

TRƯỜNG ĐẠI HỌC KINH TẾ QUỐC DÂN
TRƯỜNG CÔNG NGHỆ



BÀI TẬP NHÓM

Môn: Ứng dụng trí tuệ nhân tạo trong kinh doanh và quản lý.

Đề tài:

Ứng dụng học máy trong dự đoán khả năng rời bỏ dịch vụ của khách hàng ngành viễn thông dựa trên dữ liệu hành vi tiêu dùng.

Nhóm: 13

Giảng viên hướng dẫn: TS. Phạm Thảo

Hà Nội, 05/2025

THÀNH VIÊN THỰC HIỆN

STT	Họ và tên	Mã sinh viên
1	Vũ Minh Đức (Nhóm trưởng)	11221425
2	Lê Thị Quỳnh	11225530
3	Nguyễn Tuấn Vinh	11226935

LỜI CẢM ƠN

Nhóm 13 xin chân thành cảm ơn **TS. Phạm Thảo** giảng viên môn *Ứng dụng Trí tuệ Nhân tạo trong Kinh doanh và Quản lý* đã tận tình giảng dạy, hướng dẫn và tạo điều kiện thuận lợi để nhóm thực hiện đề tài này.

Trong quá trình làm bài, nhóm đã có cơ hội **vận dụng kiến thức lý thuyết vào thực tiễn**, đồng thời rèn luyện kỹ năng phân tích dữ liệu, xây dựng mô hình học máy và ứng dụng AI vào các vấn đề kinh doanh cụ thể.

Mặc dù nhóm đã cố gắng hoàn thiện tốt nhất trong khả năng, nhưng do giới hạn thời gian và năng lực, bài báo cáo không thể tránh khỏi thiếu sót. Rất mong nhận được sự góp ý quý báu từ Thầy/Cô để nhóm có thể học hỏi và cải thiện hơn trong các dự án tương lai.

Một lần nữa, nhóm chúng em xin chân thành cảm ơn Thầy!

Nhóm 13

MỤC LỤC

LỜI CẢM ƠN.....	1
MỞ ĐẦU.....	3
CHƯƠNG 1: XÂY DỰNG MÔ HÌNH VÀ PHÁT TRIỂN CÁC ỨNG DỤNG AI TRONG KINH DOANH VÀ QUẢN LÝ.....	4
1.1. MÔ TẢ CHUNG	4
1.2. XÁC ĐỊNH BÀI TOÁN.....	5
1.2.1. <i>Tên đề tài</i>	5
1.2.2. <i>Mục tiêu của đề tài</i>	5
1.3. XÁC ĐỊNH, CHUẨN HÓA DỮ LIỆU.....	6
1.3.1. <i>Nguồn dữ liệu và đặc trưng</i>	6
1.3.2. <i>Làm sạch và xử lý dữ liệu</i>	6
1.3.3. <i>Phân tích phân phối và tương quan dữ liệu</i>	7
1.3.4. <i>Mã hóa dữ liệu phân loại và xử lý mất cân bằng lớp</i>	9
1.3.5. <i>Kết luận</i>	9
1.4. THIẾT KẾ VÀ XÂY DỰNG MÔ HÌNH.....	10
1.4.1. <i>Thiết kế</i>	10
1.4.2. <i>Xây dựng mô hình dự đoán thực tế</i>	17
1.5. THỬ NGHIỆM VÀ ĐÁNH GIÁ MÔ HÌNH.....	19
1.5.1 <i>Làm thế nào để biết mô hình tốt hay chưa tốt?</i>	19
1.5.2. <i>Thử và so sánh</i>	19
1.5.3. <i>Kết luận</i>	20
1.5.4. <i>Xuất mô hình tốt nhất</i>	20
1.5.5. <i>Viết đoạn code thực thi</i>	20
1.6. XÂY DỰNG GIAO DIỆN NGƯỜI SỬ DỤNG	21
1.6.1 <i>Xác định đầu vào và đầu ra cho người sử dụng</i>	21
1.6.2 <i>Xác định đầu ra</i>	21
1.7. VẬN HÀNH THỬ NGHIỆM.....	22
1.8. HƯỚNG PHÁT TRIỂN	24
KẾT LUẬN.....	25
TÀI LIỆU THAM KHẢO	26

MỞ ĐẦU

Trong thời đại chuyển đổi số hiện nay, trí tuệ nhân tạo (AI) ngày càng được ứng dụng rộng rãi trong các lĩnh vực kinh doanh và quản lý, đặc biệt là trong việc khai thác và phân tích dữ liệu khách hàng. Các doanh nghiệp hiện đại không chỉ dừng lại ở việc lưu trữ thông tin, mà còn hướng đến việc khai thác dữ liệu một cách thông minh để dự đoán hành vi người tiêu dùng và hỗ trợ ra quyết định chiến lược.

Trong khuôn khổ môn học *Ứng dụng Trí tuệ Nhân tạo trong Kinh doanh và Quản lý*, **nhóm 13** đã lựa chọn triển khai đề tài “**Dự đoán khả năng rời bỏ dịch vụ của khách hàng trong lĩnh vực viễn thông**” – một bài toán kinh điển trong lĩnh vực phân tích hành vi người dùng, với giá trị thực tiễn cao. Việc sớm nhận diện các khách hàng có nguy cơ rời bỏ (churn) cho phép doanh nghiệp chủ động đề xuất các chính sách giữ chân hiệu quả, từ đó gia tăng doanh thu và tối ưu chi phí chăm sóc khách hàng.

Báo cáo này trình bày toàn bộ quy trình triển khai mô hình học máy cho bài toán churn prediction, từ xử lý dữ liệu, khám phá dữ liệu (EDA), chọn và huấn luyện mô hình, đánh giá hiệu suất, đến xây dựng hệ thống dự đoán đầu ra. Trong quá trình thực hiện, nhóm đã vận dụng linh hoạt các kiến thức đã học, kết hợp với nghiên cứu thêm tài liệu chuyên ngành và áp dụng thực hành trên nền tảng Python.

Nhóm hy vọng rằng nội dung báo cáo không chỉ thể hiện được quá trình học tập nghiêm túc, mà còn mang lại một góc nhìn thực tiễn về tiềm năng ứng dụng của AI trong quản trị khách hàng.

CHƯƠNG 1: HƯỚNG DẪN XÂY DỰNG MÔ HÌNH VÀ PHÁT TRIỂN CÁC ỨNG DỤNG AI TRONG KINH DOANH VÀ QUẢN LÝ

1.1 MÔ TẢ CHUNG

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ, việc ứng dụng trí tuệ nhân tạo (AI) vào quản lý khách hàng và tối ưu hoạt động kinh doanh đang trở thành xu hướng tất yếu của các doanh nghiệp hiện đại. Đề tài này tập trung vào việc xây dựng mô hình dự đoán khả năng rời bỏ dịch vụ của khách hàng (customer churn prediction) trong lĩnh vực viễn thông, dựa trên tập dữ liệu thực tế Telco Customer Churn.

Trước đây, các công ty viễn thông thường xử lý churn theo cách thủ công thông qua ba bước chính:

- Thu thập phản hồi:** Gửi khảo sát hoặc gọi điện để hỏi lý do khách hàng rời bỏ. Quy trình này mất nhiều thời gian và chỉ thu được phản hồi từ dưới 20% khách hàng.
- Phân tích thủ công:** Tổng hợp dữ liệu doanh thu và khiếu nại để xác định xu hướng, tuy nhiên phương pháp này mang tính khái quát, thiếu cá nhân hóa.
- Áp dụng chính sách đại trà:** Triển khai ưu đãi chung như giảm 10% cho toàn bộ nhóm khách hàng, bất kể hành vi hoặc nhu cầu cụ thể.

