BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM

KHOA CÔNG NGHỆ THÔNG TIN

…..o0o…..

LƯU ĐỨC ANH & VŨ MINH QUANG

KHÓA LUẬN TỐT NGHIỆP

PHÂN TÍCH HÀNH VI ỨNG DỤNG DI ĐỘNG

Chuyên ngành: Khoa học dữ liệu

Giảng viên hướng dẫn: TS Nguyễn Chí Kiên

BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM

KHOA CÔNG NGHỆ THÔNG TIN

…..o0o…..

LƯU ĐỨC ANH & VŨ MINH QUANG

KHÓA LUẬN TỐT NGHIỆP

PHÂN TÍCH HÀNH VI ỨNG DỤNG DI ĐỘNG

Chuyên ngành: Khoa học dữ liệu

Giảng viên hướng dẫn: TS Nguyễn Chí Kiên

Tp. Hồ Chí Minh, Tháng 5 năm 2023

INDUSTRIAL UNIVERSITY OF HO CHI MINH CITY FACULTY OF INFORMATION TECHNOLOGY

LUU DUC ANH & VU MINH QUANG

GRADUATION THESIS

MOBILE APP BEHAVIOR ANALYSIS

Major: Data Science

Supervisor: Dr Nguyen Chi Kien

Ho Chi Minh City, May 2023

CONTENT SUMMARY

Title: Mobile App Behavior Analysis

Abstract:

- Reason for writing:

- In today's market, many companies have a mobile presence. Often, these companies provide free products/services in their mobile apps in an attempt to transition their customers to a paid membership. Some examples of paid products, which originate from free ones, include YouTube Red, Pandora Premium, Audible Subscription and You Need a Budget. Since marketing efforts are never free, these companies need to know exactly who to target with their offers and promotions.

- Analyzing customer behavior to predict churn helps to gain a better understanding of the reasons and factors influencing customer attrition in mobile applications. Churn, which refers to customers leaving a service, is a critical issue for businesses, particularly in the Information Technology and mobile app industry.

- Problem:

- Use Machine Learning to predict which users subscribe or not subscribe to the paid membership, and users will continue using or churn from a mobile application.

- Analyzing behavior and applying machine learning models in mobile applications to improve user experience and make predictions.

- Method:

- Learn about user behavior while using the app

- Knowledge about Machine Learning and using to train model

- Result:

- Improved user experience and increased revenue through analyzing customer behavior in mobile applications.

- Trained classification models made predictions with high accuracy.

- Conclusuon:

- Understanding how to collect and analyze user behavior data in mobile applications; Understanding how machine learning models work.

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin chân thành cảm ơn Thầy Nguyễn Chí Kiên – giảng viên hướng dẫn, đã hỗ trợ và đồng hành cùng chúng em trong suốt quá trình thực hiện khóa luận tốt nghiệp. Đặc biệt là thầy luôn tận tâm giúp đỡ và chỉ dẫn chúng em từ việc chọn đề tài, phân tích và đưa ra giải pháp đối với vấn đề nghiên cứu cho đến việc viết báo cáo và trình bày khóa luận, truyền đạt cho chúng em nhiều kiến thức quý giá, giúp chúng em hiểu sâu hơn về lĩnh vực khoa học máy tính và đưa ra những lời khuyên hữu ích, giúp chúng em tiến bộ hơn trong quá trình nghiên cứu.

Tiếp theo, chúng em xin được bày tỏ lòng biết ơn đến toàn thể quý thầy, cô Khoa Công nghệ Thông tin đã cung cấp cho chúng em những kiến thức nền tảng và hỗ trợ chúng em trong suốt quá trình học tập và nghiên cứu tại trường. Những kiến thức và kinh nghiệm từ thầy cô truyền đạt là cơ sở giúp chúng em hoàn thành khóa luận tốt nghiệp.

Cuối cùng, chúng em xin chân thành cảm ơn Trường Đại học Công nghiệp Thành phố Hồ Chí Minh đã tạo điều kiện và cung cấp các nguồn tài nguyên cần thiết trong quá trình học tập để chúng em có thể hoàn thành khóa luận tốt nghiệp của mình. Chúng em sẽ luôn tự hào khi là một sinh viên của Nhà trường và hy vọng sẽ được đóng góp cho sự phát triển của Nhà trường trong tương lai.

Mặc dù chúng em đã hết sức cố gắng và dành nhiều thời gian, tâm huyết để thực hiện khóa luận tốt nghiệp, tuy nhiên với kiến thức và kinh nghiệm thực tiễn còn hạn chế, chắc chắn khóa luận tốt nghiệp của chúng em vẫn còn không ít thiếu sót, chính vì vậy chúng em rất mong sẽ tiếp tục lĩnh hội được ý kiến đóng góp của quý thầy cô để chúng em hoàn thiện khóa luận của mình.

Chúng em xin chân thành cảm ơn.

NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIÁO VIÊN HƯỚNG DẪN

….....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIÁO VIÊN PHẢN BIỆN

….....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

MỤC LỤC

MỤC LỤC HÌNH ẢNH

MỤC LỤC BẢNG

DANH MỤC CÁC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Từ ngữ	Ý nghĩa
AI	Trí tuệ nhân tạo
model	Mô hình
∀	Với mọi
Max, maximize	Lớn nhất
Min, minimize	Nhỏ nhất
lim	Giới hạn
bias	Thiên lệch
weight	Trọng số, tham số
CART	Classification and Regression Tree
priors	Các tiên nghiệm
Coefficient	Hệ số
XGBoost	Extreme Gradient Boosting
eKYC	Hiểu rõ về khách hàng
Churn	Sự rời đi của người dùng
AUC	Area Under the Curve
RFE	Recursive Feature Elimination
RFECV	Recursive Feature Elimination with Cross-Validation

CHƯƠNG 1: GIỚI THIỆU

1.1. Tổng quan

1.1.1. Lý do chọn đề tài

Ứng dụng dành cho thiết bị di động đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày của chúng ta, với hàng triệu người sử dụng chúng hàng ngày cho nhiều mục đích khác nhau như giao tiếp, giải trí và năng suất. Khi việc sử dụng các ứng dụng dành cho thiết bị di động tiếp tục phát triển, các nhà phân tích hành vi ứng dụng dành cho thiết bị di động ngày càng trở nên cấp bách. Nhà phân tích hành vi ứng dụng dành cho thiết bị di động là một chuyên gia nghiên cứu hành vi và tương tác của người dùng trong ứng dụng dành cho thiết bị di động, với mục đích cải thiện trải nghiệm và mức độ tương tác của người dùng.

Tính cấp thiết của chủ đề này được nhấn mạnh bởi thực tế là các ứng dụng dành cho thiết bị di động không ngừng phát triển và trở nên phức tạp hơn. Do đó, việc hiểu cách người dùng tương tác với ứng dụng của họ ngày càng trở nên quan trọng đối với các công ty. Các nhà phân tích hành vi ứng dụng dành cho thiết bị di động đóng một vai trò quan trọng trong quá trình này bằng cách giúp các công ty xác định các lĩnh vực cần cải thiện và tối ưu hóa ứng dụng của họ để thu hút người dùng tốt hơn.

Một trong những lý do chính khiến việc phân tích hành vi của ứng dụng di động trở nên cấp thiết đến vậy là do mức độ tương tác của người dùng là yếu tố quan trọng dẫn đến sự thành công của ứng dụng dành cho thiết bị di động. Nếu không có người dùng tương tác, ứng dụng sẽ gặp khó khăn trong việc tạo doanh thu và đạt được các mục tiêu kinh doanh của mình. Các nhà phân tích hành vi ứng dụng dành cho thiết bị di động có thể giúp các công ty hiểu cách người dùng tương tác với ứng dụng của họ và xác định các lĩnh vực cần cải thiện để có thể tăng mức độ tương tác và tỷ lệ giữ chân.

Một lý do khác khiến phân tích hành vi ứng dụng di động trở nên cấp thiết là nó có thể giúp các công ty vượt lên trên đối thủ. Trong thị trường ứng dụng cạnh tranh cao ngày nay, những công ty có khả năng cung cấp trải nghiệm người dùng tốt nhất là

những công ty sẽ thành công. Các nhà phân tích hành vi ứng dụng dành cho thiết bị di động có thể giúp các công ty xác định các xu hướng và kiểu hành vi của người dùng có thể được sử dụng để cải thiện ứng dụng của họ và dẫn đầu đối thủ.

