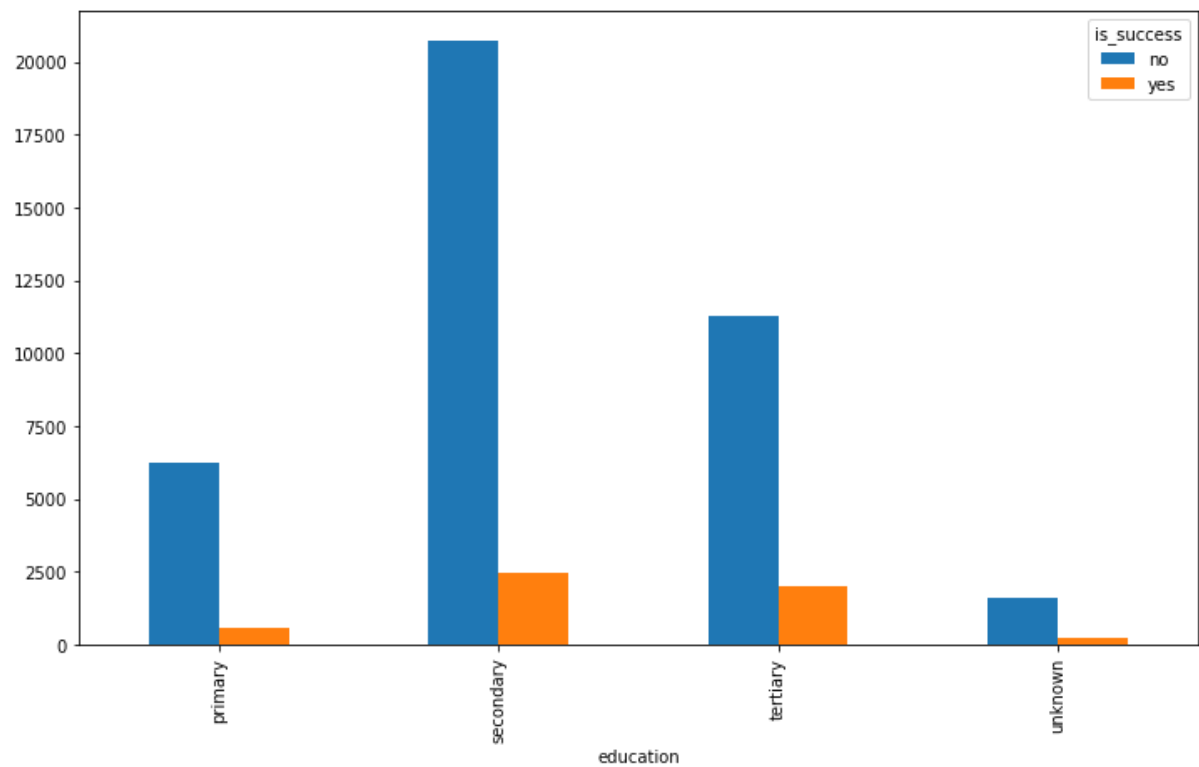
```

[83] ✓ 0.5s Python

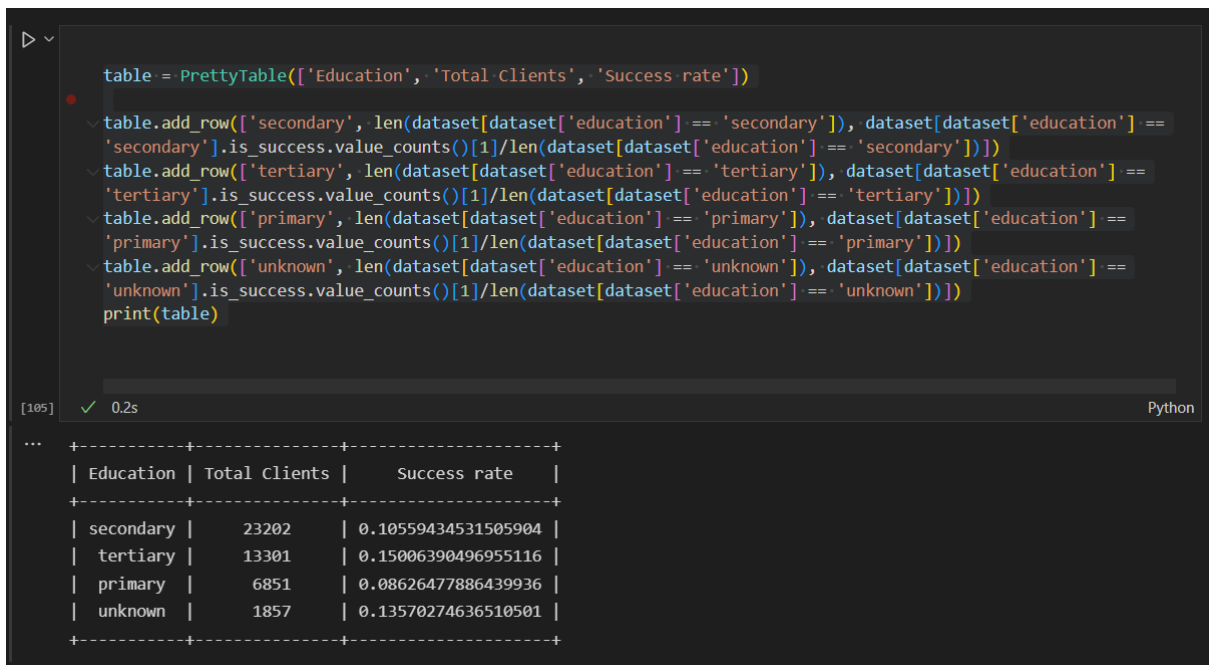
education	primary	secondary	tertiary	unknown
primary	15.153392			
secondary		51.319369		
tertiary			29.419831	
unknown				4.107407

dtype: float64

Hình 31: Tỷ lệ thành công theo trình độ học vấn



Hình 32: Biểu đồ thể hiện mối liên hệ giữa tỷ lệ thành công và trình độ học vấn



Hình 33: Bảng thể hiện mối liên hệ giữa tỷ lệ thành công và trình độ học vấn

Hầu hết những người được liên hệ đều có trình độ đại học hoặc trung học với hơn 80 phần trăm và tỷ lệ thành công cao nhất đối với những khách hàng có trình độ đại học (15% hơn 10% của nhóm trung học).

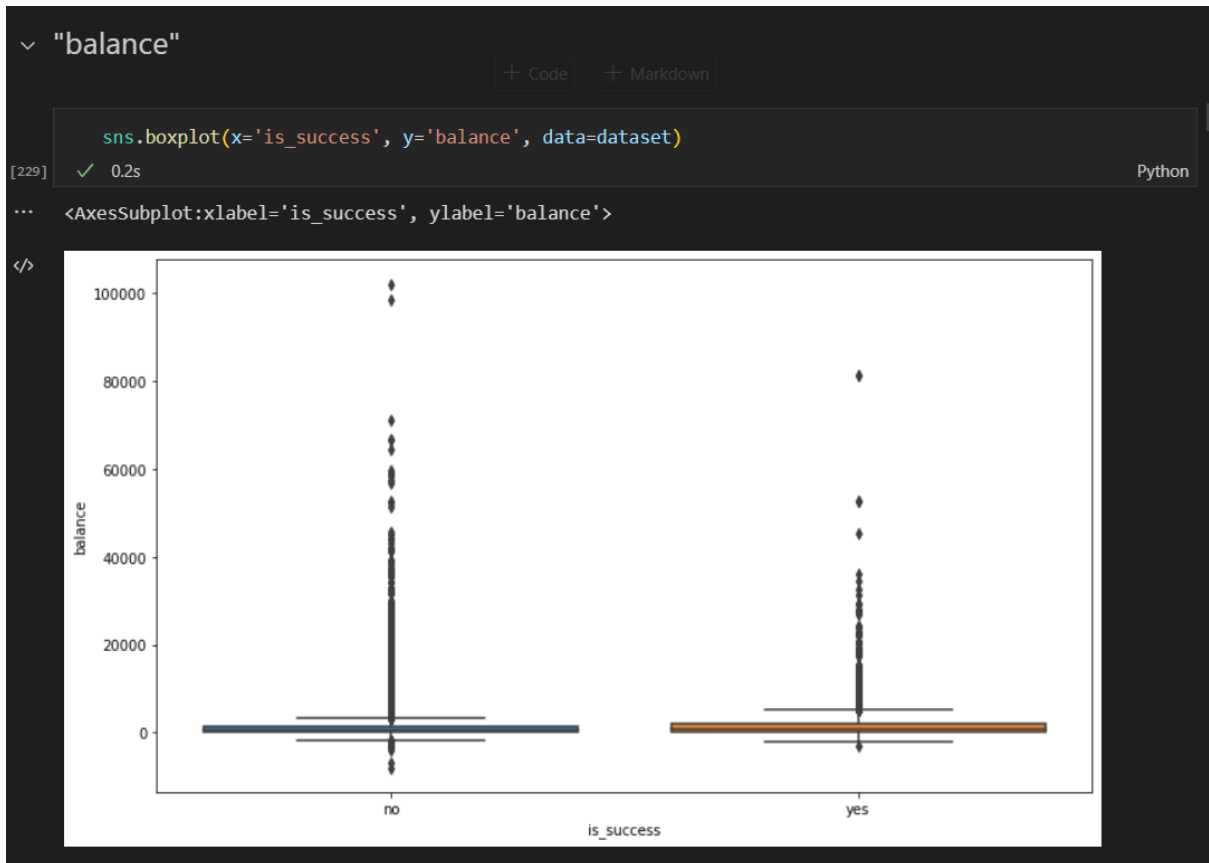
III.2.5. Default



Hình 34: Tỷ lệ thành công theo tình trạng vỡ nợ

Rất ít khách hàng được liên hệ thuộc nhóm vỡ nợ, biến này không được cân bằng nên chúng ta có thể loại bỏ.

III.2.6. Balance:



Hình 35: Biểu đồ boxplot thể hiện tỷ lệ thành công theo số dư

