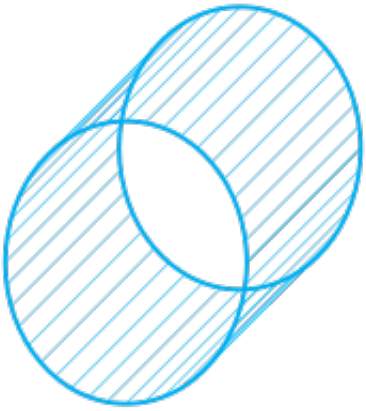
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM
KHOA TOÁN - TIN HỌC



BÁO CÁO ĐỒ ÁN CUỐI KỲ
MÔN HỌC: MÔ HÌNH TỐI ƯU TRONG KINH TẾ

ĐỀ TÀI
DỰ ĐOÁN KHÁCH HÀNG RỜI BỎ DỊCH VỤ VIỄN THÔNG

Giảng viên môn học: Cao Nghi Thục

STT	Họ và tên	MSSV
1	Trần Nguyễn Thanh Phong	22110155
2	Hồ Minh Quân	22110170
3	Trương Minh Quân	22110172

Thành phố Hồ Chí Minh, ngày 29 tháng 5 năm 2025

Mục lục

1 Phân công **3**

2 Lời mở đầu **4**
2.1 Ý tưởng . . . 4
2.2 Phương pháp tiếp cận . . . 4

3 Giới thiệu về hồi quy Logistic **5**
3.1 Sơ lược về hồi quy logistic . . . 5
3.2 Công thức . . . 7

4 Hướng dẫn truy cập công cụ để chạy file code **7**
4.1 Sử dụng Google Colab . . . 7
4.2 Cách thực hiện . . . 7

5 Tải tập dữ liệu **9**
5.1 Bối cảnh . . . 9
5.2 Nội dung . . . 10
5.3 Đọc dữ liệu . . . 10
5.4 Các đặc trưng (Features) . . . 10
5.5 Biến mục tiêu (Target Variable) . . . 11

6 Phân tích dữ liệu **11**
6.1 Kiểu dữ liệu các biến . . . 12
6.2 Kiểm tra dữ liệu thiếu . . . 12
6.3 Thống kê mô tả các biến số . . . 13
6.4 Dòng trùng lặp . . . 13
6.5 Phân tích giá trị duy nhất trong từng cột . . . 14

7 Trực quan hóa dữ liệu **15**

8 Xử lý dữ liệu **19**
8.1 Xử lý dữ liệu khuyết . . . 19
8.2 Xoá đặc trưng không cần thiết . . . 21
8.3 Mã hóa biến phân loại . . . 21
8.4 Chuẩn hóa đặc trưng (Feature Scaling) . . . 22

9 Kiểm tra đa cộng tuyến (Multicollinearity) **23**
9.1 Mục tiêu . . . 23
9.2 Phương pháp kiểm tra . . . 23
9.3 Kết quả . . . 23
9.4 Nhận xét . . . 24

10 Phân chia dữ liệu **24**
10.1 Mục tiêu . . . 24

10.2 Phương pháp 24
10.3 Kết quả 24

11 Cân bằng dữ liệu với SMOTE **25**
11.1 Giới thiệu 25
11.2 SMOTE là gì? 25
11.3 Phân tích dữ liệu trước khi áp dụng SMOTE 25
11.4 Phân tích sau khi áp dụng SMOTE 26
11.5 Lợi ích và rủi ro khi dùng SMOTE 27
11.6 Kết luận 27

12 Huấn luyện mô hình **28**
12.1 Đánh giá mô hình 28
12.2 Tối ưu siêu tham số 33

13 So sánh tổng quan các mô hình **37**

14 Tóm tắt mô hình và kết quả đạt được **39**

Tóm tắt mô hình và kết quả đạt được **39**

15 Hướng phát triển bài toán (mở rộng) **40**

Hướng phát triển **40**

16 Tài liệu tham khảo **41**

Tài liệu tham khảo **41**

17 Lời kết **41**

Lời kết **41**

1 Phân công

BẢNG PHÂN CÔNG ĐỒ ÁN CUỐI KỲ

STT	MSSV	HỌ VÀ TÊN	PHÂN CÔNG
1	22110155	Trần Nguyễn Thanh Phong	Tìm hiểu bài toán, phân tích các biến đặc trưng, khám phá dữ liệu
2	22110170	Hồ Minh Quân	Tiền xử lý dữ liệu, kiểm tra đa cộng tuyến, phân chia dữ liệu
3	22110172	Trương Minh Quân	Cân bằng dữ liệu, huấn luyện mô hình, kết luận bài toán

Tất cả thành viên đều tích cực hoàn thành công việc được phân công và đúng thời hạn

2 Lời mở đầu

Trong thời đại bùng nổ dữ liệu như hiện nay, việc khai thác và phân tích dữ liệu để hỗ trợ ra quyết định đã trở thành một yêu cầu tất yếu trong hầu hết các lĩnh vực như tài chính, y tế, marketing, giáo dục, và đặc biệt là kinh doanh dịch vụ. Trong số các phương pháp phân tích dữ liệu định lượng, mô hình hồi quy logistic nổi bật như một công cụ hiệu quả để giải quyết các bài toán phân loại nhị phân – tức là các bài toán mà đầu ra cần dự đoán chỉ nằm trong hai trạng thái (ví dụ: có/không, đúng/sai, rời bỏ/không rời bỏ, v.v.).

Hồi quy logistic không chỉ cho phép mô hình hóa mối quan hệ giữa biến phụ thuộc nhị phân và một hoặc nhiều biến độc lập, mà còn mang lại khả năng diễn giải trực quan thông qua xác suất và các chỉ số thống kê có ý nghĩa. Với ưu điểm dễ triển khai, diễn giải đơn giản và hiệu quả trong nhiều tình huống thực tế, hồi quy logistic là nền tảng của nhiều hệ thống dự đoán hiện đại, đồng thời là bước khởi đầu quan trọng cho việc tiếp cận các mô hình phức tạp hơn.

Thông qua dự án này, nhóm không chỉ củng cố kiến thức lý thuyết về hồi quy logistic mà còn rèn luyện kỹ năng xử lý dữ liệu, xây dựng mô hình và đánh giá hiệu suất mô hình thông qua các chỉ số như độ chính xác, độ nhạy. Đây cũng là cơ hội để chúng em tiếp cận gần hơn với các bài toán thực tế trong lĩnh vực phân tích dữ liệu các mô hình kinh tế sau này.

2.1 Ý tưởng

Ngành viễn thông là một lĩnh vực có mức độ cạnh tranh rất cao. Với sự hiện diện của nhiều đối thủ trên thị trường, việc quản lý mối quan hệ với khách hàng trở nên đặc biệt quan trọng đối với các nhà cung cấp dịch vụ. Các doanh nghiệp hiện nay triển khai nhiều chiến lược nhằm gia tăng doanh thu như thu hút khách hàng mới, bán chéo dịch vụ cho khách hàng hiện tại, và quan trọng nhất là giữ chân khách hàng cũ.

Hiện tượng khách hàng rời bỏ dịch vụ là tình huống khi một khách hàng ngừng sử dụng dịch vụ của một nhà cung cấp cụ thể. Về bản chất, việc khách hàng rời bỏ dịch vụ có thể được chia thành hai dạng: thứ nhất là khách hàng chủ động chuyển sang nhà cung cấp khác; thứ hai là họ ngừng sử dụng dịch vụ hoàn toàn. Khi số lượng lớn khách hàng rời đi trong thời gian ngắn, điều này có thể ảnh hưởng nghiêm trọng đến danh tiếng của doanh nghiệp, đặc biệt trong thời đại mà truyền miệng và mạng xã hội có ảnh hưởng rất lớn.

Trong khuôn khổ môn học "Mô hình tối ưu trong kinh tế", nhóm chúng em đã lựa chọn đề tài "Dự đoán khách hàng rời bỏ dịch vụ viễn thông" nhằm ứng dụng mô hình hồi quy logistic để giải quyết một bài toán có ý nghĩa thực tiễn cao. Bài toán này không chỉ giúp doanh nghiệp viễn thông nhận diện sớm nguy cơ mất khách hàng mà còn là cơ sở để đưa ra các chiến lược giữ chân phù hợp, từ đó nâng cao hiệu quả kinh doanh.

2.2 Phương pháp tiếp cận

Có hai cách tiếp cận để giữ chân khách hàng hiện tại. Thứ nhất là nâng cao chất lượng dịch vụ, xây dựng lòng trung thành thông qua các chiến dịch ưu đãi, chăm sóc khách hàng,... Tuy nhiên, cách này thường tốn kém và khó triển khai trên diện rộng. Do đó, một phương án

hiệu quả hơn là dự đoán sớm nhóm khách hàng có khả năng rời bỏ, từ đó áp dụng các biện pháp phù hợp để giữ họ lại.

Mục tiêu của các mô hình dự đoán khách hàng rời bỏ dịch vụ là nhận diện sớm những dấu hiệu cho thấy khách hàng có thể rời đi, bằng cách phân tích dữ liệu hành vi sẵn có. Có hai cách xử lý tình huống này:

1. Cách thứ nhất là phản ứng bị động, tức doanh nghiệp chỉ phản hồi khi khách hàng đã đề nghị hủy dịch vụ hoặc chuyển sang nhà cung cấp khác, sau đó cố gắng giữ chân họ bằng các khuyến mãi. Tuy nhiên, phương pháp này thường không đem lại hiệu quả.
2. Cách thứ hai là chủ động dự đoán, tức phân tích hành vi khách hàng để phát hiện sớm nguy cơ khách hàng rời bỏ dịch vụ và có giải pháp can thiệp kịp thời.

Cụ thể, nhóm dự định thực hiện:

1. Thu thập và xử lý một tập dữ liệu thực tế hoặc giả lập về hành vi khách hàng (số phút gọi, số tin nhắn, gói cước, khiếu nại, v.v.).
2. Phân tích ảnh hưởng của dữ liệu mất cân bằng đến hiệu suất của hồi quy Logistic.
3. Đề xuất một phương pháp cân bằng dữ liệu dựa trên tỷ lệ để xử lý bài toán khách hàng rời bỏ.
4. Triển khai lại thuật toán trên tập dữ liệu đã cân bằng bằng phương pháp đề xuất.
5. Đánh giá hiệu quả mô hình thông qua các chỉ số như Accuracy, Precision, Recall, F1-score.
6. Đề xuất mở rộng hướng cải thiện hiệu suất tốt hơn bằng một số thuật toán khác.

3 Giới thiệu về hồi quy Logistic

3.1 Sơ lược về hồi quy logistic

Hồi quy logistic là một mô hình thống kê được sử dụng rộng rãi để giải quyết các bài toán phân loại nhị phân, tức là khi biến mục tiêu (biến phụ thuộc) chỉ nhận một trong hai giá trị (ví dụ: có/không, đúng/sai, rời bỏ/không rời bỏ, v.v.). Không giống như hồi quy tuyến tính – vốn ước lượng giá trị đầu ra là một số thực liên tục – hồi quy logistic ước lượng xác suất của một sự kiện xảy ra, và từ đó phân loại đầu ra vào một trong hai nhóm.

Một số lợi ích của mô hình hồi quy logistic

1. **Giải quyết tốt bài toán phân loại nhị phân:** Hồi quy logistic được thiết kế để dự đoán xác suất của một sự kiện có/không xảy ra, ví dụ như khách hàng rời bỏ hay không rời bỏ dịch vụ.
2. **Dễ dàng diễn giải kết quả:** Các hệ số trong mô hình cho biết tác động tương đối của từng biến đầu vào lên xác suất xảy ra sự kiện, giúp người phân tích hiểu rõ nguyên nhân và mối liên hệ giữa các yếu tố.

3. **Không yêu cầu phân phối chuẩn:** Khác với một số mô hình thống kê khác, hồi quy logistic không đòi hỏi biến đầu vào tuân theo phân phối chuẩn, do đó có thể áp dụng trong nhiều loại dữ liệu thực tế.
4. **Tính toán nhanh và hiệu quả:** Mô hình có thể huấn luyện nhanh ngay cả trên tập dữ liệu vừa và lớn, thích hợp cho các ứng dụng thời gian thực hoặc cần phản hồi nhanh.
5. **Tính ứng dụng cao và dễ triển khai:** Logistic regression đã được triển khai thành công trong nhiều lĩnh vực như tài chính (dự đoán vỡ nợ), y tế (phân loại bệnh), marketing (dự đoán phản hồi quảng cáo), và như đề tài của bạn – dự đoán khách hàng rời bỏ dịch vụ.
6. **Có thể mở rộng sang các kỹ thuật cao hơn:** Là nền tảng cho nhiều mô hình phức tạp như mạng nơ-ron, hoặc mô hình phân loại đa lớp.

Một vài ứng dụng cụ thể của hồi quy logistic:

- **Y học và chăm sóc sức khỏe:**
  - Chẩn đoán bệnh: Dự đoán khả năng một bệnh nhân mắc bệnh (ví dụ: ung thư, tiểu đường) dựa trên các chỉ số xét nghiệm.
  - Tiên lượng kết quả điều trị: Dự đoán khả năng thành công của một phương pháp điều trị.
- **Tài chính – ngân hàng:**
  - Phân tích rủi ro tín dụng: Dự đoán khả năng vỡ nợ của khách hàng dựa trên lịch sử tín dụng, thu nhập, v.v.
  - Phát hiện gian lận giao dịch: Phân loại một giao dịch là hợp lệ hay gian lận.
- **Viễn thông & công nghệ thông tin:**
  - Dự đoán khách hàng rời bỏ dịch vụ: Phân tích hành vi người dùng để dự đoán khả năng họ hủy bỏ gói cước, đổi nhà mạng.
  - Phân loại email: Xác định email là spam hay không spam.
- **Marketing và thương mại điện tử:**
  - Dự đoán hành vi khách hàng: Ví dụ, dự đoán khả năng khách hàng sẽ mua sản phẩm khi thấy quảng cáo.
  - Cá nhân hóa nội dung: Phân tích người dùng có click vào nội dung quảng cáo hay không.
- **Khoa học xã hội và khảo sát:**
  - Phân tích kết quả khảo sát: Ví dụ, dự đoán người tham gia có ủng hộ một chính sách nào đó hay không dựa vào độ tuổi, thu nhập, trình độ học vấn,...
  - Phân tích hành vi bỏ phiếu: Dự đoán khả năng một người sẽ bầu cho ứng viên A hay B.

3.2 Công thức

Cốt lõi của mô hình là hàm logistic sigmoid:

$$P(Y = 1 \mid X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)}}$$

Trong đó:

- **Y** là biến mục tiêu nhị phân (ví dụ: khách hàng có rời bỏ hay không),
- $\mathbf{X} = (x_1, x_2, \ldots, x_n)$ là tập các biến đầu vào (đặc trưng),
- $\beta_0, \beta_1, \ldots, \beta_n$ là các hệ số cần ước lượng.

