TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO BÀI TẬP LỚN

MÔN HỌC: KHAI PHÁ DỮ LIỆU

ĐỀ TÀI: PHÂN LỚP BỘ DỮ LIỆU THÔNG TIN KHÁCH HÀNG CỦA NGÂN HÀNG BẰNG THUẬT TOÁN NAIVE BAYES VÀ SUPPORT VECTOR MACHINES ĐỂ DỰ ĐOÁN KHẢ NĂNG KHÁCH HÀNG ĐỒNG Ý GỬI TIẾT KIỆM TẠI NGÂN HÀNG

Giảng viên hướng dẫn : **Nguyễn Tiến Đạt**
Nguyễn Văn Bảy
Sinh viên thực hiện : **Tạ Thị Thiên Thanh - 2151013088**
Phạm Công Thuận - 2151013097
Lớp : **DH21CS01**

MỤC LỤC

I. MỞ ĐẦU

Gửi tiết kiệm có kỳ hạn tại ngân hàng là một trong những hình thức đầu tư phổ biến nhất hiện nay. Sau khi kết thúc kỳ hạn được ghi trong hợp đồng, khách hàng sẽ nhận lại toàn bộ số tiền gốc và tiền lãi của mình dựa trên một mức lãi suất đã được thỏa thuận với ngân hàng từ trước.

Trong bối cảnh công nghệ đang không ngừng phát triển, các ngân hàng phải liên tục chuyển đổi số, áp dụng công nghệ vào các hoạt động của mình nhằm nâng cao hiệu suất công việc. Do đó, việc xây dựng và ứng dụng các mô hình dự đoán khả năng đồng ý gửi tiết kiệm có kỳ hạn của khách hàng đang trở nên cần thiết hơn bao giờ hết.

Mục tiêu của đề tài là nghiên cứu và phát triển một mô hình dự đoán bằng các phương pháp khai phá dữ liệu như gom cụm, tìm luật kết hợp và áp dụng kĩ thuật phân lớp dữ liệu. Với các kết quả đạt được, đề tài sẽ tiến hành đánh giá để đưa ra kết luận về độ hiệu quả và tính chính xác của mô hình, từ đó xác định được các thuật toán nào là phù hợp để khai thác bộ dữ liệu của đề tài.

II. TỔNG QUAN VỀ KHAI PHÁ DỮ LIỆU

1. Khái niệm

Thuật ngữ "Khai phá dữ liệu" (*Data Mining*) ra đời vào khoảng cuối những năm 80 của thế kỷ XX [1]. Khai phá dữ liệu là một lĩnh vực liên ngành Khoa học máy tính, là quá trình tính toán sử dụng kết hợp các phương pháp trong các lĩnh vực máy học, thống kê và hệ thống cơ sở dữ liệu để tìm ra các mẫu trong các bộ dữ liệu lớn [2].

2. Quá trình khai phá dữ liệu

Bước 1. Xác định vấn đề và dữ liệu thích hợp với vấn đề cần giải quyết [2].
Bước 2. Tiền xử lý dữ liệu gồm làm sạch, tích hợp, chuẩn hóa, …
Bước 3. Xác định và lựa chọn tác vụ phù hợp.
Bước 4. Chọn thuật giải thích hợp.
Bước 5. Khai thác dữ liệu.
Bước 6. Biểu diễn, trực quan hóa tri thức tìm được bằng ngôn ngữ tự nhiên và mô hình, biểu đồ [3].
Bước 7. Đánh giá và nhận xét kết quả đạt được.

3. Các kỹ thuật Khai phá dữ liệu

- Luật kết hợp (Association Rule): Apriori, FP-Growth, Brute-force, …
- Gom cụm (Clustering): K-means, DBSCAN, Affinity Propagation, …
- Phân lớp (Classification): Naive-bayes, KNN, Decision Tree, SVM, …

4. Các ứng dụng của Khai phá dữ liệu

Hiện nay đã có rất nhiều công cụ thương mại cũng như phi thương mại triển khai nhiệm vụ của khai phá dữ liệu. Có thể thấy khai phá dữ liệu đã và đang phát triển, ứng dụng rộng rãi trong nhiều lĩnh vực [4]. Dưới đây là một số lĩnh vực phổ biến:

- *Lĩnh vực tài chính*

Phân tích hành vi khách hàng, dự đoán khả năng đồng ý vay, thanh toán, gửi tiết kiệm, …
Gom nhóm, phân loại khách hàng nhằm phục vụ các mục đích tiếp thị [4].
Phát hiện các hoạt động gian lận, đáng ngờ, có rủi ro, … [5]

- *Lĩnh vực chăm sóc sức khỏe*

Tìm ra các mối liên hệ giữa các triệu chứng, phương pháp điều trị, đặc điểm sinh học, … để đưa ra các phương án phòng, chữa bệnh phù hợp [5].

- *Lĩnh vực sinh học*

Phân tích, khai phá các cấu trúc của gen, protein, … nhằm đưa ra các kết luận về di truyền cũng như phát hiện các bất thường, rủi ro trong cơ sở dữ liệu gen để kịp thời đề ra phương án xử lý [4].

- *Lĩnh vực giáo dục*

Dự đoán hành vi học tập, xu hướng của học sinh trong tương lai, phân nhóm học sinh thông qua hành vi và năng lực để nhà trường và các tổ chức giáo dục tập trung vào chương trình giáo dục, nội dung và phương pháp giảng dạy phù hợp [5].

- *Một số ứng dụng khoa học khác*

- Phát hiện các xâm nhập bất hợp pháp, dự đoán xu hướng, thị hiếu của khách hàng, phân tích nhu cầu thị trường, ...

III. MÔ TẢ DỮ LIỆU

Bộ dữ liệu thu được sau một chiến dịch tiếp thị trực tiếp thông qua các cuộc gọi điện thoại với khách hàng của một ngân hàng tại Bồ Đào Nha từ năm 2008 đến năm 2010. Dữ liệu phục vụ cho việc dự đoán khả năng khách hàng đồng ý ký hợp đồng gửi tiền tiết kiệm có kỳ hạn dựa vào các thuộc tính cụ thể.

Các file dữ liệu bao gồm:

1. train.csv: 45211 thể hiện và 17 thuộc tính
2. test.csv: 4521 thể hiện và 17 thuộc tính, số thể hiện được chọn ngẫu nhiên từ file train.csv và bằng 10% số thể hiện của file train.csv

Các thuộc tính có trong bộ dữ liệu bao gồm:

1. age (numeric): tuổi của khách hàng

2. job: loại hình công việc (giá trị: “admin”, “unknown”, “unemployed‘, “management”, “housemaid”, “entrepreneur”, “student”. “blue-collar”, “self-employed”, “retired”, “technician”, “services”)
3. marital : tình trạng hôn nhân (giá trị: "married", "divorced", "single")
4. education: trình độ học vấn (giá trị: "unknown", "secondary", "primary", "tertiary")
5. default: đã từng bị nợ xấu (giá trị: "yes", "no")
6. balance: số dư trung bình hằng năm (euro)
7. housing: đang vay tiền để mua nhà (giá trị: "yes", "no")
8. loan: có nợ cá nhân (giá trị: "yes", "no")
9. contact: phương thức liên lạc với khách hàng (giá trị: "unknown", "telephone", "cellular")
10. day: ngày liên lạc cuối cùng trong tháng
11. month: tháng liên lạc cuối cùng trong năm (giá trị: "jan", "feb", "mar", …, "nov", "dec")
12. duration: thời lượng của lần liên lạc cuối cùng (seconds)
13. campaign: số lần liên lạc trong suốt chiến dịch
14. pdays: số ngày tính từ lần liên lạc cuối cùng của chiến dịch trước đây (khách hàng chưa từng được được liên lạc sẽ có giá trị là -1)
15. previous: số lần liên lạc trong suốt chiến dịch trước đây
16. poutcome: kết quả của chiến dịch trước đây (giá trị: "unknown", "other", "failure", "success")
17. y: khách hàng đồng ý ký hợp đồng gửi tiết kiệm có kỳ hạn (giá trị: "yes", "no")

IV. TIỀN XỬ LÝ DỮ LIỆU

1. Một số thông tin tổng quan về dữ liệu trước khi tiền xử lý

Tổng quan về dữ liệu của từng cột trong dataframe `df_train` ứng với file train.csv và dataframe `df_test` ứng với file test.csv:

#	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	y	45211 non-null	object

dtypes: int64(7), object(10)

Hình 4.1.1. Thông tin của bộ dữ liệu df_train

#	Column	Non-Null Count	Dtype
0	age	4521 non-null	int64
1	job	4521 non-null	object
2	marital	4521 non-null	object
3	education	4521 non-null	object
4	default	4521 non-null	object
5	balance	4521 non-null	int64
6	housing	4521 non-null	object
7	loan	4521 non-null	object
8	contact	4521 non-null	object
9	day	4521 non-null	int64
10	month	4521 non-null	object
11	duration	4521 non-null	int64
12	campaign	4521 non-null	int64
13	pdays	4521 non-null	int64
14	previous	4521 non-null	int64
15	poutcome	4521 non-null	object
16	y	4521 non-null	object

dtypes: int64(7), object(10)

Hình 4.1.2. Thông tin của bộ dữ liệu df_test

Cả hai bộ dữ liệu đều có 17 thuộc tính, các thuộc tính ở hai bộ dữ liệu là như nhau. Mỗi thuộc tính trong bộ dữ liệu `df_train` có 45211 dòng, không có dữ liệu null. Mỗi thuộc tính trong bộ dữ liệu `df_test` có 4521 dòng, không có dữ liệu null.

Có 10 cột dữ liệu kiểu category `['job', 'marital', 'education', 'contact', 'month', 'poutcome', 'y']`, 7 cột dữ liệu kiểu số và 3 cột `['default', 'housing', 'loan']` là kiểu dữ liệu định danh nhị phân (binary nominal) với hai giá trị `['yes', 'no']`.

	age	job	marital	education	default	balance	housing	loan	\
0	58	management	married	tertiary	no	2143	yes	no	
1	44	technician	single	secondary	no	29	yes	no	
2	33	entrepreneur	married	secondary	no	2	yes	yes	
3	47	blue-collar	married	unknown	no	1506	yes	no	
4	33	unknown	single	unknown	no	1	no	no	
...	...	...	...	...	...	...	...	...	
45206	51	technician	married	tertiary	no	825	no	no	
45207	71	retired	divorced	primary	no	1729	no	no	
45208	72	retired	married	secondary	no	5715	no	no	
45209	57	blue-collar	married	secondary	no	668	no	no	
45210	37	entrepreneur	married	secondary	no	2971	no	no	

	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	unknown	5	may	261	1	-1	0	unknown	no
1	unknown	5	may	151	1	-1	0	unknown	no
2	unknown	5	may	76	1	-1	0	unknown	no
3	unknown	5	may	92	1	-1	0	unknown	no
4	unknown	5	may	198	1	-1	0	unknown	no
...	...	...	...	...	...	...	...	...	...
45206	cellular	17	nov	977	3	-1	0	unknown	yes
45207	cellular	17	nov	456	2	-1	0	unknown	yes
45208	cellular	17	nov	1127	5	184	3	success	yes
45209	telephone	17	nov	508	4	-1	0	unknown	no
45210	cellular	17	nov	361	2	188	11	other	no

Hình 4.1.3. Bộ dữ liệu df_train

	age	job	marital	education	default	balance	housing	loan	\
0	30	unemployed	married	primary	no	1787	no	no	
1	33	services	married	secondary	no	4789	yes	yes	
2	35	management	single	tertiary	no	1350	yes	no	
3	30	management	married	tertiary	no	1476	yes	yes	
4	59	blue-collar	married	secondary	no	0	yes	no	
...	...	...	...	...	...	...	...	...	
4516	33	services	married	secondary	no	-333	yes	no	
4517	57	self-employed	married	tertiary	yes	-3313	yes	yes	
4518	57	technician	married	secondary	no	295	no	no	
4519	28	blue-collar	married	secondary	no	1137	no	no	
4520	44	entrepreneur	single	tertiary	no	1136	yes	yes	

	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	cellular	19	oct	79	1	-1	0	unknown	no
1	cellular	11	may	220	1	339	4	failure	no
2	cellular	16	apr	185	1	330	1	failure	no
3	unknown	3	jun	199	4	-1	0	unknown	no
4	unknown	5	may	226	1	-1	0	unknown	no
...	...	...	...	...	...	...	...	...	..
4516	cellular	30	jul	329	5	-1	0	unknown	no
4517	unknown	9	may	153	1	-1	0	unknown	no
4518	cellular	19	aug	151	11	-1	0	unknown	no
4519	cellular	6	feb	129	4	211	3	other	no
4520	cellular	3	apr	345	2	249	7	other	no

Hình 4.1.4. Bộ dữ liệu df_test

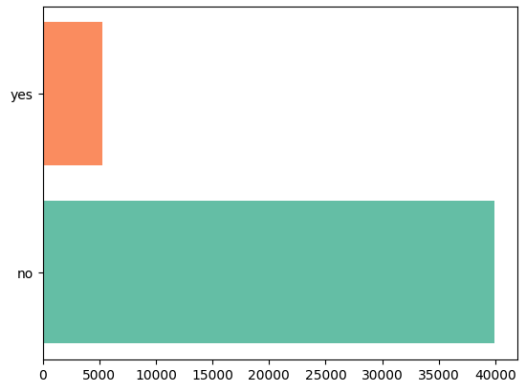
Với các cột kiểu dữ liệu kiểu số, ta có một vài thông số thống kê như sau:

	age	balance	day	duration	campaign	pdays	previous
count	45211.00	45211.00	45211.00	45211.00	45211.00	45211.00	45211.00
mean	40.94	1362.27	15.81	258.16	2.76	40.20	0.58
std	10.62	3044.77	8.32	257.53	3.10	100.13	2.30
min	18.00	-8019.00	1.00	0.00	1.00	-1.00	0.00
25%	33.00	72.00	8.00	103.00	1.00	-1.00	0.00
50%	39.00	448.00	16.00	180.00	2.00	-1.00	0.00
75%	48.00	1428.00	21.00	319.00	3.00	-1.00	0.00
max	95.00	102127.00	31.00	4918.00	63.00	871.00	275.00

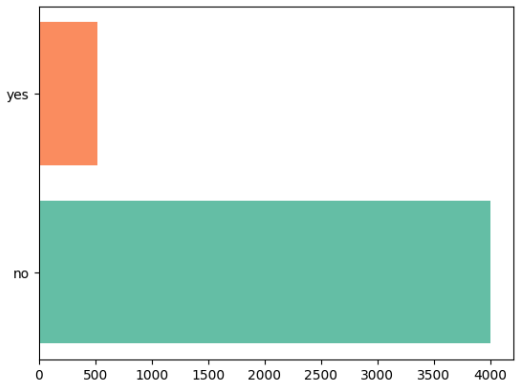
Hình 4.1.5. Một số thông số thống kê các thuộc tính kiểu số của bộ dữ liệu df_train

	age	balance	day	duration	campaign	pdays	previous
count	4521.00	4521.00	4521.00	4521.00	4521.00	4521.00	4521.00
mean	41.17	1422.66	15.92	263.96	2.79	39.77	0.54
std	10.58	3009.64	8.25	259.86	3.11	100.12	1.69
min	19.00	-3313.00	1.00	4.00	1.00	-1.00	0.00
25%	33.00	69.00	9.00	104.00	1.00	-1.00	0.00
50%	39.00	444.00	16.00	185.00	2.00	-1.00	0.00
75%	49.00	1480.00	21.00	329.00	3.00	-1.00	0.00
max	87.00	71188.00	31.00	3025.00	50.00	871.00	25.00

Hình 4.1.6. Một số thông số thống kê các thuộc tính kiểu số của bộ dữ liệu df_test



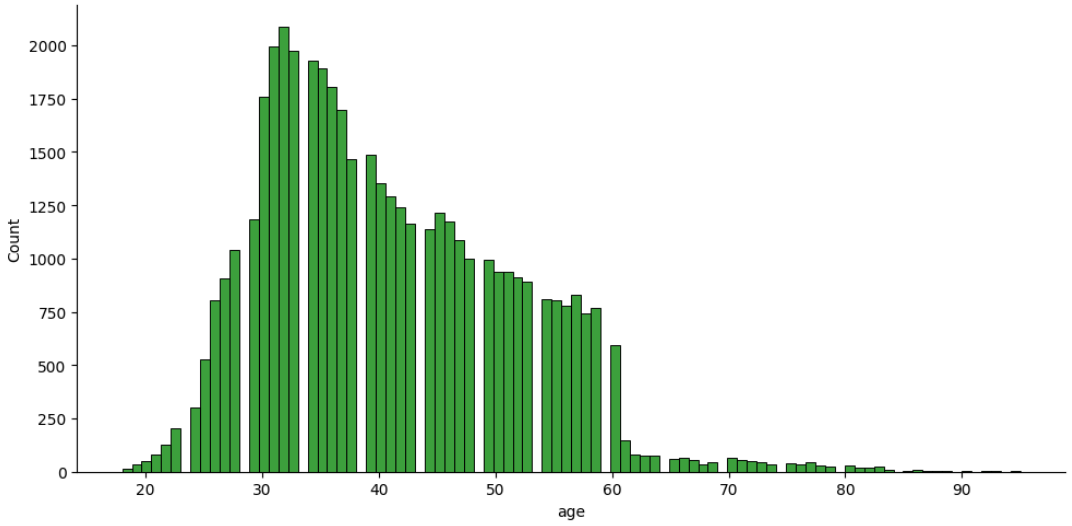
Hình 4.1.7a.