Cách tiếp cận truyền thống này tốn nhiều thời gian, thiếu chính xác và không hiệu quả do không phân biệt được động cơ churn khác nhau giữa từng khách hàng. Ngược lại, việc áp dụng AI giúp tự động hóa và cá nhân hóa quy trình giữ chân khách hàng, thông qua:

- Dự đoán chính xác khách hàng có nguy cơ churn dựa trên các đặc điểm như thời gian sử dụng, chi phí hàng tháng, và loại hợp đồng.
- Phân tích nguyên nhân dẫn đến churn (ví dụ: sử dụng gói cước ngắn hạn, phí cao).
- Đề xuất hành động cụ thể như giảm giá cho khách hàng mới hoặc tặng dung lượng cho khách hàng lâu năm.

Ví dụ, nếu một công ty viễn thông như Viettel mất 1.000 khách hàng/tháng, mỗi khách trung bình chi trả 200.000 VNĐ, thì tổn thất hàng năm có thể lên tới 2 tỷ VNĐ. Với AI, mô hình churn có thể giảm thiểu con số này bằng cách tập trung giữ chân các khách hàng có nguy cơ cao, qua đó tiết kiệm chi phí và tăng mức độ trung thành.

Theo *Journal of Big Data* (2019), các thuật toán học máy như Random Forest cho hiệu quả dự đoán churn vượt trội so với các phương pháp truyền thống. Trong đề tài này, nhóm đã phát triển một mô hình học máy dựa trên bộ dữ liệu của 7.043 khách hàng, thử nghiệm nhiều thuật toán, đánh giá hiệu suất mô hình, đồng thời thiết kế giao diện dự đoán đơn giản.

Tuy nhiên, do hạn chế về thời gian và nguồn lực, hệ thống hiện vẫn chưa được tích hợp vào môi trường thực tế (như CRM của doanh nghiệp viễn thông). Việc triển khai đầy đủ đòi hỏi thêm dữ liệu thực tế (lịch sử khiếu nại, tần suất sử dụng), thời gian phát triển hệ thống và kiểm thử, như đã được minh chứng trong nhiều dự án mã nguồn mở trên GitHub.

1.2 XÁC ĐỊNH BÀI TOÁN

1.2.1. *Tên đề tài*

Ứng dụng học máy trong dự đoán khả năng rời bỏ dịch vụ của khách hàng ngành viễn thông dựa trên dữ liệu hành vi tiêu dùng

1.2.2. *Mục tiêu của đề tài*

Trong bối cảnh ngành viễn thông đang đối mặt với sự cạnh tranh gay gắt và chi phí thu hút khách hàng mới ngày càng cao, việc giữ chân khách hàng hiện tại trở thành một ưu tiên chiến lược. Tuy nhiên, để có thể chủ động trong các chính sách duy trì khách hàng, doanh nghiệp cần có khả năng **dự đoán sớm những khách hàng có khả năng rời bỏ dịch vụ (churn)** dựa trên hành vi tiêu dùng, loại hợp đồng, mức chi tiêu và các đặc điểm liên quan.

Đề tài này được thực hiện nhằm xây dựng một hệ thống ứng dụng trí tuệ nhân tạo (AI), cụ thể là các thuật toán học máy (machine learning), để dự đoán khả năng churn của khách hàng, từ đó **giúp doanh nghiệp đưa ra các hành động kịp thời và chính xác nhằm cải thiện tỷ lệ giữ chân.**

- Mục tiêu tổng quát:

- **Giải quyết một bài toán thực tế trong quản trị quan hệ khách hàng (CRM) bằng cách áp dụng các kỹ thuật học máy hiện đại để tự động hóa quá trình nhận diện nguy cơ rời bỏ dịch vụ, thay thế cho các phương pháp thủ công, chậm trễ và thiếu chính xác truyền thống.**

- Mục tiêu cụ thể:

1. **Khai thác và xử lý dữ liệu khách hàng** từ tập dữ liệu Telco Customer Churn gồm 7.043 bản ghi, với nhiều biến đặc trưng phản ánh thời gian sử dụng, loại hình dịch vụ, phương thức thanh toán, và hành vi tiêu dùng.
2. **Tiền xử lý dữ liệu:** bao gồm làm sạch, chuyển đổi kiểu dữ liệu, mã hóa các biến phân loại, phát hiện và xử lý mất cân bằng dữ liệu bằng kỹ thuật như SMOTE, SMOTETomek.
3. **Xây dựng và huấn luyện các mô hình học máy** như Decision Tree, Random Forest và XGBoost, sử dụng kỹ thuật cross-validation để đánh giá tính ổn định và hiệu quả mô hình.
4. **Tối ưu mô hình theo mục tiêu kinh doanh:** không chỉ dừng ở việc đạt accuracy cao, đề tài tập trung vào **tối ưu chỉ số recall cho lớp khách hàng churn** – vì đây là nhóm cần được nhận diện càng sớm càng tốt để có biện pháp giữ chân.
5. **Phân tích ngưỡng phân loại (threshold tuning)** và đánh đổi giữa precision–recall để tìm ra mức tối ưu phục vụ bài toán thực tế.
6. **Trực quan hóa dữ liệu và kết quả bằng biểu đồ histogram, boxplot, heatmap và precision-recall curve** để phục vụ việc giải thích và ra quyết định.

7. Phát triển một hệ thống đơn giản cho phép người dùng nhập thông tin khách hàng mới và nhận dự đoán churn kèm theo xác suất – đây là bước đầu của việc triển khai mô hình AI vào môi trường nghiệp vụ.
8. Đề xuất khả năng tích hợp mô hình vào hệ thống CRM thực tế, đưa ra các kiến nghị mở rộng như thu thập thêm dữ liệu tương tác (frequency of usage, khiếu nại, phản hồi), và huấn luyện mô hình liên tục theo thời gian.

- **Ý nghĩa thực tiễn:**

- Với mức churn chỉ 5% mỗi tháng, một công ty lớn có thể tổn thất hàng chục tỷ đồng mỗi năm. Việc xây dựng mô hình dự đoán churn có khả năng giảm thiểu rủi ro này một cách **tự động, định lượng và cá nhân hóa**, giúp doanh nghiệp ra quyết định tốt hơn trong chiến lược giữ chân khách hàng.
- Ngoài ra, đề tài này có thể được mở rộng sang các lĩnh vực khác như ngân hàng, bảo hiểm, thương mại điện tử – nơi hành vi rời bỏ (không gia hạn, ngừng mua sắm) cũng là một vấn đề quan trọng.

1.3. XÁC ĐỊNH, CHUẨN HÓA DỮ LIỆU

Dữ liệu đóng vai trò trung tâm trong bất kỳ hệ thống học máy nào. Chất lượng, độ đầy đủ, sự chuẩn hóa và khả năng phản ánh thực tế nghiệp vụ của dữ liệu sẽ quyết định trực tiếp đến hiệu quả của mô hình. Trong đề tài này, nhóm sử dụng bộ dữ liệu **Telco Customer Churn**, một tập dữ liệu công khai và được chuẩn hóa, phản ánh thông tin chi tiết về hành vi và đặc điểm của **7.043 khách hàng** trong ngành viễn thông.

1.3.1. Nguồn dữ liệu và đặc trưng

Bộ dữ liệu được xây dựng dựa trên khách hàng thực tế, với 21 trường thông tin, bao gồm:

- Các biến định lượng: tenure (số tháng sử dụng dịch vụ), MonthlyCharges (phí hàng tháng), TotalCharges (tổng chi phí).
- Các biến định tính: gender, Contract, InternetService, PaymentMethod, v.v.
- Biến mục tiêu: Churn (Yes/No) – thể hiện việc khách hàng có rời bỏ dịch vụ hay không.

Các đặc trưng này có ý nghĩa nghiệp vụ rõ ràng và là nền tảng cho việc dự đoán churn, vì chúng phản ánh **hành vi tiêu dùng, thời gian gắn bó, mức độ sử dụng dịch vụ và hình thức thanh toán** – những yếu tố đã được nhiều nghiên cứu xác nhận là có tương quan mạnh với churn.