```
# Fixing balance column
dataset_new = dataset
dataset_new = fix_outliers(dataset=dataset_new, col='balance')
```

✓ 0.6s Python

Hình 36: Sửa giá trị ngoại lai của biến balance

```
dataset_new.balance.describe()
```

✓ 0.1s Python

```
count    45211.000000
mean      1074.055318
std       1708.752554
min       -6847.000000
25%        72.000000
50%       448.000000
75%      1322.000000
max      10483.000000
Name: balance, dtype: float64
```

Hình 37: Thống kê mô tả của biến balance

Dữ liệu balance sau khi loại bỏ outliers cho thấy trung bình số balance của khách hàng xấp xỉ 1074 với giá trị lớn nhất là 10483 và nhỏ nhất là -6847.

III.2.7. Housing



Hình 38: Biểu đồ thể hiện tỷ lệ thành công theo tình trạng nợ mua nhà

Tỷ lệ giữa những khách hàng được liên lạc có khoản vay nhà ở và những người không có khoản vay không chênh lệch nhiều (55,6% đối với những người có khoản vay mua nhà ở và 44,4 phần trăm đối với người còn lại).

```

table = PrettyTable(['Housing', 'Total Clients', 'Success rate'])

table.add_row(['Yes', len(dataset[dataset['housing'] == 'yes']), dataset[dataset['housing'] == 'yes'].is_success.value_counts()[1]/len(dataset[dataset['housing'] == 'yes'])]
table.add_row(['No', len(dataset[dataset['housing'] == 'no']), dataset[dataset['housing'] == 'no'].is_success.value_counts()[1]/len(dataset[dataset['housing'] == 'no'])]