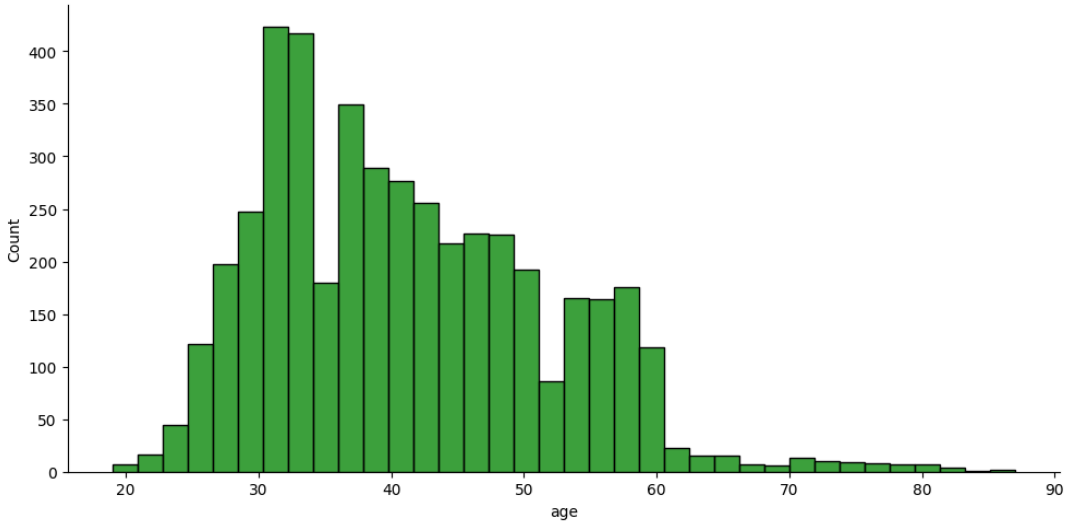
Hình 4.1.7b.

Hình 4.1.7. Biểu đồ thể hiện số lượng đồng ý/ không đồng ý của khách hàng trong hai bộ dữ liệu df_train (Hình 4.1.7a.) và df_test (Hình 4.1.7b.) dựa vào thuộc tính "y"

Hai biểu đồ trên cho thấy sự chênh lệch rất lớn giữa số lượng khách hàng đồng ý gửi tiết kiệm và không đồng ý gửi tiết kiệm tại ngân hàng. Vì thuộc tính `['y']` là thuộc tính trả về kết quả (target) nên sự chênh lệch này có thể tạo nên sự mất cân bằng dữ liệu, gây ảnh hưởng không nhỏ đến kết quả dự đoán sau này.

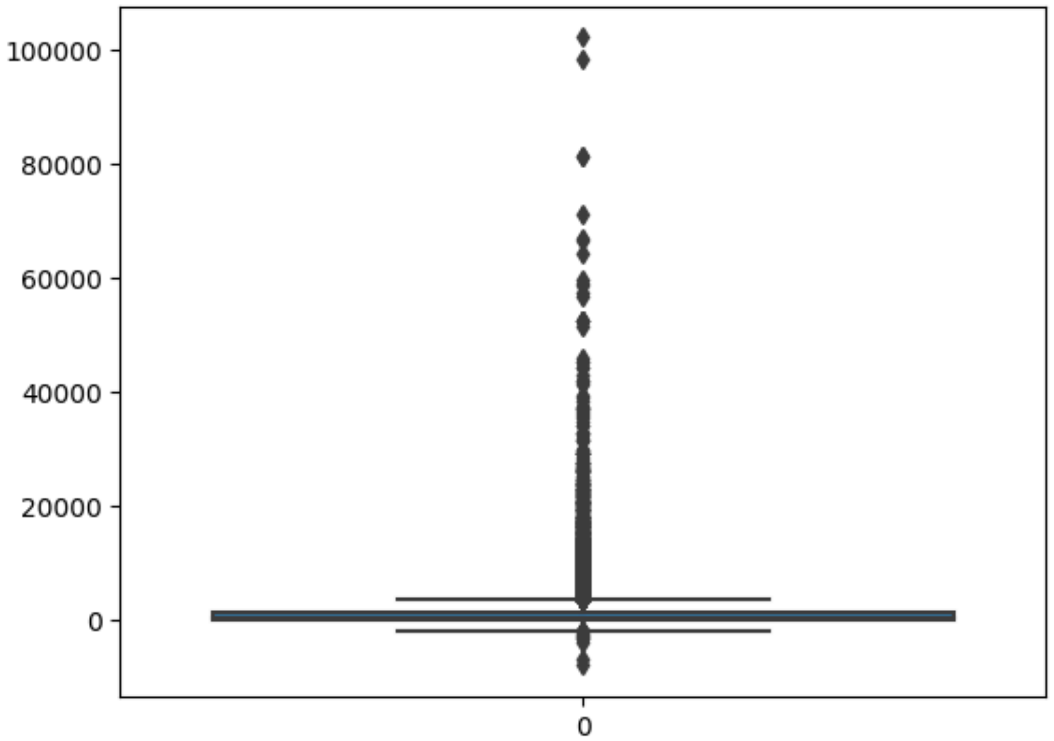


Hình 4.1.8. Biểu đồ phân bố tuổi khách hàng (thuộc tính "age") của bộ dữ liệu df_train

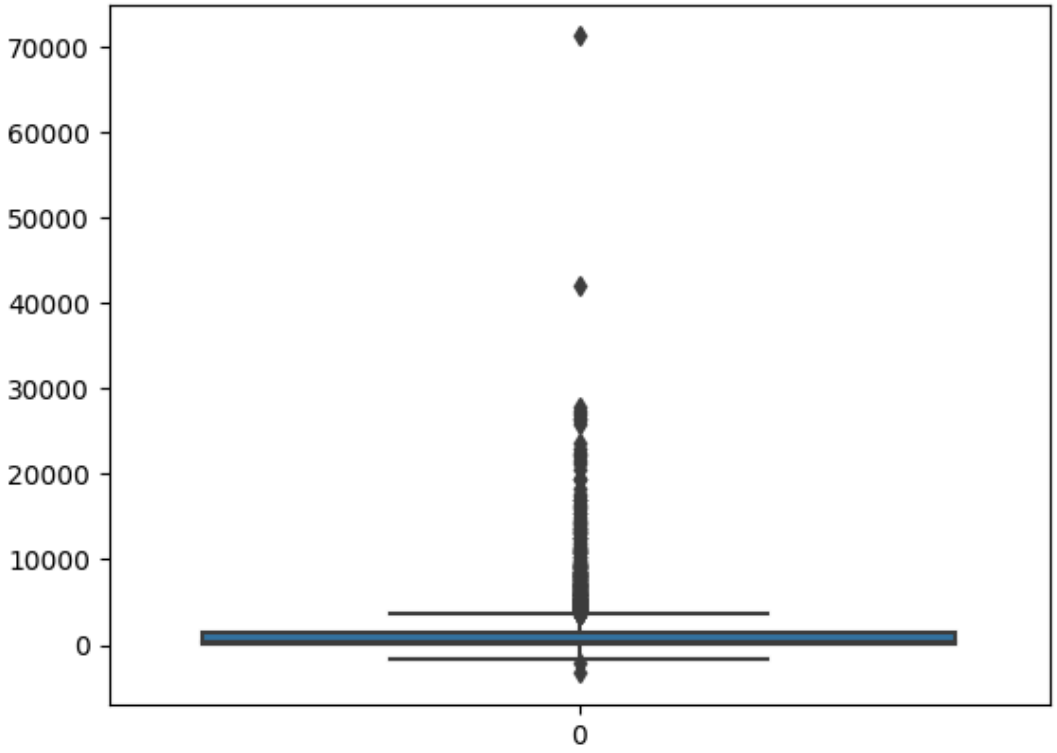


Hình 4.1.9. Biểu đồ phân bố tuổi khách hàng (thuộc tính "age") của bộ dữ liệu df_test

Qua các thông số trong biểu đồ trên, ta thấy rằng phần lớn khách hàng có độ tuổi nằm trong khoảng 30 - 45 tuổi, độ tuổi trung bình là 40. Từ đó suy ra đa số dữ liệu trong cả hai bộ dữ liệu `df_train` và `df_test` là thông tin về những người trong độ tuổi lao động.

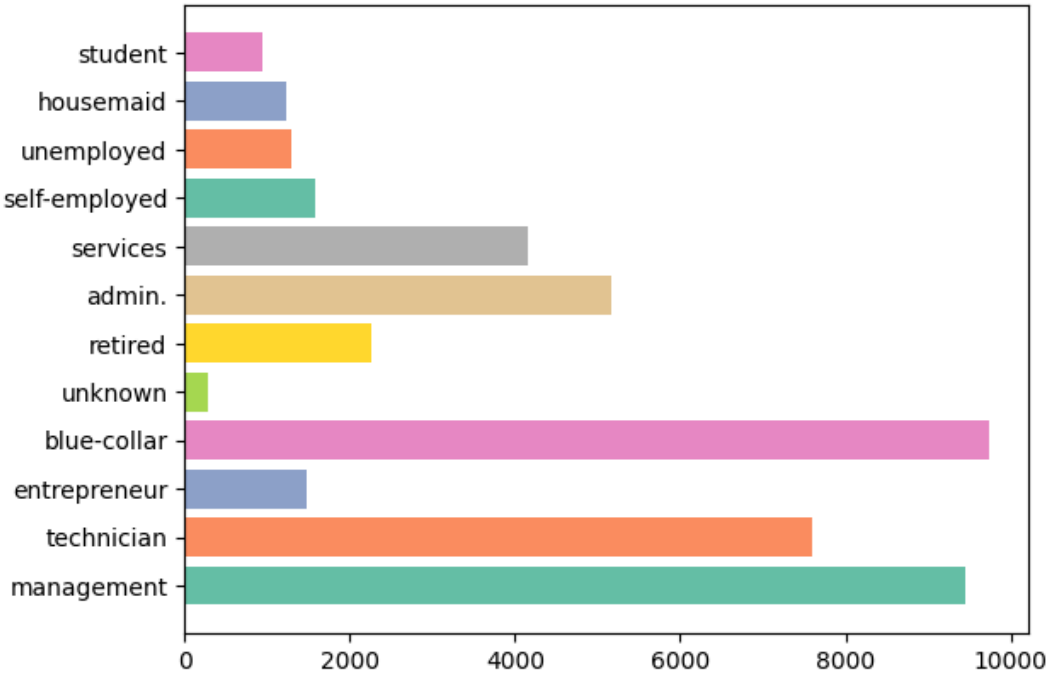


Hình 4.1.10. Biểu đồ thể hiện sự phân bố của số dư tài khoản (thuộc tính "balance") trong bộ dữ liệu df_train

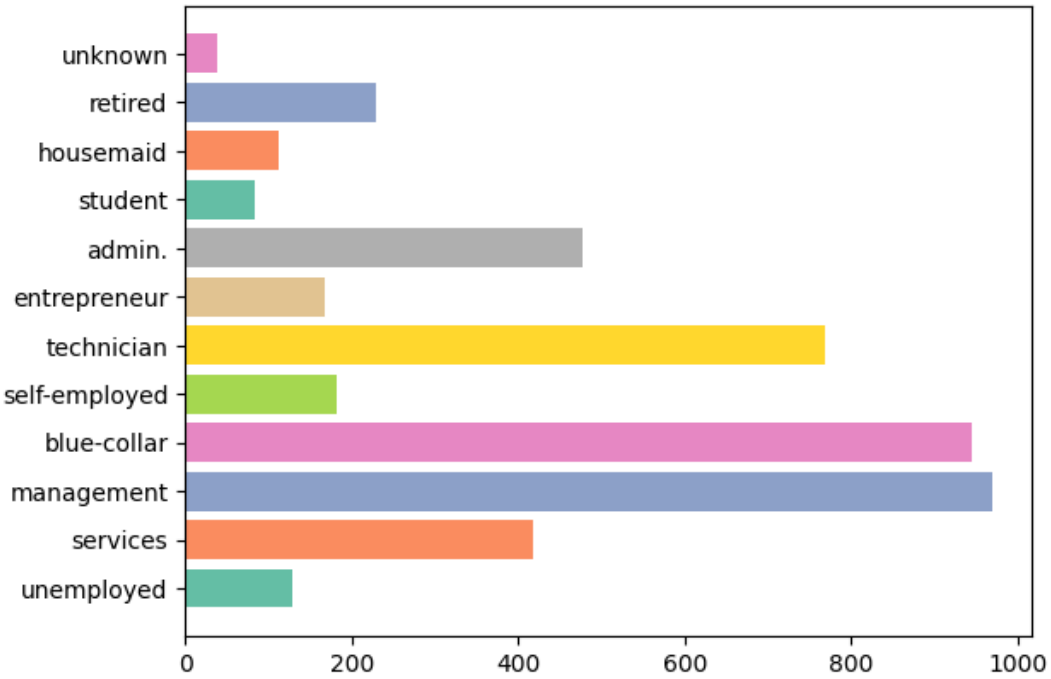


Hình 4.1.11. Biểu đồ thể hiện sự phân bố của số dư tài khoản (thuộc tính "balance") trong bộ dữ liệu df_train

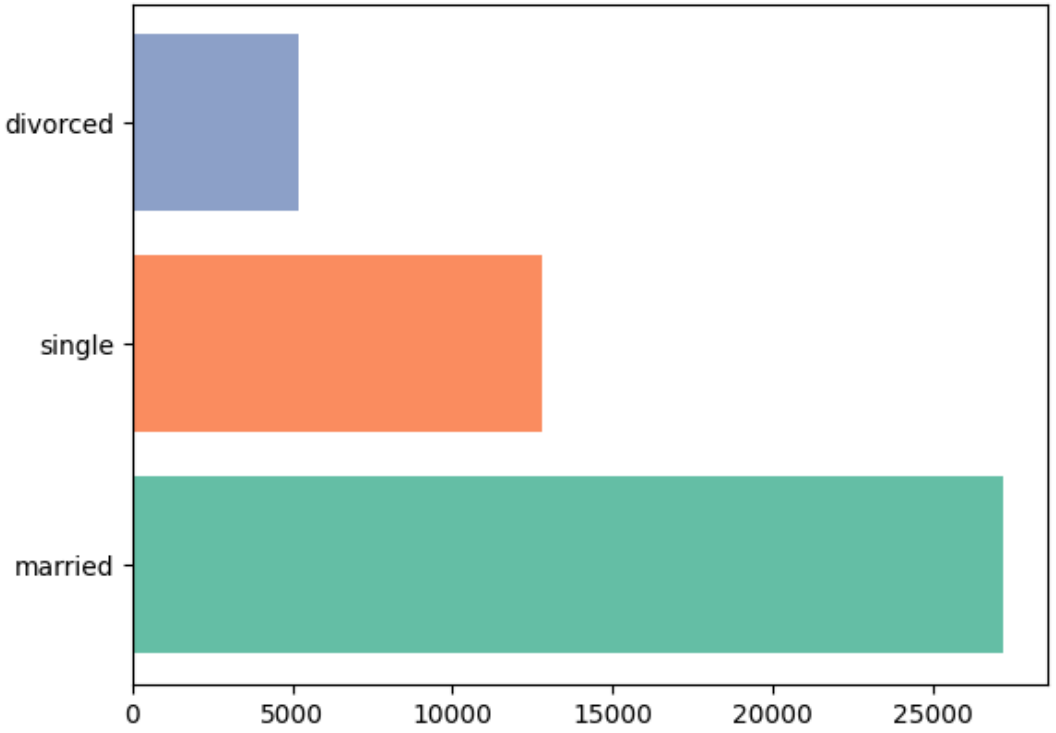
Thuộc tính `['balance']` ở cả hai bộ dữ liệu đều phân bố không đều, nhiều giá trị ngoại lai (outlier) cho thấy rằng số dư tài khoản của các khách hàng có sự chênh lệch khá lớn (từ -8019 đến 102127 euro đối với bộ dữ liệu `df_train` và từ -3313 đến 71188 euro đối với bộ dữ liệu `df_test`. Trong đó, giá trị số dư âm đại diện cho số tiền còn nợ.