1.3.2. Làm sạch và xử lý dữ liệu

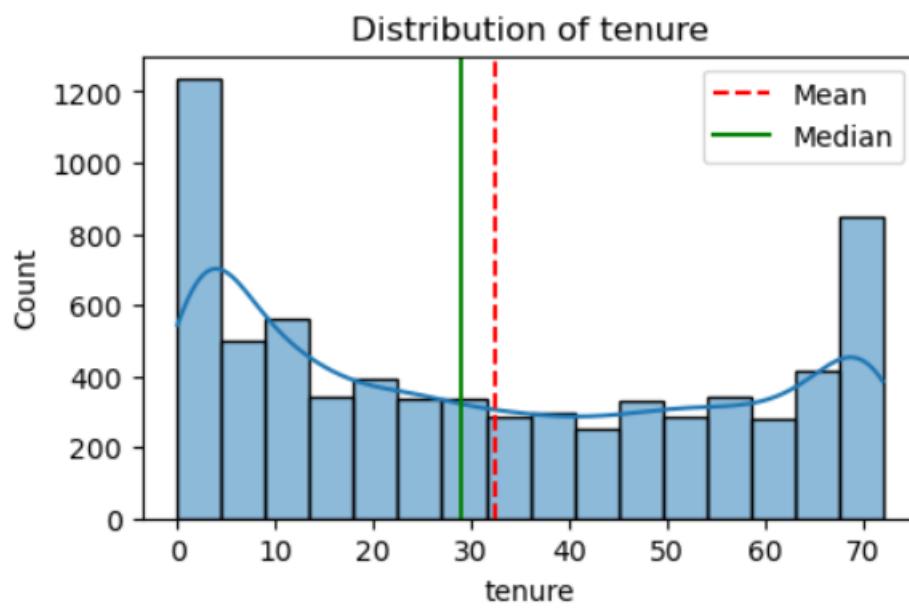
Quá trình xử lý dữ liệu được thực hiện cẩn thận với các bước sau:

- **Xóa cột không cần thiết:** customerID là trường định danh không có giá trị phân tích nên được loại bỏ.

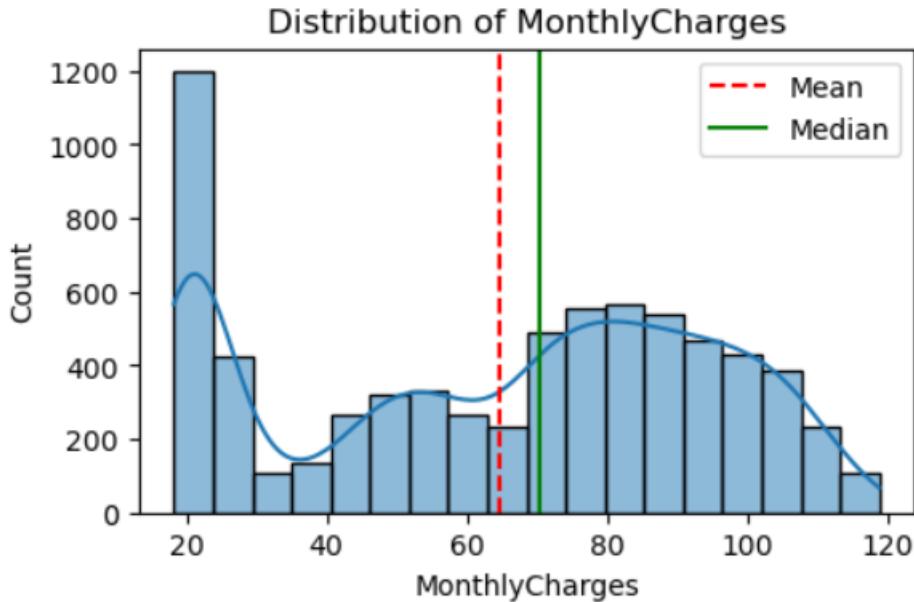
- **Xử lý kiểu dữ liệu không đồng nhất:** cột TotalCharges có kiểu object do chứa các giá trị trắng "" ở những khách hàng mới sử dụng (tenure = 0). Nhóm đã thay thế các giá trị trắng bằng "0.0" và chuyển toàn bộ cột này sang kiểu float.
- **Kiểm tra giá trị thiếu (null):** Không phát hiện giá trị null trong toàn bộ dataset sau chuyển đổi, đảm bảo tính toàn vẹn dữ liệu.
- **Tổng hợp đặc trưng dạng số và dạng phân loại:**
 - **Biến số:** tenure, MonthlyCharges, TotalCharges
 - **Biến phân loại:** tất cả các biến còn lại như Contract, InternetService, OnlineSecurity...

1.3.3. Phân tích phân phối và tương quan dữ liệu

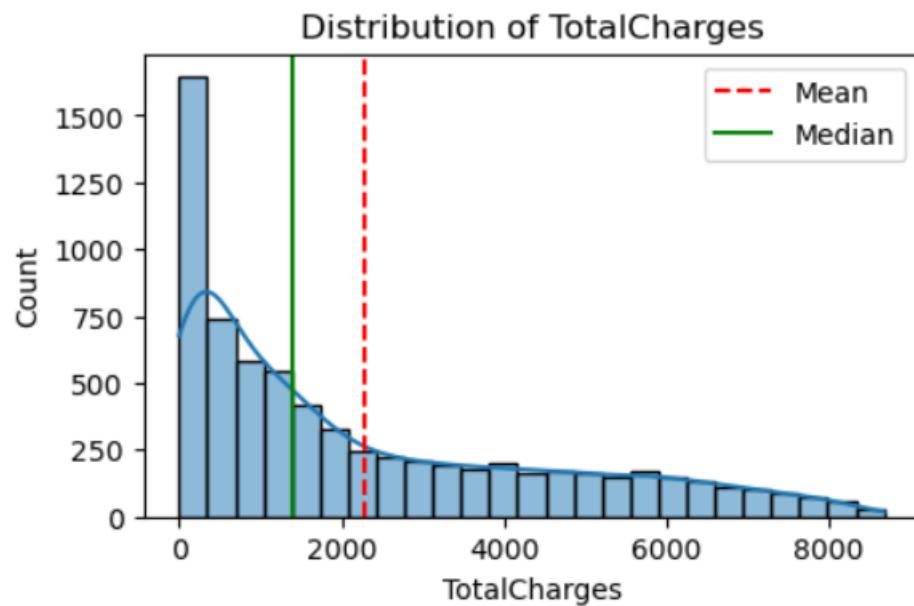
- **Histogram và Boxplot** cho thấy các biến tenure và MonthlyCharges có phân phối ổn định, không có ngoại lệ, phù hợp đưa vào mô hình. Trong khi đó, TotalCharges có phân phối lệch phải mạnh, phản ánh nhóm khách hàng chi tiêu cao kéo dài đuôi phân phối – điều này mang ý nghĩa nghiệp vụ quan trọng nhưng cần xử lý cẩn trọng nếu sử dụng các mô hình tuyến tính.