Phân tích hành vi ứng dụng dành cho thiết bị di động cũng rất quan trọng từ góc độ quyền riêng tư của người dùng. Khi các ứng dụng dành cho thiết bị di động trở nên phức tạp hơn và thu thập nhiều dữ liệu hơn, điều quan trọng là dữ liệu người dùng phải được bảo vệ và sử dụng một cách có đạo đức. Các nhà phân tích hành vi ứng dụng dành cho thiết bị di động có thể giúp đảm bảo rằng dữ liệu người dùng đang được sử dụng một cách có trách nhiệm và đạo đức, đồng thời người dùng nhận thức đầy đủ về cách dữ liệu của họ đang được sử dụng.

Tóm lại, không thể phóng đại tính cấp thiết của chủ đề "Nhà phân tích hành vi ứng dụng di động". Khi việc sử dụng các ứng dụng dành cho thiết bị di động tiếp tục phát triển và trở nên phức tạp hơn, các công ty cần hiểu cách người dùng tương tác với ứng dụng của họ. Các nhà phân tích hành vi ứng dụng dành cho thiết bị di động đóng một vai trò quan trọng trong quá trình này bằng cách giúp các công ty xác định các lĩnh vực cần cải thiện và tối ưu hóa ứng dụng của họ để thu hút người dùng tốt hơn. Bằng cách đó, họ có thể giúp các công ty vượt lên trên đối thủ cạnh tranh, cải thiện quyền riêng tư của người dùng và cuối cùng là mang lại trải nghiệm người dùng tốt hơn cho tất cả mọi người.

1.1.2. Hiện trạng của Việt Nam hiện nay

Việt Nam có thị trường ứng dụng di động đang phát triển nhanh chóng, với hàng triệu người dùng truy cập ứng dụng di động hàng ngày cho nhiều mục đích khác nhau như liên lạc, giải trí và thương mại điện tử. Sự phổ biến của ứng dụng di động cũng dẫn đến sự gia tăng số lượng nhà phát triển ứng dụng di động tại Việt Nam, với nhiều công ty khởi nghiệp mới và các công ty lâu đời tham gia thị trường.

Một trong những thách thức chính đối với thị trường ứng dụng di động ở Việt Nam là vấn đề tương tác và giữ chân người dùng. Với rất nhiều ứng dụng có sẵn, người dùng có rất nhiều tùy chọn để lựa chọn và các nhà phát triển ứng dụng có thể gặp khó khăn trong việc thu hút và giữ chân người dùng. Do đó, nhu cầu ngày càng tăng đối

với các nhà phân tích hành vi ứng dụng dành cho thiết bị di động, những người có thể giúp các công ty hiểu hành vi của người dùng và tối ưu hóa ứng dụng của họ để tương tác và duy trì tốt hơn.

Một thách thức khác đối với thị trường ứng dụng di động tại Việt Nam là vấn đề bảo vệ quyền riêng tư và dữ liệu của người dùng. Khi các ứng dụng dành cho thiết bị di động thu thập thêm dữ liệu về người dùng của họ, ngày càng có nhiều lo ngại về cách dữ liệu này đang được sử dụng và liệu dữ liệu đó có được sử dụng một cách hợp đạo đức hay không. Chính phủ Việt Nam đã đưa ra một số quy định để giải quyết những lo ngại này, bao gồm Luật An ninh mạng và Luật Bảo vệ Dữ liệu Cá nhân.

Bất chấp những thách thức này, thị trường ứng dụng di động ở Việt Nam vẫn tiếp tục phát triển với các ứng dụng và dịch vụ mới được giới thiệu thường xuyên. Với dân số trẻ và am hiểu công nghệ, thị trường ứng dụng di động tại Việt Nam có rất nhiều tiềm năng để tiếp tục mở rộng và đổi mới trong những năm tới.

1.2. Mục tiêu nghiên cứu

- Tìm hiểu rõ về khái niệm rõ về khái niệm, lý thuyết về học máy.

- Tìm hiểu về thái độ người dùng sử dụng ứng dụng điện thoại.

1.3. Phạm vi nghiên cứu

- Các kiến thức cơ bản về học máy
- Nghiên cứu về bài toán phân tích thái độ người dùng sử dụng ứng dụng di động.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Học máy

Học máy hiện là một lĩnh vực đang phát triển nhanh chóng với nhiều phát triển và ứng dụng thú vị. Một trong những động lực chính đằng sau sự tăng trưởng này là dữ liệu và tài nguyên tính toán ngày càng sẵn có, cho phép phát triển các thuật toán học máy phức tạp và phức tạp hơn.

2.1.1. Khái niệm học máy

Học máy là một lĩnh vực của AI liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Các thuật toán học máy xây dựng một mô hình dựa trên dữ liệu mẫu, được gọi là dữ liệu huấn luyện, để đưa ra dự đoán hoặc quyết định mà không cần được lập trình chi tiết về việc đưa ra dự đoán hoặc quyết định này.[1]

Có ba danh mục con của học máy:

- Các mô hình máy học có giám sát được đào tạo với các tập dữ liệu được gắn nhãn, cho phép các mô hình học hỏi và phát triển chính xác hơn theo thời gian. Ví dụ: một thuật toán sẽ được đào tạo với hình ảnh của những chú chó và những thứ khác, tất cả đều do con người dán nhãn và máy sẽ tự học cách xác định hình ảnh của những chú chó. Học máy có giám sát là loại phổ biến nhất được sử dụng ngày nay.
- Trong học máy không giám sát , một chương trình tìm kiếm các mẫu trong dữ liệu không được gắn nhãn. Học máy không giám sát có thể tìm thấy các mẫu hoặc xu hướng mà mọi người không tìm kiếm một cách rõ ràng. Ví dụ: một chương trình máy học không giám sát có thể xem qua dữ liệu bán hàng trực tuyến và xác định các loại khách hàng khác nhau đang mua hàng.
- Máy học củng cố huấn luyện máy thông qua thử và sai để thực hiện hành động tốt nhất bằng cách thiết lập một hệ thống phần thưởng. Học tăng cường có thể huấn luyện các mô hình chơi trò chơi hoặc huấn luyện các phương tiện tự lái lái bằng cách cho máy biết khi nào nó đưa ra quyết định đúng, điều này giúp máy học theo thời gian những hành động mà nó nên thực hiện.[2]

2.1.2. Khái niệm Trí tuệ nhân tạo

Trí tuệ nhân tạo (Artificial Intelligence - AI) là lĩnh vực của khoa học máy tính và kỹ thuật, nghiên cứu về cách thức thiết kế, phát triển và sử dụng các hệ thống hoặc chương trình máy tính có khả năng thực hiện các nhiệm vụ mà trước đây chỉ có con người mới có thể thực hiện được

Nhìn thấy được tiềm năng to lớn của trí tuệ nhân tạo, các ngành công nghiệp đã và đang áp dụng chúng vào rất nhiều lĩnh vực, sau đây là một số ví dụ về ứng dụng của trí tuệ nhân tạo:[23]

- Trong lĩnh vực thương mại điện tử: Hệ thống khuyến nghị và Chatbot giúp nâng cao trải nghiệm người dùng, giúp các nhà buôn bán quảng cáo được sản phẩm của họ đến đúng tay người cần. Bên cạnh đó là các mô hình phát hiện đánh giá gian lận, spam tin nhắn,..
- Trong lĩnh vực giáo dục: Các ứng dụng luyện nói tiếng anh và kiểm tra lỗi chính tả là những sản phẩm trí tuệ nhân tạo nổi bật trong lĩnh vực này.
- Trong lĩnh vực chăm sóc sức khỏe: Những mô hình thị giác máy tính phát hiện tế bào ung thư, sử dụng dữ liệu bệnh án để đưa ra những khuyến nghị chăm sóc cho bệnh nhân.
- Trong lĩnh vực trò chơi: Các mô hình trí tuệ nhân tạo học cơ chế của trò chơi, tổng hợp phân tích và đưa ra những chuyển động phù hợp.
- Trong lĩnh vực tài chính: Hệ thống đánh giá tín dụng sử dụng trí tuệ nhân tạo, hệ thống eKYC chống gian lận, hệ thống phát hiện gian lận,..
- Trong lĩnh vực nông nghiệp: Áp dụng IoT và trí tuệ nhân tạo để đưa ra được những cảnh báo về thời tiết và cây trồng, vật nuôi.
- Trong lĩnh vực không gian vũ trụ: Những mô hình trí tuệ nhân tạo hay những chú rô bốt giúp chúng ta đưa lĩnh vực này lên một tầm cao mới trong công cuộc khai phá vũ trụ.
- Trong ngành dịch vụ – du lịch: Hệ thống khuyến nghị và Chatbot giúp khách du lịch trải nghiệm dịch vụ - du lịch tốt hơn.