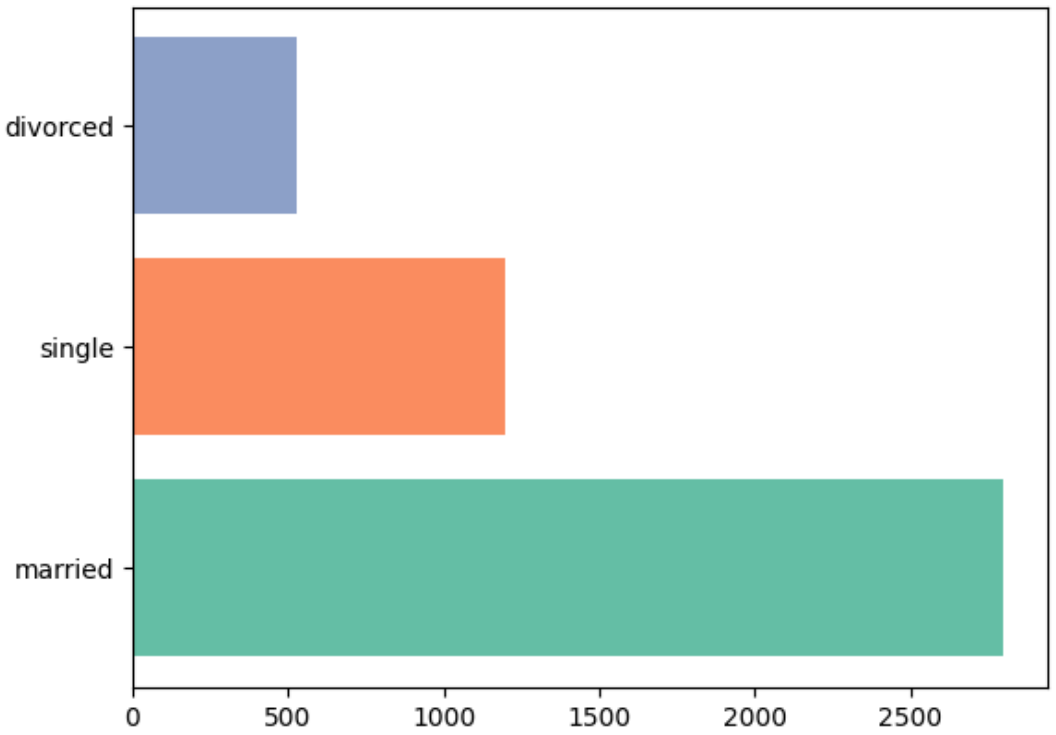
Hình 4.1.12. Biểu đồ biểu diễn số lượng khách hàng với các ngành nghề tương ứng (thuộc tính "job") trong bộ dữ liệu df_train



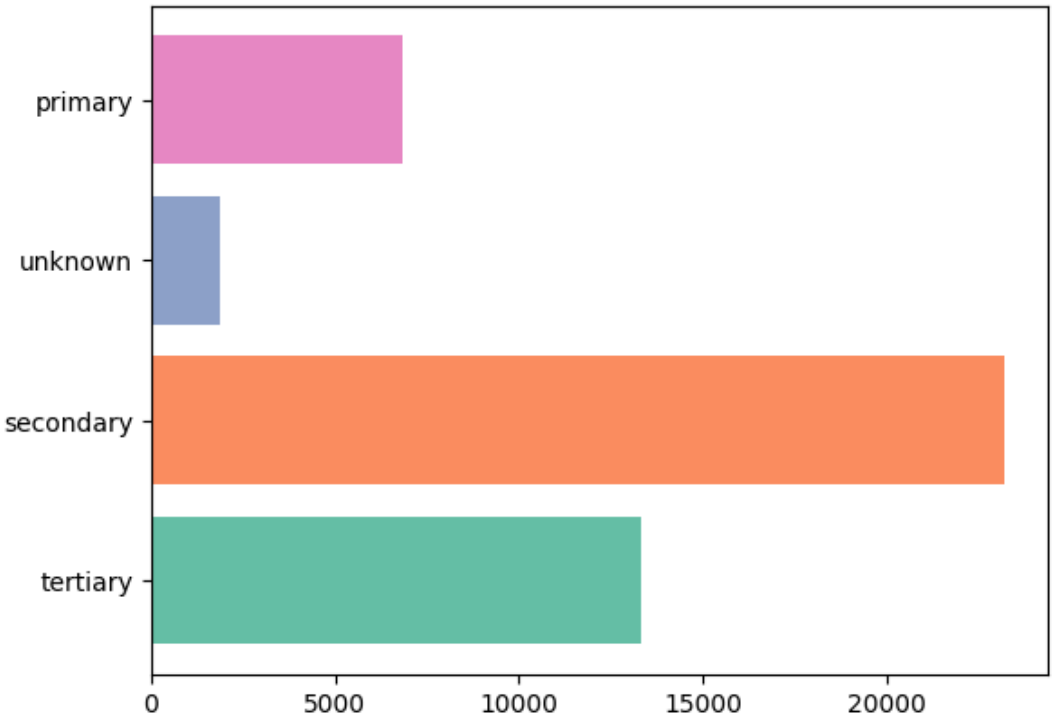
Hình 4.1.13. Biểu đồ biểu diễn số lượng khách hàng với các ngành nghề tương ứng (thuộc tính "job") trong bộ dữ liệu df_test



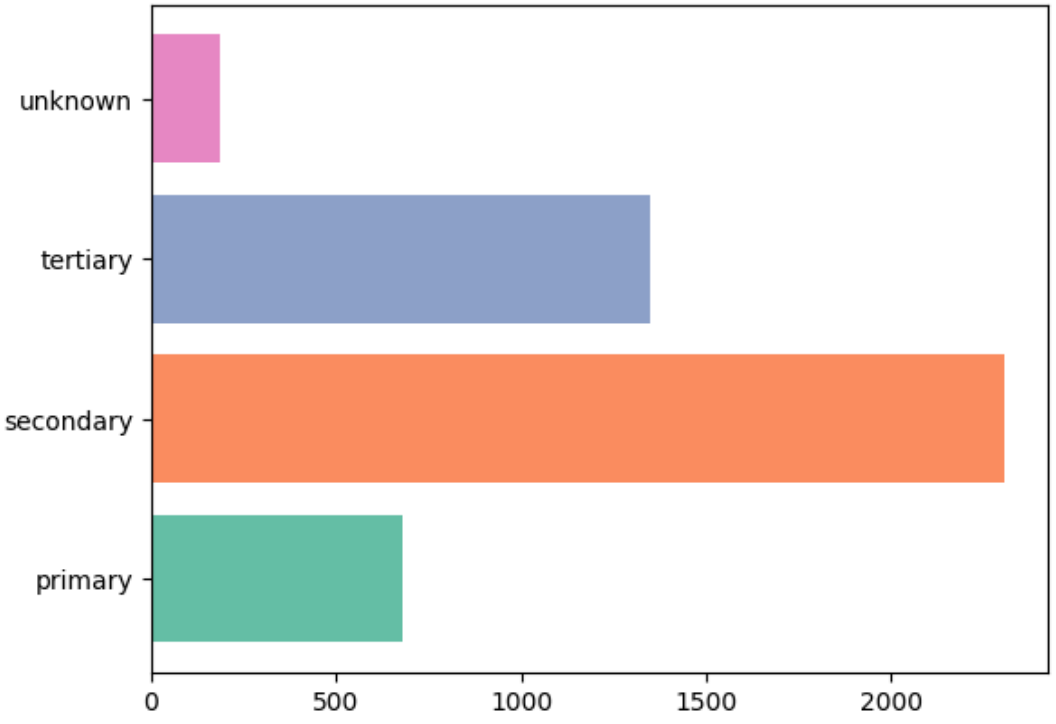
Hình 4.1.14. Biểu đồ biểu diễn số lượng khách hàng có tình trạng hôn nhân tương ứng (thuộc tính "marital") trong bộ dữ liệu df_train



Hình 4.1.15. Biểu đồ biểu diễn số lượng khách hàng có tình trạng hôn nhân tương ứng (thuộc tính "marital") trong bộ dữ liệu df_test

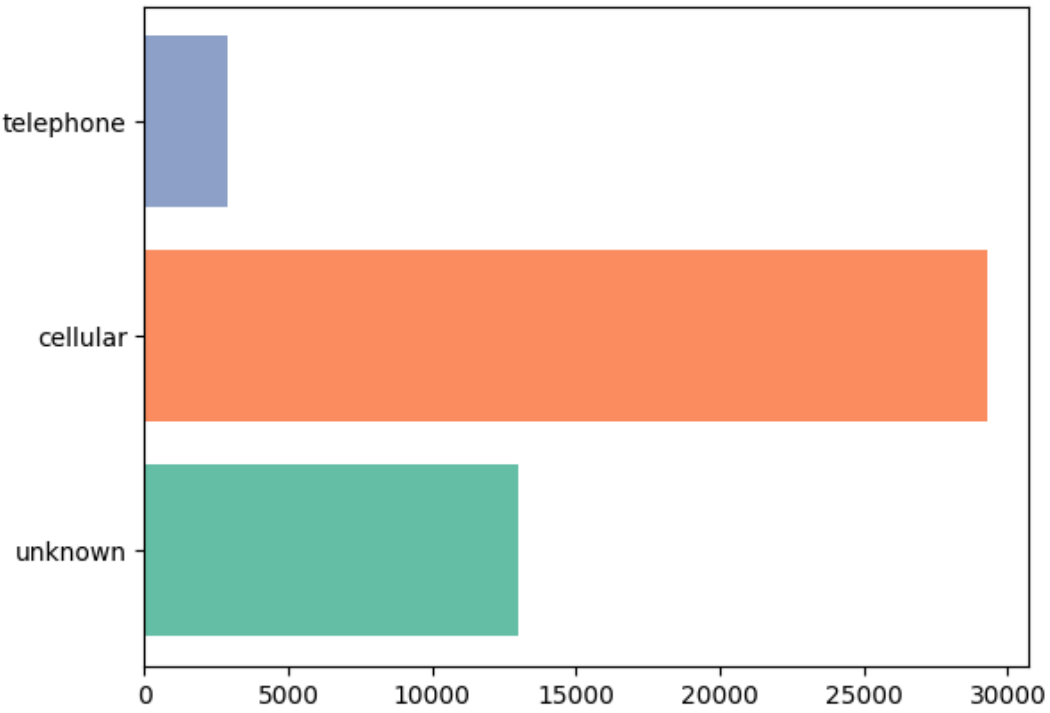


Hình 4.1.16. Biểu đồ biểu diễn số lượng khách hàng với trình độ học vấn tương ứng (thuộc tính "education") trong bộ dữ liệu df_train

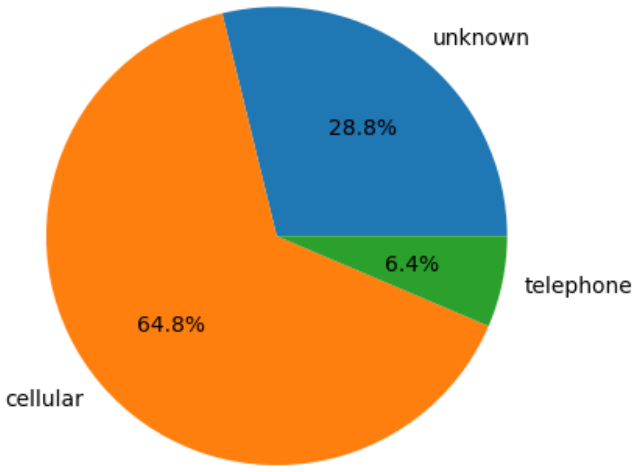


Hình 4.1.17. Biểu đồ biểu diễn số lượng khách hàng với trình độ học vấn

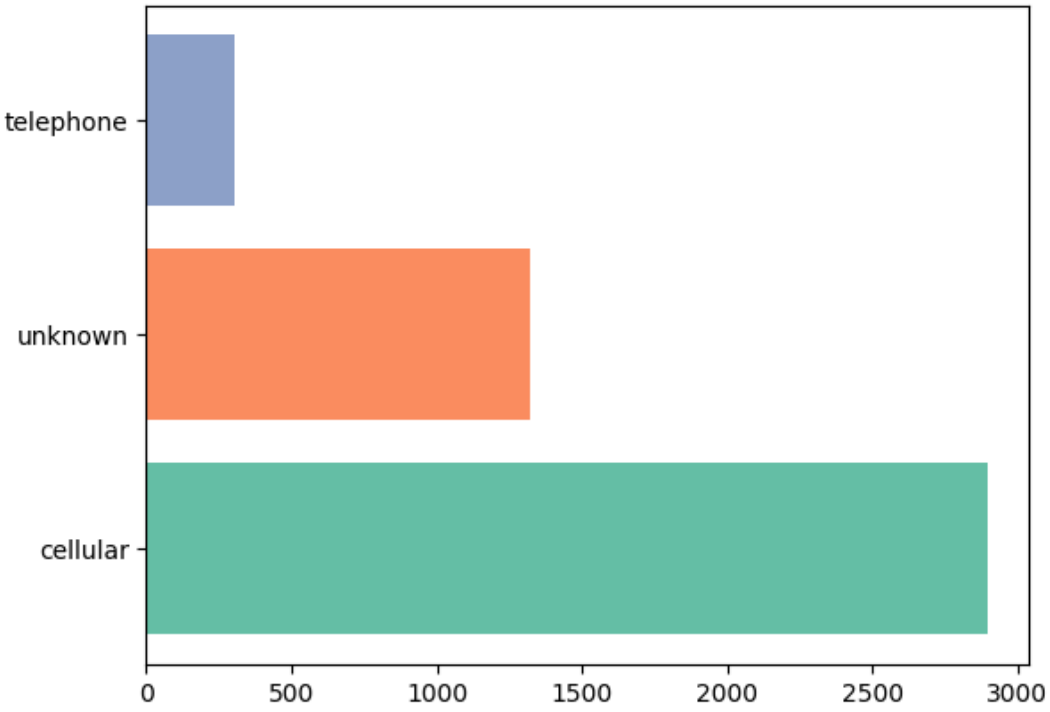
tương ứng (thuộc tính "education") trong bộ dữ liệu df_test



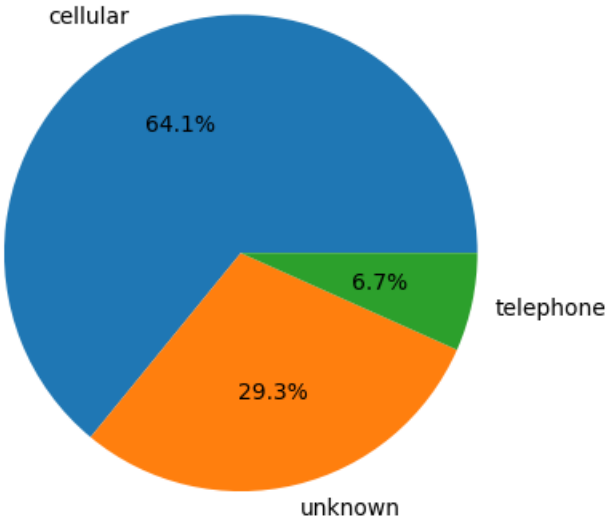
Hình 4.1.18. Biểu đồ biểu diễn số lượng khách hàng với phương thức liên lạc tương ứng (thuộc tính "contact") trong bộ dữ liệu df_train



Hình 4.1.19. Biểu đồ biểu diễn tỷ lệ các phương thức liên lạc (thuộc tính "contact") trong bộ dữ liệu df_train



Hình 4.1.20. Biểu đồ biểu diễn số lượng khách hàng với phương thức liên lạc tương ứng (thuộc tính "contact") trong bộ dữ liệu df_test



Hình 4.1.21. Biểu đồ biểu diễn tỷ lệ các phương thức liên lạc (thuộc tính "contact") trong bộ dữ liệu df_test

Các giá trị của thuộc tính `['contact']` có sự phân bố và tỷ lệ tương đương nhau ở hai bộ dữ liệu `df_train` và `df_test`. Trong đó, giá trị 'unknown'

chiếm tỷ lệ khá lớn (28.8% trong bộ dữ liệu `df_train` và 29.3% trong bộ dữ liệu `df_test`) và mang ý nghĩa mơ hồ, dễ gây ảnh hưởng đến tính chính xác của dữ liệu.

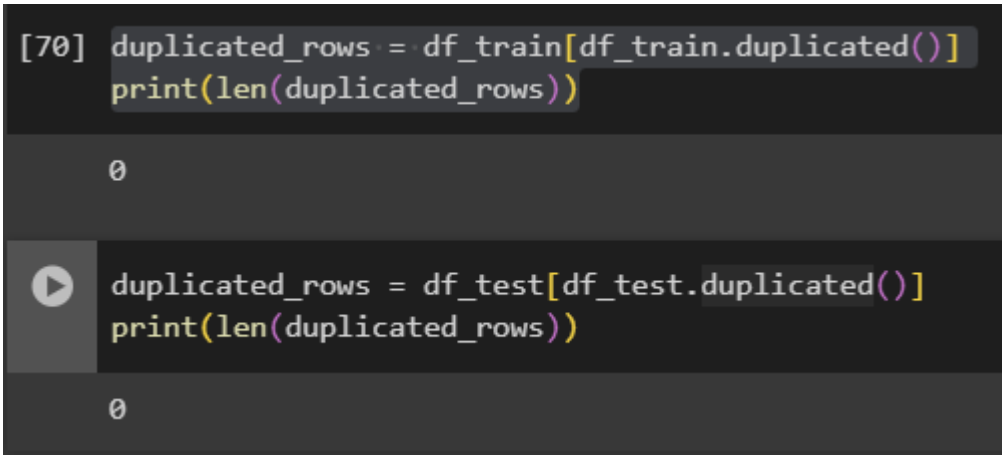
2. Tiền xử lý dữ liệu

Vì giá trị 'unknown' trong thuộc tính `['contact']` chiếm tỷ lệ khá lớn và thuộc tính `['contact']` không ảnh hưởng đáng kể đến kết quả dự đoán nên ta loại bỏ thuộc tính `['contact']` trong cả hai bộ dữ liệu `df_train` và `df_test` bằng các phương thức có sẵn của thư viện Pandas.

Vì thuộc tính `['y']` là thuộc tính trả về kết quả khách hàng có đồng ý gửi tiết kiệm tại ngân hàng không nên ta tiến hành tách cột `['y']` trong bộ dữ liệu `df_train` sang một biến `label` và cột `['y']` trong bộ dữ liệu `df_test` sang biến `label_test` bằng các phương thức có sẵn của thư viện Pandas.

Các thuộc tính có giá trị định danh nhị phân `['default', 'housing', 'loan']` đang mang các giá trị `['yes', 'no']` được chuẩn hóa sang kiểu dữ liệu luận lý `[True, False]` tương ứng bằng các phương thức có sẵn của thư viện Pandas.