Mô hình hồi quy logistic không giả định mối quan hệ tuyến tính giữa biến độc lập và biến phụ thuộc, mà thay vào đó giả định mối quan hệ tuyến tính giữa các biến độc lập và logit (log-odds) của xác suất xảy ra sự kiện:

$$\log\left(\frac{P(Y = 1)}{1 - P(Y = 1)}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

4 Hướng dẫn truy cập công cụ để chạy file code

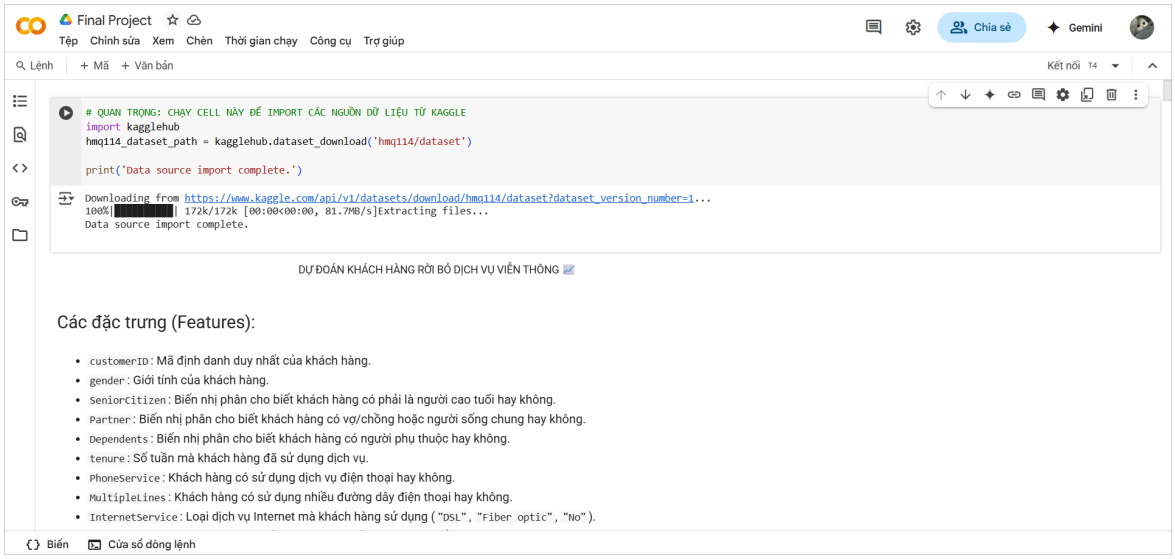
4.1 Sử dụng Google Colab

Cô có thể truy cập trực tiếp vào môi trường chạy code Google Colab thông qua đường link dưới đây:

https://colab.research.google.com/drive/1hRlofflTcxQeg_nimK8Mttj956hkhXl_?usp=sharing

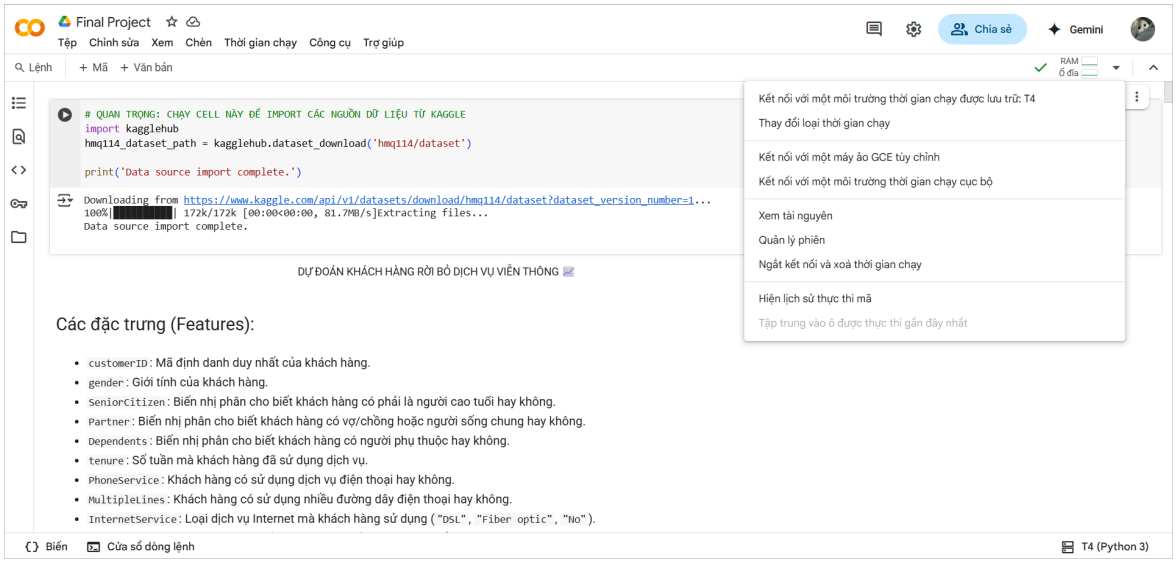
4.2 Cách thực hiện

1. Mở link Google Colab bằng cách nhấp vào đường dẫn đã cho.

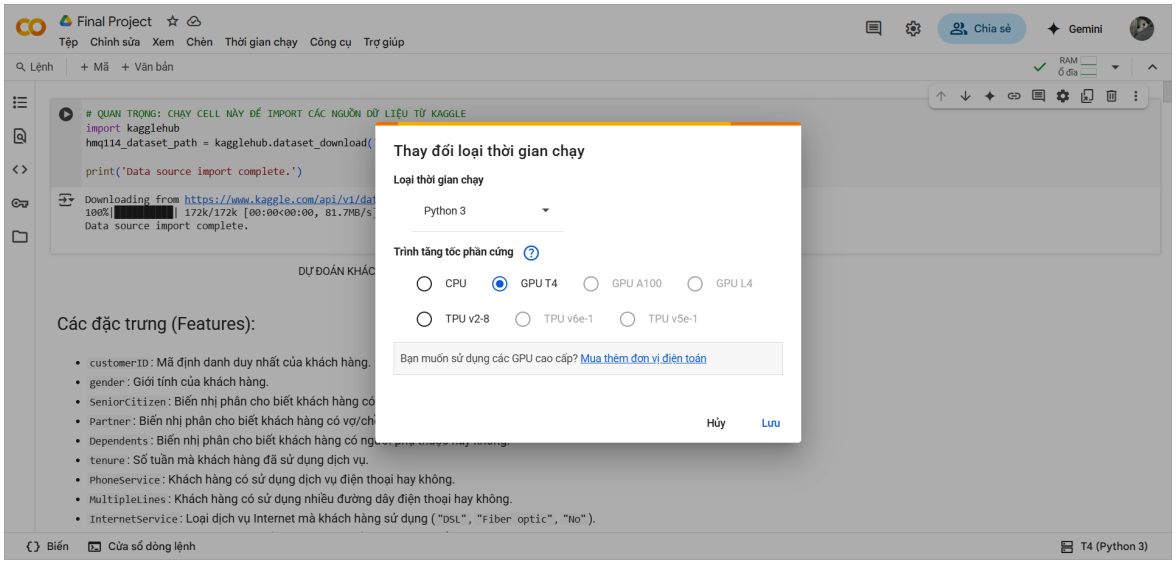


Ảnh minh họa màn hình chính của Google Colab khi mở notebook.

2. Ở góc phải phía trên của màn hình, ngay dưới chữ Gemini có dòng chữ "Kết nối T4" và hình tam giác ngược. Thì cô ấn vào biểu tượng tam giác ngược nó sẽ xuất hiện ra bảng như dưới đây thì cô ấn vào "Thay đổi loại thời gian chạy" -> tiếp đến cô tick vào ô tròn chứa chữ "GPU 4T" -> sau đó ấn "Lưu"

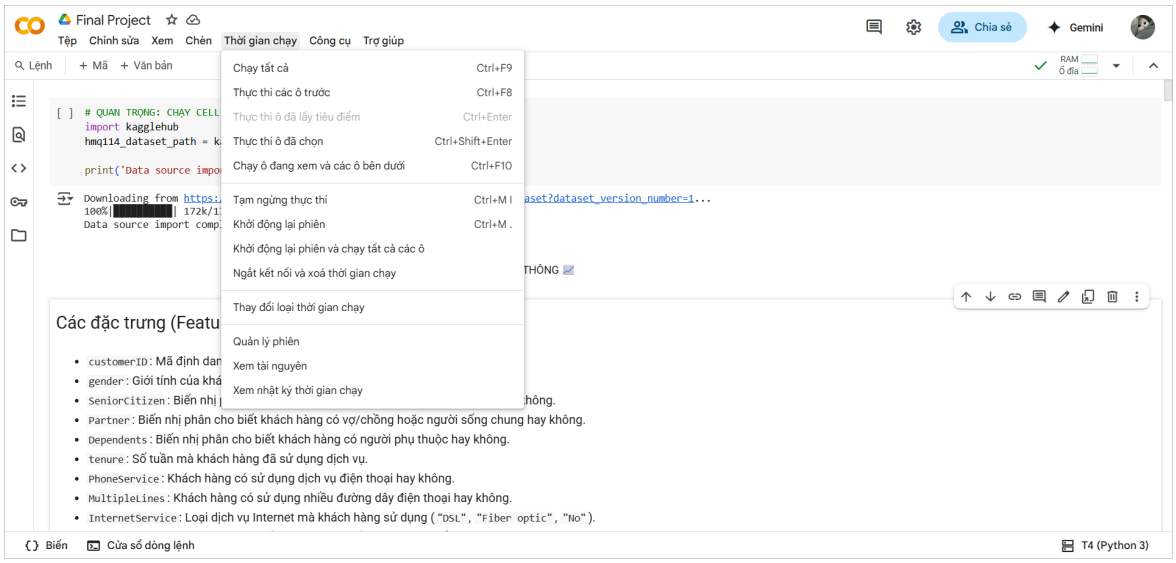


Sau khi ấn vào biểu tượng tam giác ngược



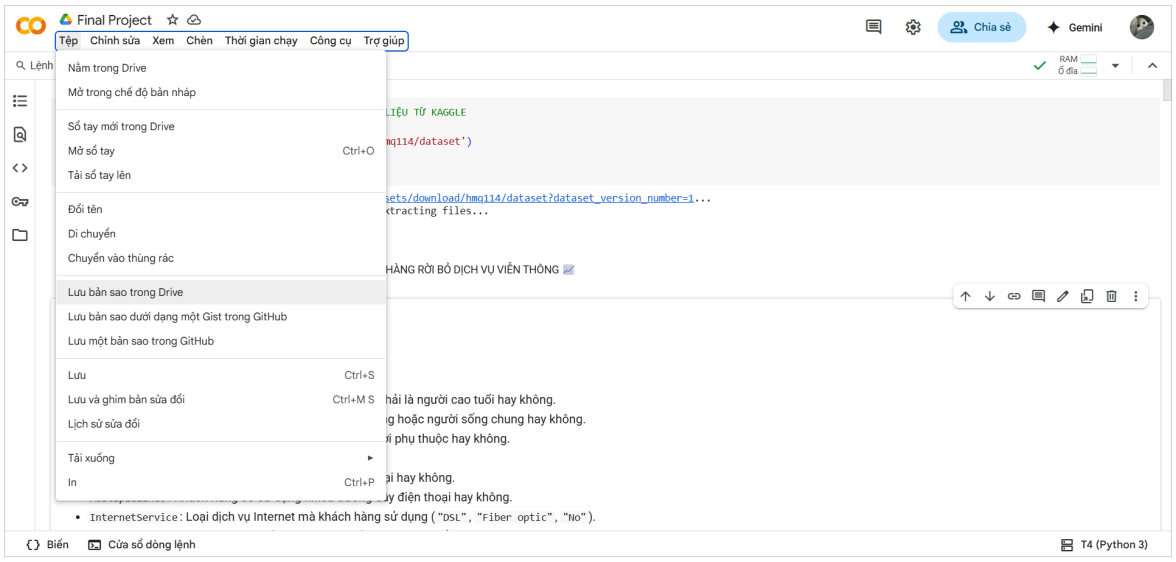
Sau khi ấn vào "Thay đổi thời gian chạy"

3. Để thuận tiện thì cô ấn vào chữ "Thời gian chạy" phía góc trên bên trái -> "Chạy tất cả" rồi chờ kết quả hiện ra.



Ảnh minh họa nút chạy code trên giao diện Colab.

4. Lưu lại bản sao notebook vào Google Drive bằng cách chọn **Tệp** → **Lưu bản sao trong Drive**. (Bước này thực hiện khi cô muốn lưu bản sao, nếu không cô có thể bỏ qua bước này)



Ảnh minh họa thao tác lưu notebook.

5 Tải tập dữ liệu

5.1 Bối cảnh

"Dự đoán hành vi để giữ chân khách hàng. Bạn có thể phân tích tất cả dữ liệu liên quan đến khách hàng và phát triển các chương trình giữ chân khách hàng tập trung." [Bộ dữ liệu mẫu của IBM]

5.2 Nội dung

Mỗi dòng đại diện cho một khách hàng, mỗi cột chứa các thuộc tính của khách hàng được mô tả trong phần Metadata của cột đó.

Bộ dữ liệu bao gồm các thông tin về:

- Khách hàng đã rời đi trong tháng vừa qua – cột này được gọi là `Churn`
- Các dịch vụ mà mỗi khách hàng đã đăng ký – điện thoại, nhiều đường dây, internet, bảo mật trực tuyến, sao lưu trực tuyến, bảo vệ thiết bị, hỗ trợ kỹ thuật, và dịch vụ xem phim, truyền hình trực tuyến
- Thông tin tài khoản khách hàng – thời gian làm khách hàng, loại hợp đồng, phương thức thanh toán, hóa đơn không dùng giấy, phí hàng tháng, và tổng phí
- Thông tin nhân khẩu học về khách hàng – giới tính, độ tuổi, và có hay không có người phối ngẫu và người phụ thuộc

5.3 Đọc dữ liệu

Cô có thể tải tập dữ liệu tại: https://www.kaggle.com/datasets/blastchar/telco-customer-churn/data

```
# QUAN TRỌNG: CHẠY CELL NÀY ĐỂ IMPORT CÁC NGUỒN DỮ LIỆU TỪ KAGGLE
import kagglehub
hmq114_dataset_path = kagglehub.dataset_download('hmq114/dataset')

print('Data source import complete.')
```

```
Downloading from https://www.kaggle.com/api/v1/datasets/download/hmq114/dataset?dataset_version_number=1...
100%|██████████| 172k/172k [00:00<00:00, 81.7MB/s]Extracting files...
Data source import complete.
```

Ở đây nhóm tụi em đã tạo link tải dữ liệu từ kaggle

5.4 Các đặc trưng (Features)

- `customerID`: Mã định danh duy nhất của khách hàng.
- `gender`: Giới tính của khách hàng.
- `SeniorCitizen`: Biến nhị phân cho biết khách hàng có phải là người cao tuổi hay không.
- `Partner`: Biến nhị phân cho biết khách hàng có vợ/chồng hoặc người sống chung hay không.
- `Dependents`: Biến nhị phân cho biết khách hàng có người phụ thuộc hay không.
- `tenure`: Số tuần mà khách hàng đã sử dụng dịch vụ.
- `PhoneService`: Khách hàng có sử dụng dịch vụ điện thoại hay không.
- `MultipleLines`: Khách hàng có sử dụng nhiều đường dây điện thoại hay không.