print(table)

```

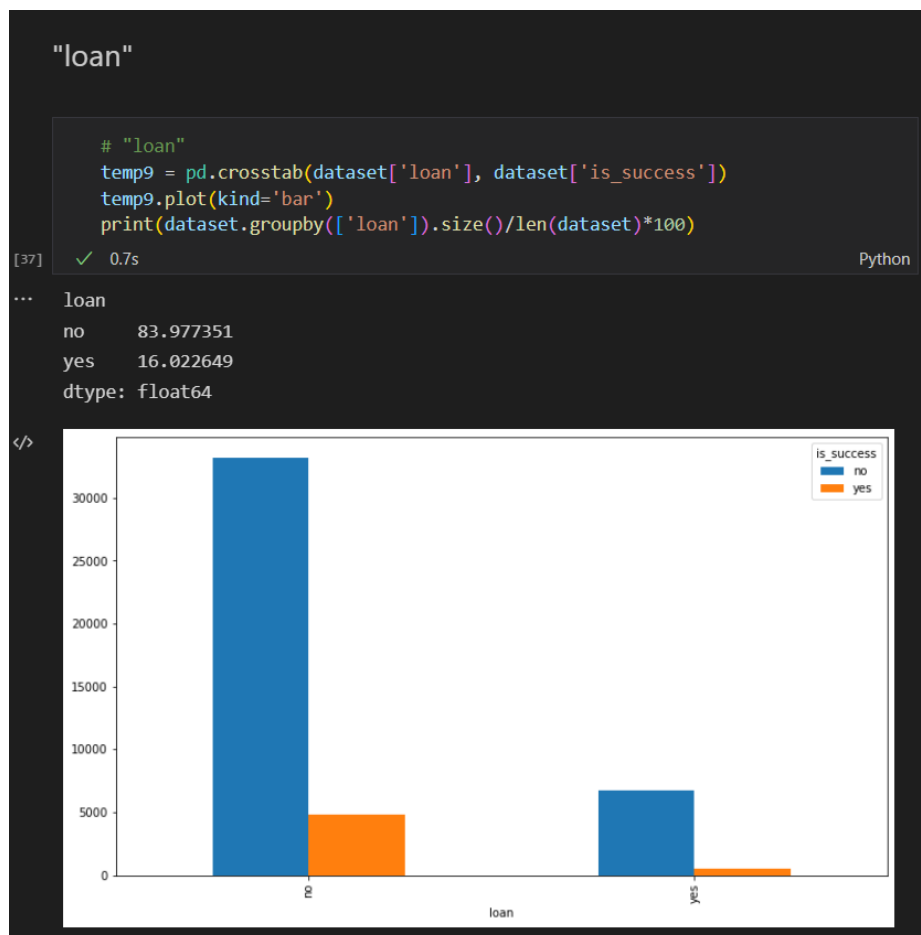
[119] ✓ 0.1s Python

Housing	Total Clients	Success rate
Yes	25130	0.07699960206923995
No	20081	0.1670235546038544

Hình 39: Bảng thể hiện tỷ lệ thành công theo tình trạng nợ mua nhà

Nhưng có một sự khác biệt lớn giữa tỷ lệ thành công của hai nhóm. Những khách hàng không có khoản vay mua nhà nào có xu hướng chấp nhận sản phẩm cao hơn nhóm còn lại (16,7% so với chỉ 7,7% của nhóm có khoản vay mua nhà).

III.2.8. Loan



Hình 40: Biểu đồ thể hiện tình trạng nợ cá nhân và tỷ lệ thành công

Giống như khoản vay mua nhà, những người không có khoản vay cá nhân là nhóm khách hàng được ngân hàng quan tâm với số lượng khách hàng được liên hệ nhiều hơn nhiều so với nhóm có khoản vay cá nhân (gần 84% khách hàng được liên hệ là những người không có khoản vay cá nhân).

```

table = PrettyTable(['Loan', 'Total Clients', 'Success rate'])

table.add_row(['No', len(dataset[dataset['loan'] == 'no']), dataset[dataset['loan'] == 'no'].is_success.value_counts()[1]/len(dataset[dataset['loan'] == 'no'])]
table.add_row(['Yes', len(dataset[dataset['loan'] == 'yes']), dataset[dataset['loan'] == 'yes'].is_success.value_counts()[1]/len(dataset[dataset['loan'] == 'yes'])]

print(table)

```

[129] ✓ 0.1s Python

```

...
+-----+-----+-----+
| Loan | Total Clients | Success rate |
+-----+-----+-----+
| No | 37967 | 0.12655727342165565 |
| Yes | 7244 | 0.06681391496410823 |
+-----+-----+-----+

```

Hình 41: Bảng thể hiện tình trạng nợ cá nhân và tỷ lệ thành công

Và tỷ lệ thành công cũng có sự chênh lệch lớn. Tỷ lệ này đối với những người không có khoản vay cá nhân cao hơn gấp đôi đối với những người có khoản vay cá nhân.

III.2.9. Contact

```

# "contact"
temp6 = pd.crosstab(dataset['contact'], dataset['is_success'])
temp6.plot(kind='bar')
print(dataset.groupby(['contact']).size()/len(dataset)*100)

```

[] Python

```

...
contact
cellular      64.774059
telephone     6.427639
unknown       28.798301
dtype: float64

```

Hình 42: Bảng thể hiện tỷ lệ thành công theo các phương thức liên lạc

```

table = PrettyTable(['Contact', 'Total Clients', 'Success rate'])

table.add_row(['cellular', len(dataset[dataset['contact'] == 'cellular']), dataset[dataset['contact'] == 'cellular'].is_success.value_counts()[1]/len(dataset[dataset['contact'] == 'cellular'])])
table.add_row(['unknown', len(dataset[dataset['contact'] == 'unknown']), dataset[dataset['contact'] == 'unknown'].is_success.value_counts()[1]/len(dataset[dataset['contact'] == 'unknown'])])
table.add_row(['telephone', len(dataset[dataset['contact'] == 'telephone']), dataset[dataset['contact'] == 'telephone'].is_success.value_counts()[1]/len(dataset[dataset['contact'] == 'telephone'])])
print(table)

```

[] Python

```

...
+-----+-----+-----+
| Contact | Total Clients | Success rate |
+-----+-----+-----+
| cellular | 29285 | 0.14918900460986853 |
| unknown | 13020 | 0.040706605222734255 |
| telephone | 2906 | 0.13420509291121818 |
+-----+-----+-----+

```

Hình 43: Bảng thể hiện tỷ lệ người dùng theo phương thức liên lạc

Hầu hết mọi người được liên lạc thông qua cellular và tỷ lệ thành công của cellular là cao nhất trong số những phương thức liên hệ này. Và vì số lượng unknown quá nhiều, chúng ta có thể bỏ biến này.

III.2.10. Month