Sau khi tiến hành kiểm tra đếm số dòng dữ liệu trùng lặp, kết quả cả hai bộ dữ liệu đều không có dữ liệu trùng nên không cần loại bỏ.



Hình 4.2.1. Kiểm tra số dòng dữ liệu trùng trong cả hai bộ dữ liệu df_train và df_test

Hoàn thành tiền xử lý dữ liệu, ta có hai bộ dữ liệu mới như sau:

```
 #   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  bool
 5   balance    45211 non-null  int64
 6   housing    45211 non-null  bool
 7   loan       45211 non-null  bool
 8   day        45211 non-null  int64
 9   month      45211 non-null  object
 10  duration   45211 non-null  int64
 11  campaign   45211 non-null  int64
 12  pdays      45211 non-null  int64
 13  previous   45211 non-null  int64
 14  poutcome   45211 non-null  object
dtypes: bool(3), int64(7), object(5)
```

Hình 4.2.2. Thông tin của bộ dữ liệu df_train sau khi tiền xử lý

```
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   age        4521 non-null   int64
 1   job        4521 non-null   object
 2   marital    4521 non-null   object
 3   education  4521 non-null   object
 4   default    4521 non-null   bool
 5   balance    4521 non-null   int64
 6   housing    4521 non-null   bool
 7   loan       4521 non-null   bool
 8   day        4521 non-null   int64
 9   month      4521 non-null   object
 10  duration   4521 non-null   int64
 11  campaign   4521 non-null   int64
 12  pdays      4521 non-null   int64
 13  previous   4521 non-null   int64
 14  poutcome   4521 non-null   object
dtypes: bool(3), int64(7), object(5)
```

Hình 4.2.3. Thông tin của bộ dữ liệu df_test sau khi tiền xử lý

Hai bộ dữ liệu sau khi tiền xử lý có 15 thuộc tính, các thuộc tính ở hai bộ dữ liệu là như nhau. Mỗi thuộc tính trong bộ dữ liệu `df_train` có 45211 dòng, không có dữ liệu null. Mỗi thuộc tính trong bộ dữ liệu `df_test` có 4521 dòng, không có dữ liệu null.

Có 5 cột dữ liệu kiểu category `['job', 'marital', 'education', 'month', 'poutcome']`, 3 cột dữ liệu kiểu luận lý `['default', 'housing', 'loan']` và 7 cột dữ liệu kiểu số.

	age	job	marital	education	default	balance	housing	\
0	58	management	married	tertiary	False	2143	True	
1	44	technician	single	secondary	False	29	True	
2	33	entrepreneur	married	secondary	False	2	True	
3	47	blue-collar	married	unknown	False	1506	True	
4	33	unknown	single	unknown	False	1	False	
...	...	...	...	...	...	...	...	
45206	51	technician	married	tertiary	False	825	False	
45207	71	retired	divorced	primary	False	1729	False	
45208	72	retired	married	secondary	False	5715	False	
45209	57	blue-collar	married	secondary	False	668	False	
45210	37	entrepreneur	married	secondary	False	2971	False	

	loan	day	month	duration	campaign	pdays	previous	poutcome
0	False	5	may	261	1	-1	0	unknown
1	False	5	may	151	1	-1	0	unknown
2	True	5	may	76	1	-1	0	unknown
3	False	5	may	92	1	-1	0	unknown
4	False	5	may	198	1	-1	0	unknown
...	...	...	...	...	...	...	...	...
45206	False	17	nov	977	3	-1	0	unknown
45207	False	17	nov	456	2	-1	0	unknown
45208	False	17	nov	1127	5	184	3	success
45209	False	17	nov	508	4	-1	0	unknown
45210	False	17	nov	361	2	188	11	other

Hình 4.2.4. Bộ dữ liệu df_train sau khi tiền xử lý

	age	job	marital	education	default	balance	housing	\
0	30	unemployed	married	primary	False	1787	False	
1	33	services	married	secondary	False	4789	True	
2	35	management	single	tertiary	False	1350	True	
3	30	management	married	tertiary	False	1476	True	
4	59	blue-collar	married	secondary	False	0	True	
...	...	...	...	...	...	...	...	
4516	33	services	married	secondary	False	-333	True	
4517	57	self-employed	married	tertiary	True	-3313	True	
4518	57	technician	married	secondary	False	295	False	
4519	28	blue-collar	married	secondary	False	1137	False	
4520	44	entrepreneur	single	tertiary	False	1136	True	

	loan	day	month	duration	campaign	pdays	previous	poutcome
0	False	19	oct	79	1	-1	0	unknown
1	True	11	may	220	1	339	4	failure
2	False	16	apr	185	1	330	1	failure
3	True	3	jun	199	4	-1	0	unknown
4	False	5	may	226	1	-1	0	unknown
...	...	...	...	...	...	...	...	...
4516	False	30	jul	329	5	-1	0	unknown
4517	True	9	may	153	1	-1	0	unknown
4518	False	19	aug	151	11	-1	0	unknown
4519	False	6	feb	129	4	211	3	other
4520	True	3	apr	345	2	249	7	other

Hình 4.2.5. Bộ dữ liệu df_test sau khi tiền xử lý

V. THỰC HIỆN GOM CỤM BẰNG THUẬT TOÁN K-MEANS

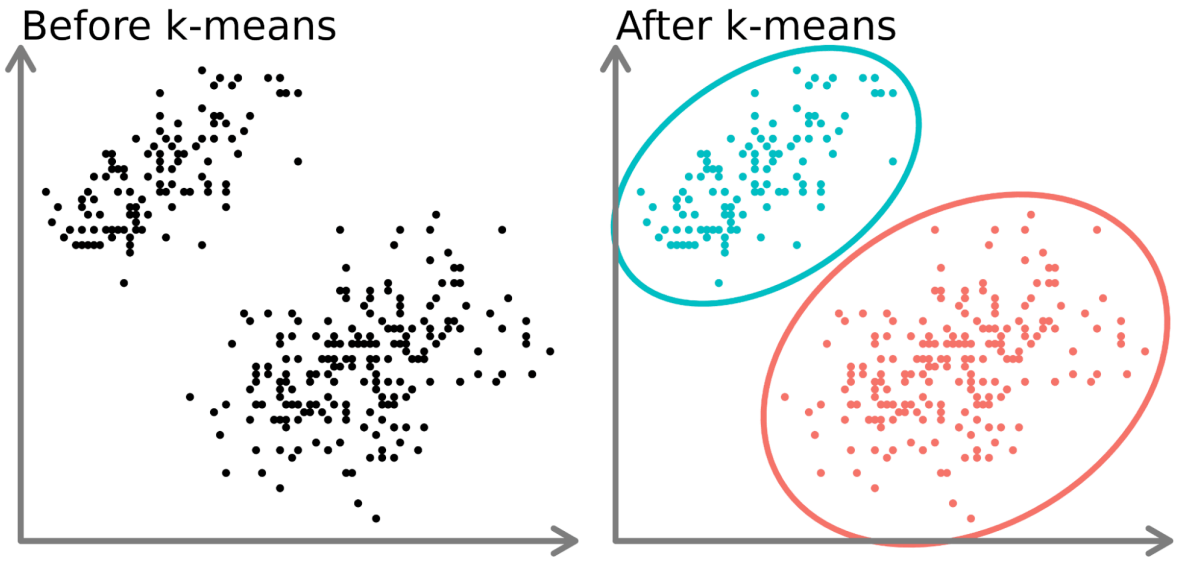
1. Khái quát thuật toán K-Means

Thuật toán K-Means là một trong những thuật toán được sử dụng phổ biến nhất trong lĩnh vực học máy (machine learning) nói chung và khai phá dữ liệu nói

riêng dùng để phân cụm dữ liệu dựa trên phương pháp học không giám sát [15].

Ta có bộ dữ liệu D = {***x1, x2,...,x***$_r$} với x_i là một vector n chiều trong không gian Euclid. Ý tưởng của thuật toán K-Means là phân dữ liệu D thành K cụm dữ liệu, trong đó:

- Mỗi cụm có một centroid (điểm trung tâm)
- K là một hằng số được xác định trước



Hình x. Dữ liệu đầu vào và đầu ra của thuật toán K-Means

2. Xử lý chuẩn hóa dữ liệu cho gom cụm

Sử dụng thư viện `sklearn.preprocessing.LabelEncoder` để mã hóa các dữ liệu dạng category về dạng số.

Bảng dữ liệu sau khi đã chuẩn hóa:

	age	job	marital	education	default	balance	housing	loan	day	month	duration	campaign	pdays	previous	poutcome
0	58	4	1	2	False	2143	True	False	5	8	261	1	-1	0	3
1	44	9	2	1	False	29	True	False	5	8	151	1	-1	0	3
2	33	2	1	1	False	2	True	True	5	8	76	1	-1	0	3
3	47	1	1	3	False	1506	True	False	5	8	92	1	-1	0	3
4	33	11	2	3	False	1	False	False	5	8	198	1	-1	0	3
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
45206	51	9	1	2	False	825	False	False	17	9	977	3	-1	0	3
45207	71	5	0	0	False	1729	False	False	17	9	456	2	-1	0	3
45208	72	5	1	1	False	5715	False	False	17	9	1127	5	184	3	2
45209	57	1	1	1	False	668	False	False	17	9	508	4	-1	0	3
45210	37	2	1	1	False	2971	False	False	17	9	361	2	188	11	1

Hình 5.2.1: Bảng dữ liệu sau khi chuẩn hóa bằng LabelEncoder

Sau khi chuẩn hóa, các dữ liệu dạng category chuyển thành dạng số nguyên từ 0 đến n với n là số lượng giá trị trong mỗi trường dữ liệu.

- Cột job:

Giá trị gốc	Giá trị sau khi chuẩn hóa
management	4
technician	9
entrepreneur	2
blue-collar	1
unknown	11
retired	5
admin.	0
services	7
self-employed	6
unemployed	10
housemaid	3
student	8

- Cột marital:

Giá trị gốc	Giá trị sau khi chuẩn hóa
married	1
single	2
divorced	0

- Cột education:

Giá trị gốc	Giá trị sau khi chuẩn hóa

tertiary	2
secondary	1
unknown	3
primary	0

- Cột month:

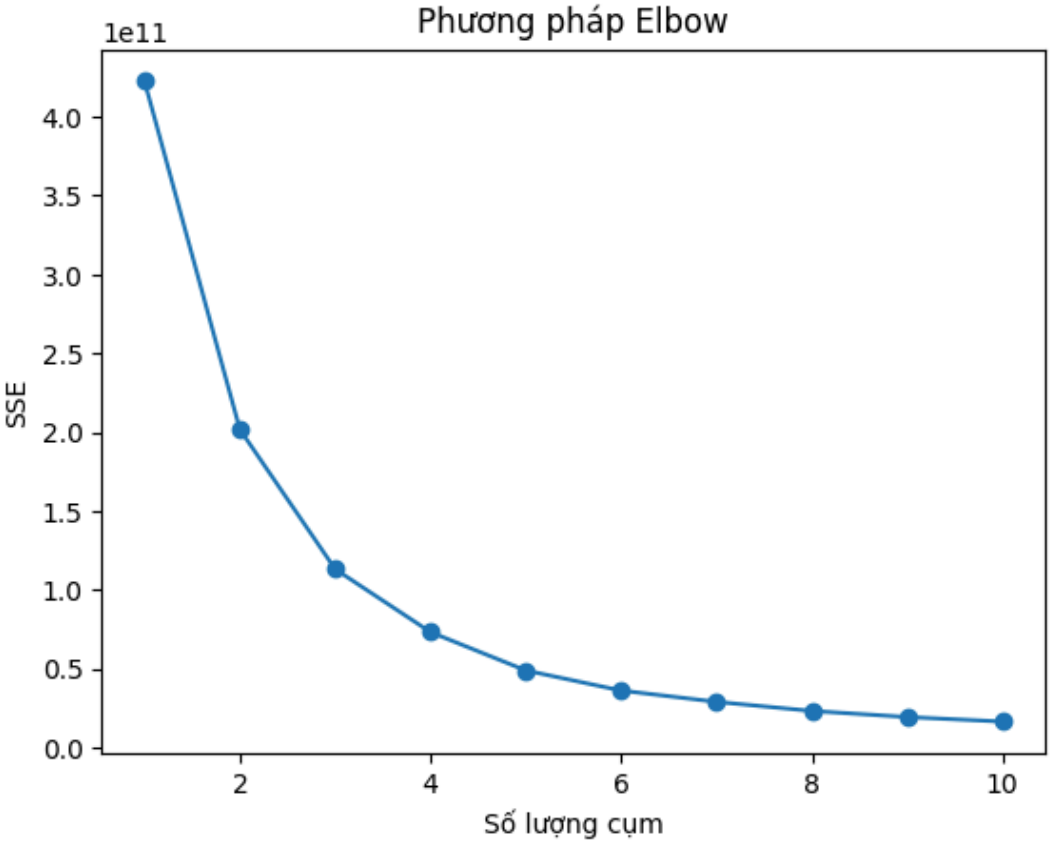
Giá trị gốc	Giá trị sau khi chuẩn hóa
may	8
jun	6
jul	5
aug	1
oct	10
nov	9
dec	3
jan	4
feb	3
mar	7
apr	0
sep	11

- Cột poutcome:

Giá trị gốc	Giá trị sau khi chuẩn hóa
unknown	3
failure	0
other	1
success	2

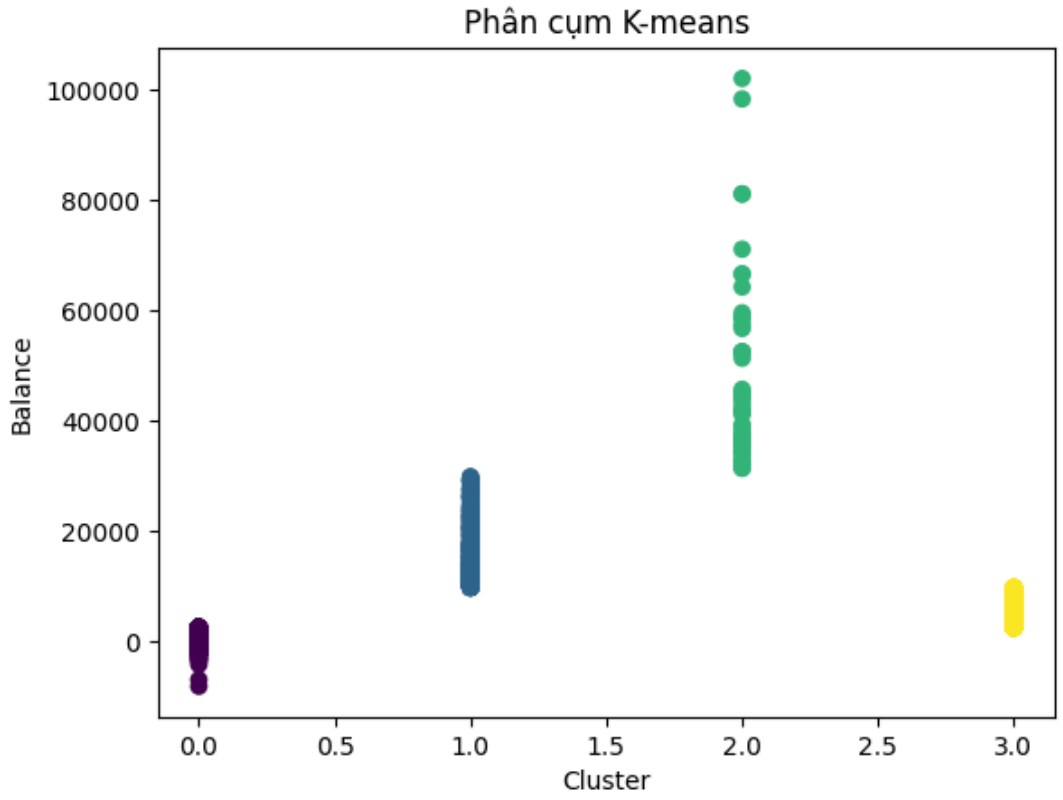
3. Thực hiện gom cụm bằng thuật toán K - Means của thư viện sklearn

Sử dụng thư viện `sklearn.cluster.KMeans` để gom cụm bằng thuật toán K-Means. Vẽ đồ thị thể hiện mối tương quan giữa số cụm và tổng khoảng cách Euclid bình phương của mỗi điểm đến trọng tâm gần nhất bằng thư viện `matplotlib.pyplot`.