- `InternetService`: Loại dịch vụ Internet mà khách hàng sử dụng (`"DSL"`, `"Fiber optic"`, `"No"`).
- `OnlineSecurity`: Khách hàng có sử dụng dịch vụ bảo mật trực tuyến hay không.
- `OnlineBackup`: Khách hàng có sử dụng dịch vụ sao lưu trực tuyến hay không.
- `DeviceProtection`: Khách hàng có sử dụng dịch vụ bảo vệ thiết bị hay không.
- `TechSupport`: Khách hàng có sử dụng dịch vụ hỗ trợ kỹ thuật hay không.
- `StreamingTV`: Khách hàng có sử dụng dịch vụ xem TV trực tuyến hay không.
- `StreamingMovies`: Khách hàng có sử dụng dịch vụ xem phim trực tuyến hay không.
- `Contract`: Loại hợp đồng của khách hàng (`'Month-to-month'`, `'One year'`, `'Two year'`).
- `PaperlessBilling`: Khách hàng có sử dụng hóa đơn điện tử hay không.
- `PaymentMethod`: Phương thức thanh toán của khách hàng.
- `MonthlyCharges`: Số tiền phí hàng tháng (đơn vị: \$).
- `TotalCharges`: Tổng số tiền đã chi trả từ trước đến nay (đơn vị: \$).

5.5 Biến mục tiêu (Target Variable)

- `Churn`: Khách hàng `"No"` (ở lại) hoặc `"Yes"` (rời đi).

6 Phân tích dữ liệu

Sau khi tải thành công tập dữ liệu *Telco Customer Churn*, bước đầu tiên trong quá trình phân tích là hiểu rõ cấu trúc, kiểu dữ liệu, tình trạng thiếu dữ liệu, đặc trưng thống kê và phân bố giá trị trong các cột. Việc này giúp hình thành chiến lược tiền xử lý và xây dựng mô hình dự báo sau này.

6.1 Kiểu dữ liệu các biến

Bảng 1: Kiểu dữ liệu của các thuộc tính

Tên biến	**Kiểu dữ liệu**
customerID	object
gender	object
SeniorCitizen	int64
Partner	object
Dependents	object
tenure	int64
PhoneService	object
MultipleLines	object
InternetService	object
OnlineSecurity	object
OnlineBackup	object
DeviceProtection	object
TechSupport	object
StreamingTV	object
StreamingMovies	object
Contract	object
PaperlessBilling	object
PaymentMethod	object
MonthlyCharges	float64
TotalCharges	object
Churn	object

Hầu hết các biến là kiểu `object` (chuỗi), cho thấy dữ liệu dạng phân loại. Các biến số bao gồm `tenure`, `MonthlyCharges`, `SeniorCitizen`. Biến `TotalCharges` tuy là số nhưng đang ở dạng chuỗi, cần xử lý lại.

6.2 Kiểm tra dữ liệu thiếu

Bảng 2: Số lượng giá trị thiếu trong mỗi cột

Tên biến	**Số lượng thiếu**
Tất cả các cột	0

Không có giá trị thiếu trong bất kỳ cột nào. Tuy nhiên, cần kiểm tra thêm các chuỗi rỗng hoặc sai định dạng ở các biến như `TotalCharges`.

6.3 Thống kê mô tả các biến số

Bảng 3: Thống kê mô tả cho các biến định lượng

	SeniorCitizen	**tenure**	**MonthlyCharges**
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000

Các thống kê cho thấy:

- `SeniorCitizen` là biến nhị phân với tỷ lệ người lớn tuổi thấp (16%).
- Khách hàng có thời gian sử dụng trung bình là 32 tháng.
- Chi phí dịch vụ hàng tháng khá đa dạng, trung bình khoảng 65.

6.4 Dòng trùng lặp

Không có dòng trùng lặp trong tập dữ liệu.

6.5 Phân tích giá trị duy nhất trong từng cột

Bảng 4: Giá trị duy nhất trong một số biến

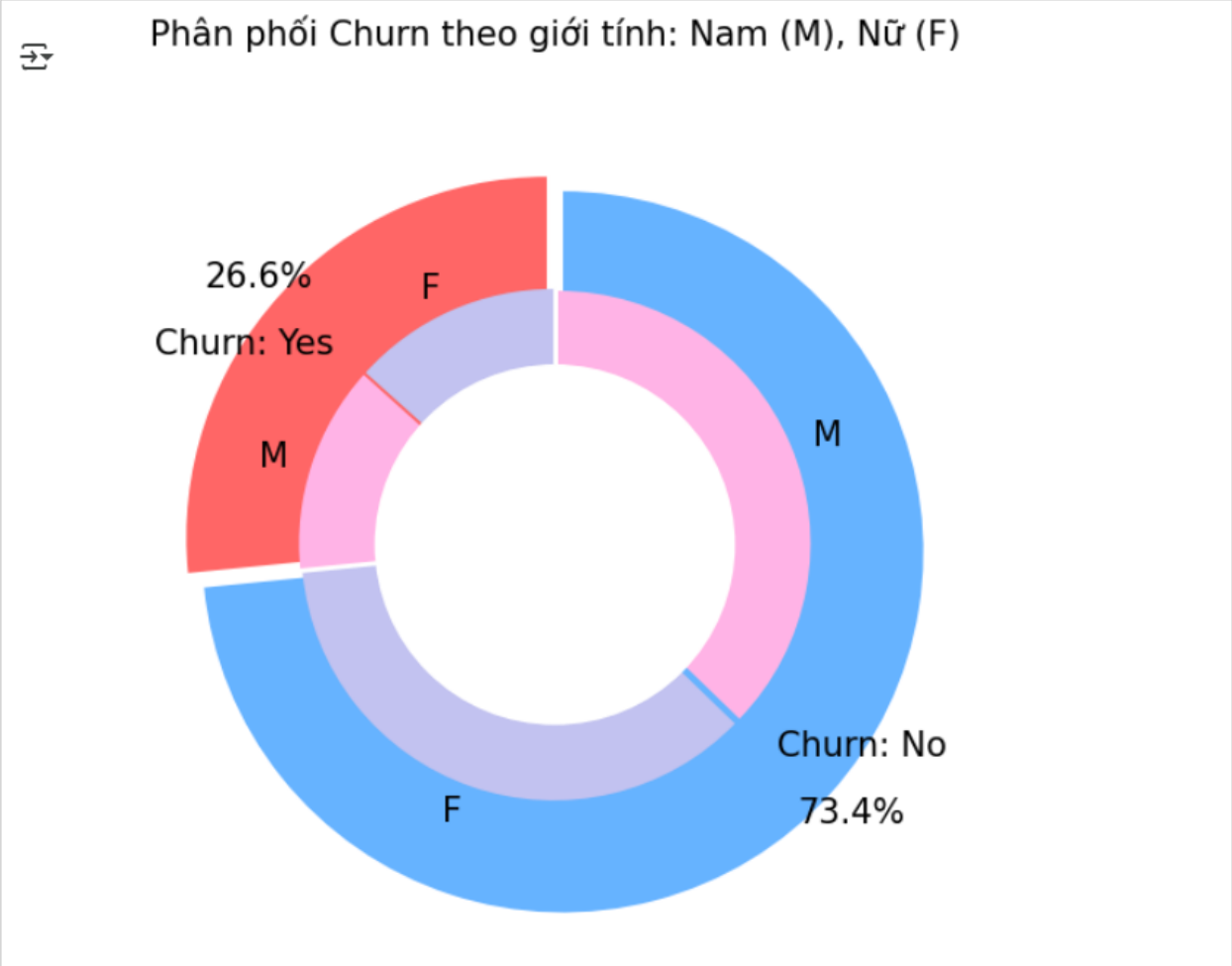
Tên biến	**Giá trị duy nhất**
gender	Female, Male
SeniorCitizen	0, 1
Partner	Yes, No
Dependents	Yes, No
tenure	Giá trị số nguyên từ 0 đến 72
PhoneService	Yes, No
MultipleLines	Yes, No, No phone service
InternetService	DSL, Fiber optic, No
OnlineSecurity	Yes, No, No internet service
OnlineBackup	Yes, No, No internet service
DeviceProtection	Yes, No, No internet service
TechSupport	Yes, No, No internet service
StreamingTV	Yes, No, No internet service
StreamingMovies	Yes, No, No internet service
Contract	Month-to-month, One year, Two year
PaperlessBilling	Yes, No
PaymentMethod	Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic)
MonthlyCharges	Các giá trị liên tục từ khoảng 18.25 đến 118.75
TotalCharges	Giá trị dạng chuỗi số, ví dụ: '29.85', '6844.5',...
Churn	Yes, No

Nhận xét:

- Phần lớn các biến đều là biến phân loại dạng chuỗi (object), đặc biệt là các dịch vụ như `InternetService`, `TechSupport`, `OnlineSecurity`, v.v.
- Các giá trị như "No internet service" hay "No phone service" có thể được gộp thành "No" để đơn giản hóa dữ liệu.
- `TotalCharges` cần được ép kiểu từ chuỗi sang số thực để phục vụ phân tích định lượng.
- `tenure` là biến thời gian khách hàng gắn bó với dịch vụ (số tháng), còn `MonthlyCharges` là chi phí phải trả hàng tháng — cả hai đều là biến số thực quan trọng trong mô hình dự đoán.

7 Trực quan hóa dữ liệu

Biểu đồ phân phối churn theo giới tính



Hình 1: Phân phối khách hàng rời bỏ dịch vụ (churn) theo giới tính: Nam (M), Nữ (F)

Biểu đồ trên thể hiện mối quan hệ giữa giới tính và tình trạng rời bỏ dịch vụ (churn) của khách hàng.

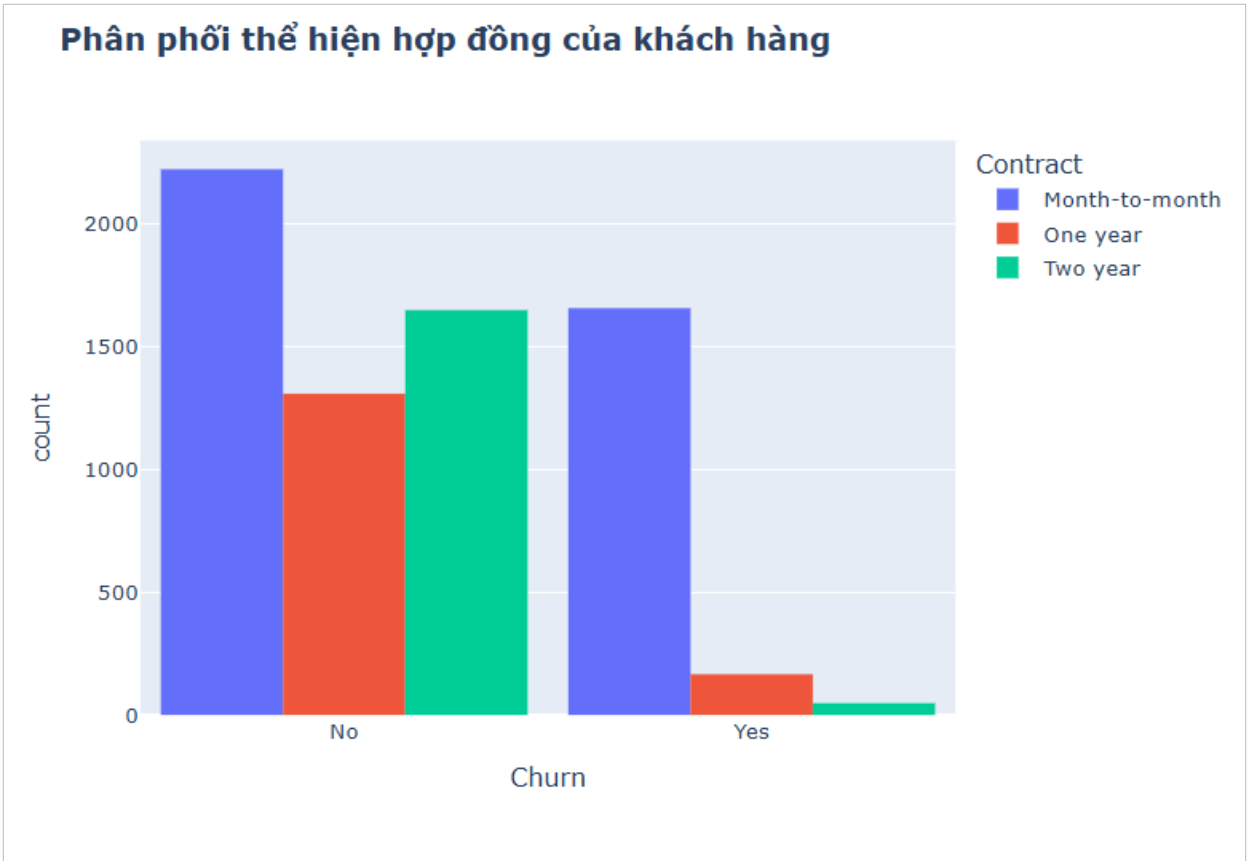
- Tầng ngoài thể hiện tỷ lệ khách hàng rời bỏ (**Churn: Yes**) và không rời bỏ (**Churn: No**).
- Tầng trong thể hiện phân phối giới tính tương ứng trong từng nhóm.

Từ biểu đồ có thể thấy:

- Có khoảng **26.6%** khách hàng đã rời bỏ dịch vụ.
- Trong cả hai nhóm rời bỏ và không rời bỏ, tỷ lệ giới tính giữa nam và nữ không có sự chênh lệch lớn.

- Cần nghiên cứu thêm các yếu tố khác như loại hợp đồng, chi phí hàng tháng để hiểu rõ hơn nguyên nhân rời bỏ.

Phân phối hợp đồng theo tình trạng rời bỏ dịch vụ



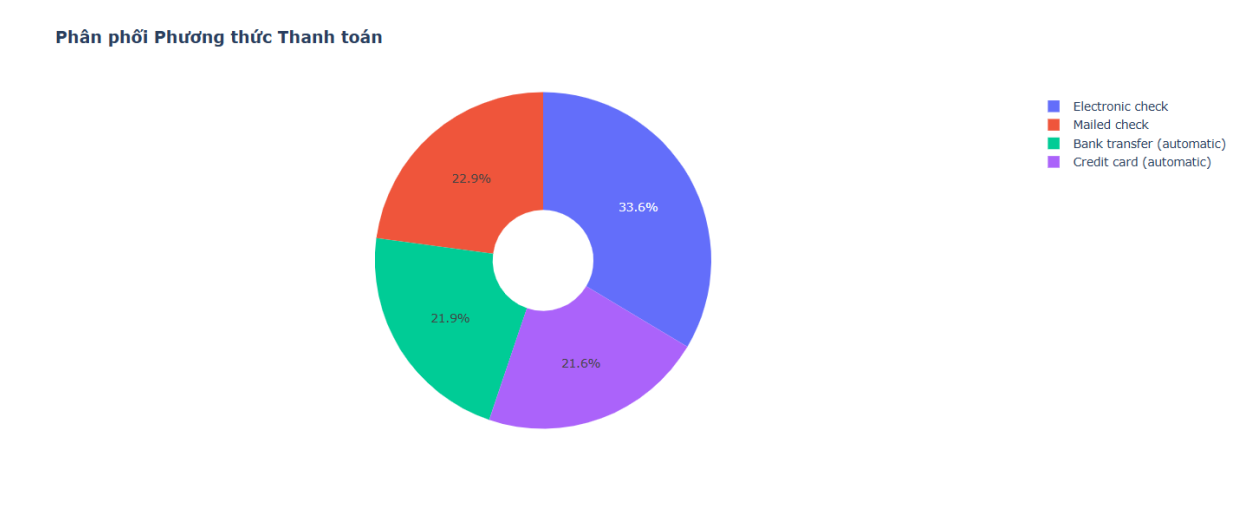
Hình 2: Phân phối loại hợp đồng của khách hàng theo tình trạng rời bỏ dịch vụ (churn)

Biểu đồ trên thể hiện số lượng khách hàng theo từng loại hợp đồng và tình trạng churn (Yes/No).