- Trong lĩnh vực Marketing: Hệ thống khuyến nghị là ứng dụng chiếm vai trò lớn lao nhất trong lĩnh vực này, tận dụng được khả năng phân tích khách hàng và sữ hỗ trợ của Chatbot giúp công việc trở nên dễ dàng hơn**.**

- Trong đời sống: Ngoài những ứng dụng trên chúng ta còn thấy nhiều ứng dụng khác của trí tuệ nhân tạo trong đời sống như là những chú rô bốt hút bụi nhỏ xíu, những chiếc máy chấm công bằng nhận diện khuôn mặt,..

2.1.3. Ứng dụng của học máy

- Nhận dạng hình ảnh và giọng nói: Các thuật toán học máy có thể được đào tạo để nhận dạng và phân loại hình ảnh, cũng như để hiểu và xử lý lời nói của con người.

- Xử lý ngôn ngữ tự nhiên (NLP): NLP là một lĩnh vực phụ của học máy tập trung vào việc dạy máy tính hiểu, diễn giải và tạo ra ngôn ngữ của con người.

- Phân tích dự đoán: Máy học có thể được sử dụng để phân tích các tập dữ liệu lớn và đưa ra dự đoán về kết quả trong tương lai.

- Phát hiện gian lận: Các thuật toán học máy có thể được sử dụng để xác định các kiểu hành vi gian lận và phát hiện các điểm bất thường trong các giao dịch tài chính.

- Hệ thống đề xuất: Máy học có thể được sử dụng để xây dựng các công cụ đề xuất cung cấp các đề xuất được cá nhân hóa cho các sản phẩm, dịch vụ và nội dung dựa trên hành vi trong quá khứ của người dùng

- Xe tự lái: Máy học là công nghệ then chốt đằng sau xe tự lái và các phương tiện tự lái khác.

- Chăm sóc sức khỏe: Máy học đang được sử dụng để cải thiện chẩn đoán y tế và phát triển kế hoạch điều trị cá nhân hóa.

- Dịch vụ tài chính: Máy học được sử dụng trong lĩnh vực tài chính để chấm điểm tín dụng, phát hiện gian lận và tối ưu hóa danh mục đầu tư.

- Tiếp thị: Máy học được sử dụng trong tiếp thị để phân khúc khách hàng, quảng cáo được cá nhân hóa và tối ưu hóa chiến dịch.

- Chơi game: Máy học được sử dụng trong chơi game cho AI trò chơi, tạo nội dung theo thủ tục và mô hình hóa trình phát.

Đây chỉ là một vài ứng dụng của học máy. Khi công nghệ máy học tiếp tục phát triển và trở nên dễ tiếp cận hơn, chúng ta có thể kỳ vọng sẽ thấy nhiều ứng dụng sáng tạo hơn nữa trong tương lai.

2.2. Các mô hình của học máy

2.2.1. Hồi quy Logistic

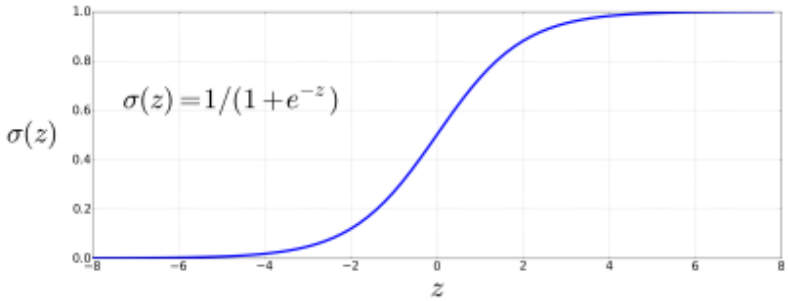
a) Khái niệm

Hồi quy logistic là một thuật toán học máy được giám sát để hoàn thành các nhiệm vụ phân loại nhị phân bằng cách dự đoán xác suất của một kết quả, sự kiện hoặc quan sát. Mô hình mang lại kết quả nhị phân hoặc phân đôi giới hạn ở hai kết quả có thể xảy ra: có/không, 0/1 hoặc đúng/sai.

Hồi quy logic phân tích mối quan hệ giữa một hoặc nhiều biến độc lập và phân loại dữ liệu thành các lớp rời rạc. Nó được sử dụng rộng rãi trong mô hình dự đoán, trong đó mô hình ước tính xác suất toán học về việc liệu một trường hợp có thuộc một danh mục cụ thể hay không.

b) Hàm Sigmoid

Hồi quy logistic giải quyết nhiệm vụ này bằng cách học, từ một tập huấn luyện, một vectơ trọng số và một thuật ngữ b. Mỗi trọng số wi là một số thực và được liên kết với một trong các tính năng đầu vào xi . Trọng số wi thể hiện mức độ quan trọng của tính năng đầu vào đó đối với quyết định phân loại và có thể dương (cung cấp bằng chứng cho thấy cá thể được phân loại thuộc loại dương) hoặc âm (cung cấp bằng chứng cho thấy cá thể được phân loại thuộc loại âm).[3]



Hình 1: Hàm sigmoid σ(z) nhận một giá trị thực và ánh xạ nó vào khoảng (0,1).

Để tạo xác suất, chúng ta sẽ chuyển z qua hàm sigmoid, σ(z). Các hàm sigmoid (được đặt tên vì nó trông giống chữ s) cũng được gọi là hàm logistic và đặt tên cho hồi quy logistic. Hàm sigmoid được thể hiện phương trình sau

$$\sigma(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+\exp(-z)}$$

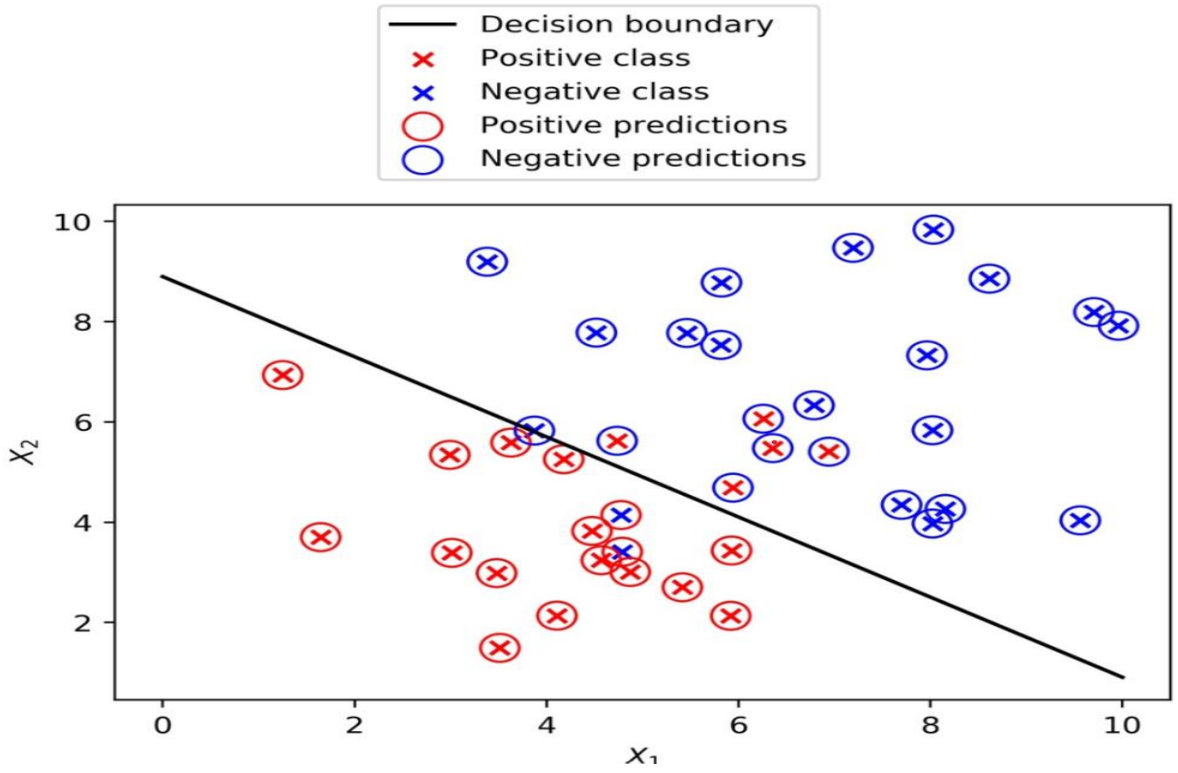
c) Ranh giới quyết định

Trong hồi quy logistic nhị phân, ranh giới quyết định là ranh giới phân tách hai lớp hoặc danh mục mà mô hình đang cố gắng dự đoán. Đó là một đường thẳng hoặc một siêu phẳng trong không gian đặc trưng biểu diễn điểm tại đó xác suất dự đoán thuộc về một lớp bằng với xác suất dự đoán thuộc về lớp kia.