Hình 5.3.1. Đồ thị hàm biến dạng của thuật toán k-Means

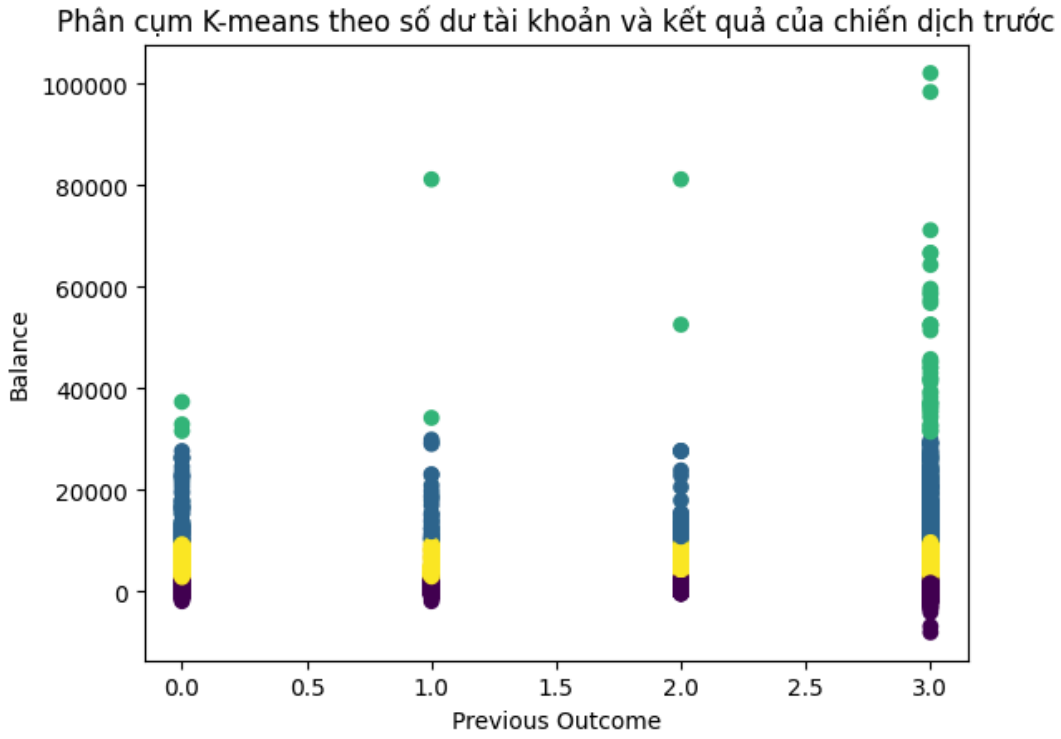
Đối với phương pháp Elbow dùng để xác định số cụm tối ưu, ta cần chọn giá trị tại điểm mà đồ thị bắt đầu có sự suy giảm ổn định và không quá đáng kể, còn gọi là "điểm khuỷu tay". Trong trường hợp này, chúng ta có thể kết luận số cụm tối ưu để phân cụm dữ liệu là 4 cụm.



Hình 5.3.2. Kết quả sau khi phân cụm bộ dữ liệu thành 4 cụm, dựa vào thuộc tính "balance"

Đồ thị trên cho thấy:

- Cụm 1: Các khách hàng có số dư trung bình hằng năm từ dưới 0 đến 5000 euro.
- Cụm 2: Các khách hàng có số dư trung bình hằng năm từ 5000 đến 10000 euro.
- Cụm 3: Các khách hàng có số dư trung bình hằng năm từ 10000 đến 30000 euro.
- Cụm 4: Các khách hàng có số dư trung bình hằng năm từ 30000 đến trên 100000 euro (đa số nằm trong khoảng từ 25000 đến 45000 euro).



Hình 5.3.3. Kết quả sau khi phân cụm bộ dữ liệu thành 4 cụm, dựa vào thuộc tính "balance" và "poutcome"

Kết quả đồ thị cho ta thấy như sau:

- Cụm 1: Các khách hàng có số dư trung bình hằng năm từ dưới 0 đến 5000 euro. Kết quả từ chiến dịch trước đa số là chưa rõ.
- Cụm 2: Các khách hàng có số dư trung bình hằng năm từ 5000 đến 10000 euro. Kết quả từ chiến dịch trước có tỷ lệ đồng đều nhưng tỷ lệ thất bại vẫn còn cao.
- Cụm 3: Các khách hàng có số dư trung bình hằng năm từ 10000 đến 30000 euro. Kết quả từ chiến dịch trước đa số là thất bại và chưa rõ, tỷ lệ thành công thấp.
- Cụm 4: Các khách hàng có số dư trung bình hằng năm từ dưới 30000 đến trên 100000 euro. Kết quả từ chiến dịch trước hầu như là chưa rõ, tỷ lệ thành công thấp.

Từ kết quả được biểu diễn qua *hình 5.2.3*, ta có thể kết luận rằng phần lớn khách hàng đồng ý ký hợp đồng gửi tiết kiệm có kỳ hạn là khách hàng có số dư trung bình hằng năm từ 10000 đến 30000 euro. Với kết luận trên, ngân hàng có thể đưa ra quyết định chú trọng tiếp thị cho nhóm khách hàng này một cách nhiệt tình và thường xuyên hơn để tăng khả năng thành công của chiến dịch, góp phần mang lại doanh thu cho ngân hàng.

VI. TÌM LUẬT KẾT HỢP BẰNG APRIORI

1. Khái quát thuật toán Apriori

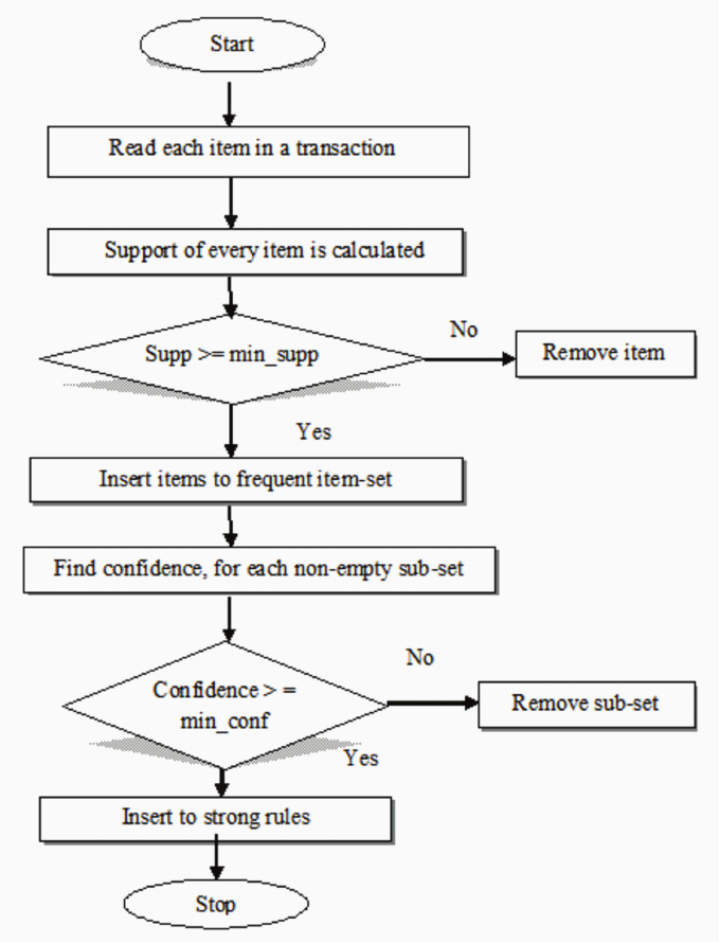
Thuật toán Apriori là một trong những thuật toán quan trọng được sử dụng trong lĩnh vực khai phá dữ liệu để phân tích các tập dữ liệu lớn và đưa ra các luật kết hợp. Ý tưởng của thuật toán là tìm tất cả tập các hạng mục (itemsets) có độ hỗ trợ (support) lớn hơn hoặc bằng độ hỗ trợ tối thiểu được quy định trước (minSupp) [16].

Độ hỗ trợ (support) là "độ đo" cho 1 tập itemset. Những tập có độ hỗ trợ càng lớn thì tỷ lệ xuất hiện càng cao và được xem là những tập "đáng quan tâm" để tiến hành khai thác. Độ hỗ trợ của tập S được tính theo công thức:

$$Supp(S) = Count(S) / N$$

Độ tin cậy (confidence) là "độ đo" cho 1 luật (rule) có dạng L -> R dùng để xác định tỷ lệ R xuất hiện mỗi khi L xuất hiện. Độ tin cậy của luật L -> R được tính như sau:

$$Conf(L \rightarrow R) = Count(L \cup R) / Count(L)$$



Hình 6.1.1. Sơ đồ khối biểu diễn các bước của Apriori

2. Thực hiện tìm luật kết hợp

Tạo một bộ dữ liệu mới df_ap từ bộ dữ liệu df_train với các cột `['job', 'marital', 'education', 'default', 'balance', 'poutcome']`.

	job	marital	education	default	balance	poutcome
0	management	married	tertiary	False	2143	unknown
1	technician	single	secondary	False	29	unknown
2	entrepreneur	married	secondary	False	2	unknown
3	blue-collar	married	unknown	False	1506	unknown
4	unknown	single	unknown	False	1	unknown

Hình 6.2.1. Bộ dữ liệu df_ap

```
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   job        45211 non-null  object
 1   marital    45211 non-null  object
 2   education  45211 non-null  object
 3   default    45211 non-null  bool
 4   balance    45211 non-null  int64
 5   poutcome   45211 non-null  object
dtypes: bool(1), int64(1), object(4)
```

Hình 6.2.2. Thông tin bộ dữ liệu df_ap

Bộ dữ liệu đều có 6 thuộc tính, mỗi thuộc tính trong bộ dữ liệu có 45211 dòng, không có dữ liệu null. Có 4 cột dữ liệu kiểu category `['job', 'marital', 'education', 'poutcome']`, 1 cột dữ liệu kiểu luận lý và 1 cột dữ liệu kiểu số.

Chuyển đổi bộ dữ liệu thành kiểu dữ liệu list bằng các phương thức có sẵn của thư viện Pandas. List mới lưu vào biến `transactions`, với 45211 phần tử, mỗi phần tử là một list các giá trị trong một hàng của bộ dữ liệu `df_ap`.

Cài đặt thư viện `apyori` và sử dụng để tìm luật kết hợp từ bộ dữ liệu `df_ap`.

Tiến hành tìm luật kết hợp bằng phương thức apriori với các giá trị đầu vào transactions = `transactions` (list vừa tạo ở trên từ bộ dữ liệu `df_ap`), min_support = 0.01, min_confidence = 0.6, min_lift = 3, min_length = 2.

Ta có kết quả như sau:

	items	support	ordered_statistics
0	(single, student)	0.019420	[((student), (single), 0.9360341151385927, 3.3...
1	(single, False, student)	0.019376	[((student), (False, single), 0.93390191897654...
2	(student, single, secondary)	0.010772	[((student, secondary), (single), 0.9586614173...
3	(unknown, single, student)	0.014355	[((student), (single, unknown), 0.691897654584...
4	(unknown, management, 0, tertiary)	0.013183	[((management, 0), (tertiary, unknown), 0.7095...
5	(student, single, False, secondary)	0.010772	[((student, secondary), (False, single), 0.958...
6	(single, unknown, False, student)	0.014311	[((student), (False, single, unknown), 0.68976...
7	(management, False, tertiary, 0, unknown)	0.012608	[((False, management, 0), (unknown, tertiary),...

Hình 6.2.3a.

ordered_statistics
frozenset({'student'}),frozenset({'single'}),0.9360341151385927,3.3087598420274364
frozenset({'student'}),frozenset({'False', 'single'}),0.9339019189765457,3.3640857030394877,frozenset({'False', 'student'}),frozenset({'single'}),0.9368983957219251,3.3118149623912396
frozenset({'student', 'secondary'}),frozenset({'single'}),0.9586614173228346,3.3887444361675274
frozenset({'student'}),frozenset({'single', 'unknown'}),0.6918976545842217,3.031729488409309,frozenset({'unknown', 'student'}),frozenset({'single'}),0.9297994269340975,3.2867210235431963
frozenset({'management', '0'}),frozenset({'tertiary', 'unknown'}),0.7095238095238096,3.0123280075482164
frozenset({'student', 'secondary'}),frozenset({'False', 'single'}),0.9586614173228346,3.453273949373172,frozenset({'student', 'False', 'secondary'}),frozenset({'single'}),0.9586614173228346,3.3887444361675274
frozenset({'student'}),frozenset({'False', 'single', 'unknown'}),0.6897654584221747,3.0873167152484844,frozenset({'False', 'student'}),frozenset({'single', 'unknown'}),0.6919786096256685,3.0320842139742297,frozenset({'unknown', 'student'}),frozenset({'False', 'single'}),0.9269340974212034,3.3389863340379273,frozenset({'unknown', 'False', 'student'}),frozenset({'single'}),0.9309352517985611,3.290736017909675
frozenset({'False', 'management', '0'}),frozenset({'unknown', 'tertiary'}),0.7071960297766751,3.0024452720662276

Hình 6.2.3b.

Hình 6.2.3. Bảng kết quả sau khi tìm luật kết hợp từ bộ dữ liệu df_ap

Từ *Hình 6.2.3.* ta có được giá trị độ hỗ trợ (*support*) lớn nhất là 0.0194, tức tần suất xuất hiện các luật kết hợp chưa đến 2%. Từ đó cho thấy bộ dữ liệu không tồn tại itemset nào có tần suất xuất hiện đáng kể. Kết quả sau khi thực hiện tìm luật kết hợp bằng thuật toán Apriori không có ý nghĩa đối với bộ dữ liệu hiện tại.

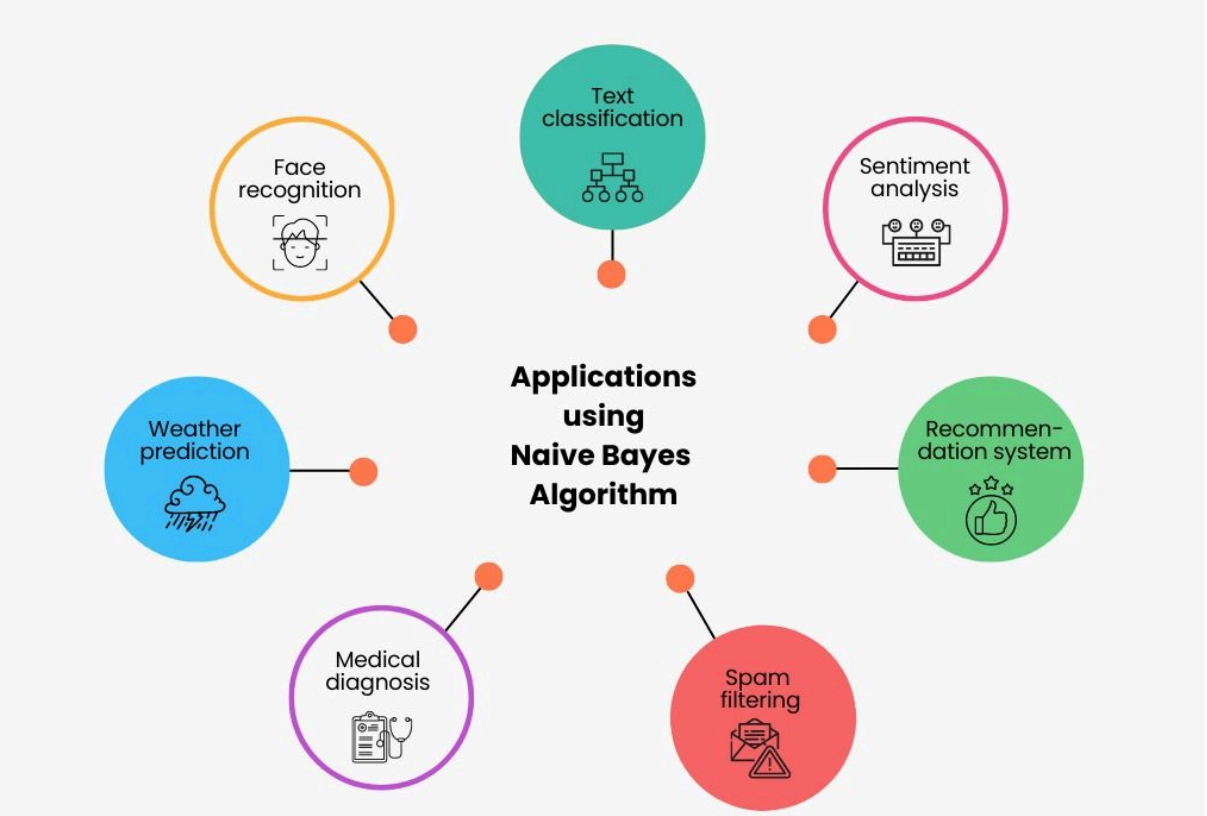
VII. THUẬT TOÁN PHÂN LỚP NAIVE BAYES

1. Khái quát thuật toán Naive Bayes

Naive Bayes classifier là một thuật toán thuộc nhóm các thuật toán áp dụng định lý Bayes và giả định "ngây thơ" (naive). Giả định này cho rằng mọi thuộc tính đầu vào (ví dụ: độ tuổi, nghề nghiệp, tình trạng hôn nhân,...) đều độc lập

với nhau, giá trị của thuộc tính này không liên quan và không làm ảnh hưởng đến giá trị của một thuộc tính nào khác [12].

Naive Bayes thường được sử dụng trong các bài toán phân loại dữ liệu dạng văn bản, nhận diện văn bản spam, xây dựng các mô hình dự đoán và nhiều ứng dụng khác trong lĩnh vực học máy (*machine learning*).



Hình 7.1.1. Một số ứng dụng của thuật toán Naive Bayes

Đề tài sử dụng phân phối Gaussian Naive Bayes để áp dụng cho bộ dữ liệu mà thành phần của nó là các biến liên tục. Gaussian Naive Bayes là một phân phối phổ biến, dễ dàng thực hiện sau khi đã tính được giá trị trung bình và độ lệch chuẩn từ bộ dữ liệu "train" của đề tài.