- Phần lớn khách hàng rời bỏ dịch vụ sử dụng hợp đồng **tháng một** (*Month-to-month*).
- Trong khi đó, khách hàng với hợp đồng *One year* và *Two year* có tỷ lệ churn thấp hơn đáng kể.
- Điều này cho thấy rằng các hợp đồng dài hạn có xu hướng giữ chân khách hàng hiệu quả hơn.

Kết quả này gợi ý rằng doanh nghiệp nên xem xét cung cấp ưu đãi cho các gói hợp đồng dài hạn để giảm tỷ lệ churn.

Phân phối phương thức thanh toán của khách hàng



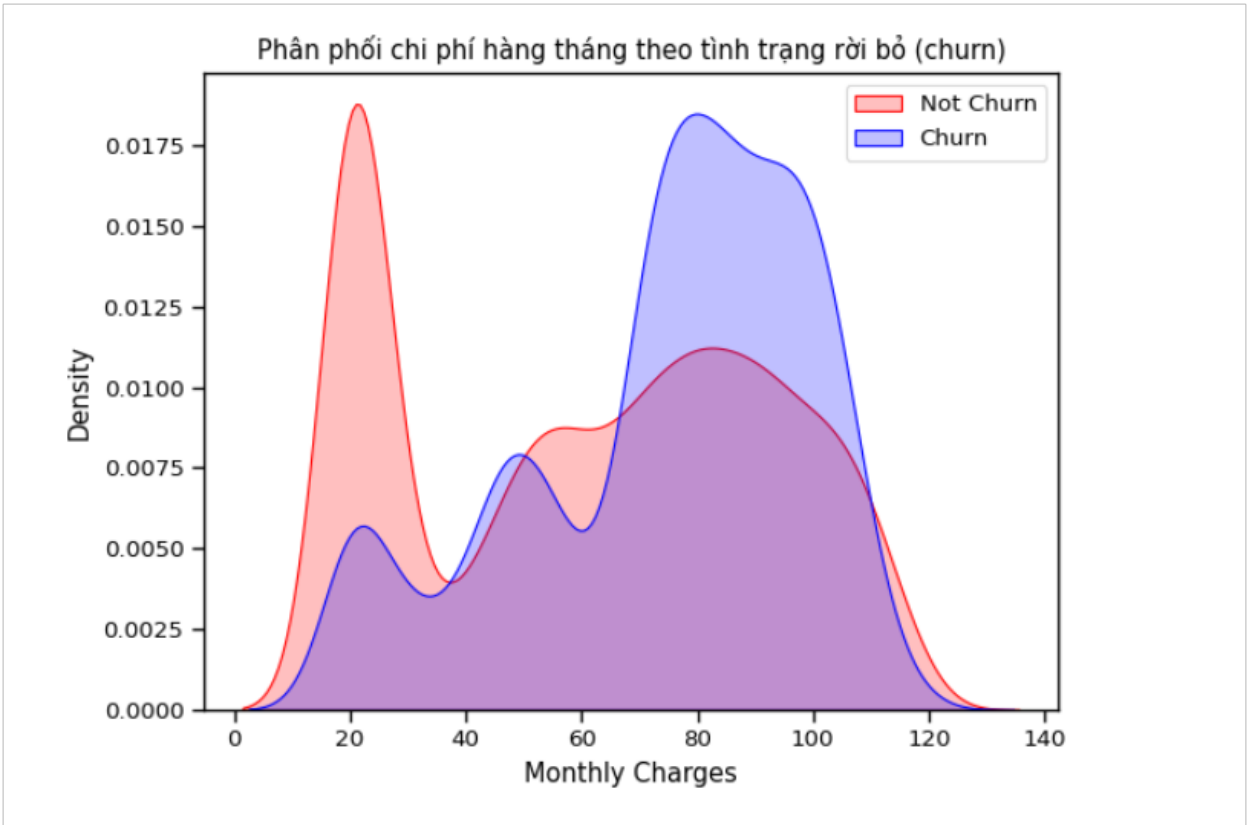
Hình 3: Tỷ lệ các phương thức thanh toán được sử dụng bởi khách hàng

Biểu đồ trên thể hiện phân phối tỷ lệ khách hàng sử dụng từng phương thức thanh toán khác nhau.

- **Electronic check** là phương thức được sử dụng phổ biến nhất với tỷ lệ **33.6%**.
- Các phương thức thanh toán tự động như *Bank transfer (automatic)* và *Credit card (automatic)* có tỷ lệ tương đối đồng đều, chiếm khoảng 21.9% và 21.6%.
- **Mailed check** cũng được sử dụng bởi một bộ phận không nhỏ khách hàng, chiếm 22.9%.

Sự phổ biến của hình thức *electronic check* có thể liên quan đến tỷ lệ churn cao được quan sát trong các phân tích khác. Điều này cho thấy doanh nghiệp nên khuyến khích các phương thức thanh toán tự động để tăng mức độ cam kết và giảm churn.

Phân phối chi phí hàng tháng theo trạng thái rời bỏ (Churn)



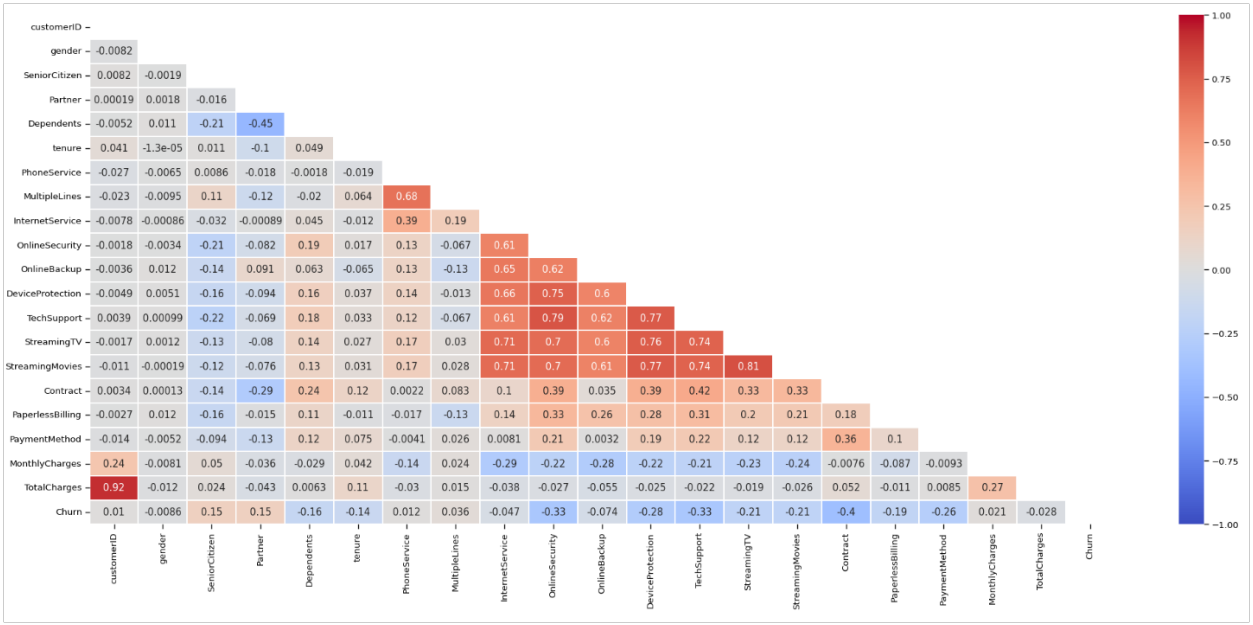
Hình 4: Phân phối chi phí hàng tháng theo trạng thái rời bỏ (churn)

Biểu đồ trên thể hiện phân phối mật độ chi phí hàng tháng (Monthly Charges) của hai nhóm khách hàng: rời bỏ (Churn) và không rời bỏ (Not Churn).

- Khách hàng rời bỏ (**Churn**) có xu hướng phân bố chi phí nhiều ở vùng giá từ 70 đến 100 USD mỗi tháng.
- Trong khi đó, khách hàng **không rời bỏ** có một tỷ lệ đáng kể tập trung ở vùng giá thấp (dưới 30 USD).
- Ngoài ra, vùng trung bình từ 60 đến 90 USD có mật độ tương đối cao ở cả hai nhóm, nhưng vẫn có sự lệch về phía nhóm churn.

Từ biểu đồ có thể thấy rằng, khách hàng với chi phí hàng tháng cao có xu hướng rời bỏ dịch vụ cao hơn. Do đó, việc phân tầng và đưa ra chính sách ưu đãi phù hợp cho nhóm khách hàng có mức chi tiêu cao là cần thiết nhằm giảm churn.

Ma trận tương quan giữa các biến



Hình 5: Ma trận tương quan giữa các biến

Biểu đồ trên minh họa hệ số tương quan giữa các cặp biến trong tập dữ liệu. Một số điểm đáng chú ý:

- **TotalCharges** và **MonthlyCharges** có tương quan dương rất cao với hệ số khoảng 0.66.
- **Contract** có tương quan âm khá mạnh với **Churn** (-0.4), cho thấy khách hàng có hợp đồng dài hạn ít rời bỏ hơn.
- Các dịch vụ như **TechSupport**, **OnlineSecurity**, và **StreamingTV** có tương quan âm với churn, cho thấy khách hàng sử dụng nhiều dịch vụ hơn có xu hướng gắn bó hơn.
- **PaperlessBilling** và **MonthlyCharges** có tương quan dương nhẹ với churn.

Từ biểu đồ, có thể xác định các yếu tố quan trọng ảnh hưởng đến khả năng rời bỏ của khách hàng, hỗ trợ trong việc xây dựng mô hình dự báo churn và đề xuất chiến lược giữ chân khách hàng hiệu quả hơn.

8 Xử lý dữ liệu

8.1 Xử lý dữ liệu khuyết

Tên biến	Số lượng thiếu
customerID	0
gender	0
SeniorCitizen	0
Partner	0
Dependents	0
tenure	0
PhoneService	0
MultipleLines	0
InternetService	0
OnlineSecurity	0
OnlineBackup	0
DeviceProtection	0
TechSupport	0
StreamingTV	0
StreamingMovies	0
Contract	0
PaperlessBilling	0
PaymentMethod	0
MonthlyCharges	0
TotalCharges	**11**
Churn	0

Có thể chọn một trong các hướng xử lý sau cho dữ liệu thiếu ở cột `TotalCharges`:

- Loại bỏ 11 dòng có giá trị thiếu.
- Thay thế giá trị thiếu bằng 0 (nếu hợp lý theo ngữ cảnh).
- Thay thế bằng giá trị trung bình, trung vị hoặc giá trị suy đoán từ các biến liên quan như `tenure` và `MonthlyCharges`.

⇒ Việc phát hiện và xử lý dữ liệu khuyết là một bước quan trọng trong tiền xử lý dữ liệu. Trong trường hợp này, nhờ sử dụng `pd.to_numeric` với tham số thích hợp, nhóm chúng em đã phát hiện 11 giá trị không hợp lệ trong cột `TotalCharges` và có thể tiến hành xử lý tiếp theo phù hợp với mục tiêu phân tích.

Phân tích tỷ lệ dữ liệu khuyết và xử lý

Sau khi xác định có 11 giá trị thiếu trong cột `TotalCharges`, nhóm chúng em tiếp tục đánh giá tỷ lệ phần trăm của dữ liệu thiếu so với toàn bộ tập dữ liệu:

- Số dòng có giá trị thiếu: 11
- Tổng số dòng: 7043

- Phần trăm dữ liệu thiếu: 0.16%

Với tỷ lệ thiếu nhỏ hơn 5%, việc loại bỏ các dòng này được xem là chấp nhận được và không gây ảnh hưởng đáng kể đến chất lượng phân tích. Do đó, nhóm em đã thực hiện loại bỏ các dòng có giá trị thiếu ở biến `TotalCharges`.

Sau khi loại bỏ:

- Đã xoá 11 dòng có dữ liệu thiếu.
- Số dòng còn lại: 7032

Việc xử lý này đảm bảo tập dữ liệu không còn thiếu giá trị nào và sẵn sàng cho các bước phân tích tiếp theo như khám phá dữ liệu, trực quan hoá hoặc xây dựng mô hình.

8.2 Xoá đặc trưng không cần thiết

Trong quá trình tiền xử lý dữ liệu, việc loại bỏ những đặc trưng không đóng góp thông tin hữu ích cho mô hình học máy là điều cần thiết để giảm nhiễu và tăng hiệu quả huấn luyện. Trong trường hợp này, cột `customerID` chỉ đóng vai trò là mã định danh cho từng khách hàng mà không mang ý nghĩa phân tích hay dự đoán.

⇒ Do đó, nhóm đã loại bỏ cột `customerID` khỏi tập dữ liệu.

Sau khi loại bỏ, danh sách các đặc trưng còn lại là:

```
['gender', 'SeniorCitizen', 'Partner', 'Dependents', 'tenure',
 'PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity',
 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
 'StreamingMovies', 'Contract', 'PaperlessBilling', 'PaymentMethod',
 'MonthlyCharges', 'TotalCharges', 'Churn']
```

8.3 Mã hóa biến phân loại

Xác định loại biến

Tập dữ liệu được chia thành ba nhóm biến chính:

- **Biến số (numerical columns):** `tenure`, `MonthlyCharges`, `TotalCharges`.
- **Biến nhị phân (binary columns):** Bao gồm các cột có hai giá trị phân loại như: `gender`, `Partner`, `Dependents`, `PhoneService`, `PaperlessBilling`.
- **Biến phân loại đa giá trị (multi-category columns):** Bao gồm các cột như: `MultipleLines`, `InternetService`, `OnlineSecurity`, `OnlineBackup`, `DeviceProtection`, `TechSupport`, `StreamingTV`, `StreamingMovies`, `Contract`, `PaymentMethod`.

Mã hóa biến nhị phân

Đối với các biến nhị phân, ta sử dụng phương pháp **Label Encoding** để chuyển đổi các giá trị chuỗi thành các số nguyên (0 hoặc 1).

Mã hóa biến phân loại đa giá trị

Với các biến phân loại có nhiều hơn hai giá trị, ta sử dụng kỹ thuật **One-hot Encoding** để tạo ra các cột đặc trưng nhị phân đại diện cho từng giá trị trong biến ban đầu.