Ranh giới quyết định được xác định bởi các hệ số hoặc trọng số của mô hình hồi quy logistic, được học từ dữ liệu huấn luyện. Mục tiêu của mô hình là tìm ra ranh giới quyết định tối đa hóa khả năng xảy ra của dữ liệu huấn luyện, hoặc tương đương, giảm thiểu hàm mất mát hậu cần.

Khi ranh giới quyết định được thiết lập, các quan sát mới có thể được phân loại dựa trên phía nào của ranh giới mà chúng rơi vào. Nếu xác suất dự đoán thuộc về một lớp lớn hơn 0,5, quan sát được gán cho lớp đó và nếu nó nhỏ hơn 0,5, thì quan sát được gán cho lớp khác.

$$Decision = \begin{cases} 1 & if\ P(y = 1|x) > 0.5 \\ 0 & ngược\ lại \end{cases}$$



Hình 2.2: Các lớp thực, các lớp dự đoán và ranh giới quyết định của hồi quy logistic.

d) Hàm mất Entropy

- Entropy

Entropy của biến ngẫu nhiên X là mức độ không chắc chắn vốn có trong kết quả có thể xảy ra của biến.

Đối với p(x) — phân phối xác suất và một biến ngẫu nhiên X, entropy được định nghĩa:

$$H(x) = \begin{cases} -\int p(x)\ log\ p(x)\ \ nếu\ \ x\ liên\ tục \\ -\sum p(x)\ log\ p(x)\ nếu\ x\ rời\ rạc \end{cases}$$

Giá trị của entropy H(x) càng lớn, độ không đảm bảo cho phân bố xác suất càng lớn và giá trị càng nhỏ thì độ không đảm bảo càng ít.

- Hàm mất Entropy chéo

Mỗi xác suất lớp dự đoán được so sánh với đầu ra mong muốn thực tế của lớp 0 hoặc 1 và điểm/thua được tính toán để xử phạt xác suất dựa trên khoảng cách của nó so với giá trị dự kiến thực tế. Hình phạt về bản chất là logarit mang lại điểm số lớn cho sự khác biệt lớn gần bằng 1 và điểm số nhỏ cho sự khác biệt nhỏ có xu hướng bằng 0.

Mất entropy chéo được sử dụng khi điều chỉnh trọng số mô hình trong quá trình đào tạo. Mục đích là để giảm thiểu tổn thất, nghĩa là tổn thất càng nhỏ thì mô hình càng tốt. Một mô hình hoàn hảo có tổn thất entropy chéo bằng 0.

- Hàm mất Entropy nhị phân

Phân loại nhị phân chia hàm mất mát thành 2 phần, một phần tính toán chi phí cho mẫu dương (y=1) một phần tính toán chi phí cho mẫu âm (y=0)

$$\begin{cases} -log(p) & nếu\ y = 1 \\ -log(1\ -\ p)\ nếu\ y = 0 \end{cases}$$

Với -log(t) sẽ có giá trị lớn nếu t gần giá trị 0 vậy nên hàm mất mát sẽ có giá trị lớn nếu model dự đoán xác suất p với giá trị gần 0 cho mẫu dương (y=1) và ngược lại với mẫu âm (y=0) .

$$L(\theta) = \frac{1}{m}\sum_{1}^{m}[y^i\ log(p^i) + (1-y^i)\ log\ (1-p^i)]$$

e) Hàm giảm độ dốc

Giảm độ dốc là một thuật toán tối ưu hóa lặp đi lặp lại, thuật toán này tìm giá trị nhỏ nhất của một hàm khả vi. Trong quá trình này, chúng tôi thử các giá trị khác nhau và cập nhật chúng để đạt được giá trị tối ưu, giảm thiểu đầu ra. Có thể tìm ra giải pháp tối ưu giảm thiểu chi phí so với tham số mô hình: $\min J(\theta)$

Giả sử chúng ta có tổng cộng n tính năng. Trong trường hợp này, chúng ta có n tham số cho θ vector. Để giảm thiểu chức năng chi phí của chúng tôi, chúng tôi cần chạy giảm độ dốc trên từng tham số θ:

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

lặp qua các tham số θ_0 , θ_1 , …, θ_n trong vector θ. Ta có: [19]

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^i) - y^i)\ x_j^i$$

f) Hàm chuẩn hoá

Chuẩn hoá là một kỹ thuật được sử dụng trong hồi quy logistic để ngăn chặn quá mức và cải thiện hiệu suất tổng quát hóa của mô hình. Nó hoạt động bằng cách thêm một điều khoản phạt vào hàm chi phí hoặc tổn thất để khuyến khích các trọng số nhỏ hơn và ít phức tạp hơn.

Hai thuật ngữ chuẩn hóa phổ biến, được thêm vào để phạt các hệ số cao, là định mức L_1 hoặc bình phương của định mức L_2 nhân với ½, điều này thúc đẩy các tên chính quy hóa L_1 và L_2

L_1 được định nghĩa như sau: $R(\beta) = ||\beta||_1 = \sum_{i=0}^{n} |\beta_i|$

L_2 được định nghĩa như sau: $R(\beta) = \frac{1}{2}||\beta||_2^2 = \frac{1}{2}\sum_{i=0}^{n}\beta_i^2$

Trong đó : β là coefficent [20]

2.2.2. K – láng giềng gần nhất (KNN)

a) Khái niệm

Thuật toán k-láng giềng gần nhất, còn được gọi là KNN hoặc k-NN, là một thuật toán phân loại học có giám sát, phi tham số, sử dụng khoảng cách gần để phân loại hoặc dự đoán về việc nhóm một điểm dữ liệu riêng lẻ. Mặc dù nó có thể được sử dụng cho các vấn đề hồi quy hoặc phân loại, nhưng nó thường được sử dụng như một thuật toán phân loại, dựa trên giả định rằng các điểm tương tự có thể được tìm thấy gần nhau.

b) Số liệu khoảng cách đơn giản

Vì KNN cần tính toán khoảng cách giữa các cặp điểm dữ liệu, một phương pháp đơn giản là sử dụng một giá trị xác định

công thức.

- Khoảng cách Minkowski

Khoảng cách Minkowski của bậc p (trong đó p là số nguyên) giữa hai điểm x = $\{x_1, x_2, \ldots, xn\}$ và y = $\{y_1, y_2, \ldots, yn\}$ được định nghĩa là

$$d(x,y) = (\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}}$$

Với p trong khoảng cách Minkowski khác nhau, chúng ta có thể nhận được các số liệu khoảng cách khác nhau. [4]

Đối với trường hợp đặc biệt của khoảng cách Minkowski p = 1, số liệu Minkowski cho khoảng cách khối thành phố, p = 2, số liệu Minkowski cho khoảng cách Euclide và p = ∞, số liệu Minkowski cho khoảng cách Chebychev.[5]

- Khoảng cách Eculidean

- Đối với 1 chiều:

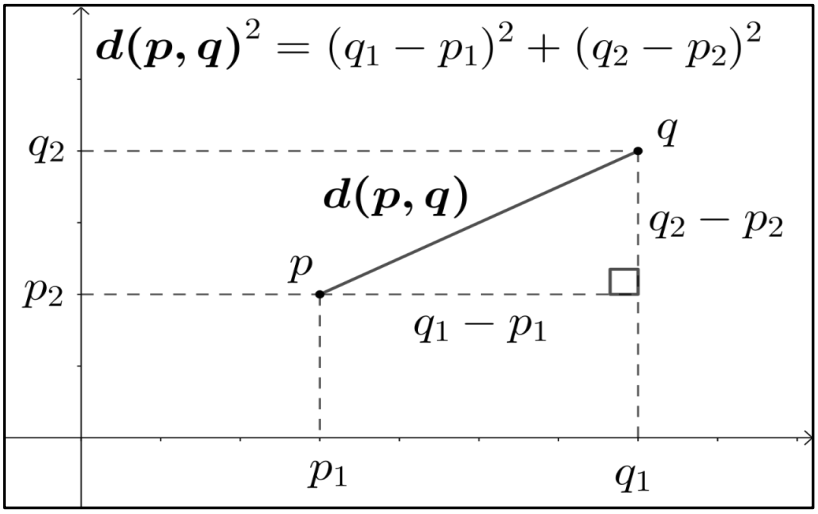
Khoảng cách giữa hai điểm bất kỳ trên đường thẳng thực là giá trị tuyệt đối của hiệu số tọa độ của chúng, hiệu tuyệt đối của chúng. Do đó, nếu x và y là hai điểm trên đường thẳng thực, thì khoảng cách giữa chúng được cho bởi:[6]

$$d(x,y) = |x - y| = \sqrt{(x - y)^2}$$

- Đối với 2 chiều:

Đặt điểm p có tọa độ (x1 , x2) và cho điểm q có tọa độ (y_1 , y_2). Khi đó khoảng cách giữa p và q được cho bởi [7]

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$



Hình 2.3: Khoảng cách Euclide 2 chiều

- Đối với đa chiều:

Nói chung, đối với các điểm được cho bởi tọa độ Descartes trong không gian Euclide n chiều, khoảng cách là:

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2}$$

Khoảng cách Euclide cũng có thể được biểu diễn gọn hơn theo chuẩn Euclide của hiệu vectơ Euclide:

$$d(x\,,y) = ||x - y||$$

- Khoảng cách Manhattan

Khoảng cách giữa hai điểm bằng tổng các hiệu tuyệt đối của tọa độ Descartes của chúng. [8]

$$d(x,y) = \sum_{i=1}^{n} |xi - y_i|$$

- Khoảng cách Cosine

Độ tương tự cosine đo độ tương tự giữa hai vectơ của một không gian tích bên trong. Nó được đo bằng cosin của góc giữa hai vectơ và xác định xem hai vectơ có hướng gần cùng hướng.

$$\cos(x\ ,y) = \frac{x^T y}{||x||||y||}$$

Độ tương tự kết quả dao động từ −1 nghĩa là hoàn toàn ngược lại, đến 1 nghĩa là hoàn toàn giống nhau, với 0 biểu thị tính trực giao hoặc suy giảm tương quan. Do đó, khoảng cách cosin được định nghĩa là:

$$d(x,y) = 1 - \cos\,(x,y)$$

Khi x và y được chuẩn hóa thành đơn vị độ dài, ‖x‖ = ‖y‖ = 1, mối quan hệ giữa khoảng cách Euclide và độ tương tự cosin là: [4]

$$d(x,y) = \sqrt{2 - 2\cos\,(x,y)}$$

- Khoảng cách Chebyshev

Khoảng cách Chebyshev giữa hai vectơ hoặc hai điểm x và y, với tọa độ chuẩn lần lượt là x_{i} và y_{i} là

$$d(x,y) = \max\ |xi - yi|$$

Điều này bằng với giới hạn của chỉ số Lp:

$$\lim_{p\to\infty} (\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}}$$

2.2.3 Naive Bayes

a) Phân loại Bayes

Phân loại Bayes đại diện cho một sự giám sát phương pháp học cũng như phương pháp thống kê để phân loại. Giả định một mô hình xác suất cơ bản và nó cho phép chúng tôi nắm bắt được sự không chắc chắn về mô hình theo cách có nguyên tắc bằng cách xác định xác suất của các kết quả. Nó có thể giải quyết các vấn đề chẩn đoán và dự đoán. Bộ phân loại là một quy tắc gán cho một quan sát một dự đoán hoặc ước tính về nhãn không được quan sát thực sự là gì. Theo thuật ngữ lý thuyết, bộ phân loại là một hàm có thể đo lường được, với cách hiểu rằng C phân loại điểm x vào lớp C(x). Xác suất phân loại sai, hoặc rủi ro.

Bắt đầu bằng cách xem xét cách thiết kế các thuật toán học tập dựa trên quy tắc Bayes. Xem xét một bài toán học có giám sát trong đó chúng ta muốn tính gần đúng một hàm mục tiêu chưa biết f: X → Y hoặc tương đương P(Y|X). Để bắt đầu, chúng ta sẽ giả sử Y là một biến ngẫu nhiên có giá trị boolean và X là một vectơ chứa n thuộc tính boolean. Nói cách khác, X = X_1 , X_2 . . . , Xn , trong đó X_k là biến ngẫu nhiên boolean biểu thị thuộc tính thứ k của X.

$$P(Y = y_i, X = x_k) = \frac{P(Y = y_i, X = x_k)P(Y = y_i)}{\sum_j P\big(Y = y_j, X = x_k\big)P(Y = y_j)}$$

Trong đó y_i biểu thị giá trị có thể thứ i của Y , x_i biểu thị giá trị vectơ có thể thứ i cho X và trong đó tổng ở mẫu số lớn hơn tất cả các giá trị hợp pháp của biến ngẫu nhiên Y. Một cách để tìm hiểu P(Y |X) là sử dụng dữ liệu huấn luyện để ước tính P(X|Y) và P(Y). Sau đó, chúng ta có thể sử dụng các ước tính này, cùng với quy tắc Bayes ở trên, để xác định P(Y |X = x_k) cho bất kỳ trường hợp mới nào x_k.[15]

b) Phân loại Naïve Bayes:

- Gaussian Naive Bayes

Khi xử lý dữ liệu liên tục, một giả định điển hình là các giá trị liên tục được liên kết với mỗi lớp được phân phối theo phân phối chuẩn (hoặc Gaussian).

Giả sử dữ liệu huấn luyện chứa một thuộc tính liên tục x. Trước tiên, dữ liệu được phân đoạn theo lớp, sau đó giá trị trung bình và phương sai của x được tính trong

mỗi lớp. Để μ_k là giá trị trung bình của x liên kết với lớp C_i, và để σ_k^2 là phương sai hiệu chỉnh Bessel của giá trị x liên kết với lớp C_i. Khi đó, mật độ xác suất của v đưa 1 lớp C_i có thể được tính bằng cách đưa v vào phương trình cho một phân phối bình thường được tham số hóa bởi μ_k và C. [16] Ta có:

$$p(x = v|C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} * e^{\frac{-(v-\mu_k)^2}{2\sigma_k^2}}$$

- Naive Bayes đa thức

Với mô hình sự kiện đa thức, các mẫu (vectơ đặc trưng) biểu thị tần suất mà một số sự kiện nhất định đã được tạo bởi một biến cố đa thức (p_1 , p_2 , … , p_n) trong đó p_i là xác suất để biến cố i xảy ra (hoặc K đa thức như vậy trong trường hợp đa lớp). Đây là mô hình sự kiện thường được sử dụng để phân loại tài liệu, với các sự kiện đại diện cho sự xuất hiện của một từ trong một tài liệu. Khả năng quan sát biểu đồ x được đưa ra bởi:

$$p(x|C_k) = \frac{(\sum_{i=1}^{n} x_i)!}{\prod_{i=1}^{n} x_i!} \prod_{i=1}^{n} p_{ki}^{x_i}$$

Với $p_{ki}^{x_i}$ = p(x_i | C_k)

Trình phân loại Naïve Bayes đa thức trở thành một trình phân loại tuyến tính khi được biểu thị trong không gian log:

$$\log p(C_k|x) = \log (p(C_k) \prod_{i=1}^{n} p_{ki}^{x_i})$$

$$= \log p(C_k) + \sum_{i=1}^{n} x_i * logC$$

$$= b + w_k^T * x$$

Với b = log p(C_k) và w_{ki} = log p_{ki} [16]

- Bernoulli naive Bayes

Trong mô hình sự kiện Bernoulli đa biến, các đặc trưng là các Boolean độc lập (biến nhị phân) mô tả đầu vào. Giống như mô hình đa thức, mô hình này phổ biến cho các tác vụ phân loại tài liệu. Trong đó các tính năng xuất hiện thuật ngữ nhị phân được sử dụng thay vì tần suất thuật ngữ. Nếu x_i là một phép toán luận thể hiện sự xuất hiện hoặc vắng mặt của thuật ngữ thứ i trong từ vựng, thì khả năng tài liệu được cấp một lớp C_k được cho bởi: [16]

$$p(x|C_k) = \prod_{i=1}^{n} p_{ki}^{x_i} (1 - p_{ki})^{(1-x_i)}$$

trong đó p_{ki} là xác suất của lớp C_k tạo thuật ngữ x_i

2.2.4 Máy véc tơ hỗ trợ

a) Khái niệm

Support Vector Machine (SVM) là một thuật toán máy học được giám sát mạnh mẽ được sử dụng cho các nhiệm vụ phân loại và hồi quy. Nó hoạt động bằng cách tìm siêu phẳng tối ưu trong không gian đặc trưng nhiều chiều để phân tách các lớp khác nhau hoặc dự đoán giá trị đầu ra cho dữ liệu đầu vào.