2. Định lý Bayes

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

Hình 7.2.1. Công thức tính xác suất có điều kiện

Công thức trên giúp chúng ta xác định được tần suất của A xảy ra khi điều kiện B đã xảy ra, ký hiệu là P(A|B). Xác suất P(A|B) có thể được tính dựa vào xác suất P(B|A), P(A) và P(B) [12]. Trong đó:

- P(A|B): Xác suất A xảy ra khi B đã xảy ra
- P(A): Xác suất A xảy ra
- P(B|A): Xác suất B xảy ra khi A đã xảy ra
- P(B): Xác suất B xảy ra

3. Ưu điểm

Bộ phân lớp Naive Bayes có một số điểm nổi trội hơn so với các thuật toán phân lớp khác như sau:

- Thuật toán đơn giản và dễ triển khai nên chỉ cần số lượng ít dữ liệu để huấn luyện cho mô hình.
- Naive Bayes có khả năng xử lý dữ liệu lớn với hiệu suất cao. Khi số lượng thuộc tính hay số thể hiện của bộ dữ liệu tăng lên thì hiệu suất suy giảm không đáng kể.
- Có thể đưa ra dự đoán cho bộ dữ liệu test một cách nhanh chóng,

4. Nhược điểm

Bên cạnh ưu điểm thì Naive Bayes cũng có những hạn chế:

- Naive Bayes giả định rằng tất cả thuộc tính đều độc lập, điều này hầu như ít xảy ra trong thực tế, làm giảm độ chính xác của mô hình.
- Nếu gặp một giá trị chưa từng xuất hiện trong bộ dữ liệu huấn luyện, Naive Bayes sẽ tính ra xác suất bằng 0 và không thể dự đoán trong tương lai.

5. Áp dụng vào dự đoán khả năng đồng ý của khách hàng

Sử dụng thư viện `sklearn.preprocessing.LabelEncoder` để mã hóa các dữ liệu dạng category về dạng số.

Bảng dữ liệu sau khi đã chuẩn hóa:

	age	job	marital	education	default	balance	housing	loan	day	month	duration	campaign	pdays	previous	poutcome
0	58	4	1	2	False	2143	True	False	5	8	261	1	-1	0	3
1	44	9	2	1	False	29	True	False	5	8	151	1	-1	0	3
2	33	2	1	1	False	2	True	True	5	8	76	1	-1	0	3
3	47	1	1	3	False	1506	True	False	5	8	92	1	-1	0	3
4	33	11	2	3	False	1	False	False	5	8	198	1	-1	0	3
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
45206	51	9	1	2	False	825	False	False	17	9	977	3	-1	0	3
45207	71	5	0	0	False	1729	False	False	17	9	456	2	-1	0	3
45208	72	5	1	1	False	5715	False	False	17	9	1127	5	184	3	2
45209	57	1	1	1	False	668	False	False	17	9	508	4	-1	0	3
45210	37	2	1	1	False	2971	False	False	17	9	361	2	188	11	1

Hình 7.5.1: Bảng dữ liệu sau khi chuẩn hóa bằng LabelEncoder

Sau khi chuẩn hóa, các dữ liệu dạng category chuyển thành dạng số nguyên từ 0 đến n với n là số lượng giá trị trong mỗi trường dữ liệu (tương tự phần xử lý dữ liệu ở mục V.2).

Để tăng tính thống nhất cho dữ liệu, ta chuẩn hóa bộ dữ liệu `nb_test` (bản sao từ bộ dữ liệu `df_test`) tương ứng các giá trị đã chuẩn hóa trên.

	age	job	marital	education	default	balance	housing	loan	day	month	duration	campaign	pdays	previous	poutcome
0	30	10	1	0	False	1787	False	False	19	10	79	1	-1	0	3
1	33	7	1	1	False	4789	True	True	11	8	220	1	339	4	0
2	35	4	2	2	False	1350	True	False	16	0	185	1	330	1	0
3	30	4	1	2	False	1476	True	True	3	6	199	4	-1	0	3
4	59	1	1	1	False	0	True	False	5	8	226	1	-1	0	3
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4516	33	7	1	1	False	-333	True	False	30	5	329	5	-1	0	3
4517	57	6	1	2	True	-3313	True	True	9	8	153	1	-1	0	3
4518	57	9	1	1	False	295	False	False	19	1	151	11	-1	0	3
4519	28	1	1	1	False	1137	False	False	6	3	129	4	211	3	1
4520	44	2	2	2	False	1136	True	True	3	0	345	2	249	7	1

Hình 7.5.2: Bảng dữ liệu nb_test sau khi chuẩn hóa

Sử dụng thư viện `sklearn.naive_bayes` để thực hiện phân lớp và dự đoán khả năng khách hàng đồng ý cho vay tại ngân hàng bằng thuật toán Gaussian Naive Bayes.

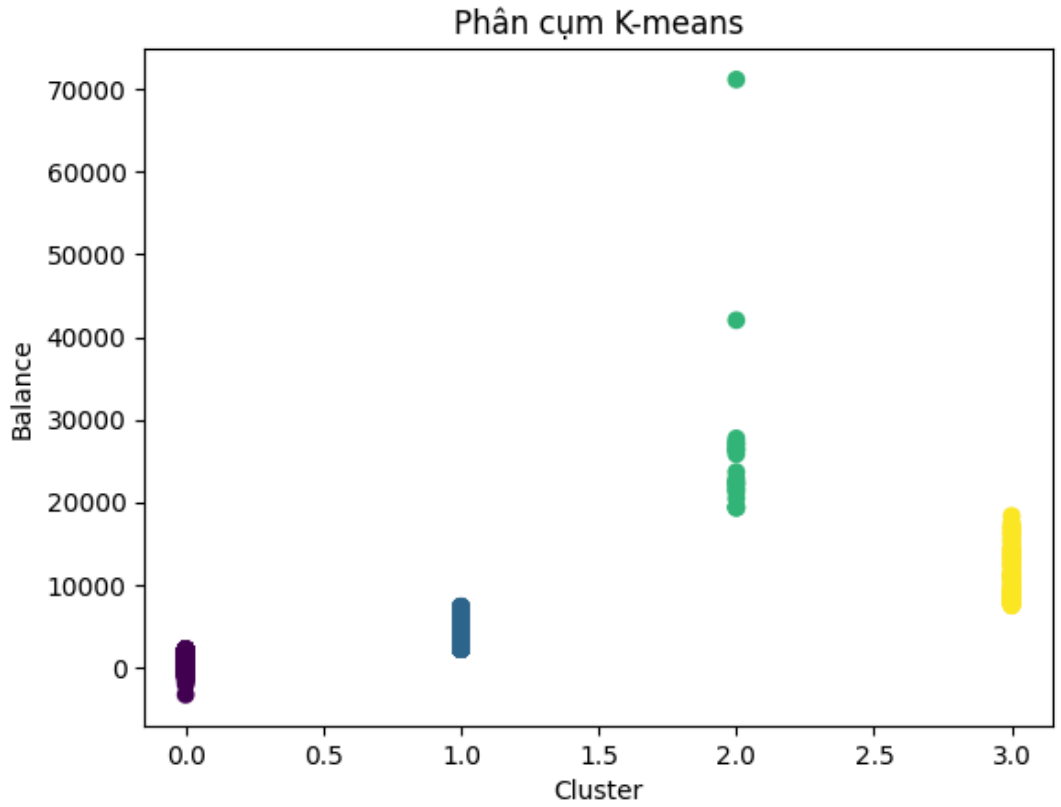
Dùng phương thức GaussianNB() và fit() có sẵn trong thư viện để tiến hành phân lớp dữ liệu với các tham số X là bộ dữ liệu đã chuẩn hóa ở *Hình 7.5.1.* và Y là biến `label` đã tách ra từ bộ dữ liệu `df_train` ban đầu (mục IV.2).

Sử dụng phương thức predict() có sẵn trong thư viện, truyền vào phương thức bộ dữ liệu `nb_test` để dự đoán khả năng đồng ý gửi tiết kiệm tại ngân hàng của từng khách hàng. Kết quả cho ra mảng một chiều chứa giá trị kết quả dự đoán `['yes', 'no']`. Gán kết quả vừa dự đoán được vào biến `result_nb` và thêm thuộc tính `['new_y']` vào cuối bộ dữ liệu `nb_test` sau khi chuyển đổi dữ liệu đã chuẩn hóa bằng thư viện `sklearn.preprocessing.LabelEncoder` về ban đầu.

	age	job	marital	education	default	balance	housing	loan	day	month	duration	campaign	pdays	previous	poutcome	new_y
0	30	unemployed	married	primary	False	1787	False	False	19	oct	79	1	-1	0	unknown	no
1	33	services	married	secondary	False	4789	True	True	11	may	220	1	339	4	failure	no
2	35	management	single	tertiary	False	1350	True	False	16	apr	185	1	330	1	failure	yes
3	30	management	married	tertiary	False	1476	True	True	3	jun	199	4	-1	0	unknown	no
4	59	blue-collar	married	secondary	False	0	True	False	5	may	226	1	-1	0	unknown	no
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4516	33	services	married	secondary	False	-333	True	False	30	jul	329	5	-1	0	unknown	no
4517	57	self-employed	married	tertiary	True	-3313	True	True	9	may	153	1	-1	0	unknown	no
4518	57	technician	married	secondary	False	295	False	False	19	aug	151	11	-1	0	unknown	no
4519	28	blue-collar	married	secondary	False	1137	False	False	6	feb	129	4	211	3	other	no
4520	44	entrepreneur	single	tertiary	False	1136	True	True	3	apr	345	2	249	7	other	no

Hình 7.5.3: Bảng dữ liệu nb_test sau khi phân lớp và dự đoán bằng thuật toán Gaussian Naive Bayes

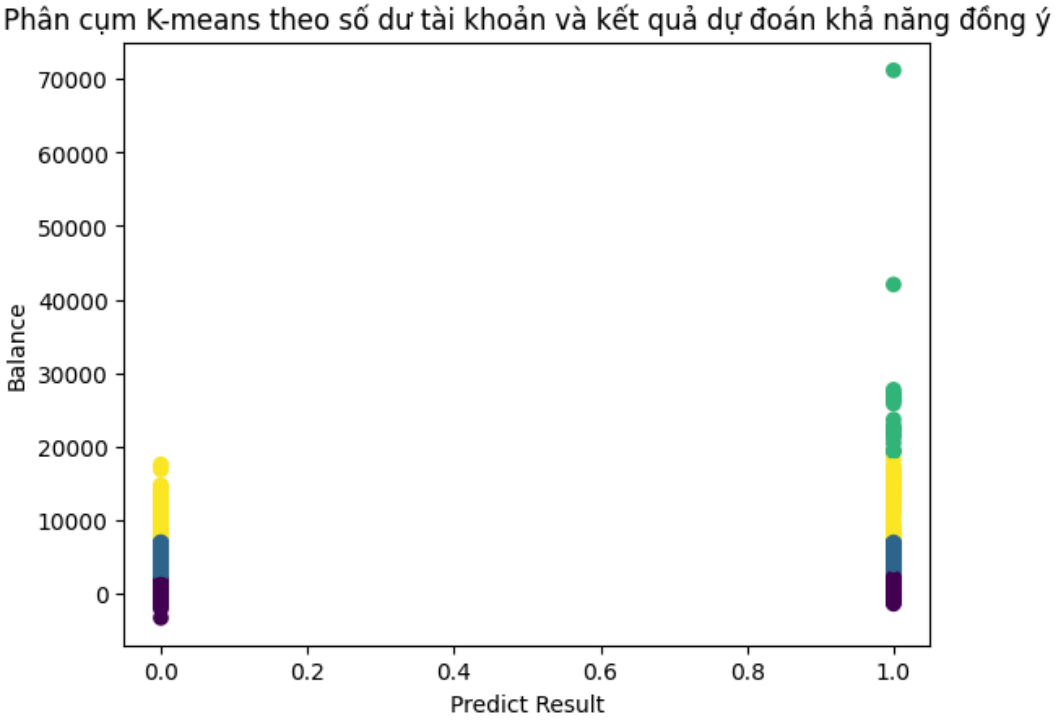
Coi `nb_test` là bộ dữ liệu hoàn chỉnh. Sử dụng thư viện `sklearn.cluster.KMeans` để gom cụm bằng thuật toán K-Means (tương tự như đã làm ở mục V.3). Sau khi gom cụm, ta thấy được một số mối tương quan giữa các thuộc tính như sau:



Hình 7.5.4. Biểu đồ gom cụm bộ dữ liệu nb_test theo thuộc tính "balance"

Từ *Hình 7.5.4.*, ta có thể xác định được các thông tin sau thông qua 4 cụm dữ liệu:

- Cụm 1: Các khách hàng có số dư trung bình hằng năm từ dưới 0 đến 5000 euro.
- Cụm 2: Các khách hàng có số dư trung bình hằng năm từ 5000 đến dưới 10000 euro.
- Cụm 3: Phần lớn khách hàng có số dư trung bình hằng năm từ 18000 đến 25000 euro. Một bộ phận nhỏ có số dư trung bình hằng năm trên 40000 euro và trên 70000 euro.
- Cụm 4: Các khách hàng có số dư trung bình hằng năm từ 8000 đến dưới 20000 euro.

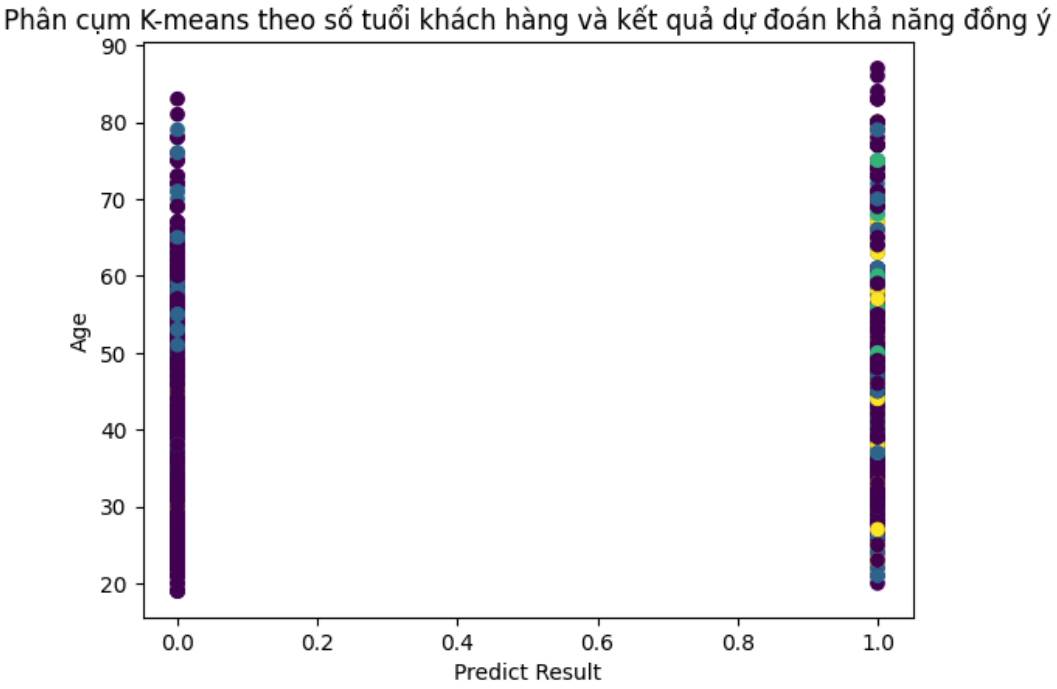


Hình 7.5.5. Biểu đồ gom cụm bộ dữ liệu nb_test theo hai thuộc tính "balance" và "y"

Từ *Hình 7.5.5.*, ta xác định được 4 cụm dữ liệu như sau:

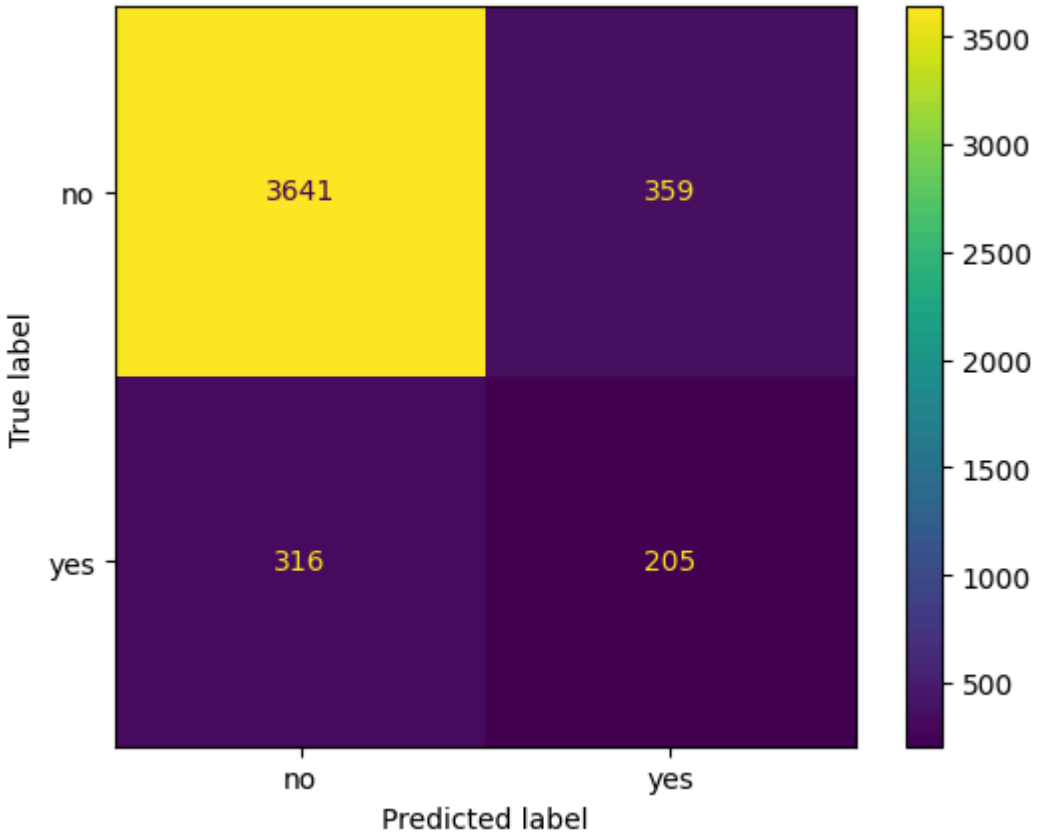
- Cụm 1: Các khách hàng có số dư trung bình hằng năm từ dưới 0 đến 2000 euro. Đa số khách hàng từ chối ký hợp đồng gửi tiết kiệm.
- Cụm 2: Các khách hàng có số dư trung bình hằng năm từ 2000 đến 8000 euro. Đa số khách hàng từ chối ký hợp đồng gửi tiết kiệm.
- Cụm 3: Các khách hàng có số dư trung bình hằng năm từ 8000 đến dưới 20000 euro. Tỷ lệ khách hàng từ chối ký hợp đồng và đồng ý ký hợp đồng có sự chênh lệch không đáng kể.
- Cụm 4: Phần lớn khách hàng có số dư trung bình hằng năm từ 20000 đến dưới 30000 euro. Một bộ phận nhỏ khách hàng có số dư trên 40000 euro và trên 70000 euro. Tất cả khách hàng đều đồng ý ký hợp đồng gửi tiết kiệm có kỳ hạn.