Mã hóa biến mục tiêu (target variable)

Biến mục tiêu `Churn` cũng được mã hóa bằng phương pháp **Label Encoding**, giúp chuẩn hóa nhãn đầu ra thành định dạng số.

Kết quả

Sau khi thực hiện mã hóa, hình dạng của tập dữ liệu đã thay đổi như sau:

- **Số lượng dòng:** 7032
- **Số lượng cột:** 41

Việc xử lý và mã hóa đúng định dạng sẽ giúp mô hình học máy dễ dàng tiếp cận và khai thác được thông tin từ các biến phân loại.

8.4 Chuẩn hóa đặc trưng (Feature Scaling)

Mục tiêu

Chuẩn hóa các biến số (numerical features) giúp mô hình học máy hội tụ nhanh hơn và cải thiện hiệu quả dự đoán, đặc biệt là với các thuật toán nhạy cảm với thang đo như Hồi quy Logistic, SVM hoặc mạng nơ-ron.

Phương pháp sử dụng

Biến số được chuẩn hóa bằng phương pháp **StandardScaler**, tức là đưa các giá trị về dạng phân phối chuẩn với:

$$x' = \frac{x - \mu}{\sigma}$$

Trong đó:

- x là giá trị ban đầu
- μ là giá trị trung bình
- σ là độ lệch chuẩn

Lưu mô hình và thông tin

Sau khi chuẩn hóa, các đối tượng quan trọng được lưu bằng thư viện `pickle` để sử dụng trong các bước dự đoán sau này:

- `label_encoders.pkl`: Lưu các encoder cho biến nhị phân
- `target_encoder.pkl`: Lưu encoder cho biến mục tiêu `Churn`
- `feature_scaler.pkl`: Lưu đối tượng `StandardScaler` đã được huấn luyện
- `column_info.pkl`: Lưu thông tin về các cột như danh sách các biến số, biến nhị phân, biến phân loại nhiều giá trị và tất cả đặc trưng đầu vào

Lưu dữ liệu

Bộ dữ liệu sau khi xử lý và chuẩn hóa được lưu vào file `processed_churn_data.csv`, sẵn sàng cho bước huấn luyện mô hình.

9 Kiểm tra đa cộng tuyến (Multicollinearity)

9.1 Mục tiêu

Đa cộng tuyến là hiện tượng xảy ra khi các biến đầu vào (đặc biệt là biến số) có mối tương quan tuyến tính mạnh với nhau, điều này có thể gây ảnh hưởng đến độ ổn định và khả năng giải thích của mô hình học máy, đặc biệt là các mô hình tuyến tính.

9.2 Phương pháp kiểm tra

Sử dụng chỉ số **Variance Inflation Factor (VIF)** để đo lường mức độ đa cộng tuyến giữa các biến. Giá trị VIF được tính theo công thức:

$$\text{VIF}_i = \frac{1}{1 - R_i^2}$$

Trong đó R_i^2 là hệ số xác định từ hồi quy biến i theo các biến còn lại.

Ngưỡng đánh giá:

- VIF ≤ 5: không có vấn đề đa cộng tuyến nghiêm trọng
- VIF > 10: có thể có đa cộng tuyến nghiêm trọng, nên xem xét loại bỏ biến

9.3 Kết quả

Sau khi tính VIF cho các biến số, ta thu được bảng sau:

STT	Biến số	VIF
1	tenure	5.844466
2	MonthlyCharges	3.225293
3	TotalCharges	9.526697

9.4 Nhận xét

Không có biến nào có giá trị VIF vượt quá 10, do đó không cần loại bỏ biến nào để xử lý đa cộng tuyến. Các biến đầu vào được giữ nguyên để tiếp tục huấn luyện mô hình.

10 Phân chia dữ liệu

10.1 Mục tiêu

Sau khi xử lý và mã hóa dữ liệu hoàn chỉnh, bước tiếp theo là phân chia dữ liệu thành tập huấn luyện và tập kiểm tra nhằm phục vụ cho việc huấn luyện và đánh giá mô hình học máy.

10.2 Phương pháp

Sử dụng hàm `train_test_split` từ thư viện `scikit-learn` để chia dữ liệu thành hai phần:

- **Tập huấn luyện (training set)**: chiếm 80% tổng dữ liệu, dùng để huấn luyện mô hình.
- **Tập kiểm tra (test set)**: chiếm 20% tổng dữ liệu, dùng để đánh giá mô hình.

Tham số `stratify=y` được sử dụng để đảm bảo phân bố của biến mục tiêu (`Churn`) là tương đương giữa hai tập dữ liệu, giúp tăng tính đại diện của tập kiểm tra.

10.3 Kết quả

Dữ liệu sau khi phân chia có kích thước như sau:

- **Tổng số mẫu**: 7032
- **Số đặc trưng (features)**: 40
- **Kích thước biến đầu vào (X)**: (7032, 40)
- **Kích thước biến mục tiêu (y)**: (7032,)

Tập huấn luyện và tập kiểm tra được tạo ra để sẵn sàng cho bước huấn luyện mô hình trong giai đoạn tiếp theo.

11 Cân bằng dữ liệu với SMOTE

11.1 Giới thiệu

Một thách thức lớn trong việc dự đoán churn là vấn đề dữ liệu mất cân bằng – tức là sự chênh lệch rõ rệt giữa số lượng khách hàng ở hai nhóm (ở lại và rời đi). Trong trường hợp này, mô hình dễ học sai lệch về lớp chiếm số đông, dẫn đến dự đoán thiếu chính xác. Đây là vấn đề phổ biến trong nhiều bài toán như phát hiện gian lận, chẩn đoán y khoa hay phân loại văn bản. Trong bối cảnh dự đoán churn, hậu quả của việc phân loại sai có thể rất nghiêm trọng: nếu nhầm khách hàng trung thành là khách hàng sắp rời đi, doanh nghiệp sẽ tốn chi phí giữ chân không cần thiết; còn nếu ngược lại, sẽ mất khách hàng mà không có biện pháp nào can thiệp. Vì vậy, việc xử lý và cân bằng lại dữ liệu là bước thiết yếu trước khi áp dụng học máy.

Hai lớp bị mất cân bằng trong bộ dữ liệu là:

- Class 0 (Không rời bỏ): số lượng lớn.
- Class 1 (Rời bỏ): số lượng nhỏ hơn đáng kể.

Việc mất cân bằng này có thể khiến mô hình học máy nghiêng về việc dự đoán class chiếm đa số, làm giảm hiệu quả dự đoán đối với class thiểu số.

Do đó, trong nghiên cứu này, nhóm em đề xuất một phương pháp cân bằng dữ liệu mới – cân bằng dữ liệu dựa trên tỷ lệ, và áp dụng nó vào bài toán dự đoán churn sử dụng tập dữ liệu tải từ Kaggle. Sau bước tiền xử lý, nhóm em sử dụng thuật toán Logistic Regression để xây dựng mô hình dự đoán. Kết quả dự đoán của các thuật toán được so sánh trên hai phiên bản của tập dữ liệu: một phiên bản xử lý dựa trên tập dữ liệu gốc mà không cân bằng dữ liệu và một phiên bản sử dụng phương pháp SMOTE để cân bằng dữ liệu.

11.2 SMOTE là gì?

SMOTE (Synthetic Minority Over-sampling Technique) là một kỹ thuật sinh mẫu dữ liệu tổng hợp cho lớp thiểu số bằng cách:

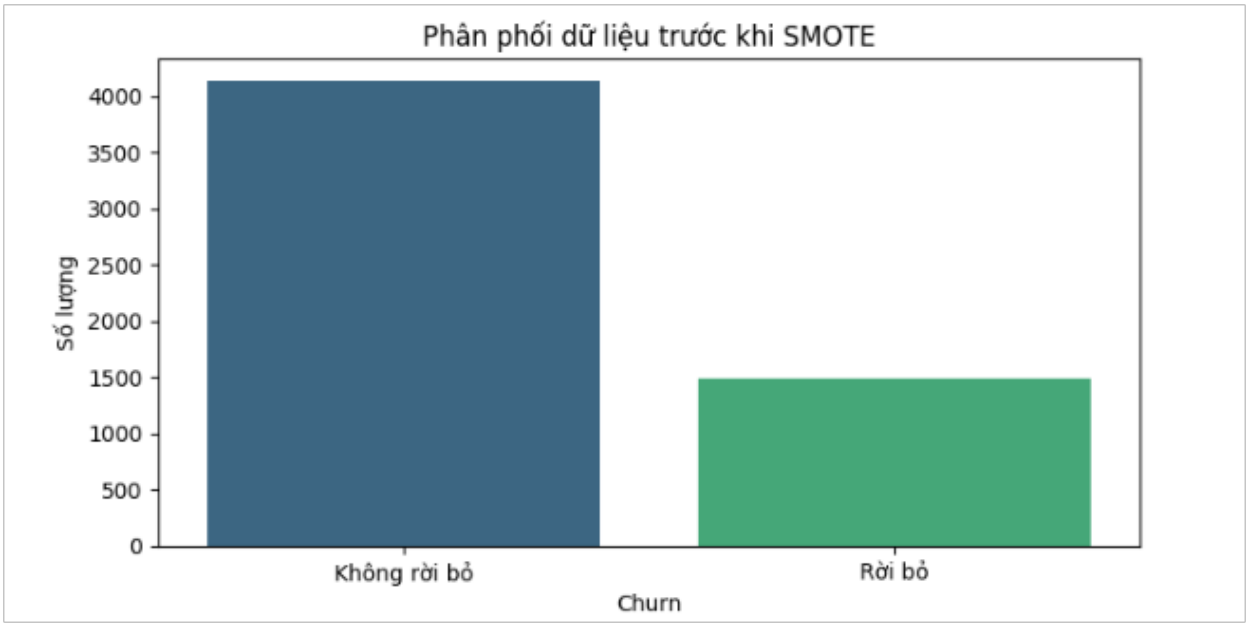
- Chọn ngẫu nhiên một mẫu dữ liệu từ lớp thiểu số.
- Tìm các hàng xóm gần nhất (k-nearest neighbors).
- Tạo các điểm dữ liệu mới nằm giữa điểm hiện tại và hàng xóm gần đó.

Mục tiêu là tạo ra các mẫu giả lập có ý nghĩa, giúp mô hình học được tốt hơn về đặc trưng của lớp thiểu số.

11.3 Phân tích dữ liệu trước khi áp dụng SMOTE

Thống kê phân bố lớp trước SMOTE

Class 0 (Không rời bỏ) : 4130
Class 1 (Rời bỏ) : 1495



Hình 6: Phân phối dữ liệu trước khi áp dụng SMOTE

Nhận xét: Lớp 1 chỉ chiếm khoảng 27% tổng số mẫu, điều này dễ gây ra tình trạng học lệch nếu không xử lý.

Áp dụng SMOTE

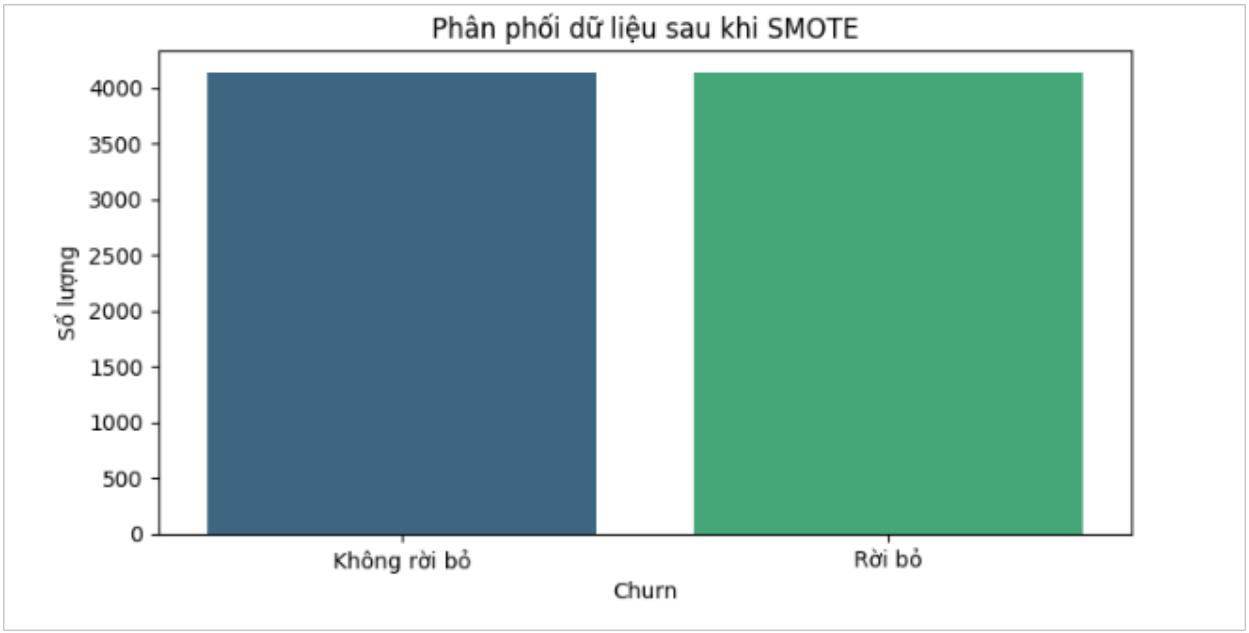
SMOTE được áp dụng trên tập huấn luyện bằng cách:

- Giữ nguyên số lượng mẫu của lớp đa số (Class 0).
- Tăng số lượng mẫu của lớp thiểu số (Class 1) bằng cách sinh thêm các điểm tổng hợp, sao cho cân bằng với Class 0.
- Sử dụng `SMOTE(random_state=42)` để đảm bảo tái lập kết quả.

11.4 Phân tích sau khi áp dụng SMOTE

Thống kê phân bố lớp sau SMOTE

Class 0 (Không rời bỏ) : 4130
Class 1 (Rời bỏ) : 4130
Tổng số mẫu huấn luyện : 8260



Hình 7: Phân phối dữ liệu sau khi áp dụng SMOTE

Nhận xét: Dữ liệu đã được cân bằng hoàn toàn giữa hai lớp, điều này giúp mô hình có cơ hội học tốt hơn cả hai lớp.

11.5 Lợi ích và rủi ro khi dùng SMOTE

Lợi ích

- Cải thiện khả năng học và phân biệt lớp thiểu số.
- Giảm hiện tượng thiên lệch mô hình.
- Không làm mất dữ liệu thật như kỹ thuật undersampling.

Rủi ro tiềm ẩn

- Dễ gây overfitting nếu sinh quá nhiều mẫu không tự nhiên.
- Có thể làm tăng thời gian huấn luyện.
- Không nên áp dụng trên toàn bộ tập dữ liệu (chỉ nên trên tập huấn luyện).

11.6 Kết luận

Việc sử dụng SMOTE giúp cải thiện độ cân bằng giữa các lớp trong dữ liệu huấn luyện, từ đó nâng cao khả năng tổng quát của mô hình đối với lớp thiểu số. Đây là một bước tiền xử lý quan trọng trong các bài toán phân loại với dữ liệu mất cân bằng.