```

dataset.month.value_counts()

```

[] Python

```

...
may    13766
jul     6895
aug     6247
jun     5341
nov     3970
apr     2932
feb     2649
jan     1403
oct      738
sep      579
mar      477
dec      214
Name: month, dtype: int64

```

Hình 44: Phân bố số lượng khách hàng được liên hệ theo từng tháng

```

print('Success rate and total clients contacted for different months:')
print('Clients contacted in January: {}, Success rate: {}'.format(len(dataset[dataset['month'] ==
'jan']), dataset[dataset['month'] == 'jan'].is_success.value_counts()[1]/len(dataset[dataset['month'] == 'jan'])))
print('Clients contacted in February: {}, Success rate: {}'.format(len(dataset[dataset['month'] == 'feb'
]), dataset[dataset['month'] == 'feb'].is_success.value_counts()[1]/len(dataset[dataset['month'] == 'feb'])))
print('Clients contacted in March: {}, Success rate: {}'.format(len(dataset[dataset['month'] == 'mar'
]), dataset[dataset['month'] == 'mar'].is_success.value_counts()[1]/len(dataset[dataset['month'] == 'mar'])))
print('Clients contacted in April: {}, Success rate: {}'.format(len(dataset[dataset['month'] == 'apr'
]), dataset[dataset['month'] == 'apr'].is_success.value_counts()[1]/len(dataset[dataset['month'] == 'apr'])))
print('Clients contacted in May: {}, Success rate: {}'.format(len(dataset[dataset['month'] == 'may'
]), dataset[dataset['month'] == 'may'].is_success.value_counts()[1]/len(dataset[dataset['month'] == 'may'])))
print('Clients contacted in June: {}, Success rate: {}'.format(len(dataset[dataset['month'] == 'jun'
]), dataset[dataset['month'] == 'jun'].is_success.value_counts()[1]/len(dataset[dataset['month'] == 'jun'])))
print('Clients contacted in July: {}, Success rate: {}'.format(len(dataset[dataset['month'] == 'jul'
]), dataset[dataset['month'] == 'jul'].is_success.value_counts()[1]/len(dataset[dataset['month'] == 'jul'])))
print('Clients contacted in August: {}, Success rate: {}'.format(len(dataset[dataset['month'] == 'aug'
]), dataset[dataset['month'] == 'aug'].is_success.value_counts()[1]/len(dataset[dataset['month'] == 'aug'])))
print('Clients contacted in September: {}, Success rate: {}'.format(len(dataset[dataset['month'] == 'sep'
]), dataset[dataset['month'] == 'sep'].is_success.value_counts()[1]/len(dataset[dataset['month'] == 'sep'])))
print('Clients contacted in October: {}, Success rate: {}'.format(len(dataset[dataset['month'] ==
'oct']), dataset[dataset['month'] == 'oct'].is_success.value_counts()[1]/len(dataset[dataset['month'] == 'oct'])))
print('Clients contacted in November: {}, Success rate: {}'.format(len(dataset[dataset['month'] == 'nov'
]), dataset[dataset['month'] == 'nov'].is_success.value_counts()[1]/len(dataset[dataset['month'] == 'nov'])))
print('Clients contacted in December: {}, Success rate: {}'.format(len(dataset[dataset['month'] == 'dec'
]), dataset[dataset['month'] == 'dec'].is_success.value_counts()[1]/len(dataset[dataset['month'] == 'dec'])))

```

Hình 45: Xử lý phân bố số lượng khách hàng theo tháng