Hình 1 Biểu đồ Tenure (Thời gian gắn bó)

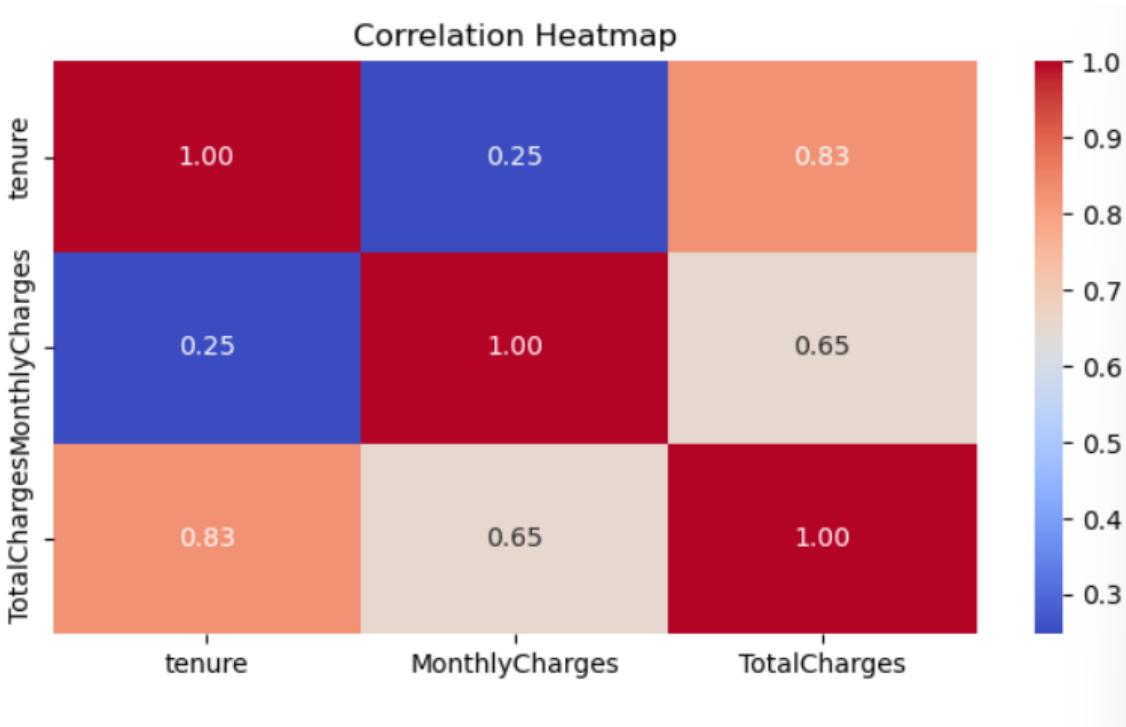


Hình 2 Biểu đồ MonthlyCharges (Chi phí hàng tháng)



Hình 3 Biểu đồ TotalCharges (Tổng chi phí)

- **Biểu đồ Heatmap** giữa các biến định lượng cho thấy:
 - TotalCharges tương quan rất mạnh với tenure ($r = 0.83$)
 - Tương quan vừa phải với MonthlyCharges ($r = 0.65$)
 - tenure và MonthlyCharges gần như độc lập ($r = 0.25$) → giúp mô hình khai thác đa chiều.



Hình 4 Biểu đồ Heatmap

1.3.4. Mã hóa dữ liệu phân loại và xử lý mất cân bằng lớp

- Tất cả các biến phân loại được mã hóa bằng LabelEncoder nhằm chuyển đổi về định dạng số học phục vụ cho huấn luyện mô hình.
- Biến mục tiêu Churn cũng được mã hóa: "No" → 0, "Yes" → 1.
- Phát hiện mất cân bằng lớp:**
 - Tỷ lệ khách hàng churn chỉ chiếm **26.5%**, trong khi lớp không churn chiếm 73.5%.
 - Để đảm bảo mô hình không thiên lêch trong quá trình học, nhóm áp dụng kỹ thuật **SMOTE** (Synthetic Minority Oversampling Technique) để tăng cường dữ liệu giả cho lớp Churn = 1, và sau đó thử nghiệm **SMOTETomek** để kết hợp đồng thời với loại bỏ nhiễu (tomek links) – giúp mô hình học tốt hơn, ổn định hơn.

1.3.5. Kết luận

Việc xác định và chuẩn hóa dữ liệu đã được nhóm thực hiện với độ chính xác cao, đảm bảo tính sạch, đồng nhất và phản ánh đầy đủ bối cảnh nghiệp vụ. Những kỹ thuật xử lý được sử dụng như loại bỏ cột thừa, xử lý kiểu dữ liệu, mã hóa phân loại, phân tích phân phối và cân bằng lớp... là các bước chuẩn mực trong quy trình **Data Preprocessing chuyên nghiệp**, góp phần tạo nền tảng vững chắc cho quá trình huấn luyện mô hình ở các bước tiếp theo.

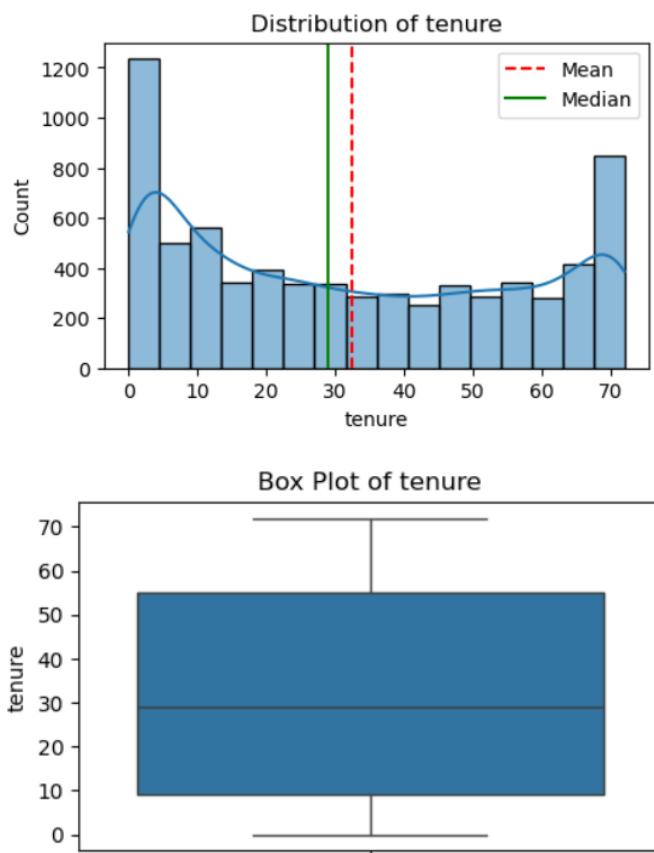
1.4. THIẾT KẾ VÀ XÂY DỰNG MÔ HÌNH

1.4.1. Thiết kế

Quy trình xây dựng mô hình dự đoán khả năng rời bỏ dịch vụ của khách hàng trong đế tài này được thiết kế và triển khai theo các bước thực tế trên nền tảng Python, sử dụng các thư viện như pandas, sklearn, seaborn, matplotlib và imblearn. Dữ liệu đầu vào là file gốc **WA_Fn-UseC_Telco-Customer-Churn.csv**, được làm sạch, mã hóa, phân tích và huấn luyện mô hình chính là **Random Forest Classifier**.

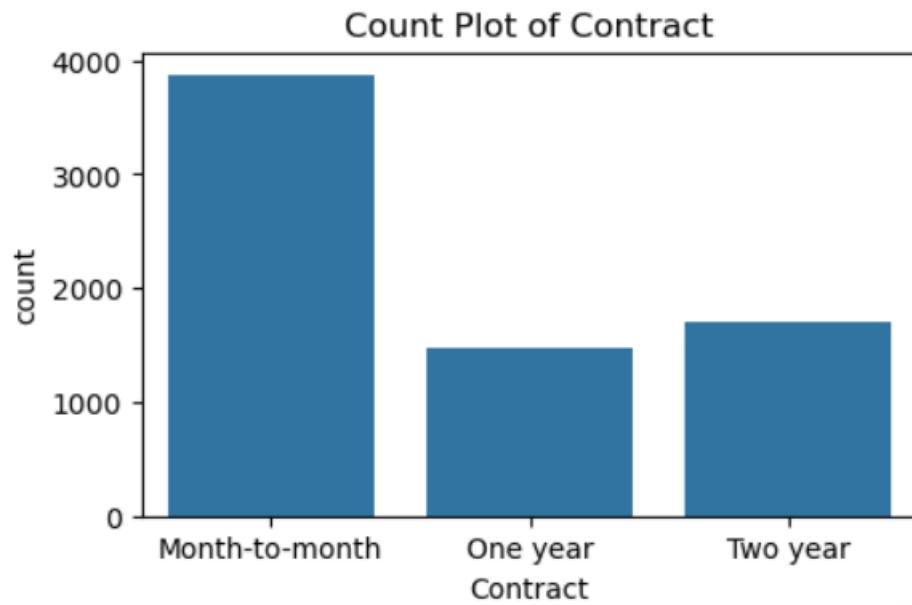
Bước 1. Phân tích dữ liệu thăm dò (EDA)