Trong bài toán phân loại nhị phân, siêu phẳng tối ưu là siêu phẳng tối đa hóa khoảng cách giữa hai lớp. Lề là khoảng cách giữa siêu phẳng và các điểm dữ liệu gần nhất từ mỗi lớp. Các điểm dữ liệu gần siêu phẳng nhất được gọi là vectơ hỗ trợ và chúng xác định vị trí và hướng của siêu phẳng.

SVM có thể xử lý cả dữ liệu có thể phân tách tuyến tính và phi tuyến tính bằng cách sử dụng các hàm nhân khác nhau để ánh xạ các tính năng đầu vào vào một không gian có chiều cao hơn.

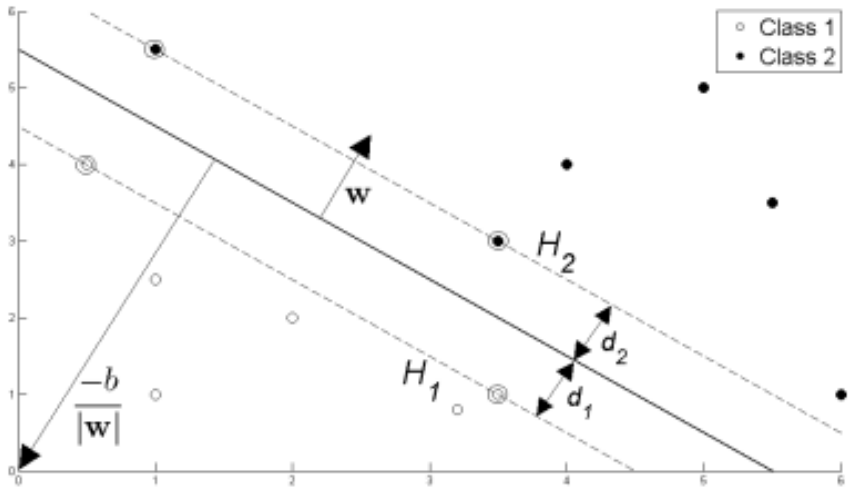
b) Trường hợp tách tuyến tính

Trực giác của SVM là đặt một siêu phẳng ở giữa hai lớp sao cho khoảng cách đến ví dụ dương hoặc âm gần nhất là lớn nhất. Bỏ qua phân phối lớp p(x|y) và giống với hồi quy logistic hơn. Hàm phân biệt SVM có dạng:

$$f(x) = w^T x + b$$

Trong đó: w là vectơ tham số, b là độ lệch [10]

Khoảng cách vuông góc từ siêu phẳng đến gốc tọa độ: $\frac{b}{||w||}$



Hình 2.4: Siêu phẳng thông qua hai lớp có thể phân tách tuyến tính

Việc triển khai một SVM tập trung vào việc chọn các biến w và b để dữ liệu đào tạo của chúng tôi có thể được mô tả bằng:

$$\begin{cases} xi * w + b \geq 1 \; nếu \; yi = +1 \\ xi * w + b < 1 \; nếu \; yi = -1 \end{cases}$$

Những phương trình này có thể được kết hợp thành:

$$yi(xi * w + b) - 1 \geq 0[9]$$

Cho một siêu phẳng phân tách a thỏa mãn các điều kiện, đối với khoảng cách giữa α và điểm x ε s , ta có:

$$dist\ (\alpha, x) = \frac{|w^T + b|}{||w||} \geq \frac{1}{||w||}$$

Do đó, siêu làn tối ưu là siêu làn $\frac{1}{||w||}$ có cực đại đảm bảo biên độ rộng tối ưu $\frac{2}{||w||}$vì đạt được đẳng thức ít nhất cho một điểm và ít nhất một điểm. Độ cực đại $\frac{1}{||w||}$ tương đương với việc giảm thiểu$\frac{1}{2}||w||^2$. Vấn đề bây giờ trở thành [11]

$$\min \quad \frac{1}{2}||w||^2.$$

s.t. yi(w >xi + b) ≥ 1 i = 1 . . . n.

Để phục vụ cho các ràng buộc trong quá trình tối thiểu hóa này, chúng ta cần cấp phát cho chúng các hệ số nhân Lagrange α, trong đó αi ≥ 0 ∀i

$$Lp = \frac{1}{2}\,||w||^2 - \alpha[yi(xi * w + b) - 1]\ \forall i$$

$$= \frac{1}{2}\,||w||^2 - \sum_{i=1}^{L} \alpha i\,[yi(xi * w + b) - 1$$

$$= \frac{1}{2}\,||w||^2 - \sum_{i=1}^{L} \alpha i\, yi(xi * w + b) + \sum_{i=1}^{L} \alpha i$$

Để tìm w và b cực tiểu và α cực đại trong khi vẫn giữ cho αi ≥ 0 ∀i. Có thể làm điều này bằng cách lấy đạo hàm L_P theo w và b và đặt đạo hàm bằng 0:[9]

$$\frac{\partial Lp}{\partial w} = 0 => w = \sum_{i=1}^{L} \alpha i\, yi\, xi$$

$$\frac{\partial Lp}{\partial b} = 0 => w = \sum_{i=1}^{L} \alpha i\, yi$$

Công thức mới này L_D được gọi là dạng kép của L_P . Điều đáng chú ý là dạng Dual chỉ yêu cầu tính toán tích vô hướng của mỗi vectơ đầu vào xi, điều này rất quan trọng đối với Thủ thuật hạt nhân được mô tả trong phần thứ tư.

$$Ld = \sum_{i=1}^{L} \alpha i - \frac{1}{2}\sum_{i.j} \alpha i\, \alpha j\, yi\, yj\, xi\, xj, \qquad s.t\ \alpha i \geq 0\ \forall i, \sum_{i=1}^{L} \alpha i\, yi = 0$$

$$= \sum_{i=1}^{L} \alpha i - \frac{1}{2}\sum_{i.j} \alpha i\, \alpha j\, Hij \ \ với\ Hij = yi\, yj\, xi\, xj$$

$$= \sum_{i=1}^{L} \alpha i - \frac{1}{2}\alpha^T H\alpha, \qquad s.t\ \alpha i \geq 0\ \forall i, \sum_{i=1}^{L} \alpha i\, yi = 0$$

Sau khi chuyển từ giảm thiểu L_P sang tối đa hóa L_D, chúng ta cần tìm:

$$\max\left[\sum_{i=1}^{L} \alpha i - \frac{1}{2}\alpha^T H\alpha\right], \quad s.t\ \alpha i \geq 0\ \forall i, \sum_{i=1}^{L} \alpha i\, yi = 0$$