```

... Success rate and total clients contacted for different months:
Clients contacted in January: 1403, Success rate: 0.10121168923734854
Clients contacted in February: 2649, Success rate: 0.1664779161947905
Clients contacted in March: 477, Success rate: 0.480083857442348
Clients contacted in April: 2932, Success rate: 0.19679399727148705
Clients contacted in May: 13766, Success rate: 0.06719453726572715
Clients contacted in June: 5341, Success rate: 0.10222804718217562
Clients contacted in July: 6895, Success rate: 0.09093546047860769
Clients contacted in August: 6247, Success rate: 0.11013286377461182
Clients contacted in September: 579, Success rate: 0.46459412780656306
Clients contacted in October: 738, Success rate: 0.43766937669376693
Clients contacted in November: 3970, Success rate: 0.10151133501259446
Clients contacted in December: 214, Success rate: 0.4672897196261682

```

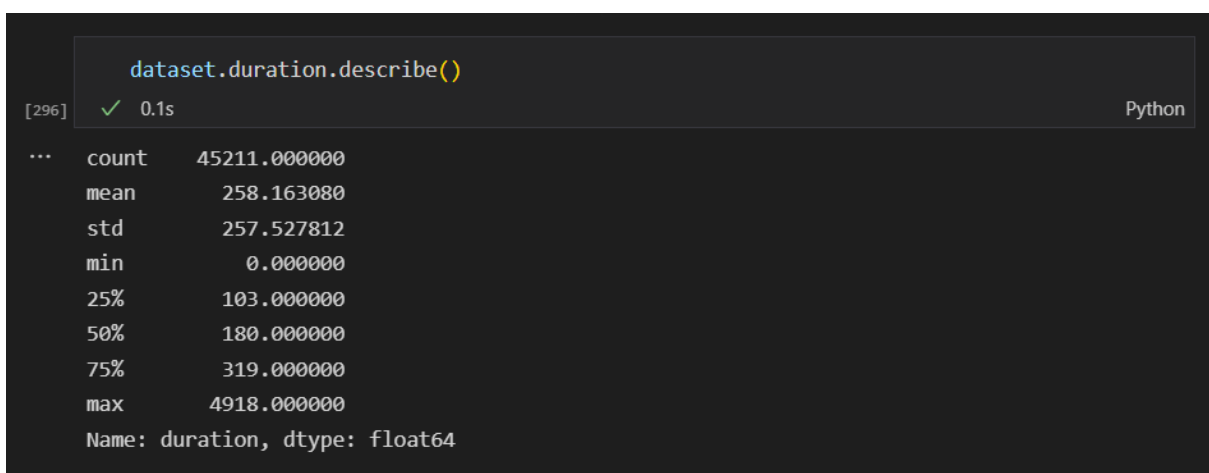
Hình 46: Tỷ lệ thành công của khách hàng theo từng tháng

Khách hàng được liên hệ nhiều nhất vào tháng 5 nhưng tỷ lệ thành công cao nhất vào tháng 3 và tháng 12 (48% và 46,7%).

III.2.11. Duration



Hình 47: Biểu đồ boxplot thể hiện tỷ lệ thành công theo thời lượng liên lạc với khách hàng



Hình 48: Thống kê mô tả của biến duration

Giá trị trung bình thời gian các cuộc gọi xấp xỉ 258 giây với cuộc gọi dài nhất trong 4918 giây và ngắn nhất là 0 giây (chưa liên hệ được). Với những khách hàng chưa liên hệ được thì mặc định kết quả là không thành công.

III.2.12. Campaign



Hình 49: Biểu đồ boxplot thể hiện giữa tỷ lệ thành công và số lần thực hiện cuộc gọi với khách hàng



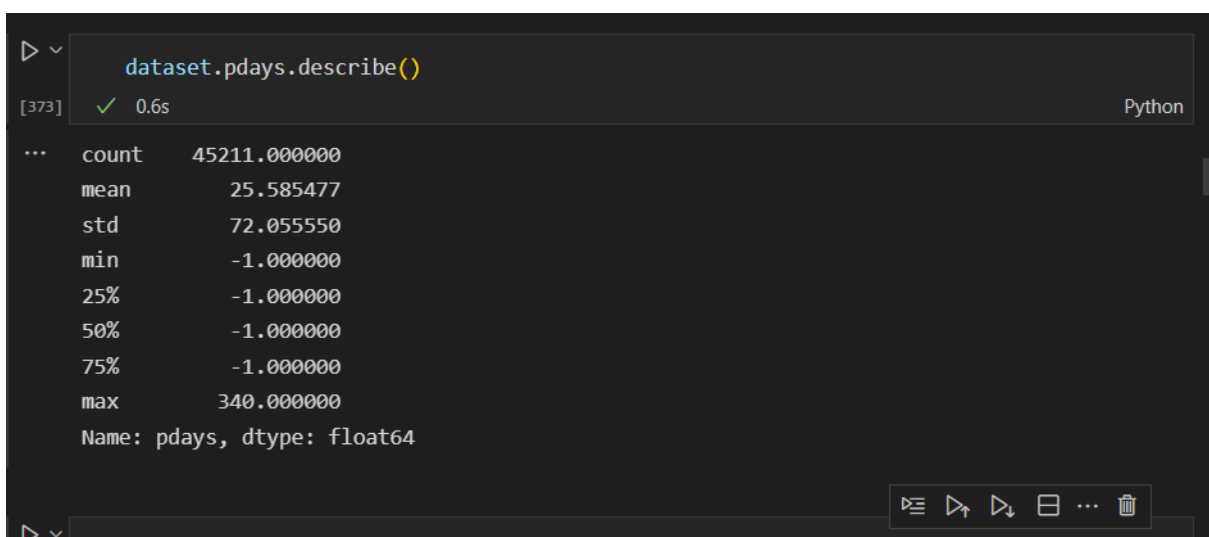
Hình 50: Thống kê mô tả của biến campaign

Số lượng cuộc gọi trung bình xấp xỉ 3 cuộc đối với mỗi khách hàng. Số cuộc lớn nhất là 63 và nhỏ nhất là 1 cuộc.

III.2.13. Pdays:



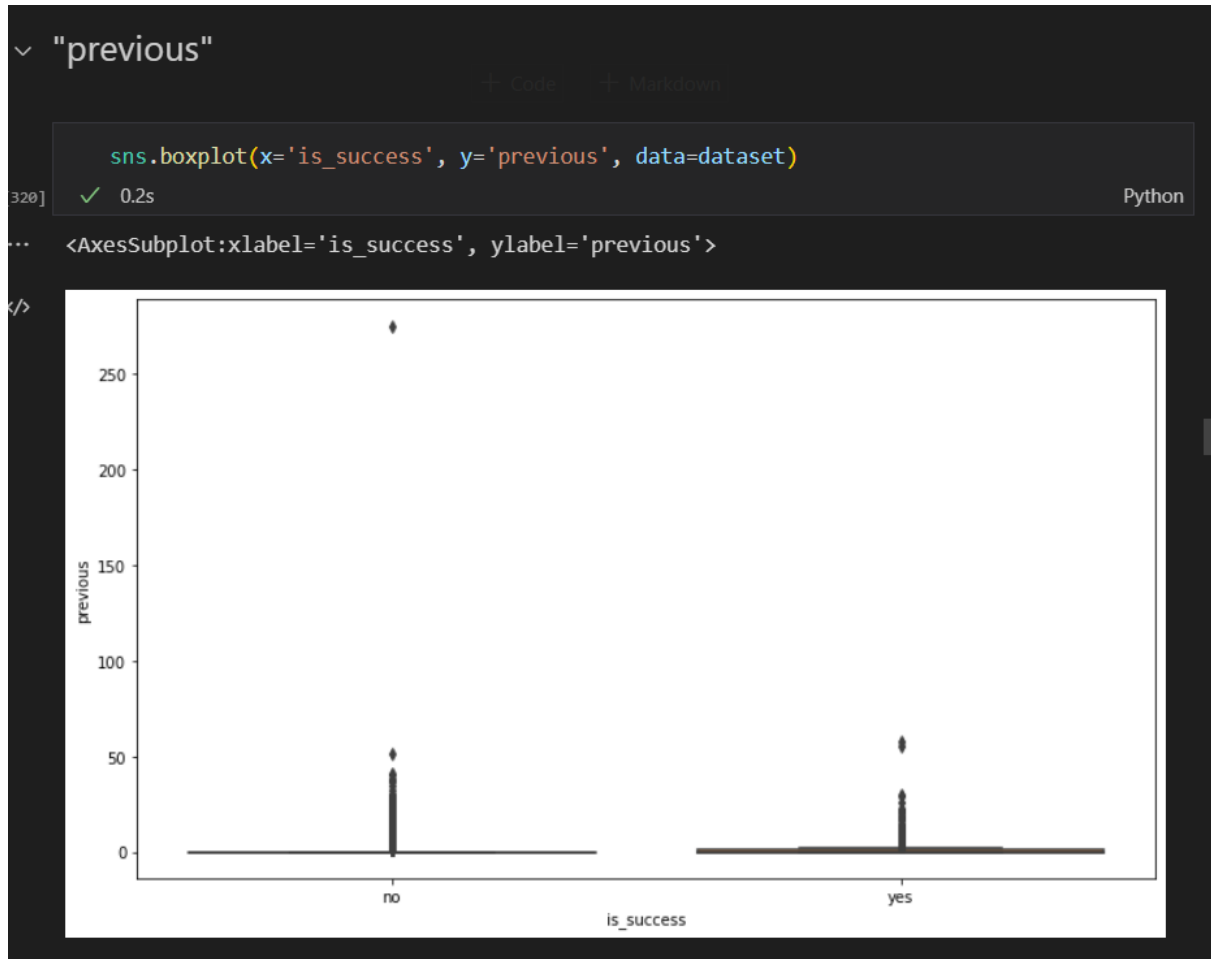
Hình 51: Biểu đồ boxplot thể hiện tỷ lệ thành công và số ngày từ ngày cuối thực hiện chiến dịch.



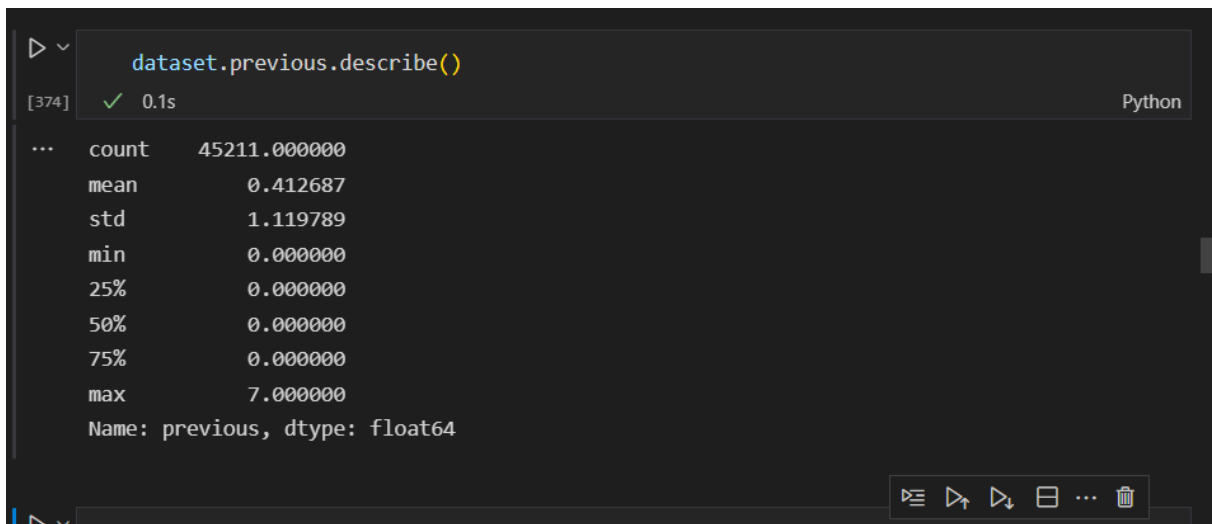
Hình 52: Thống kê mô tả biến pdays

Phần lớn giá trị của biến pdays có giá trị bằng -1, có nghĩa là khách hàng chưa được liên hệ lần nào trước đó.

III.2.14. Previous



Hình 53: Biểu đồ boxplot thể hiện tỷ lệ thành công và số lần liên lạc trước chiến dịch