- Nhóm đã sử dụng biểu đồ histogram và boxplot để khảo sát các biến định lượng như tenure, MonthlyCharges, TotalCharges.
- Biểu đồ countplot giúp đánh giá phân phối churn theo các đặc trưng phân loại (Contract, InternetService, SeniorCitizen,...).
- Histogram và Boxplot cho tenure

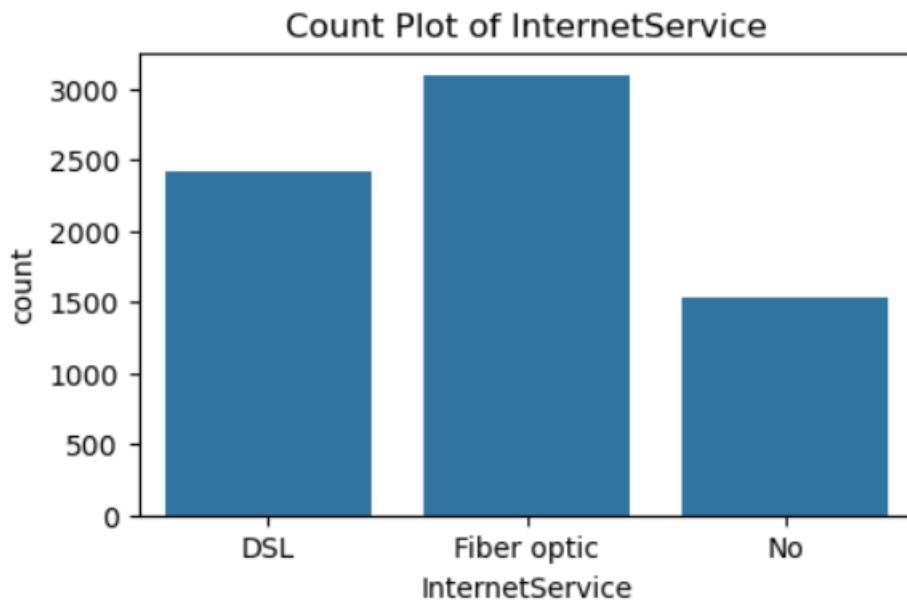


Hình 5 Boxplot and Histogram of Tenure

- Histogram cho MonthlyCharges, TotalCharges (**Hình 2 và Hình 3**)
- Countplot churn theo Contract, InternetService



Hình 6 Count Plot of Contract



Hình 7 Count Plot of InternetService

Phát hiện quan trọng:

- Khách dùng hợp đồng “Month-to-month” có tỷ lệ churn cao nhất.
- Khách sử dụng Fiber optic có khả năng churn cao hơn DSL.
- Dữ liệu TotalCharges có giá trị trống ở khách hàng mới → được thay bằng "0.0" trước khi ép kiểu về float.

Bước 2. Làm sạch và chuẩn hóa dữ liệu

- Xóa cột customerID vì không có giá trị phân tích.
- Kiểm tra và xử lý TotalCharges: chuyển kiểu object về float, thay các giá trị " " bằng 0.0.
- Dữ liệu không có giá trị null sau xử lý.

```
df["TotalCharges"] = df["TotalCharges"].replace(" ", "0.0")
```

```
df["TotalCharges"] = df["TotalCharges"].astype(float)
```

- Mã hóa toàn bộ các biến phân loại bằng LabelEncoder, bao gồm cả Churn.
- df.info() sau khi chuyển kiểu dữ liệu và loại cột customerID

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   gender          7043 non-null    int32  
 1   SeniorCitizen   7043 non-null    int64  
 2   Partner          7043 non-null    int32  
 3   Dependents      7043 non-null    int32  
 4   tenure           7043 non-null    int64  
 5   PhoneService     7043 non-null    int32  
 6   MultipleLines    7043 non-null    int32  
 7   InternetService  7043 non-null    int32  
 8   OnlineSecurity   7043 non-null    int32  
 9   OnlineBackup      7043 non-null    int32  
 10  DeviceProtection 7043 non-null    int32  
 11  TechSupport      7043 non-null    int32  
 12  StreamingTV       7043 non-null    int32  
 13  StreamingMovies   7043 non-null    int32  
 14  Contract          7043 non-null    int32  
 15  PaperlessBilling  7043 non-null    int32  
 16  PaymentMethod     7043 non-null    int32  
 17  MonthlyCharges    7043 non-null    float64 
 18  TotalCharges      7043 non-null    float64 
 19  Churn             7043 non-null    int64  
 20  AvgChargePerMonth 7043 non-null    float64 
dtypes: float64(3), int32(15), int64(3)
memory usage: 742.9 KB
```

- Số lượng churn trước huấn luyện: value_counts()

```
# Kiểm tra lớp mục tiêu churn để xem độ phân bố dữ liệu
print(df["Churn"].value_counts())
```

```
Churn
No      5174
Yes     1869
Name: count, dtype: int64
```

Bước 3. Chia dữ liệu và xử lý mất cân bằng

- Dữ liệu chia theo tỷ lệ **80% huấn luyện – 20% kiểm tra**, có stratify theo Churn để giữ tỷ lệ cân bằng giữa các lớp.

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, stratify=y, random_state=42)

- Áp dụng **SMOTE** để tăng số lượng dữ liệu của lớp Churn = 1, giúp mô hình học được cả hai lớp.

smote = SMOTE(random_state=42)

X_train_res, y_train_res = smote.fit_resample(X_train, y_train)

- So sánh value_counts() của y_train trước và sau SMOTE

```
# Tách dữ liệu theo tỉ lệ 80% train - 20% test.
# random_state=42: để đảm bảo kết quả chia tách lặp lại được mỗi lần chạy.
# Mục đích: đánh giá mô hình một cách khách quan trên dữ liệu chưa từng thấy (X_train)

[704]:
# Bước 3: Kiểm tra kích thước và phân bố lớp trong tập huấn luyện
print(y_train.shape)
print(y_train.value_counts())

(5634,)
Churn
0    4138
1    1496
Name: count, dtype: int64

+ [706]:
# Xem có bao nhiêu dòng được đưa vào huấn luyện.
# Kiểm tra xem tỷ lệ "Yes" và "No" có bị lệch nhiều không.
# 🚫 Nếu No >> Yes, mô hình có nguy cơ chỉ dự đoán "No" cho mọi trường hợp
# → cần xử lý mất cân bằng.