12 Huấn luyện mô hình

Mục tiêu của thí nghiệm này là đánh giá hiệu quả của kỹ thuật tăng cường dữ liệu SMOTE (Synthetic Minority Over-sampling Technique) trong việc huấn luyện mô hình Logistic Regression, sử dụng độ đo ROC-AUC với phương pháp Cross-validation (5-Fold).

Mô hình baseline

Hàm `train_baseline_model` được sử dụng để huấn luyện mô hình Logistic Regression với các tham số sau:

- `random_state = 42` để đảm bảo tính tái lập.
- `max_iter = 1000` để đảm bảo mô hình hội tụ.
- Đánh giá mô hình bằng độ đo ROC-AUC thông qua 5-Fold Cross-validation.

Kết quả thực nghiệm

Mô hình với SMOTE

Dữ liệu huấn luyện được tăng cường bằng kỹ thuật SMOTE trước khi huấn luyện mô hình. Kết quả:

- **5-Fold CV ROC-AUC: 0.9155 ± 0.1024**

Mô hình không sử dụng SMOTE

Dữ liệu huấn luyện không được tăng cường. Kết quả:

- **5-Fold CV ROC-AUC: 0.8460 ± 0.0379**

Nhận xét

Việc sử dụng SMOTE đã giúp cải thiện đáng kể hiệu năng của mô hình Logistic Regression trên độ đo ROC-AUC, tăng từ 0.8460 lên 0.9155. Tuy nhiên, độ lệch chuẩn trong kết quả với SMOTE cũng cao hơn, cho thấy mức độ dao động giữa các fold lớn hơn. Điều này cần được xem xét thêm khi triển khai mô hình thực tế.

12.1 Đánh giá mô hình

Mô hình Baseline: Logistic Regression không dùng SMOTE

1.1 Các chỉ số đánh giá

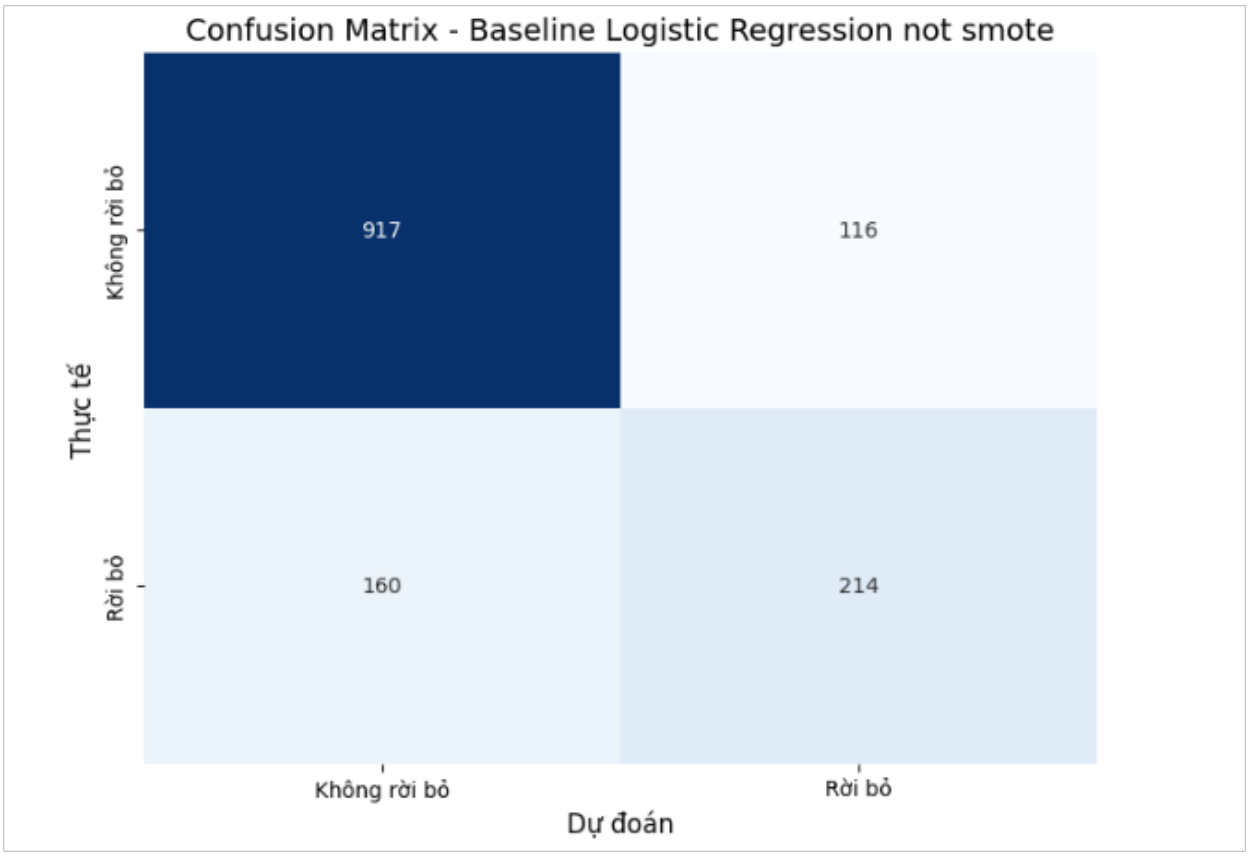
- Accuracy: 0.8038
- Balanced Accuracy: 0.7299
- Precision: 0.6485

- Recall: 0.5722
- F1 Score: 0.6080
- F2 Score (ưu tiên recall): 0.5860
- ROC AUC: 0.8359

1.2 Báo cáo phân loại

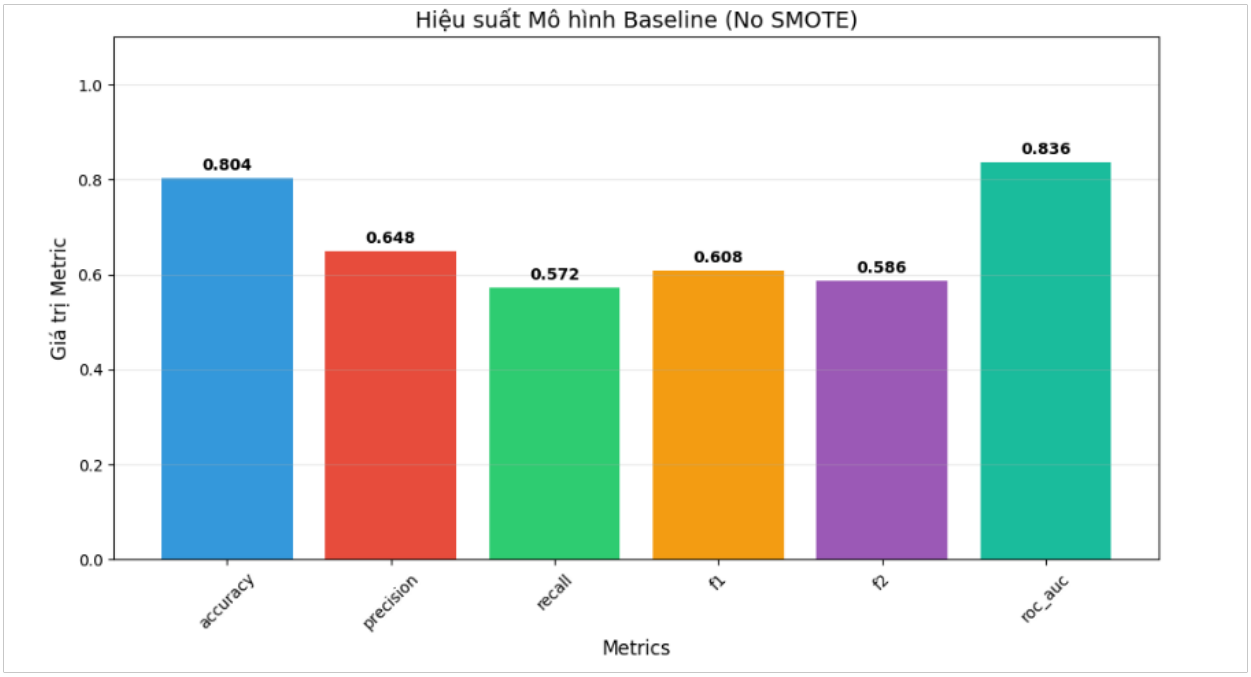
	precision	recall	f1-score	support
Không rời bỏ	0.85	0.89	0.87	1033
Rời bỏ	0.65	0.57	0.61	374
Accuracy			0.80	1407
Macro avg	0.75	0.73	0.74	1407
Weighted avg	0.80	0.80	0.80	1407

1.3 Ma trận nhầm lẫn



Hình 8: Confusion Matrix - Logistic Regression (Không dùng SMOTE)

1.4 Biểu đồ hiệu suất mô hình



Hình 9: Biểu đồ các chỉ số đánh giá - Không dùng SMOTE

1.5 Nhận xét

- Mô hình đạt độ chính xác cao (accuracy 80.38%) nhưng độ **recall** của lớp "Rời bỏ" còn thấp (57.2%), dẫn đến bỏ sót nhiều khách hàng rời bỏ.
- Precision ở mức trung bình (64.85%) – nhiều dự đoán "rời bỏ" là sai.
- AUC khá tốt (0.8359) cho thấy mô hình phân biệt được hai lớp khá ổn.
- Có sự mất cân bằng giữa các lớp: số lượng "Rời bỏ" ít hơn nhiều so với "Không rời bỏ".

Mô hình Logistic Regression có dùng SMOTE

2.1 Các chỉ số đánh giá

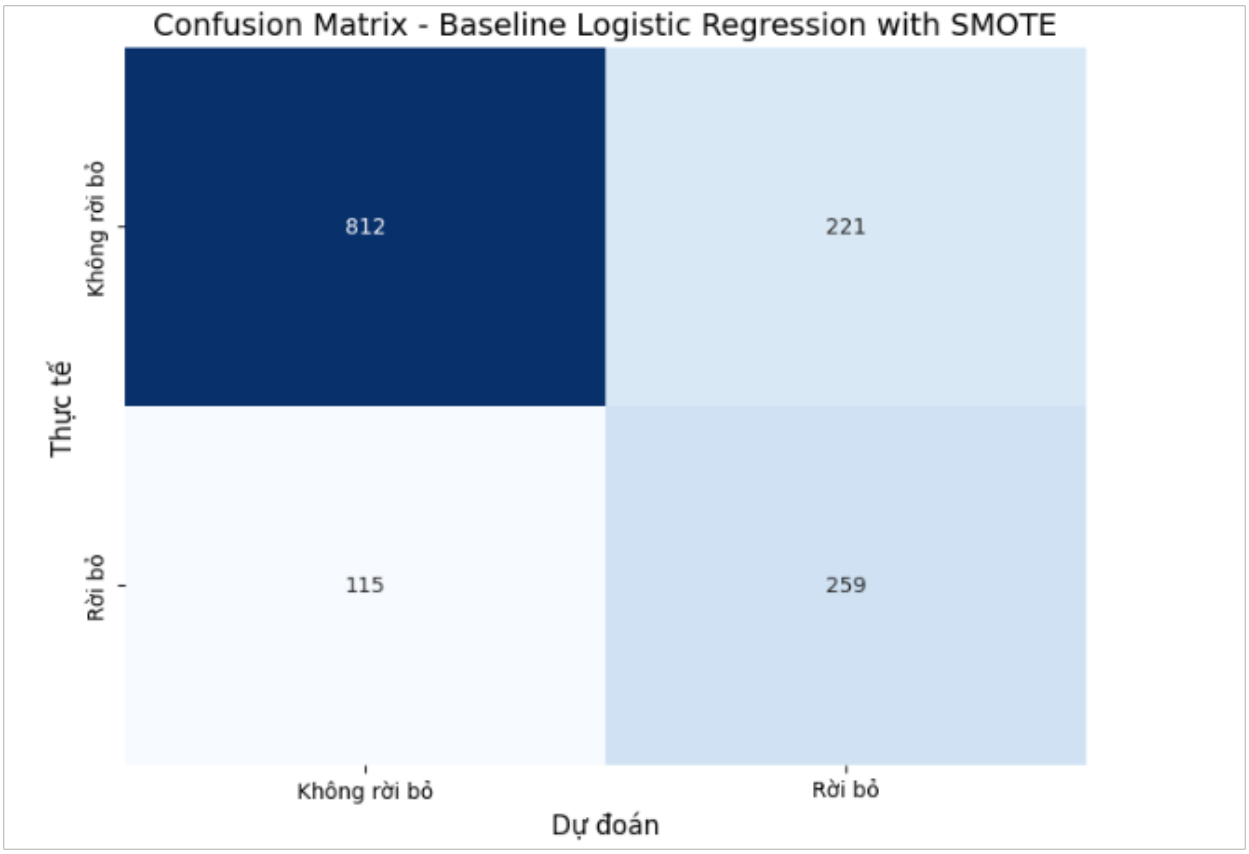
- Accuracy: 0.7612
- Balanced Accuracy: 0.7393
- Precision: 0.5396
- Recall: 0.6925
- F1 Score: 0.6066
- F2 Score (ưu tiên recall): 0.6554

- ROC AUC: 0.8264

2.2 Báo cáo phân loại

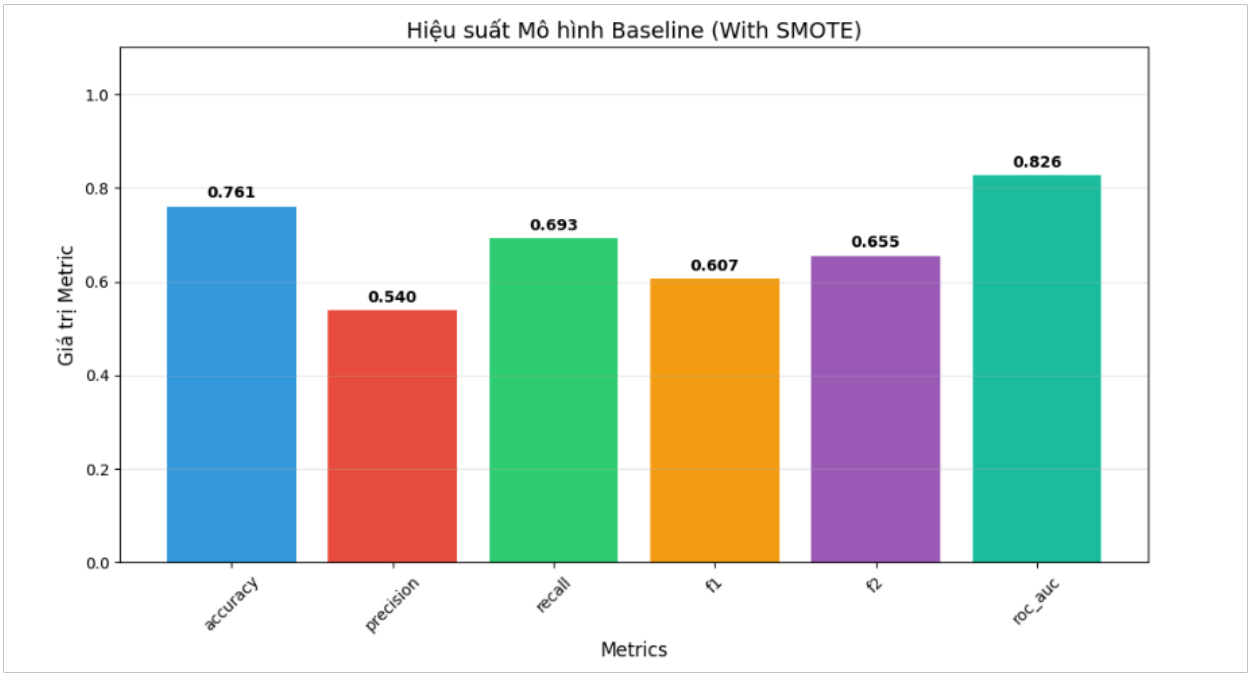
	precision	recall	f1-score	support
Không rời bỏ	0.88	0.79	0.83	1033
Rời bỏ	0.54	0.69	0.61	374
Accuracy			0.76	1407
Macro avg	0.71	0.74	0.72	1407
Weighted avg	0.79	0.76	0.77	1407