Từ 4 cụm dữ liệu trên, ta có thể kết luận nhóm khách hàng có số dư trung bình hằng năm từ 8000 đến dưới 20000 euro chiếm tỷ lệ cao nhất trong số các khách hàng đồng ý ký hợp đồng gửi tiết kiệm. Nhóm khách hàng có số dư trung bình hằng năm từ 20000 đến dưới 30000 euro có tỷ lệ đồng ý ký hợp đồng gửi tiết kiệm cao nhất. Để tăng doanh thu và hiệu quả của chiến dịch, ngân hàng có thể đưa ra chiến lược hợp lý để tập trung vào 2 nhóm khách hàng trên.



Hình 7.5.6. Biểu đồ gom cụm bộ dữ liệu nb_test theo hai thuộc tính "age" và "y"

Hình 7.5.6. cho thấy các cụm dữ liệu có sự phân hóa không đồng đều. Do đó có thể kết luận rằng độ tuổi của khách hàng không ảnh hưởng đến khả năng đồng ý hay từ chối ký hợp đồng gửi tiết kiệm có kỳ hạn.



Hình 7.5.7. Ma trận sai lầm của kỹ thuật phân lớp Gaussian Naive Bayes

Từ ma trận sai lầm trong *Hình 7.5.6*, ta tính toán được các chỉ số sau:

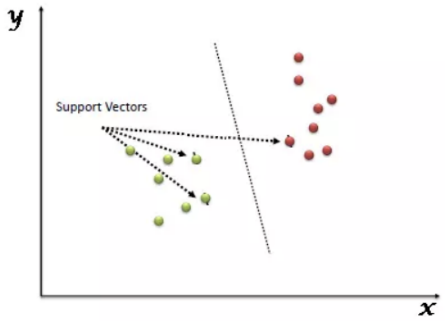
- Độ chính xác (Accuracy) = $\frac{3641 + 205}{3641 + 359 + 316 + 205} = 0.85$
- Tỷ lệ đồng ý đoán đúng (PPV) = $\frac{205}{205 + 316} = 0.4$
- Tỷ lệ từ chối đoán đúng (NPV) = $\frac{3641}{3641 + 359} = 0.91$
- Tỷ lệ đồng ý giả (FNR) = $\frac{316}{316 + 205} = 0.61$
- Tỷ lệ từ chối giả (FPR) = $\frac{359}{3641 + 359} = 0.09$

VIII. THUẬT TOÁN PHÂN LỚP SUPPORT VECTOR MACHINES (SVM)

1. Khái quát thuật toán SVM

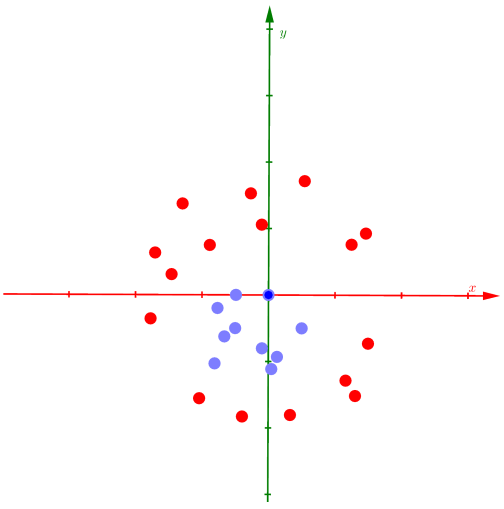
Trong lĩnh vực học máy (*machine learning*), Support Vector Machine (*SVM*) là mô hình tìm biên tối đa có giám sát (*supervised max-margin model*) phân tích dữ liệu nhằm sử dụng cho phân lớp, hồi quy và phát hiện các ngoại lệ [7]. Tuy nhiên SVM được sử dụng phổ biến trong phân lớp.

Support vector có thể được hiểu là các điểm (đối tượng) trên tọa độ. Còn support vector machine là biên giới để phân chia các điểm trên một cách tối ưu nhất [8].

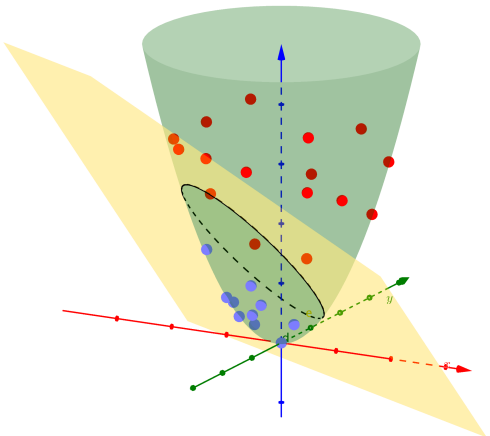


Hình 8.1.1. Hình minh họa hoạt động của SVM qua đồ thị tọa độ [8]

Ý tưởng cơ bản của SVM là chuyển đổi dữ liệu đầu vào thành dữ liệu có chiều không gian cao hơn giúp dễ dàng phân lớp dữ liệu [11]. SVM có thể xử lý cả dữ liệu phân tách tuyến tính (*linearly separable*) và phi tuyến tính (*non-linearly separable*) bằng cách dùng các loại hàm kernel (*kernel function*) khác nhau [10].



Hình 8.1.2. Dữ liệu phi tuyến tính trong không gian hai chiều [11]



Hình 8.1.3. Dữ liệu đã được chuyển đổi thành dữ liệu phân tách tuyến tính trong không gian ba chiều [11]

2. Hàm Kernel

Hàm Kernel là các hàm toán học có chức năng chuyển đổi dữ liệu đầu vào thành không gian có chiều cao hơn, khi đó dữ liệu trở nên dễ phân tách tuyến

tính. Kernel SVM hoạt động bằng việc tìm một hàm số $\Phi(x)$ biến đổi dữ liệu đầu vào x từ không gian ban đầu thành dữ liệu trong không gian mới [11].

Tính chất các hàm Kernel $k()$:

- Đối xứng: $k(x, z) = k(z, x)$
- Về mặt lý thuyết, cần phải thỏa mãn điều kiện Mercer:

$$\sum_{n=1}^{N}\sum_{m=1}^{N} k(x_m x_n)c_n c_m \geq 0, \forall c_i \in R, i = 1, 2, ..., N \quad (1)$$

- Trong thực hành, $k()$ có thể không thỏa mãn điều kiện Mercer nhưng vẫn cho ra kết quả nên vẫn được gọi là hàm Kernel.

Nếu hàm Kernel thỏa mãn điều kiện (1), xét $c_n = y_n \lambda_n$, ta có:

$$\lambda^T K \lambda = \sum_{n=1}^{N}\sum_{m=1}^{N} k(x_m, x_n) y_n y_m \lambda_n \lambda_m \geq 0, \forall \lambda_n \quad (2)$$

Trong đó:

- K là ma trận đối xứng
- $k_{nm} = y_n y_m k(x_n, x_m)$ là phần tử hàng n cột m của K [11]

Tên	Công thức	kernel	Thiết lập hệ số
linear	$\mathbf{x}^T\mathbf{z}$	'linear'	không có hệ số
polynomial	$(r + \gamma\mathbf{x}^T\mathbf{z})^d$	'poly'	d: degree, γ: gamma, r: coef0
sigmoid	$\tanh(\gamma\mathbf{x}^T\mathbf{z} + r)$	'sigmoid'	γ: gamma, r: coef0
rbf	$\exp(-\gamma\|\|\mathbf{x} - \mathbf{z}\|\|_2^2)$	'rbf'	$\gamma > 0$: gamma

Hình 8.2.1. Bảng tóm tắt các hàm Kernel thông dụng và cách sử dụng trong sklearn [11]

3. Ưu điểm

SVM là một kỹ thuật được sử dụng phổ biến với những ưu điểm như sau:

- Hoạt động hiệu quả trên không gian nhiều chiều: với cơ chế hoạt động tăng chiều không gian của dữ liệu đầu vào, SVM thích hợp với các bài toán phân loại văn bản và phân tích quan điểm với số chiều lớn [13].
- Tính linh hoạt: SVM được áp dụng vào cả phân lớp và hồi quy với nhiều ứng dụng trong các lĩnh vực NLP (*Natural Language Processing*), thị giác máy tính (*Computer Vision*), … Thêm vào đó, Kernel trong SVM có thể xử lý linh động trên cả dữ liệu phân tách tuyến tính và phi tuyến tính làm tăng hiệu suất phân lớp [14].

- Khả năng chống nhiễu: SVM thuần (*Hard Margin SVM*) chưa thể hiện tốt khả năng xử lý nhiễu nhưng Soft Margin SVM đã khắc phục rất tốt bằng cách hy sinh điểm nhiễu để được một margin tốt hơn [14].

4. Nhược điểm

Những ưu điểm trên cũng đi kèm theo những hạn chế:

- Chưa thể hiện rõ tính xác suất: SVM chỉ hoạt động theo cơ chế tách dữ liệu bằng siêu phẳng và xác định dựa vào margin từ điểm dữ liệu đến siêu phẳng chứ chưa giải thích được xác suất xuất hiện của từng thể hiện trong tập dữ liệu [13, 14].
- Khó khăn trong lựa chọn Kernel: việc lựa chọn kernel ảnh hưởng rất lớn đến hiệu suất của SVM vì thế việc xác định kernel theo đặc điểm từng bộ dữ liệu là rất khó và quan trọng [14].

5. Áp dụng vào dự đoán khả năng đồng ý của khách hàng

Sử dụng thư viện `sklearn.preprocessing.LabelEncoder` để mã hóa các dữ liệu dạng category về dạng số. Bảng dữ liệu sau khi chuẩn hóa tương tự với *Hình 7.5.1*.

Sau khi chuẩn hóa, các dữ liệu dạng category chuyển thành dạng số nguyên từ 0 đến n với n là số lượng giá trị trong mỗi trường dữ liệu (tương tự phần xử lý dữ liệu ở mục V.2).

Để tăng tính thống nhất cho dữ liệu, ta chuẩn hóa bộ dữ liệu `svm_test` (bản sao từ bộ dữ liệu `df_test`) tương ứng các giá trị đã chuẩn hóa trên.

Sử dụng thư viện `sklearn.svm` để thực hiện phân lớp và dự đoán khả năng khách hàng đồng ý cho vay tại ngân hàng bằng thuật toán Support Vector Machine.

Dùng phương thức SVC() với tham số kernel mặc định là ‘rbf’ và fit() có sẵn trong thư viện để tiến hành phân lớp dữ liệu với các tham số X là bộ dữ liệu đã chuẩn hóa trên. và Y là biến `label` đã tách ra từ bộ dữ liệu `df_train` ban đầu (mục IV.2).

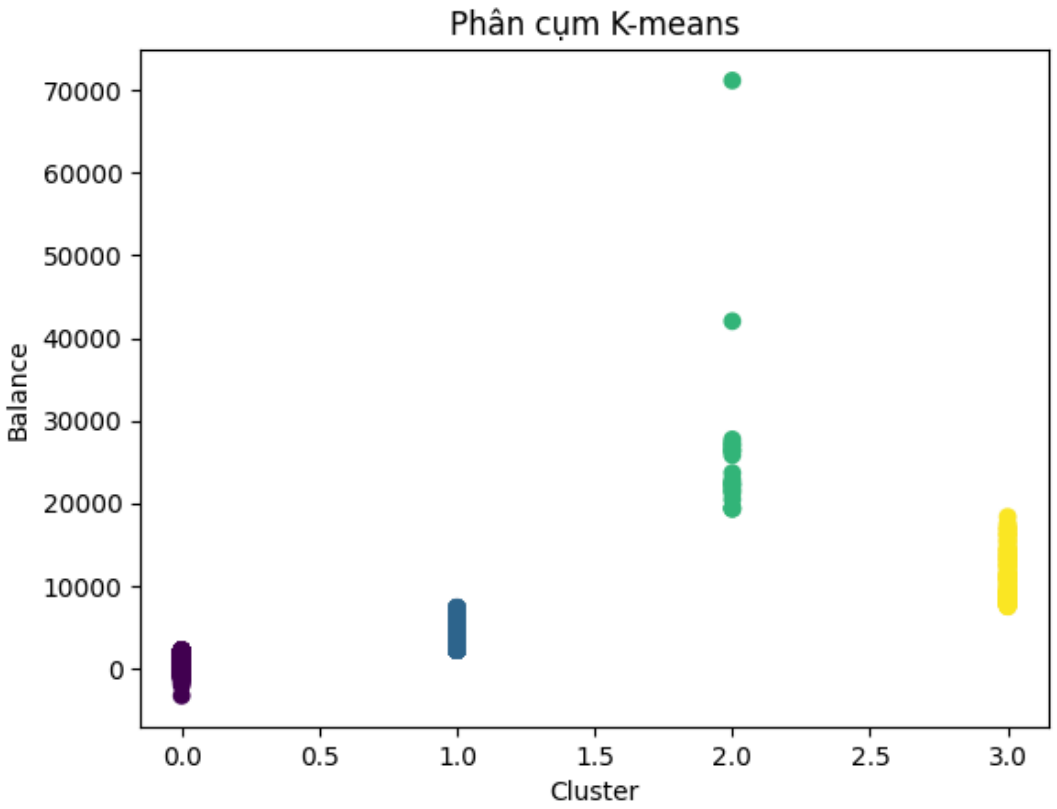
Sử dụng phương thức predict() có sẵn trong thư viện, truyền vào phương thức bộ dữ liệu `svm_test` để dự đoán khả năng đồng ý gửi tiết kiệm tại ngân hàng của từng khách hàng. Kết quả cho ra mảng một chiều chứa giá trị kết quả dự đoán `['yes', 'no']`. Gán kết quả vừa dự đoán được vào biến `svm_result` và thêm thuộc tính `['new_y']` vào cuối bộ dữ liệu

`svm_test` sau khi chuyển đổi dữ liệu đã chuẩn hóa bằng thư viện `sklearn.preprocessing.LabelEncoder` về ban đầu.

	age	job	marital	education	default	balance	housing	loan	day	month	duration	campaign	pdays	previous	poutcome	new_y
0	30	unemployed	married	primary	False	1787	False	False	19	oct	79	1	-1	0	unknown	no
1	33	services	married	secondary	False	4789	True	True	11	may	220	1	339	4	failure	no
2	35	management	single	tertiary	False	1350	True	False	16	apr	185	1	330	1	failure	no
3	30	management	married	tertiary	False	1476	True	True	3	jun	199	4	-1	0	unknown	no
4	59	blue-collar	married	secondary	False	0	True	False	5	may	226	1	-1	0	unknown	no
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4516	33	services	married	secondary	False	-333	True	False	30	jul	329	5	-1	0	unknown	no
4517	57	self-employed	married	tertiary	True	-3313	True	True	9	may	153	1	-1	0	unknown	no
4518	57	technician	married	secondary	False	295	False	False	19	aug	151	11	-1	0	unknown	no
4519	28	blue-collar	married	secondary	False	1137	False	False	6	feb	129	4	211	3	other	no
4520	44	entrepreneur	single	tertiary	False	1136	True	True	3	apr	345	2	249	7	other	no

Hình 8.5.1. Bảng dữ liệu svm_test sau khi phân lớp và dự đoán bằng thuật toán Support Vector Machine