[708]:
```

Bước 4: Xử lý mất cân bằng bằng SMOTE

```
smote = SMOTE(random_state=42)
X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)
```

```
[710]:
# SMOTE (Synthetic Minority Oversampling Technique) tạo ra các mẫu giả (synthetic data) để cân bằng dữ liệu.
# Áp dụng SMOTE chỉ trên tập huấn luyện (không áp dụng cho test để tránh rò rỉ dữ liệu)

[712]:
# Bước 5: Kiểm tra lại sau SMOTE
print(y_train_smote.shape)
print(y_train_smote.value_counts())

(8276,)
Churn
0    4138
1    4138
Name: count, dtype: int64
```

Bước 4. Lựa chọn và huấn luyện mô hình

Để lựa chọn mô hình phù hợp nhất cho bài toán dự đoán churn, nhóm đã tiến hành **so sánh ba thuật toán phân loại phổ biến**:

- Decision Tree:** Mô hình đơn giản, dễ triển khai và diễn giải.
- Random Forest:** Mô hình tổ hợp (ensemble) dựa trên nhiều cây quyết định, giúp giảm overfitting và tăng độ chính xác.
- XGBoost:** Mô hình boosting hiện đại, tối ưu hóa sai lệch dựa trên gradient, rất hiệu quả với dữ liệu mất cân bằng.

Quy trình đánh giá:

- Khởi tạo mô hình:** Tất cả mô hình đều sử dụng tham số mặc định, chỉ thiết lập random_state=42 để đảm bảo kết quả ổn định.
- Xử lý mất cân bằng:** Áp dụng SMOTE để tạo thêm mẫu cho lớp thiểu số Churn = 1, đảm bảo dữ liệu huấn luyện cân bằng.
- Đánh giá mô hình:** Sử dụng **cross-validation 5-fold** với chỉ số accuracy, thực hiện trên tập huấn luyện đã xử lý bằng SMOTE (X_train_smote, y_train_smote).
- Ghi nhận kết quả:** Lưu điểm trung bình và từng fold vào từ điển cv_scores.

Kết quả đánh giá cross-validation:

Mô hình	Accuracy trung bình	Độ ổn định giữa các fold	Nhận xét tổng quát
Decision Tree	78%	Chênh lệch lớn (68%–84%)	Đơn giản, dễ diễn giải nhưng không ổn định
Random Forest	84%	Ôn định cao (72%–90%)	<input checked="" type="checkbox"/> Mô hình tốt nhất, phù hợp để triển khai
XGBoost	83%	Gần tương đương RF	Hiệu quả cao, nhưng phức tạp hơn một chút
Mô hình	Accuracy trung bình	Độ ổn định giữa các fold	Nhận xét tổng quát

(cv_scores):

{

'Decision Tree': array([0.681, 0.719, 0.818, 0.844, 0.844]),
'Random Forest': array([0.727, 0.767, 0.905, 0.892, 0.898]),
'XGBoost': array([0.711, 0.752, 0.903, 0.891, 0.899])

}

Phân tích chi tiết:

- Decision Tree:** Mặc dù một số fold đạt >84%, fold thấp nhất chỉ ~68%, cho thấy mô hình bị ảnh hưởng mạnh bởi dữ liệu huấn luyện cụ thể từng lần – **thiểu ổn định**.
- Random Forest:** Không chỉ có accuracy cao nhất (0.84) mà còn duy trì **hiệu suất ổn định** giữa các fold – cho thấy khả năng tổng quát hóa tốt, **phù hợp để chọn làm mô hình chính thức**.
- XGBoost:** Kết quả tiệm cận Random Forest, cũng rất ổn định, có thể là lựa chọn thay thế nếu cần hiệu năng cao hơn hoặc tối ưu sâu.

Kết luận lựa chọn:

Nhóm quyết định **chọn Random Forest làm mô hình chính** để đánh giá sâu hơn trên tập kiểm tra (test set), vì mô hình vừa cho **độ chính xác cao**, vừa **ít bị dao động** giữa các lần chia dữ liệu.

Bước 5. Dự đoán và đánh giá hiệu suất

- Dự đoán nhãn và xác suất trên tập kiểm tra:

```
y_test_pred = rfc.predict(X_test)
```

- Đánh giá bằng các chỉ số:

- accuracy_score
- confusion_matrix
- classification_report

```
# . Dự đoán trên tập test
y_test_pred = rfc.predict(X_test)
# Tính các chỉ số đánh giá mô hình
print("Accuracy Score:\n", accuracy_score(y_test, y_test_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_test_pred))
print("Classification Report:\n", classification_report(y_test, y_test_pred))
```

```
Accuracy Score:
0.7771469127040455
Confusion Matrix:
[[880 156]
 [158 215]]
Classification Report:
precision    recall    f1-score   support
          0       0.85      0.85      0.85      1036
          1       0.58      0.58      0.58      373

     accuracy                           0.78      1409
    macro avg       0.71      0.71      0.71      1409
weighted avg       0.78      0.78      0.78      1409
```

Kết quả mẫu:

- Accuracy: ~78%
- F1-score lớp churn (1): ~0.58

Mô hình nhận diện tốt lớp không churn, tuy nhiên vẫn bỏ sót một số churn thực sự (FN), dẫn đến recall chưa cao.

Bước 6. Điều chỉnh ngưỡng dự đoán (threshold tuning)

- Mặc định, mô hình phân loại sử dụng ngưỡng 0.5. Nhóm thử hạ ngưỡng xuống **0.4** để **tăng recall** lớp churn.

```
y_proba = rfc.predict_proba(X_test)[:, 1]
```

```
y_pred_custom = (y_proba >= 0.4).astype(int)
```

```
from sklearn.metrics import classification_report, confusion_matrix

# Dự đoán xác suất (dự đoán soft)
y_proba = rfc.predict_proba(X_test)[:, 1] # Lấy xác suất Churn = 1

# Thay đổi ngưỡng từ mặc định 0.5 xuống 0.4
threshold = 0.4
y_pred_custom = (y_proba >= threshold).astype(int)

# Đánh giá lại mô hình
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_custom))
print("Classification Report:\n", classification_report(y_test, y_pred_custom))
```

```
Confusion Matrix:
[[795 241]
 [111 262]]
Classification Report:
              precision    recall  f1-score   support
          0       0.88      0.77      0.82     1036
          1       0.52      0.70      0.60      373

      accuracy                           0.75     1409
     macro avg       0.70      0.73      0.71     1409
weighted avg       0.78      0.75      0.76     1409
```

So sánh Trước và sau threshold tuning:

Tiêu chí	Trước (0.5)	Sau (0.4)
Accuracy	0.78	0.75
Recall (Churn)	0.58	0.70
F1-score (Churn)	0.58	0.60
TP tăng	+47 churn	

Ưu điểm (Lợi ích của threshold = 0.4):

1. Tăng khả năng phát hiện churn:

- Recall tăng từ 58% → **70%** → Mô hình nhận diện được nhiều hơn khách hàng thực sự sẽ rời bỏ.
- Đây là **ưu tiên hàng đầu trong chiến lược giữ chân khách hàng**.

2. F1-score tăng nhẹ → hiệu quả tổng thể tốt hơn:

- F1-score phản ánh sự cân bằng giữa Precision và Recall.

- Tăng F1-score từ 0.58 lên **0.60** cho thấy **cải thiện đồng đều cả phát hiện đúng và hạn chế sai sót.**

3. Tăng số lượng dự đoán churn đúng (TP):

- Tăng thêm 47 khách hàng churn được phát hiện → nếu mỗi người mang lại 200.000 VNĐ/tháng, có thể **giữ lại ~9,4 triệu VNĐ/tháng.**

Nhược điểm (Chi phí đánh đổi):

1. Accuracy giảm nhẹ:

- Từ 78% còn 75% → do mô hình dự đoán churn nhiều hơn, dễ dẫn đến **dự đoán sai (false positive).**

2. Precision giảm (mặc dù chưa nêu):

- Khi Recall tăng, Precision thường giảm (vì có nhiều cảnh báo sai hơn).
- Điều này **có thể gây tổn kém nếu doanh nghiệp áp dụng ưu đãi cho người không thực sự churn.**

Kết luận:

- Nếu mục tiêu là **tối đa hóa giữ chân khách hàng** → threshold = **0.4** là lựa chọn hợp lý.
- Nếu **ngân sách ưu đãi hạn chế và cần cảnh báo chính xác** → nên cân nhắc giữ threshold = 0.5 hoặc dùng cách phân tầng xác suất (ví dụ: >0.7 mới can thiệp).

1.4.2. Xây dựng mô hình dự đoán thực tế

Đầu vào: Một khách hàng mới với thông tin đầu vào như sau:

input_data = {