2.3 Ma trận nhầm lẫn



Hình 10: Confusion Matrix - Logistic Regression (Có dùng SMOTE)

2.4 Biểu đồ hiệu suất mô hình

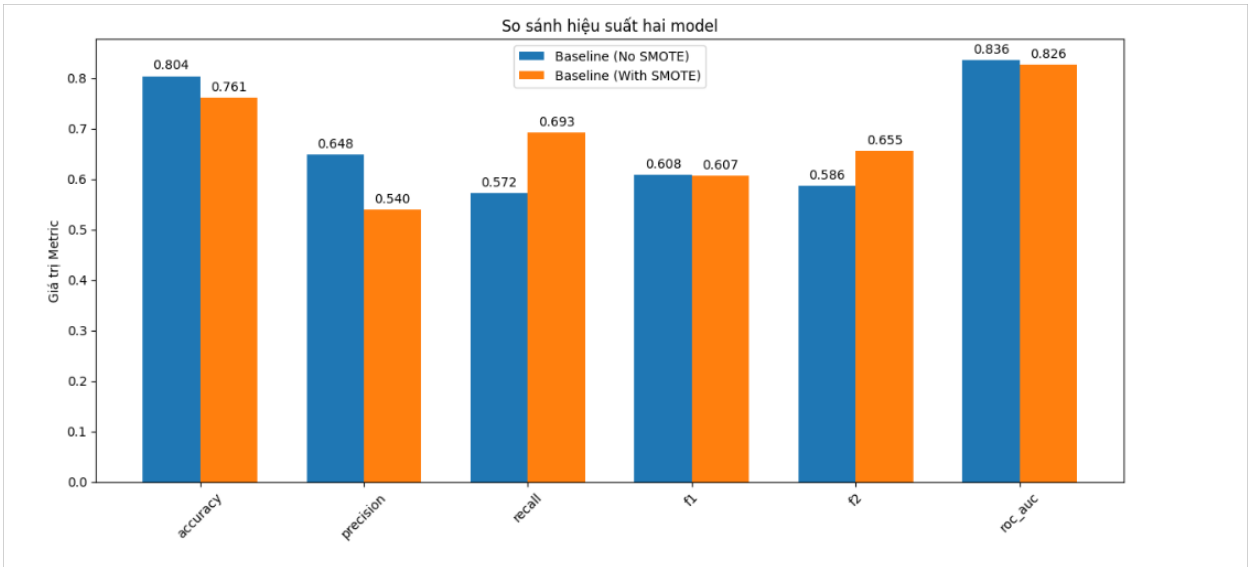


Hình 11: Hiệu suất Mô hình Logistic Regression (Có dùng SMOTE)

2.5 Nhận xét

- SMOTE giúp cải thiện **recall** của lớp "Rời bỏ" từ 57.2% lên 69.2% – tức là giảm số lượng khách hàng rời bỏ bị dự đoán sai.
- Tuy nhiên, precision giảm xuống còn 53.96%, nghĩa là tăng số lượng false positives.
- F2 Score (ưu tiên recall) tăng đáng kể, từ 0.5860 lên 0.6554.
- ROC AUC chỉ giảm nhẹ (0.8359 xuống 0.8264), cho thấy mô hình vẫn phân biệt tốt giữa hai lớp.

So sánh mô hình Logistic Regression (Có và không dùng SMOTE)



Hình 12: So sánh hiệu suất hai mô hình Logistic Regression (Có và Không dùng SMOTE)

- Mô hình **không dùng SMOTE** có **accuracy**, **precision**, và **ROC AUC** cao hơn – cho thấy mô hình này dự đoán tổng thể chính xác hơn và ít false positives hơn.
- Mô hình **có dùng SMOTE** có **recall** và **F2 Score** cao hơn – thể hiện khả năng nhận diện tốt hơn các trường hợp khách hàng rời bỏ, điều này rất quan trọng nếu mục tiêu là giảm churn.
- **F1 Score** giữa hai mô hình gần như tương đương, cho thấy mức độ cân bằng giữa precision và recall không thay đổi nhiều.
- ROC AUC của mô hình có SMOTE chỉ giảm nhẹ (từ 0.836 xuống 0.826), nghĩa là khả năng phân biệt hai lớp vẫn được giữ ở mức tốt.

Kết luận: Việc chọn mô hình phụ thuộc vào mục tiêu kinh doanh. Nếu mục tiêu là **phát hiện tối đa khách hàng rời bỏ** (ưu tiên recall), mô hình dùng SMOTE là lựa chọn hợp lý. Ngược lại, nếu muốn đảm bảo **dự đoán tổng thể chính xác hơn** và **giảm false positives**, mô hình không dùng SMOTE sẽ phù hợp hơn.

12.2 Tối ưu siêu tham số

*** Tối ưu siêu tham số cho Logistic Regression**

1. Tham số tốt nhất

Sau quá trình tìm kiếm bằng GridSearchCV với 5-fold cross-validation trên 180 tổ hợp siêu tham số (tổng cộng 900 lượt huấn luyện), mô hình Logistic Regression có bộ tham số tối ưu như sau:

- `C`: 1
- `class_weight`: None
- `penalty`: l2
- `solver`: newton-cholesky

F2 score tốt nhất trên tập validation (cross-validation): **0.8316**

2. Kết quả đánh giá mô hình sau khi tuning

- Accuracy: 0.7598
- Balanced Accuracy: 0.7383
- Precision: 0.5373
- Recall: 0.6925
- F1 Score: 0.6081
- F2 Score (ưu tiên recall): 0.6547
- ROC AUC: 0.8263

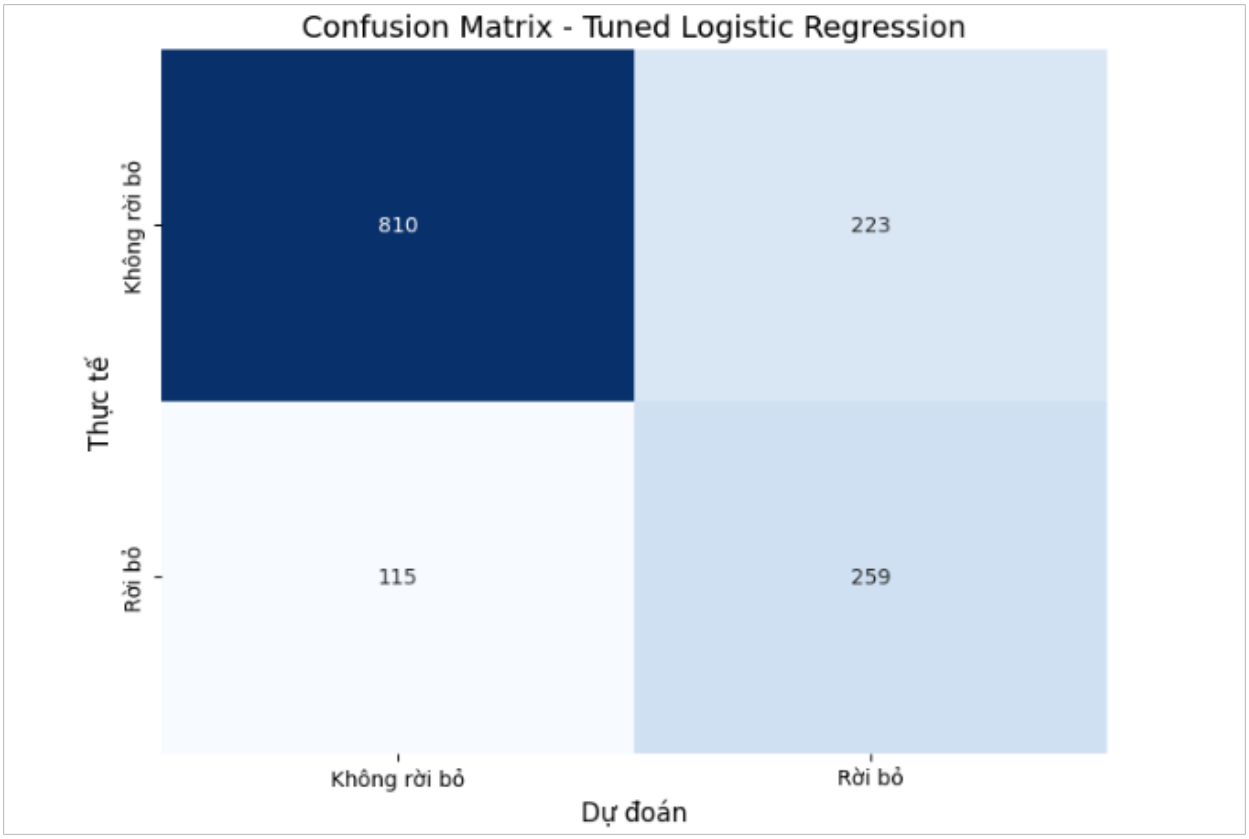
3. Báo cáo phân loại chi tiết

	precision	recall	f1-score	support
Không rời bỏ	0.88	0.78	0.83	1033
Rời bỏ	0.54	0.69	0.61	374
Accuracy			0.76	1407
Macro avg	0.71	0.74	0.72	1407
Weighted avg	0.79	0.76	0.77	1407

Nhận xét:

- Sau khi tối ưu siêu tham số, mô hình không cải thiện nhiều về accuracy (0.7598 so với 0.7612 trước đó).
- Tuy nhiên, các chỉ số như **recall** và **F2 Score** vẫn giữ ở mức cao, phù hợp với mục tiêu ưu tiên phát hiện khách hàng rời bỏ.
- Precision vẫn còn thấp (0.5373), đồng nghĩa với việc mô hình vẫn có nhiều false positives.
- ROC AUC đạt 0.8263 – mô hình vẫn có khả năng phân biệt tốt hai lớp.
- Nhìn chung, quá trình tuning giúp đảm bảo mô hình hoạt động ổn định và giữ được cân bằng giữa các chỉ số, đặc biệt nếu mục tiêu là tối ưu F2 score.

4 Ma trận nhầm lẫn



Hình 13: Confusion Matrix - Tuned Logistic Regression

Diễn giải ma trận:

- **Dự đoán đúng (True Negatives)** – Không rời bỏ và được dự đoán đúng: **810**
- **Dự đoán sai (False Positives)** – Không rời bỏ nhưng bị dự đoán là rời bỏ: **223**
- **Dự đoán sai (False Negatives)** – Rời bỏ nhưng bị dự đoán là không rời bỏ: **115**
- **Dự đoán đúng (True Positives)** – Rời bỏ và được dự đoán đúng: **259**

Nhận xét:

- Mô hình có khả năng nhận diện khách hàng rời bỏ tốt hơn trước, với **259/374** trường hợp được dự đoán đúng (recall lớp rời bỏ đạt 69.2%).
- Tuy nhiên, số lượng false positive vẫn còn tương đối lớn (223 trường hợp), phản ánh qua chỉ số precision chưa cao.
- Tổng thể, mô hình đã giảm thiểu sai sót quan trọng là bỏ sót khách hàng rời bỏ – phù hợp với mục tiêu kinh doanh nếu ưu tiên giữ chân khách hàng.

*** Logistic Regression sau khi tuning – không dùng SMOTE**

- **Tham số tốt nhất:** `C=1, class_weight='balanced', penalty='l2', solver='liblinear'`
- **F2 score tốt nhất (Cross-Validation):** 0.7261

Các chỉ số đánh giá trên tập kiểm tra:

- Accuracy: 0.7249
- Balanced Accuracy: 0.7400
- Precision: 0.4893
- Recall: 0.7941
- F1 Score: 0.6055
- F2 Score (ưu tiên recall): 0.7061
- ROC AUC: 0.8351

Báo cáo phân loại:

	precision	recall	f1-score	support
Không rời bỏ	0.90	0.79	0.84	1033
Rời bỏ	0.49	0.79	0.61	374
Accuracy			0.72	1407
Macro avg	0.70	0.75	0.72	1407
Weighted avg	0.79	0.74	0.76	1407

Nhận xét:

Mô hình Logistic Regression sau khi được tối ưu siêu tham số và không áp dụng kỹ thuật SMOTE cho thấy hiệu quả phân loại tốt hơn so với các mô hình trước đó. Cụ thể:

- **Recall** đạt 79.41%, tăng đáng kể so với mô hình baseline và cả mô hình có áp dụng SMOTE. Điều này cho thấy mô hình có khả năng phát hiện khách hàng rời bỏ tốt hơn, rất quan trọng trong bối cảnh ứng dụng thực tế.
- **F2 Score** là 0.7061, cao nhất trong tất cả các mô hình đã thử nghiệm, chứng minh rằng việc ưu tiên recall đã đạt được hiệu quả cao sau khi tuning với `class_weight='balanced'`.
- Mặc dù **precision** chỉ đạt 48.93%, điều này là chấp nhận được trong trường hợp bài toán chú trọng đến việc hạn chế bỏ sót khách hàng có khả năng rời bỏ (tức là tăng recall).
- **ROC AUC** đạt 0.8351 – cao nhất trong các mô hình – cho thấy mô hình có khả năng phân biệt hai lớp khách hàng rõ rệt.

- **Báo cáo phân loại** cũng cho thấy độ chính xác đối với lớp "Rời bỏ" đã được cải thiện mạnh mẽ (recall 79%, f1-score 61%), trong khi độ chính xác tổng thể vẫn giữ ở mức hợp lý (accuracy 72.49%).

Ma trận nhầm lẫn:

	Dự đoán: Không rời bỏ	**Dự đoán: Rời bỏ**
Thực tế: Không rời bỏ	723	310
Thực tế: Rời bỏ	77	297

Nhận xét:

- Mô hình đã dự đoán đúng 297/374 khách hàng rời bỏ, đạt recall khoảng **79.41%**, cho thấy hiệu quả trong việc phát hiện các trường hợp cần quan tâm.
- Tuy vẫn có 310 khách hàng không rời bỏ bị phân loại nhầm là sẽ rời bỏ (false positive), nhưng điều này là chấp nhận được khi mục tiêu ưu tiên là không bỏ sót khách hàng thật sự sẽ rời bỏ.
- Số lượng dự đoán sai loại "rời bỏ" là 77, chiếm một tỷ lệ thấp so với tổng số khách hàng rời bỏ, cho thấy độ chính xác trong phân loại nhóm nhạy cảm này đã được cải thiện nhờ tuning và sử dụng `class_weight='balanced'`.