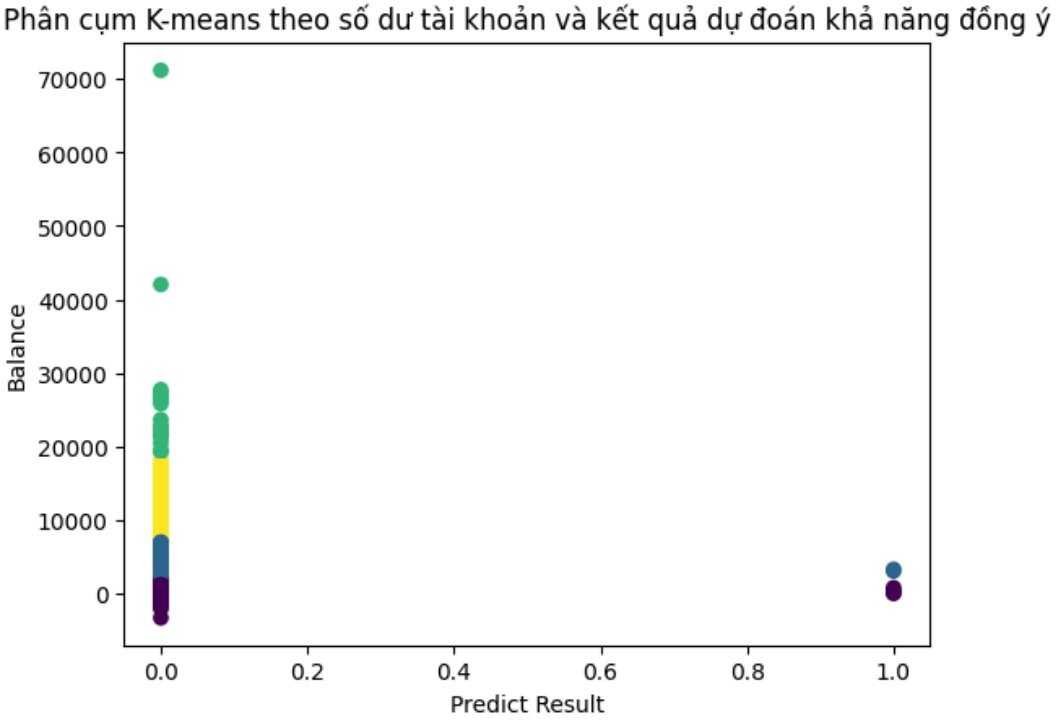
Coi `svm_test` là bộ dữ liệu hoàn chỉnh. Sử dụng thư viện `sklearn.cluster.KMeans` để gom cụm bằng thuật toán K-Means (tương tự như đã làm ở mục V.3). Sau khi gom cụm, ta thấy được một số mối tương quan giữa các thuộc tính như sau:



Hình 8.5.2. Biểu đồ gom cụm bộ dữ liệu svm_test theo thuộc tính "balance"

Từ *Hình 8.5.2.*, ta có thể xác định được các thông tin sau thông qua 4 cụm dữ liệu:

- Cụm 1: Các khách hàng có số dư trung bình hằng năm từ dưới 0 đến 5000 euro.
- Cụm 2: Các khách hàng có số dư trung bình hằng năm từ 5000 đến dưới 10000 euro.
- Cụm 3: Phần lớn khách hàng có số dư trung bình hằng năm từ 18000 đến 28000 euro. Một bộ phận nhỏ có số dư trung bình hằng năm trên 40000 euro và trên 70000 euro.
- Cụm 4: Các khách hàng có số dư trung bình hằng năm từ 8000 đến dưới 20000 euro.



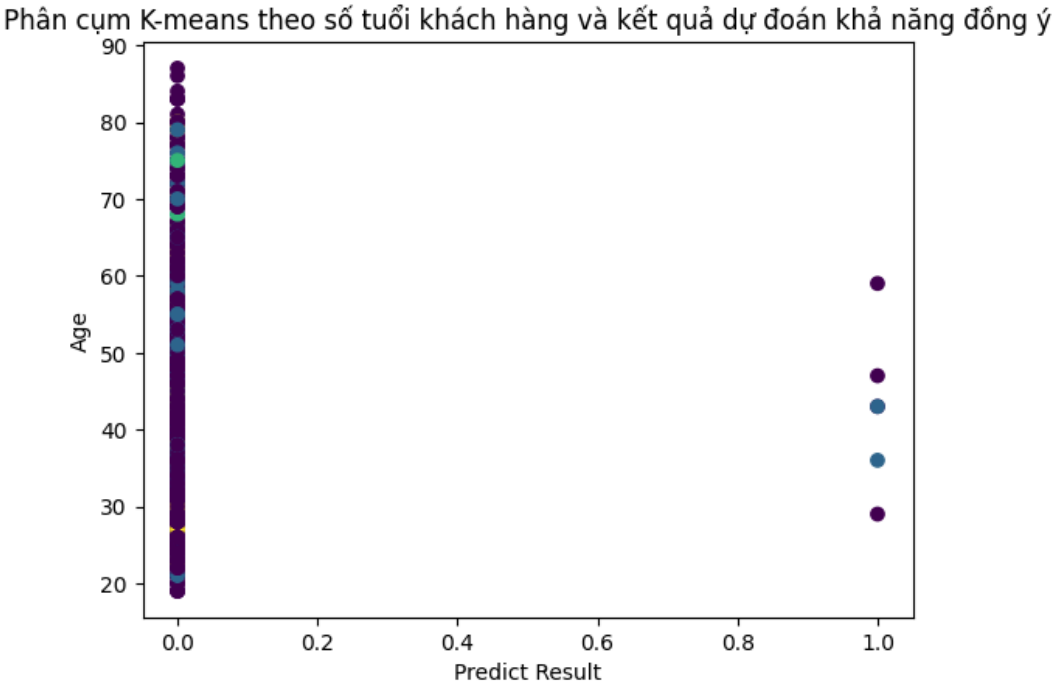
Hình 8.5.3. Biểu đồ gom cụm bộ dữ liệu svm_test theo hai thuộc tính "balance" và "y"

Từ *Hình 8.5.3.*, ta xác định được 4 cụm dữ liệu như sau:

- Cụm 1: Các khách hàng có số dư trung bình hằng năm từ dưới 0 đến 2000 euro. Đa số khách hàng từ chối ký hợp đồng gửi tiết kiệm.
- Cụm 2: Các khách hàng có số dư trung bình hằng năm từ 2000 đến 8000 euro. Đa số khách hàng từ chối ký hợp đồng gửi tiết kiệm.
- Cụm 3: Các khách hàng có số dư trung bình hằng năm từ 8000 đến dưới 20000 euro. Tất cả khách hàng đều không đồng ý ký hợp đồng gửi tiết kiệm có kỳ hạn.

- Cụm 4: Phần lớn khách hàng có số dư trung bình hằng năm từ 20000 đến dưới 30000 euro. Một bộ phận nhỏ khách hàng có số dư trên 40000 euro và trên 70000 euro. Tất cả khách hàng đều không đồng ý ký hợp đồng gửi tiết kiệm có kỳ hạn.

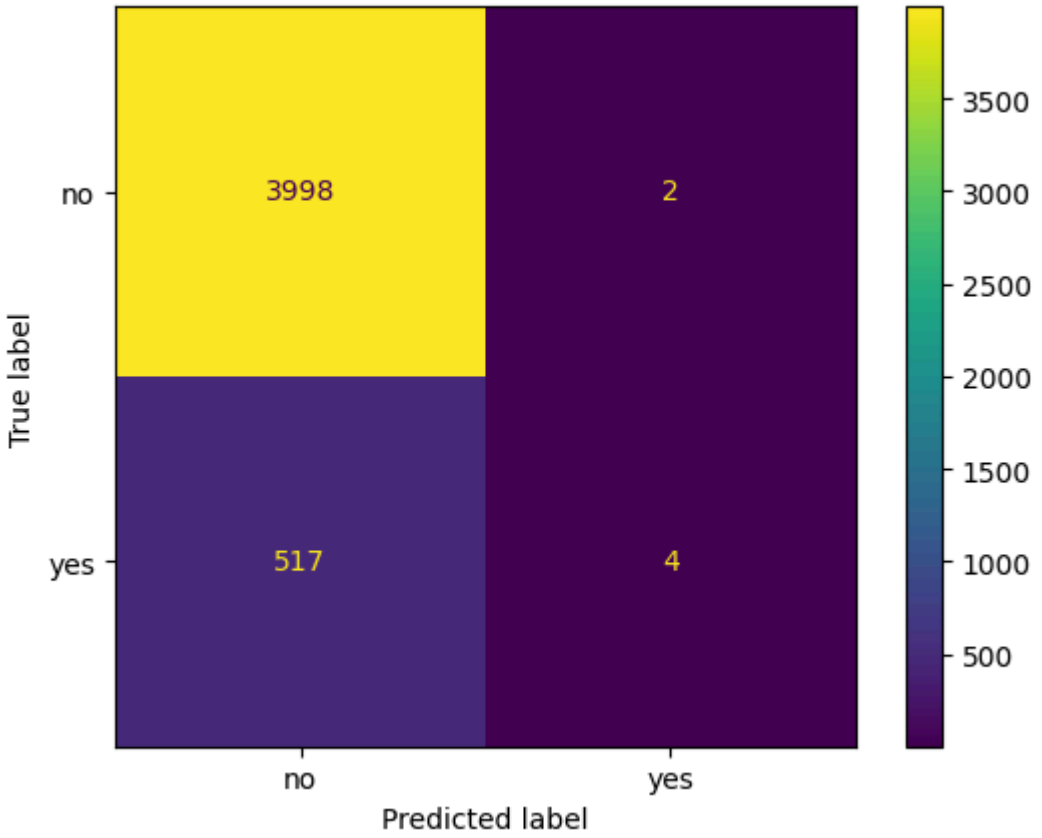
Từ 4 cụm dữ liệu trên, ta có thể thấy sự chênh lệch khá lớn giữa tỷ lệ đồng ý và không đồng ý trong từng cụm. Trong đó, chỉ có cụm 1 và cụm 2 có khách hàng đồng ý gửi tiết kiệm tại ngân hàng, hai cụm còn lại đều từ chối. Sự chênh lệch đáng kể sẽ ảnh hưởng không nhỏ đến tính chính xác của dự đoán. Kết quả tìm được cho thấy việc gom cụm từ kết quả dự đoán của thuật toán Support Vector Machine không có ý nghĩa đối với bộ dữ liệu hiện tại.



Hình 8.5.4. Biểu đồ gom cụm bộ dữ liệu svm_test theo hai thuộc tính "age" và "y"

Hình 8.5.4. cho thấy các cụm dữ liệu có sự phân hóa không đồng đều. Do đó có thể kết luận rằng độ tuổi của khách hàng không ảnh hưởng đến khả năng đồng ý hay từ chối ký hợp đồng gửi tiết kiệm có kỳ hạn.

Sử dụng phương thức from_predictions() có sẵn trong thư viện `sklearn.metrics.ConfusionMatrixDisplay` để trực quan hóa ma trận sai lầm của kỹ thuật phân lớp Support Vector Machine.



Hình 8.5.5. Ma trận sai lầm của kỹ thuật phân lớp Support Vector Machine

Từ ma trận sai lầm trong Hình 8.5.x, ta tính toán được các chỉ số sau:

- Độ chính xác (Accuracy) = $\frac{3998 + 4}{3998 + 4 + 2 + 517} = 0.89$
- Tỷ lệ đồng ý đoán đúng (PPV) = $\frac{4}{4 + 517} = 0.008$
- Tỷ lệ từ chối đoán đúng (NPV) = $\frac{3998}{3998 + 2} = 0.99$
- Tỷ lệ đồng ý giả (FNR) = $\frac{517}{4 + 517} = 0.99$
- Tỷ lệ từ chối giả (FPR) = $\frac{2}{3998 + 2} = 0.0005$

IX. KẾT LUẬN CHUNG

Thuật toán phân lớp Naive Bayes sử dụng phân phối Gaussian Naive Bayes là thuật toán phù hợp để khai phá bộ dữ liệu của đề tài nhằm đưa ra dự đoán về khả năng đồng ý ký hợp đồng gửi tiết kiệm có kỳ hạn của đề tài. Mặc dù thuật toán đơn giản nhưng có Accuracy khá tốt (85%), NPV cao (91%) và FPR rất thấp (< 1%) có thể giúp ngân hàng nhận biết được nhóm khách hàng thường xuyên từ chối ký hợp đồng để không tập trung quá nhiều vào nhóm khách hàng này. Bên cạnh những lợi ích trên thì thuật toán còn hạn chế về chỉ số PPV thấp (40%) và FNR khá cao (61%).

Thuật toán phân lớp Support Vector Machines (SVM) có Accuracy khá cao (89%), NPV gần như tuyệt đối (99%) và FPR gần như không đáng kể (0.05%). Qua đó, ta thấy được thuật toán SVM có hiệu quả tương tự như thuật toán Naive Bayes giúp ngân hàng nhận biết nhóm khách hàng thường xuyên từ chối ký hợp đồng, các chỉ số trên của thuật toán SVM cũng ấn tượng hơn so với thuật toán Naive Bayes. Tuy nhiên, điểm hạn chế của thuật toán SVM là chỉ số PPV rất thấp (0.08%) và FNR gần như tuyệt đối (99%) dẫn đến không nhận biết được nhóm khách hàng tiềm năng để ngân hàng có thể tập trung hướng đến. Với lý do trên, thuật toán SVM không phù hợp để áp dụng vào khai phá bộ dữ liệu của đề tài vì không mang lại được nhiều thông tin có ích cho ngân hàng.

❖ CÁC TÀI LIỆU THAM KHẢO

[1] N.T.V. Hà, "Tổng quan về khai phá dữ liệu và phương pháp khai phá luật kết hợp trong cơ sở dữ liệu", 2020. [Trực tuyến]. Địa chỉ: https://tapchicongthuong.vn/bai-viet/tong-quan-ve-khai-pha-du-lieu-va-phuong-phap-khai-pha-luat-ket-hop-trong-co-so-du-lieu-69634.htm. [Truy cập 19/12/2023].

[2] Wikipedia, "Khai phá dữ liệu", 2023. [Trực tuyến]. Địa chỉ: https://vi.wikipedia.org/wiki/Khai_ph%C3%A1_d%E1%BB%AF_li%E1%BB%87u. [Truy cập 19/12/2023].

[3] Van Bien's blog, "Quy trình Khai phá dữ liệu (Process of Data mining)", 2013. [Trực tuyến]. Địa chỉ: https://bienuit.wordpress.com/2013/09/07/quy-trinh-khai-pha-du-lieu-process-of-data-mining/. [Truy cập 19/12/2023].

[4] chucvn, "Khai phá dữ liệu: Ứng dụng, hướng nghiên cứu và công cụ", 2014. [Trực tuyến]. Địa chỉ: https://bis.net.vn/forums/t/815.aspx. [Truy cập 27/12/2023].

[5] Viện IBS, "Data Mining: Ứng dụng của Data Mining trong các lĩnh vực", 2023. [Trực tuyến]. Địa chỉ: https://insight.isb.edu.vn/ung-dung-cua-data-mining-trong-cac-linh-vuc/. [Truy cập 27/12/2023].

[6] S. Moro, R. Laureano và P. Cortez, "A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems", ??? https://www.kaggle.com/datasets/prakharrathi25/banking-dataset-marketing-targets

[7] Wikipedia, "Support vector machine", 2023. [Trực tuyến]. Địa chỉ: https://en.wikipedia.org/wiki/Support_vector_machine. [Truy cập 29/12/2023].

[8] H.C. Trung, "Giới thiệu về Support Vector Machine (SVM)", 2020. [Trực tuyến]. Địa chỉ: https://viblo.asia/p/gioi-thieu-ve-support-vector-machine-svm-6J3ZgPVElmB. [Truy cập 29/12/2023].

[9] Scikit-learn, “Support Vector Machine”, 2023. [Trực tuyến]. Địa chỉ: https://scikit-learn.org/stable/modules/svm.html. [Truy cập 29/12/2023].

[10] F. Tabsharani, “support vector machine (SVM)”, 2023. [Trực tuyến]. Địa chỉ: https://www.techtarget.com/whatis/definition/support-vector-machine-SVM#:~:text=A%20support%20vector%20machine%20(SVM)%20is%20a%20type%20of%20supervised,data%20set%20into%20two%20groups. [Truy cập 29/12/2023].

[11] V.H. Tiệp, “Bài 21: Kernel Support Vector Machine”, 2017. [Trực tuyến]. Địa chỉ: https://machinelearningcoban.com/2017/04/22/kernelsmv/#-ham-so-kernel. [Truy cập 29/12/2023].

[12] V. Nga, “Tìm hiểu Naive Bayes Classification - Phần 1”, 2022. [Trực tuyến]. Địa chỉ: https://200lab.io/blog/tim-hieu-naive-bayes-classification-phan-1/. [Truy cập 5/1/2024].

[13] P.V. Toàn, “Support Vector Machine trong học máy - Một cái nhìn đơn giản hơn”, 2016. [Trực tuyến]. Địa chỉ: https://viblo.asia/p/support-vector-machine-trong-hoc-may-mot-cai-nhin-don-gian-hon-XQZkxoQmewA. [Truy cập 8/1/2024].

[14] goelaparna1520, “Support vector machine in Machine Learning”, 2023. [Trực tuyến]. Địa chỉ: https://www.geeksforgeeks.org/support-vector-machine-in-machine-learning/. [Truy cập 8/1/2024].

[15] N. Quy, “Thuật toán phân cụm K-Means”, 9/9/2921. [Trực tuyến]. Địa chỉ: https://ndquy.github.io/posts/thuat-toan-phan-cum-kmeans/. [Truy cập 10/1/2024].

[16] N.M. Đức, “Thuật toán Apriori khai phá luật kết hợp trong Data Mining”, 20/8/2019. [Trực tuyến]. Địa chỉ: https://viblo.asia/p/thuat-toan-apriori-khai-pha-luat-ket-hop-trong-data-mining-3P0lPEv85ox. [Truy cập 10/1/2024].