```
dataset.previous.describe()

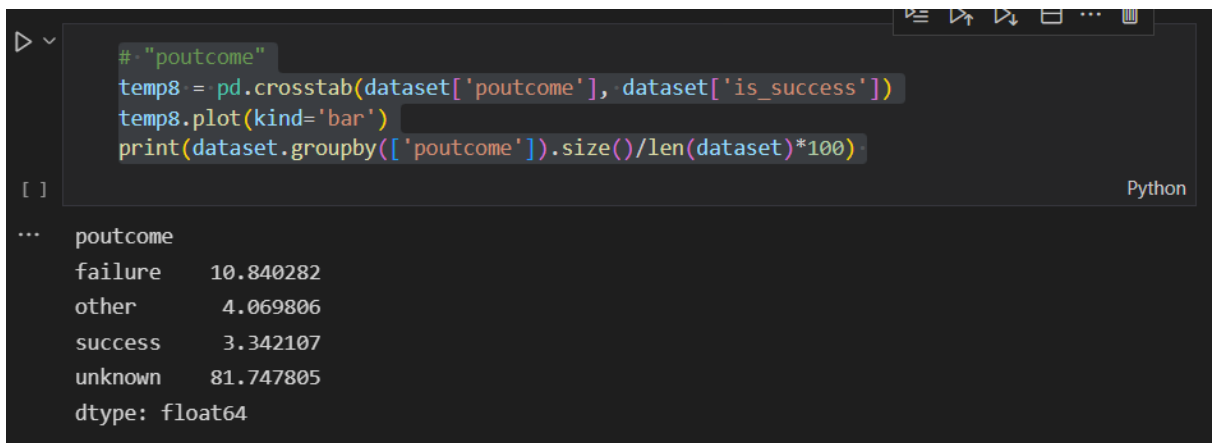
[374] ✓ 0.1s Python

... count      45211.000000
     mean         0.412687
     std          1.119789
     min          0.000000
     25%          0.000000
     50%          0.000000
     75%          0.000000
     max          7.000000
     Name: previous, dtype: float64
```

Hình 54: Thống kê mô tả biến previous

Phần lớn dữ liệu thuộc giá trị 0, cho thấy khách hàng chưa được liên hệ lần nào trước chiến dịch hiện tại. Từ nhận định trên, liên hệ với biến pdays cho thấy đa số khách hàng được liên hệ là khách hàng mới.

III.2.15. Poutcome



```
# "poutcome"
temp8 = pd.crosstab(dataset['poutcome'], dataset['is_success'])
temp8.plot(kind='bar')
print(dataset.groupby(['poutcome']).size()/len(dataset)*100)