Như vậy, việc tuning mô hình Logistic Regression mà không cần áp dụng SMOTE đã mang lại hiệu quả rõ rệt trong việc phát hiện khách hàng rời bỏ, đồng thời vẫn duy trì được độ chính xác cân bằng và độ phân biệt tốt.

13 So sánh tổng quan các mô hình

Bảng tổng hợp kết quả đánh giá các mô hình:

Model	**F2 Score**	**Recall**	**Precision**	**Accuracy**	**ROC AUC**
Tuned_No_SMOTE	**0.7061**	**0.7941**	0.4893	0.7249	0.8351
Baseline_SMOTE	0.6554	0.6925	0.5396	0.7612	0.8264
Tuned_SMOTE	0.6547	0.6925	0.5373	0.7598	0.8263
Baseline_No_SMOTE	0.5860	0.5722	**0.6485**	**0.8038**	**0.8359**

Phân tích chi tiết:

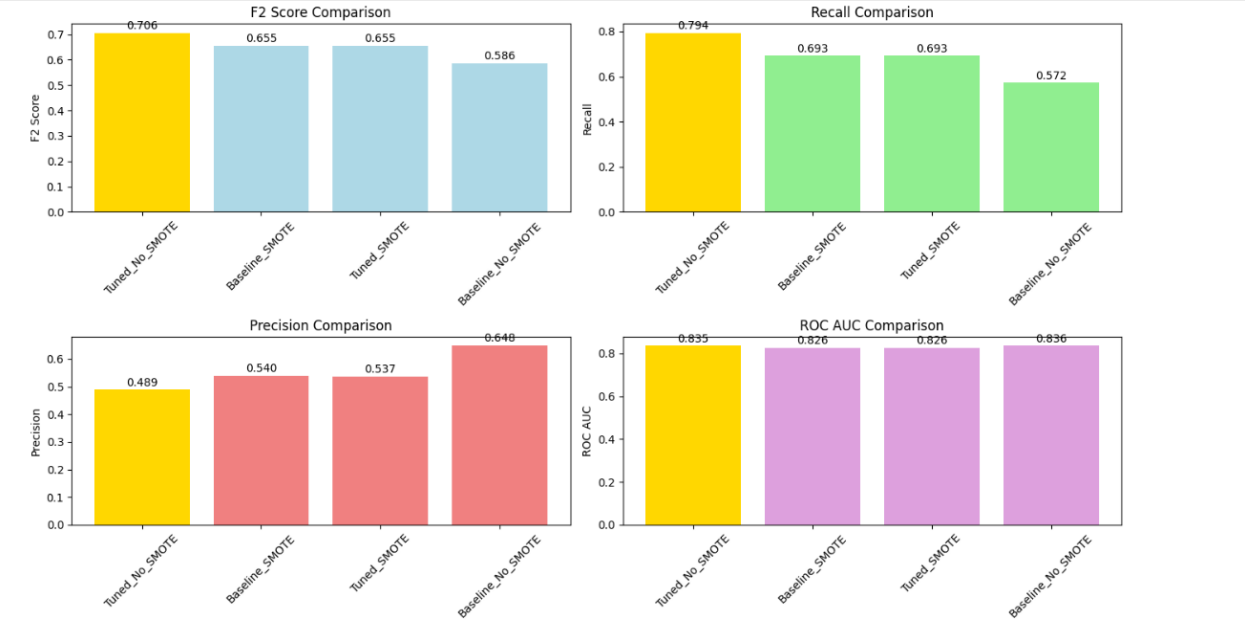
- **F2 Score:** Đây là chỉ số chính trong bài toán vì nhấn mạnh vào *Recall* – yếu tố quan trọng khi mục tiêu là phát hiện tối đa khách hàng có nguy cơ rời bỏ. Mô hình **Tuned_No_SMOTE** đạt F2 Score cao nhất (0.7061), vượt trội so với cả các mô hình baseline lẫn mô hình có sử dụng SMOTE. Điều này cho thấy việc điều chỉnh siêu tham số mà không sử dụng kỹ thuật sinh dữ liệu đã giúp tối ưu khả năng nhận diện chính xác các trường hợp quan trọng.
- **Recall:** Mô hình Tuned_No_SMOTE cũng đạt Recall cao nhất (0.7941), chứng tỏ khả năng phát hiện tốt nhất các khách hàng thực sự rời bỏ. So với các mô hình khác

(Recall chỉ khoảng 0.6925 hoặc thấp hơn), mô hình này giảm thiểu tối đa số lượng khách hàng rời bỏ bị bỏ sót – điều rất quan trọng trong ứng dụng thực tế (ví dụ như chăm sóc khách hàng hoặc giữ chân khách hàng).

- **Precision:** Tuy Precision của mô hình tốt nhất chỉ đạt 0.4893 (thấp hơn các mô hình khác), điều này thể hiện sự đánh đổi hợp lý giữa độ chính xác và khả năng phát hiện. Mô hình ưu tiên phát hiện thật nhiều các trường hợp rời bỏ, dù có chấp nhận một phần dự đoán sai (false positives).
- **Accuracy:** Dù không phải chỉ số quan trọng nhất trong bài toán mất cân bằng dữ liệu, nhưng có thể thấy mô hình Tuned_No_SMOTE vẫn đạt accuracy khá (0.7249), chỉ thấp hơn mô hình Baseline_No_SMOTE. Tuy nhiên, mô hình Baseline_No_SMOTE lại có Recall thấp (0.5722), nghĩa là bỏ sót nhiều khách hàng rời bỏ.
- **ROC AUC:** Mô hình Tuned_No_SMOTE đạt điểm AUC là 0.8351 – rất sát với các mô hình còn lại (dao động từ 0.8263 đến 0.8359), cho thấy mô hình vẫn giữ được khả năng phân biệt tốt giữa hai lớp dữ liệu.

Nhận xét tổng quan:

- Việc sử dụng kỹ thuật SMOTE trong cả hai mô hình (baseline và tuned) không giúp cải thiện đáng kể F2 Score hay Recall. Điều này cho thấy trong trường hợp cụ thể của tập dữ liệu này, việc sinh thêm dữ liệu từ lớp thiểu số chưa mang lại hiệu quả rõ ràng.
- Mặc dù mô hình Baseline_No_SMOTE đạt Accuracy và Precision cao nhất, nhưng lại có Recall và F2 Score thấp nhất – điều này là không phù hợp với mục tiêu tối đa hóa khả năng phát hiện khách hàng rời bỏ.
- Nhờ quá trình tuning kỹ lưỡng, mô hình Logistic Regression không dùng SMOTE đã được cải thiện hiệu suất rõ rệt so với phiên bản baseline. Điều này khẳng định tầm quan trọng của việc điều chỉnh siêu tham số phù hợp thay vì chỉ áp dụng kỹ thuật cân bằng dữ liệu.



Hình 14: So sánh các chỉ số F2 Score, Recall, Precision và ROC AUC giữa các mô hình

Kết luận:

Mô hình **Tuned_No_SMOTE** là lựa chọn tối ưu cho bài toán phân loại rời bỏ khách hàng. Mô hình đạt F2 Score cao nhất và Recall vượt trội, đồng thời duy trì hiệu suất tổng thể tốt trên nhiều chỉ số đánh giá. Mặc dù Precision không cao, nhưng điều này là chấp nhận được trong bối cảnh mục tiêu là phát hiện tối đa các trường hợp rời bỏ.

14 Tóm tắt mô hình và kết quả đạt được

Trong đề tài này, nhóm đã lựa chọn mô hình **Hồi quy Logistic** (Logistic Regression) để giải quyết bài toán phân loại nhị phân — cụ thể là xác định khả năng khách hàng rời bỏ dịch vụ viễn thông.

Mô hình Logistic Regression là một phương pháp thống kê cổ điển nhưng vẫn rất hiệu quả trong các bài toán phân loại, đặc biệt là khi mục tiêu là diễn giải và phân tích tác động của các đặc trưng đầu vào lên xác suất xảy ra sự kiện. Hàm sigmoid (logistic function) giúp chuyển đổi giá trị đầu ra thành một xác suất trong khoảng từ 0 đến 1, thuận tiện cho việc dự đoán nhị phân.

Trong quá trình thực hiện, nhóm đã:

- Tiền xử lý dữ liệu: bao gồm xử lý dữ liệu thiếu, loại bỏ đặc trưng không cần thiết, mã hóa biến phân loại và chuẩn hóa đặc trưng.
- Áp dụng kỹ thuật SMOTE để cân bằng dữ liệu khi lớp khách hàng "rời bỏ" (churn) chiếm tỷ lệ nhỏ.

- So sánh hiệu quả của mô hình Logistic Regression trong các kịch bản: có và không dùng SMOTE, có và không tuning siêu tham số.

Kết quả cuối cùng cho thấy:

- Mô hình **tốt nhất** là Logistic Regression có tuning siêu tham số nhưng **không áp dụng SMOTE**, với:
  - F2 Score: **0.7061**
  - Recall: **79.41%**
  - ROC AUC: **0.8351**
- Điều này cho thấy việc tối ưu hóa siêu tham số và sử dụng trọng số cân bằng (class_weight='balanced') có thể hiệu quả hơn so với tạo mẫu nhân tạo.

Mô hình đã chứng minh được khả năng phát hiện sớm khách hàng có nguy cơ rời bỏ cao — một yếu tố cực kỳ quan trọng trong chiến lược giữ chân khách hàng của doanh nghiệp.

15 Hướng phát triển bài toán (mở rộng)

Để nâng cao hiệu quả của hệ thống dự đoán churn và mở rộng khả năng ứng dụng trong thực tế, nhóm đề xuất một số hướng phát triển như sau:

1. **Thử nghiệm thêm các mô hình nâng cao:** Các mô hình mạnh như *Random Forest*, *XGBoost*, *Gradient Boosting* hoặc *Neural Networks* có thể được áp dụng để cải thiện độ chính xác và khả năng học phi tuyến.
2. **Tích hợp thêm yếu tố thời gian:** Sử dụng dữ liệu chuỗi thời gian (ví dụ: lịch sử sử dụng dịch vụ theo tháng) và áp dụng các mô hình như *LSTM* hoặc *Time Series Classification* để phản ánh xu hướng hành vi khách hàng.
3. **Ứng dụng Explainable AI (XAI):** Áp dụng các kỹ thuật như *SHAP*, *LIME* giúp giải thích quyết định của mô hình, từ đó hỗ trợ bộ phận marketing và CSKH hiểu nguyên nhân rời bỏ và đưa ra giải pháp phù hợp.
4. **Đóng gói mô hình dưới dạng API/Web service:** Triển khai mô hình thành REST API để dễ dàng tích hợp vào hệ thống quản lý khách hàng hiện có của doanh nghiệp.
5. **Phân tầng khách hàng:** Kết hợp với kỹ thuật phân cụm (clustering) như KMeans hoặc DBSCAN để chia khách hàng thành các nhóm, từ đó thiết kế chính sách ưu đãi phù hợp với từng phân khúc.

Những hướng phát triển này không chỉ giúp cải thiện mô hình mà còn tăng tính ứng dụng thực tế, góp phần đưa giải pháp vào hệ thống quản lý khách hàng hiện đại.

16 Tài liệu tham khảo

1. IBM Telco Customer Churn Dataset. Truy cập tại: https://www.kaggle.com/datasets/blastchar/telco-customer-churn
2. Hastie, T., Tibshirani, R., Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.
3. Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media.
4. Brownlee, J. (2020). *Imbalanced Classification with Python*. Machine Learning Mastery.
5. Documentation Scikit-learn: https://scikit-learn.org/
6. Documentation imbalanced-learn: https://imbalanced-learn.org/

17 Lời kết

Sau quá trình tìm hiểu, phân tích và triển khai bài toán "Dự đoán khách hàng rời bỏ dịch vụ viễn thông", nhóm chúng em đã có cơ hội áp dụng những kiến thức lý thuyết được học trong môn **Mô hình tối ưu trong kinh tế** vào một vấn đề thực tế có tính ứng dụng cao trong doanh nghiệp hiện đại. Đây không chỉ là một bài toán kỹ thuật về mô hình hóa, mà còn là một ví dụ điển hình cho việc kết nối giữa phân tích dữ liệu và chiến lược kinh doanh.

Thông qua đề tài, chúng em đã tiếp cận đầy đủ quy trình của một dự án học máy: từ tiền xử lý dữ liệu, phân tích đặc trưng, xử lý dữ liệu mất cân bằng, huấn luyện và đánh giá mô hình, cho đến việc đề xuất các hướng phát triển và ứng dụng thực tế. Đặc biệt, nhóm đã nhận ra rằng một mô hình đơn giản như Logistic Regression, khi được xử lý dữ liệu cẩn thận và tối ưu siêu tham số phù hợp, hoàn toàn có thể đạt được hiệu quả rất cao, đủ để hỗ trợ ra quyết định trong doanh nghiệp.

Trong quá trình thực hiện, nhóm cũng đã gặp không ít khó khăn, đặc biệt là việc xử lý dữ liệu không đồng nhất, cân bằng giữa các chỉ số đánh giá trong bối cảnh dữ liệu mất cân bằng, và lựa chọn mô hình tối ưu phù hợp với mục tiêu kinh doanh. Tuy nhiên, chính những thử thách đó đã giúp chúng em rèn luyện kỹ năng giải quyết vấn đề, tư duy hệ thống và học cách đánh đổi trong mô hình hóa thực tế.

Chúng em xin gửi lời cảm ơn chân thành và sâu sắc đến **cô Cao Nghi Thục** – giảng viên hướng dẫn môn học, vì sự hướng dẫn tận tình, những góp ý quý báu và sự hỗ trợ nhiệt tình trong suốt học kỳ. Sự kiên nhẫn và phương pháp giảng dạy truyền cảm hứng của cô đã giúp chúng em hiểu rõ hơn về giá trị của các phương pháp xử lý hồi quy tuyến tính, hồi quy logistic từ đó áp dụng được trong kinh tế.

Nhóm cũng xin cảm ơn **Trường Đại học Khoa học Tự nhiên – ĐHQG-HCM**, cùng **Khoa Toán - Tin học** đã tạo điều kiện cho sinh viên tiếp cận với những môn học thực tiễn, hiện đại và đầy thách thức, góp phần xây dựng nền tảng vững chắc cho sự phát triển nghề nghiệp sau này.

Cuối cùng, chúng em xin cảm ơn sự phối hợp, nỗ lực và tinh thần trách nhiệm của các thành viên trong nhóm. Từng cá nhân đã hoàn thành tốt phần việc được giao, đồng thời hỗ trợ nhau trong suốt quá trình làm việc. Chính tinh thần làm việc nhóm này là yếu tố then chốt giúp bài báo cáo được hoàn thiện đúng tiến độ và đạt chất lượng tốt.

Mặc dù nhóm đã nỗ lực hết sức để hoàn thành đề tài một cách trọn vẹn, chắc chắn không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự góp ý từ cô và để tiếp tục hoàn thiện hơn trong các dự án nghiên cứu sau này.

Thành phố Hồ Chí Minh, ngày 29 tháng 5 năm 2025
Nhóm thực hiện