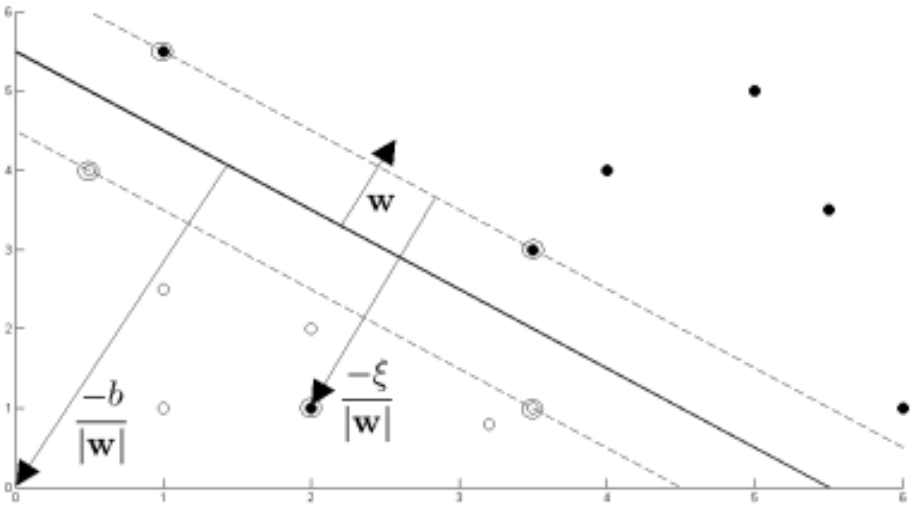
c) Trường hợp tuyến tính không thể tách rời

Trong trường hợp tập không tách tuyến tính thì không thể xây dựng một siêu phẳng tách mà không cho phép sai số phân lớp. Biên độ phân tách giữa các lớp được cho là mềm nếu các điểm dữ liệu huấn luyện vi phạm điều kiện của khả năng phân tách tuyến tính và vấn đề tối ưu hóa ban đầu được thay đổi bằng cách sử dụng các biến trùng. [12]

Cái gọi là biến chùng ξ đo độ lệch giữa vị trí thực và vị trí lý tưởng của mỗi điểm . nếu ξ = 0 , điểm nằm ở phía bên phải, ξ > 1có nghĩa là phía sai, và 0 < ξ < 1 có nghĩa là dải biên, ta có:

$$\text{yi}(\text{w}^T * xi + b) \geq +1 - \xi$$

với ξi ≥ 0 ∀i



Hình 2.5: Siêu phẳng thông qua hai lớp tuyến tính không thể tách rời

Trong SVM lề mềm này, các điểm dữ liệu ở phía không chính xác của lề ranh giới tăng theo khoảng cách từ nó. cố gắng giảm số lượng phân loại sai, một cách hợp lý để thích nghi hàm mục tiêu, là tìm:

$$\min \frac{1}{2}||w||^2 + C\sum_{i=1}^{L} \xi i$$

Trong đó tham số C kiểm soát sự đánh đổi giữa hình phạt biến chùng và kích thước của biên. Là một Lagrangian, cần tối thiểu hóa đối với w, b và ξi và tối đa hóa đối với α

$$Lp = \frac{1}{2}||w||^2 + C\sum_{i=1}^{L} \alpha i\,[yi(xi * w + b) - 1 + \xi i] - \sum_{i=1}^{L} \mu i\ \xi i$$

Vi phân đối với w, b và ξi và đặt các đạo hàm bằng 0, ta có:

$$\frac{\partial Lp}{\partial w} = 0 => w = \sum_{i=1}^{L} \alpha i\, yi\, xi$$

$$\frac{\partial Lp}{\partial b} = 0 => \sum_{i=1}^{L} \alpha i\, yi = 0$$

$$\frac{\partial Lp}{\partial \xi i} = 0 => C = \alpha i + \mu i$$

Thay những thứ này vào, Ld có dạng giống như trường hợp tách tuyến tính, tuy nhiên cùng với μi ≥ 0 ∀i, α ≥ C. Ta có: [13]

$$\max\left[\sum_{i=1}^{L} \alpha i - \frac{1}{2}\alpha^T H\alpha\right], \quad s.t\ 0 \le \alpha i \le C\ \forall i, \sum_{i=1}^{L} \alpha i\, yi = 0$$

d) Trường hợp phi tuyến tính

Trong trường hợp phi tuyến tính, bài toán tối ưu ở dạng đối ngẫu như sau:

$$Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n} \alpha_i \alpha_j y_i\, y_j K(x_i^T x_j)$$

Do đó, có thể sử dụng kernel K(X,Y) để xây dựng siêu phẳng tối ưu trong không gian đặc trưng mà không cần phải xem xét chính không gian đặc trưng đó ở dạng rõ ràng. Siêu phẳng tối ưu hiện được định nghĩa là:

$$f(x) = \sum_{j=1}^{n} \alpha_j \, y_j \, K(x_j, x) + b$$

Cuối cùng, hàm quyết định phi tuyến tính được xác định bởi mối quan hệ sau: [12]

$$F(x) = sign(w^T K(x_j, x) + b)$$

2.2.5 Rừng ngẫu nhiên

a) Cây quyết định

Phân loại cây quyết định là một thuật toán học máy phổ biến được sử dụng cho cả bài toán phân loại và hồi quy. Nó hoạt động bằng cách phân vùng đệ quy dữ liệu thành các tập hợp con dựa trên các giá trị của một trong các tính năng đầu vào, cho đến khi đáp ứng tiêu chí dừng. Bộ phân loại cây nhị phân được coi là trường hợp đặc biệt của bộ phân loại cây quyết định. Các điều kiện tách thích hợp khác nhau giữa các ứng dụng. Một nút được gọi là nút cuối khi nó chỉ chứa một quyết định của lớp

Ba phương pháp được sử dụng rộng rãi trong thiết kế cây là entropy, gini và twoing. Trong phương pháp đầu tiên, entropy là được sử dụng như một thước đo cơ bản của lượng thông tin. Các thông tin dự kiến cần thiết để phân loại một véc tơ quan sát D được cho bởi:

$$Info(D) = -\sum_{i=1}^{n} p_i * \log p_i$$

Trong đó p_i là xác suất mà một véc tơ quan sát trong D thuộc về lớp C_i

Gini là xác suất rằng một lớp được dán nhãn ngẫu nhiên, có tính đến lớp phân phối và priors, được dán nhãn không chính xác. thu được thông tin sử dụng chỉ số gini được định nghĩa là:

$$G(D) = 1 - \sum_{i=1}^{n} p_i$$

Trong đó p_i là xác suất mà một véc tơ quan sát trong D thuộc về lớp C_i

Một phương pháp khác được sử dụng để phân tách là twoing, sử dụng một chiến lược khác để tìm ra cách phân chia tốt nhất giữa các trường hợp.

Độ lợi thông tin được định nghĩa là sự khác biệt giữa yêu cầu thông tin ban đầu và yêu cầu thông tin mới thu được sau khi phân vùng trên thuộc tính A

$$Gain(A) = Info(D) - Info_A(D)$$

Với $Info_A(D) = \sum_{i=1}^{n} \frac{D_i}{D} * Info(D)$

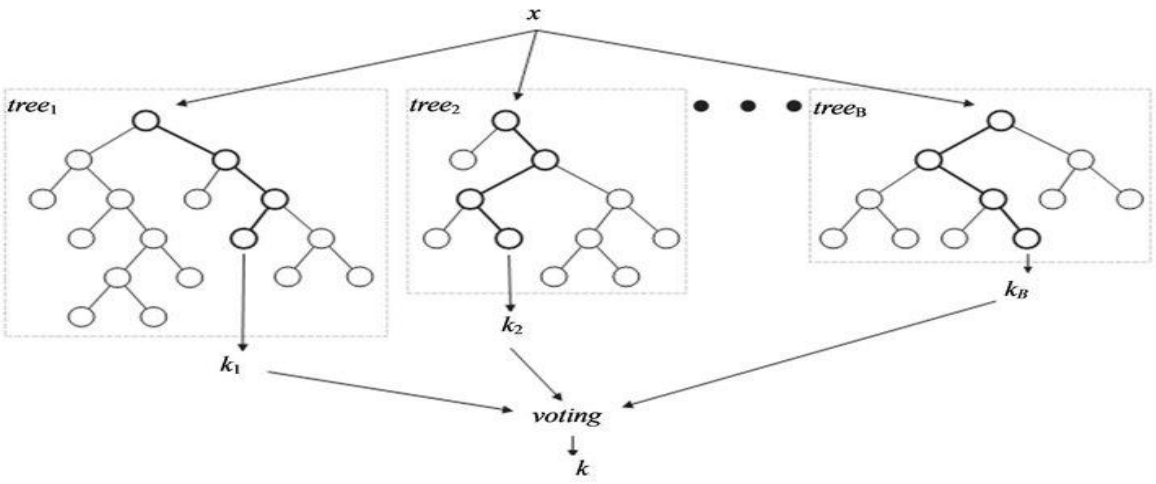
Info(D) có thể được tính bằng một trong các phương trình được hiển thị ở trên tùy thuộc vào phương pháp chia tách mong muốn [17]

b) Rừng ngẫu nhiên

Giới thiệu về CART: Cây quyết định là một thuật toán học máy có giám sát được sử dụng để lập mô hình dự đoán của một biến phụ thuộc (mục tiêu) dựa trên đầu vào của một số biến độc lập. Nó có cấu trúc dạng cây với gốc ở trên cùng. CART viết tắt của Phân loại và Cây hồi quy được sử dụng làm thuật ngữ để chỉ các loại cây quyết định sau:

- Phân loại cây: trong đó biến mục tiêu là cố định hoặc phân loại, thuật toán này được sử dụng để xác định lớp/loại mà mục tiêu có nhiều khả năng nhất sẽ rơi vào trong đó.
- Cây hồi quy: trong đó biến mục tiêu là liên tục và thuật toán/cây được sử dụng để dự đoán giá trị của nó, ví dụ: dự đoán thời tiết.