[ ] Python

... poutcome
     failure    10.840282
     other       4.069806
     success     3.342107
     unknown    81.747805
     dtype: float64
```

Hình 55: Bảng phân bố tỷ lệ thành công theo kết quả đầu ra của chiến dịch trước

Biến "poutcome" cũng có hơn 81% giá trị "unknown" nên chúng ta cũng có thể loại bỏ biến này..

III.2.16. *Is_success*:


```
# Target variable distribution
count = dataset.groupby('is_success').size()
percent = count/len(dataset)*100
print(percent)
```

[7] ✓ 0.1s Python

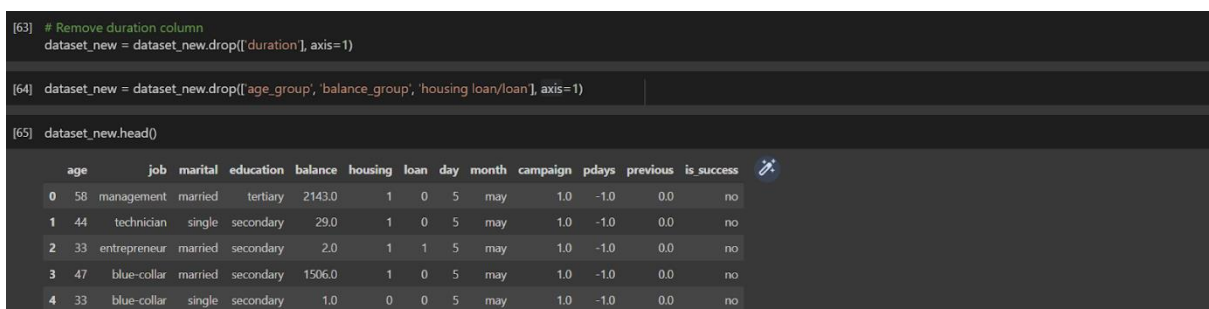
```
... is_success
no    88.30152
yes   11.69848
dtype: float64
```

Hình 56: Phân bố tỷ lệ biến phụ thuộc y.

Từ phân phối của biến mục tiêu: "is_success", chúng ta thấy rằng dữ liệu không cân bằng vì có khoảng 88% là "no" và 12% là "yes".

Chương IV: Xây dựng mô hình hồi quy logistic dự báo biến phụ thuộc.

Sau quá trình Khai phá dữ liệu, nhóm nhận thấy biến 'duration' gây ảnh hưởng lớn đến biến đầu ra 'is_success' của bài toán, vì biến này chỉ có giá trị khi ngân hàng tiến hành gọi cho khách hàng tuy nhiên với những khách hàng chưa được gọi bất kỳ cuộc gọi nào thì điều này là hoàn toàn vô ích.



```
[63] # Remove duration column
dataset_new = dataset_new.drop(['duration'], axis=1)

[64] dataset_new = dataset_new.drop(['age_group', 'balance_group', 'housing loan/loan'], axis=1)

[65] dataset_new.head()
```

	age	job	marital	education	balance	housing	loan	day	month	campaign	pdays	previous	is success
0	58	management	married	tertiary	2143.0	1	0	5	may	1.0	-1.0	0.0	no
1	44	technician	single	secondary	29.0	1	0	5	may	1.0	-1.0	0.0	no
2	33	entrepreneur	married	secondary	2.0	1	1	5	may	1.0	-1.0	0.0	no
3	47	blue-collar	married	secondary	1506.0	1	0	5	may	1.0	-1.0	0.0	no
4	33	blue-collar	single	secondary	1.0	0	0	5	may	1.0	-1.0	0.0	no

Hình 57: Xử lý các cột không cần thiết

Do máy không thể hiểu được các dữ liệu kiểu category ở dạng kí tự nên ta phải dummy ra thành các biến con mới để tiện việc huấn luyện và dự đoán mô hình.

```
[67] #Seperating Target variable from other variables
dataset_Y = dataset_new['is_success']
dataset_X = dataset_new[dataset_new.columns[0:12]]

[69] #converting Independent Categorical into Numerical by creating Dummy variables
dataset_X_dummy = pd.get_dummies(dataset_X)
print(dataset_X_dummy.info())

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 37 columns):
#   Column                Non-Null Count  Dtype
---  -
0   age                   45211 non-null  int64
1   balance               45211 non-null  float64
2   housing               45211 non-null  int64
3   loan                  45211 non-null  int64
4   day                   45211 non-null  int64
5   campaign              45211 non-null  float64
6   pdays                 45211 non-null  float64
7   previous              45211 non-null  float64
8   job_admin              45211 non-null  uint8
9   job_blue-collar       45211 non-null  uint8
10  job_entrepreneur       45211 non-null  uint8
11  job_housemaid          45211 non-null  uint8
12  job_management         45211 non-null  uint8
13  job_retired            45211 non-null  uint8
14  job_self-employed      45211 non-null  uint8
15  job_services           45211 non-null  uint8
16  job_student            45211 non-null  uint8
17  job_technician         45211 non-null  uint8
18  job_unemployed         45211 non-null  uint8
19  marital_divorced       45211 non-null  uint8
20  marital_married        45211 non-null  uint8
21  marital_single         45211 non-null  uint8
22  education_primary      45211 non-null  uint8
23  education_secondary    45211 non-null  uint8
24  education_tertiary     45211 non-null  uint8
25  month_apr              45211 non-null  uint8
26  month_aug              45211 non-null  uint8
27  month_dec              45211 non-null  uint8
28  month_feb              45211 non-null  uint8
```

Hình 58: Dummy các biến category

=> Kết quả cho ra tổng cộng 37 biến (thêm 24 biến mới)

Từ tập dữ liệu ban đầu ta chia ra thành 2 tập dữ liệu nhỏ là tập dữ liệu huấn luyện và tập dữ liệu để kiểm thử với tỷ lệ là 80% dữ liệu cho tập huấn luyện và 20% cho tập kiểm thử. Trong quá trình kiểm tra dữ liệu của biến phụ thuộc, nhóm thấy rằng dữ liệu không cân bằng, cụ thể là số lượng nhãn ‘no’ lên đến 31942 (chiếm 88,3% toàn bộ tập train) và nhãn ‘yes’ chỉ chiếm 11,7% số nhãn của bộ dữ liệu này. Điều này sẽ gây ảnh hưởng ít nhiều đến kết quả đầu ra của mô hình. Vì vậy, nhóm sử dụng phương pháp SMOTE để chuyển bộ data này về lại cân bằng.