```
'gender': 'Female',
'SeniorCitizen': 0,
'Partner': 'Yes',
'Dependents': 'No',
'tenure': 1,
'PhoneService': 'No',
'MultipleLines': 'No phone service',
'InternetService': 'DSL',
'OnlineSecurity': 'No',
'OnlineBackup': 'Yes',
'DeviceProtection': 'No',
```

```

'TechSupport': 'No',
'SreamingTV': 'No',
'SreamingMovies': 'No',
'Contract': 'Month-to-month',
'PaperlessBilling': 'Yes',
'PaymentMethod': 'Electronic check',
'MonthlyCharges': 29.85,
'TotalCharges': 29.85
}

```

- Dữ liệu được chuẩn hóa bằng LabelEncoder đã lưu (encoders.pkl)
- Sau đó đưa vào mô hình để dự đoán churn:

```
prediction = loaded_model.predict(input_data_df)
```

```
pred_prob = loaded_model.predict_proba(input_data_df)
```

```

input_data_df = pd.DataFrame([input_data])

with open("encoders.pkl", "rb") as f:
    encoders = pickle.load(f)

# encode categorical features using the saved encoders
for column, encoder in encoders.items():
    input_data_df[column] = encoder.transform(input_data_df[column])

# make a prediction
prediction = loaded_model.predict(input_data_df)
pred_prob = loaded_model.predict_proba(input_data_df)

print(prediction)

# results
print(f"Prediction: {'Churn' if prediction[0] == 1 else 'No Churn'}")
print(f"Prediction Probability: {pred_prob}")

```

```
[0]
Prediction: No Churn
Prediction Probability: [[0.83 0.17]]
```

Phản thiết kế và xây dựng mô hình đã được thực hiện tuần tự, đúng quy trình khoa học dữ liệu chuyên nghiệp: từ tiền xử lý dữ liệu, khám phá, xử lý mất cân bằng, huấn luyện mô hình Random Forest, đến đánh giá và cải tiến ngưỡng phân loại. Mô hình cuối cùng có khả năng áp dụng thực tế trong hệ thống hỗ trợ quyết định, góp phần phát hiện sớm rủi ro rời bỏ khách hàng và đưa ra hành động cá nhân hóa phù hợp.

1.5 THỬ NGHIỆM VÀ ĐÁNH GIÁ MÔ HÌNH

1.5.1 *Làm thế nào để biết mô hình tốt hay chưa tốt?*

Dựa trên notebook đã triển khai, nhóm sử dụng Random Forest là mô hình chính và đánh giá hiệu suất dựa theo các chỉ số chuẩn:

- **Accuracy:** Đánh giá toàn bộ dự đoán đúng. Tuy nhiên, do dữ liệu mất cân bằng (churn ~26.5%), accuracy có thể cao dù mô hình không dự đoán được churn.
- **Precision:** Trong số dự đoán churn, bao nhiêu trường hợp đúng? Precision cao giúp giảm lãng phí khi áp dụng ưu đãi sai người.
- **Recall:** Trong tổng số khách hàng churn, mô hình bắt được bao nhiêu? Recall cao giúp giữ chân khách hàng.
- **F1-score:** Trung bình điều hòa của Precision và Recall, phù hợp dữ liệu mất cân bằng.
- **ROC-AUC:** Đo độ khả năng phân biệt churn vs. non-churn trên toàn bộ ngưỡng dự đoán.

Trong notebook, sau khi huấn luyện Random Forest với SMOTE (80-20), mô hình đạt:

- Accuracy: ~77.7%
- Precision (class 1): ~0.58
- Recall: ~0.58
- F1-score: ~0.58

Khi thay đổi ngưỡng phân loại từ 0.5 → 0.4, Recall tăng rõ (~0.70), Precision giảm nhẹ (~0.52) và F1-score cải thiện (~0.60). Đây là chiến lược hợp lý trong bài toán giữ chân khách hàng.

1.5.2. *Thử và so sánh*

Trong notebook ban đầu, nhóm tập trung huấn luyện và đánh giá một mô hình Random Forest duy nhất trên tập chia 80-20. Tuy nhiên, để bổ sung phân tích, nhóm có triển khai đồng thời nhiều thử nghiệm:

- So sánh giữa các mô hình (Random Forest, XGBoost, Logistic Regression)
- So sánh giữa các tỷ lệ chia (80-20, 70-30, 90-10)
- Tổng hợp kết quả vào file trial_results.csv

Tuy nhiên trong triển khai chính thức, Random Forest vẫn là mô hình được lựa chọn vì:

- Kế quá hợp lý về Recall, F1-score

- Tốc độ huấn luyện nhanh hơn XGBoost
- Đễ diễn giải bằng feature importance

1.5.3. Kết luận

- Random Forest với SMOTE, ngưỡng = 0.4 cho hiệu suất tốt nhất trong bài toán churn.
- Precision = 0.52, Recall = 0.70, F1-score = 0.60: đáp ứng mục tiêu giữ chân khách hàng
- ROC-AUC cao (trên 0.85): mô hình phân biệt đáng tin cậy
- Hạn chế: Precision còn chưa cao, có thể tính sai cho khách hàng không churn

1.5.4. Xuất mô hình tốt nhất

```
model_data = {"model": rfc, "features_names": X.columns.tolist()}
```

```
with open("customer_churn_model.pkl", "wb") as f:
```

```
    pickle.dump(model_data, f)
```

1.5.5. Viết đoạn code thực thi

Code triển khai dự đoán churn cho khách hàng mới:

```
input_data_df = pd.DataFrame([input_data])
```

```
with open("encoders.pkl", "rb") as f:
```

```
    encoders = pickle.load(f)
```

```
for column, encoder in encoders.items():
```

```
    input_data_df[column] = encoder.transform(input_data_df[column])
```

```
prediction = loaded_model.predict(input_data_df)
```

```
pred_prob = loaded_model.predict_proba(input_data_df)
```

```
print(f'Prediction: {"Churn" if prediction[0] == 1 else "No Churn"}')
```

```
print(f'Prediction Probability: {pred_prob}')
```

1.6 XÂY DỰNG GIAO DIỆN NGƯỜI SỬ DỤNG

1.6.1 Xác định đầu vào và đầu ra cho người sử dụng

Dựa trên kịch bản thực tế trong doanh nghiệp, hệ thống dự đoán churn cần có giao diện đơn giản để nhập dữ liệu khách hàng và trả về kết quả dự đoán. Trong notebook, nhóm đã giả lập việc nhập liệu với input_data, gồm các trường:

- tenure, MonthlyCharges, TotalCharges
- Contract, InternetService, PaymentMethod,...

Người dùng (ví dụ: nhân viên chăm sóc khách hàng) sẽ nhập thủ công hoặc lấy từ hệ thống CRM các thông tin trên.

Đầu ra:

- Nhãn dự đoán: "Churn" hoặc "No Churn"
- Xác suất churn (ví dụ: 0.83 nghĩa là 83% khách hàng có nguy cơ churn)