```
[ ] X = dataset_X_dummy
Y = dataset_Y

# Split-out validation dataset
X_train, X_validation, Y_train, Y_validation = model_selection.train_test_split(X, Y, test_size=0.2, random_state=0)

[ ] from imblearn.over_sampling import SMOTE
from collections import Counter

# summarize class distribution
counter = Counter(Y_train)
print(counter)

# transform the dataset
oversample = SMOTE()
x_train_smote, y_train = oversample.fit_resample(X_train, Y_train)

# summarize the new class distribution
counter = Counter(y_train)
print(counter)
```

Hình 59: Phân chia tập dữ liệu

Trong bước chọn biến để bắt đầu huấn luyện mô hình nhóm có sử dụng phương pháp Recursive Feature Elimination (RFE), nghĩa là dùng mô hình mẫu để huấn luyện dựa trên các biến và tìm ra một số lượng biến có xếp hạng cao nhất. Mô hình mẫu này sẽ huấn luyện và đưa ra các biến có độ quan trọng cao, thấp và lặp đi lặp lại với nhiều tổ hợp biến, việc này để tìm ra một số lượng biến đã thiết lập trong hàm trước (`n_features_to_select`) với độ xếp hạng cao nhất.

```
[72] from sklearn.feature_selection import RFE

[75] logistic_regression_model = LogisticRegression()
logistic_regression_model.fit(x_train_smote, y_train)
logistic_regression_model

/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
LogisticRegression())

[76] def selectFeatureRFE(n_features):
    rfe = RFE(logistic_regression_model, step=25, n_features_to_select=n_features)
    rfe = rfe.fit(x_train_smote, y_train)
    rfe_ = X_train.columns[rfe.support_]
    # rfe_
    selected_columns = x_train_smote.columns[rfe.support_]
    return selected_columns.tolist()
```

Hình 60: Trích chọn đặc trưng cho mô hình

Các biến được chọn thông qua hàm RFE sẽ được lưu lại và tiến hành đưa vào mô hình để bắt đầu huấn luyện. Nhóm sử dụng số lượng biến lần lượt là 7,15,20 để xây dựng mô hình, từ đó tiến hành chấm điểm để chọn ra mô hình tốt nhất.

```
[1] def trainModel(featuresList):
    y_train.value_counts()
    # x_train_smote.info()
    X_train_final = x_train_smote[featuresList]
    y_train_final = y_train

    logreg = LogisticRegression()
    logreg.fit(X_train_final, y_train_final)

    X_test_final = X_validation[featuresList]
    y_test_final = Y_validation
    y_pred = logreg.predict(X_test_final)
    # print('Accuracy of logistic regression classifier on test set: {:.2f}'.format(logreg.score(X_test_final, y_test_final)))
    return y_pred
```

Hình 61: Xây dựng mô hình Logistics Regression

Chương V: Đánh giá kết quả và lựa chọn mô hình

Logistic regression model 1 với

- 7 biến: 'loan', 'job_technician', 'education_primary', 'education_secondary', 'education_tertiary', 'month_jul', 'month_may'
- Confusion matrix:


```
Confusion matrix
[[5565 2415]
 [ 550  513]]
```

- Classification report:

	Precision	Recall	F1-score	Support
No	0.91	0.73	0.81	7980
Yes	0.19	0.47	0.27	1063
Accuracy			0.70	9043
Macro avg	0.55	0.60	0.54	9043
Weighted avg	0.83	0.70	0.75	9043

Bảng 3: Bảng kết quả phân loại của mô hình 1

- Accuracy: 0.7
- F1 score: 0.6
- Precision: 0.55
- Recall: 0.6

Logistic regression model 2

- 15 biến: 'housing', 'loan', 'previous', 'job_blue-collar', 'job_management', 'job_technician', 'marital_married', 'marital_divorced', 'education_primary', 'education_secondary', 'education_tertiary', 'month_aug', 'month_jul', 'month_jun', 'month_may'.
- Confusion matrix

```
Confusion matrix
[[7137 843]
 [ 734 329]]
```

- Classification report:

	Precision	Recall	F1-score	Support
No	0.91	0.89	0.90	7980
Yes	0.28	0.31	0.29	1063

Accuracy			0.83	9043
Macro avg	0.59	0.60	0.60	9043
Weighted avg	0.83	0.83	0.83	9043

Bảng 4: Bảng kết quả phân loại của mô hình 2

- Accuracy: 0.83
- F1 score: 0.59
- Precision: 0.59
- Recall: 0.6

Logistic regression model 3

- 20 biến: 'housing', 'loan', 'previous', 'job_admin.', 'job_blue-collar', 'job_management', 'job_services', 'job_technician', 'marital_divorced', 'marital_married', 'marital_single', 'education_primary', 'education_secondary', 'education_tertiary', 'month_aug', 'month_feb', 'month_jul', 'month_jun', 'month_may', 'month_nov'
- Confusion matrix:

Confusion matrix
[[7493 487]
[800 263]]

- Classification report:

	Precision	Recall	F1-score	Support
No	0.90	0.94	0.92	7980
Yes	0.35	0.25	0.29	1063

Accuracy			0.86	9043
Macro avg	0.63	0.59	0.61	9043
Weighted avg	0.84	0.86	0.85	9043

Bảng 5: Bảng kết quả phân loại của mô hình 3

- Accuracy: 0.86
- F1 score: 0.6

- Precision: 0.62
- Recall: 0.59

Với bài toán này dự đoán nhầm sẽ tốt hơn là bỏ sót: dự đoán một người có khả năng đăng ký dịch vụ sẽ có lợi hơn là bỏ qua người này - vì đó một cơ hội tốt để tăng tiền gửi cho ngân hàng. Nên nhóm muốn tập trung vào việc dự đoán càng nhiều giá trị dương càng tốt, do đó giá trị recall sẽ được ưu tiên.

Dựa vào những thông số dự đoán, nhóm đưa ra nhận xét **logistic regression model 3** là model tốt nhất. Điểm recall của các model gần tương đương nhau, nên xét theo các thông số về accuracy, precision, F1 score... thì model 3 vượt trội hơn.