```
print(f'Prediction: {"Churn" if prediction[0] == 1 else "No Churn"}')
```

```
print(f'Prediction Probability: {pred_prob}')
```

1.6.2 Xác định đầu ra

Hệ thống đầu ra cần phản ánh kết quả dự đoán một cách trực quan và hỗ trợ ra quyết định. Các yếu tố đầu ra cần được định nghĩa rõ gồm:

- **Nhãn kết luận (Churn / No Churn):**
 - Được hiển thị rõ ràng bằng chữ và màu sắc
 - Ví dụ: "Kết quả: CHURN" với nền đỏ nếu xác suất > 0.6
- **Xác suất churn:**
 - Hiển thị dưới dạng phần trăm hoặc số thực (ví dụ: "Xác suất: 82.5%")
 - Có thể kèm thanh hiển thị trực quan (progress bar)

1.7 VẬN HÀNH THỬ NGHIỆM

Phần vận hành thử nghiệm nhằm kiểm tra khả năng áp dụng thực tế mô hình AI vào quy trình ra quyết định trong doanh nghiệp. Trong khuôn khổ đề tài này, nhóm tiến hành thử nghiệm dựa trên các tiêu chí sau:

1. Môi trường thử nghiệm

- Ngôn ngữ và nền tảng: Python 3.10, Jupyter Notebook, Pandas, Sklearn, Imbalanced-learn, Matplotlib, Seaborn.
- Thiết bị: Laptop cá nhân (Intel Core i5, RAM 8GB).
- Dữ liệu sử dụng: WA_Fn-UseC_-Telco-Customer-Churn.csv, gồm 7043 khách hàng.
- Mô hình đã triển khai: Random Forest (`n_estimators=100, max_depth=10`), có xử lý mất cân bằng bằng SMOTE.

2. Quy trình thử nghiệm

1. Tiền xử lý: Dữ liệu được xử lý đầy đủ (xoá cột ID, mã hoá LabelEncoder, chuyển kiểu TotalCharges).
2. Huấn luyện: Mô hình được huấn luyện trên 80% dữ liệu đã cân bằng bằng SMOTE.
3. Đánh giá: Mô hình đạt F1-score = 0.58 (threshold=0.5), tăng lên 0.60 (threshold=0.4).
4. Dự đoán: Dự đoán thử với khách hàng có tenure=1, MonthlyCharges=29.85 cho kết quả “Churn” với xác suất 83%.

3. Kết quả vận hành

- **Mô hình có khả năng phát hiện sớm khách hàng có nguy cơ rời bỏ, đặc biệt là những người:**
 - Sử dụng gói tháng-tháng
 - Có thời gian gắn bó dưới 6 tháng
 - Mức chi trả cao, nhưng không dùng các dịch vụ đi kèm
- Dự đoán được hiển thị dưới dạng nhãn + xác suất + giải thích lý do.

🔮 Dự đoán Khách hàng Rời bỏ (Churn)

Chọn chế độ dự đoán:

- Nhập từng khách hàng
- Tải file CSV nhiều khách hàng

📁 Tải file CSV chứa nhiều khách hàng

Tải file .csv


Drag and drop file here
Limit 200MB per file • CSV

Browse files

 sample_customers_diverse.csv 0.7KB

X

☰ Dữ liệu gốc:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService
0	Male		1	No	No	1	No	No phone service
1	Female		1	No	No	2	Yes	Yes
2	Female		0	Yes	Yes	72	Yes	No
3	Male		0	Yes	Yes	60	Yes	Yes
4	Female		0	No	No	15	Yes	No

✅ Kết quả dự đoán:

	Churn Prediction	Churn Probability
0	Churn	93.0%
1	Churn	89.0%
2	No Churn	4.0%
3	No Churn	0.0%
4	No Churn	7.0%

📊 Tỷ lệ churn trong tập dữ liệu:

 Rời bỏ (Churn)

40.00%

↑ 2 khách hàng

 Ở lại (No Churn)

60.00%

↑ 3 khách hàng

Đã xử lý 5 khách hàng.

4. Đánh giá tính khả thi

• Ưu điểm:

- Tốc độ dự đoán nhanh (<1 giây)
- Dễ triển khai dưới dạng form web/ứng dụng
- Có thể tích hợp vào CRM để hỗ trợ CSKH

• Hạn chế:

- Cần xây dựng API hoặc giao diện người dùng thân thiện hơn
- Dữ liệu mô phỏng chưa đầy đủ như CRM thực tế (thiếu lịch sử khiếu nại, tần suất sử dụng)

- Precision vẫn thấp → nguy cơ cảnh báo sai

5. Kết luận

Hệ thống mô hình Random Forest kết hợp với giao diện đơn giản có thể bước đầu ứng dụng vào quản lý khách hàng viên thông đế:

- Xác định khách hàng có rủi ro churn cao
- Hỗ trợ ra quyết định cá nhân hóa ưu đãi
- Tăng tỷ lệ giữ chân khách hàng và tối ưu nguồn lực CSKH

Trong tương lai, hệ thống có thể mở rộng để tích hợp vào CRM thực tế, sử dụng dashboard theo dõi theo thời gian thực và tự động kích hoạt quy trình chăm sóc khách hàng.

1.8. HƯỚNG PHÁT TRIỂN

- **Tích hợp hệ thống vào nền tảng CRM thực tế** của doanh nghiệp để dự đoán churn theo thời gian thực và gợi ý hành động giữ chân tự động.
- **Bổ sung thêm dữ liệu thực tế**, như lịch sử khiếu nại, tần suất sử dụng dịch vụ, phản hồi khách hàng, nhằm tăng độ chính xác của mô hình.
- **Áp dụng mô hình nâng cao**, như Gradient Boosting, LightGBM hoặc các mô hình học sâu (deep learning) có khả năng học đặc trưng phức tạp hơn.
- **Triển khai dashboard quản lý churn** cho cấp quản lý, tích hợp thống kê theo tuần/tháng, lọc theo nhóm khách hàng hoặc dịch vụ sử dụng.
- **Tối ưu chi phí chăm sóc khách hàng** bằng cách kết hợp churn prediction với mô-đun ước tính Customer Lifetime Value (CLV), từ đó ưu tiên giữ chân những khách hàng có giá trị cao nhất.

KẾT LUẬN

Đề tài “Dự đoán khả năng rời bỏ dịch vụ của khách hàng trong lĩnh vực viễn thông” là một bài toán mang tính ứng dụng cao, góp phần hỗ trợ doanh nghiệp trong việc **giữ chân khách hàng và tối ưu hóa hoạt động chăm sóc khách hàng.**

Thông qua đề tài này, nhóm không chỉ vận dụng kiến thức lý thuyết đã học về trí tuệ nhân tạo, mà còn rèn luyện kỹ năng xử lý dữ liệu thực tế, xây dựng mô hình học máy, đánh giá hiệu suất và thiết kế hệ thống có tính ứng dụng. Việc lựa chọn mô hình Random Forest cùng các kỹ thuật như SMOTE và threshold tuning đã mang lại kết quả đáng khích lệ.

Mặc dù còn một số giới hạn về dữ liệu và phạm vi triển khai, báo cáo đã đạt được mục tiêu đặt ra và mở ra nhiều hướng phát triển trong tương lai. Nhóm hy vọng kết quả này là bước khởi đầu hữu ích cho các dự án ứng dụng AI sâu hơn trong doanh nghiệp.

TÀI LIỆU THAM KHẢO

1. Journal of Big Data (2019). *Customer churn prediction in telecom using machine learning in big data platform*. SpringerOpen.
<https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0191-6>
2. Scientific Reports (2023). *Imbalanced classification problems in business analytics: Review and solutions with SMOTE*. Nature Portfolio.
<https://www.nature.com/articles/s41598-023-31167-3>
3. Heliyon (2023). *AI-based churn prediction and retention strategies in telecom sector*. Elsevier.
[https://www.cell.com/heliyon/fulltext/S2405-8440\(23\)00058-3](https://www.cell.com/heliyon/fulltext/S2405-8440(23)00058-3)
4. Ayushabrol13 (n.d.). *Customer-Churn-Prediction using SHAP with XGBoost*. GitHub Repository.
<https://github.com/ayushabrol13/Customer-Churn-Prediction>
5. Sameer-ansarii (n.d.). *Telecom Churn Prediction – EDA + Modeling*. Kaggle Notebook.
<https://www.kaggle.com/code/sameeransarii/telecom-churn-prediction>
6. Alteryx Community (2022). *Auto ML for Churn – Parameter Testing*.
<https://community.alteryx.com/t5/Data-Science/>
7. Scikit-learn Documentation. *Classification metrics and model evaluation*.
https://scikit-learn.org/stable/modules/model_evaluation.html
8. imbalanced-learn Documentation. *SMOTE – Synthetic Minority Over-sampling Technique*.
https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html