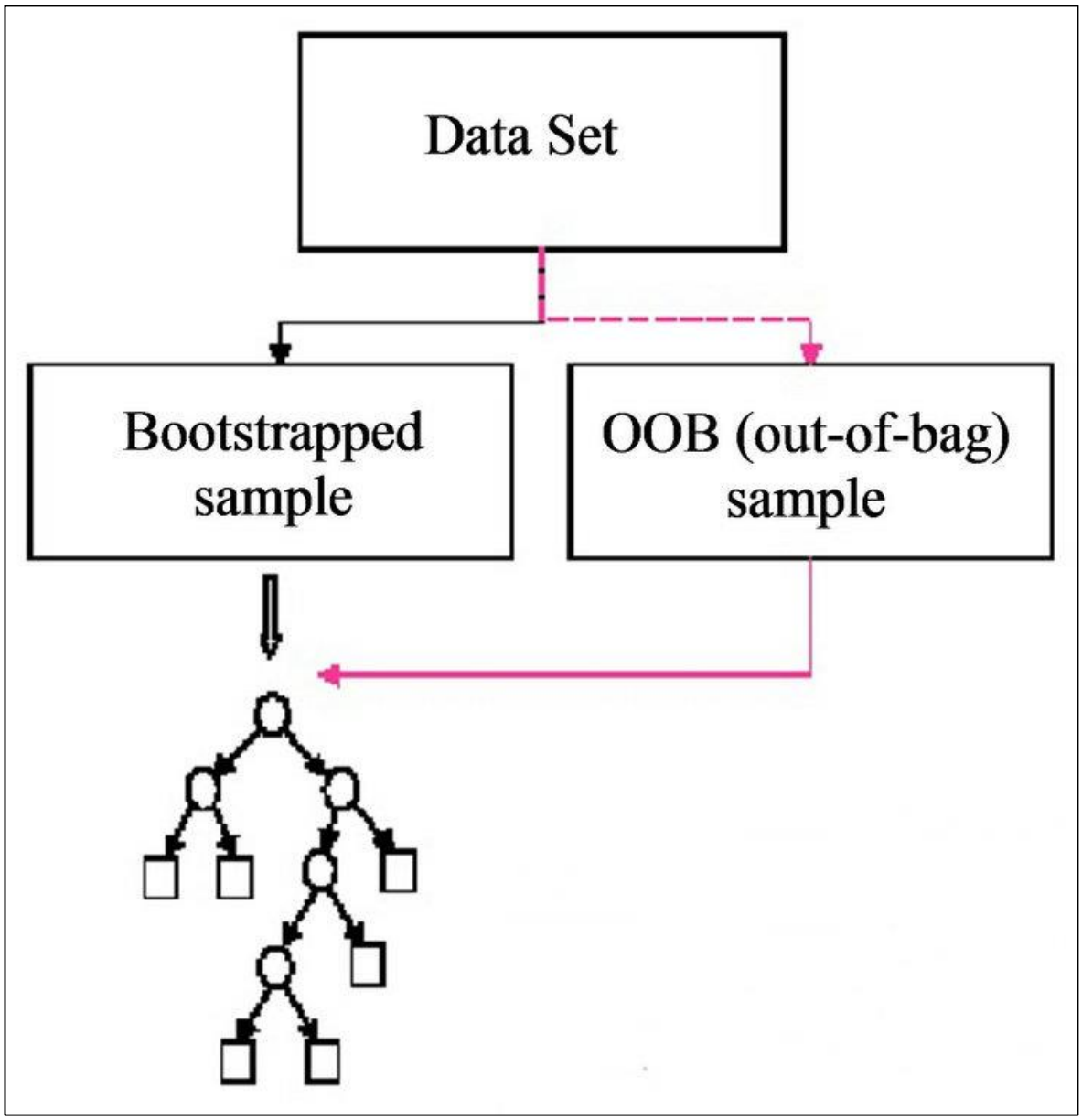
Các thuật toán rừng ngẫu nhiên (RF) tạo thành một nhóm các phương pháp phân loại dựa trên sự kết hợp của một số cây quyết định. Điểm đặc biệt của Nhóm phân loại như vậy là các thành phần dựa trên cây của chúng được phát triển từ một mức độ ngẫu nhiên nhất định. Dựa trên ý tưởng này, RF được định nghĩa là một nguyên tắc chung của tập hợp ngẫu nhiên các cây quyết định. Đơn vị cơ bản của RF là một cây nhị phân được xây dựng bằng cách sử dụng phân vùng đệ quy



Hình 2.6: Kiến trúc của rừng ngẫu nhiên

Trình học cơ sở cây RF thường được phát triển bằng cách sử dụng phương pháp của CART một phương pháp trong đó phân tách nhị phân theo cách đệ quy phân vùng cây thành các nút đầu cuối đồng nhất hoặc gần đồng nhất (các đầu của cây). Một phép chia nhị phân tốt sẽ đẩy dữ liệu từ nút cây mẹ sang hai nút con của nó để tính đồng nhất tiếp theo trong các nút con được cải thiện từ nút mẹ. RF thường là một tập hợp từ hàng trăm đến hàng nghìn cây, trong đó mỗi cây được trồng bằng cách sử dụng mẫu bootstrap của dữ liệu gốc.. Ngoài việc ngẫu nhiên hóa được giới thiệu bằng cách phát triển cây bằng cách sử dụng mẫu bootstrap của dữ liệu gốc, lớp ngẫu nhiên hóa thứ hai được đưa ra ở cấp độ nút khi phát triển cây. Thay vì phân tách một nút cây bằng cách sử dụng tất cả các biến, RF chọn tại mỗi nút của mỗi cây một tập hợp con ngẫu nhiên của các biến và chỉ những biến đó được sử dụng làm ứng cử viên để tìm ra cách phân chia tốt nhất cho nút. Mục đích của việc ngẫu nhiên hóa hai bước này là loại bỏ tương quan giữa các cây để quần thể rừng sẽ có phương sai thấp.[18]

Khoảng một phần ba bootstrap bị loại bỏ và được coi là dữ liệu ngoài túi (OOB). Ngoài ra, không phải mọi tính năng đều được sử dụng để xây dựng cây. Một lựa chọn ngẫu nhiên các tính năng được đánh giá trong mỗi nút. Dữ liệu OOB được sử dụng để nhận tỷ lệ lỗi phân loại khi cây được thêm vào rừng và để đo tầm quan trọng của biến (tính năng) đầu vào. Sau khi rừng hoàn thành, một trường hợp có thể được phân loại bằng cách lấy đa số phiếu bầu trong số tất cả các cây trong rừng giống như ý tưởng tổng hợp bootstrap.[17]



Hình 2.7: Mẫu Boostrap và mẫu OOB

Dữ liệu OOB cũng được sử dụng để ước tính tầm quan trọng của các biến. Hai ước tính này (ước tính lỗi tập kiểm tra và tầm quan trọng của biến) là sản phẩm phụ rất hữu ích của RF.

2.2.6 XGBoost

a) Tăng cường độ dốc

Tăng cường độ dốc là một mô hình phân lớp dựa trên việc tối ưu hàm mất mát bằng cách sử dụng gradient descent và thêm vào các cây quyết định trong từng vòng

lặp để cải thiện dự đoán phân loại. Gradient Boosting có thể giúp tối ưu mô hình dự đoán và cải thiện độ chính xác của mô hình.

Gradient Boosting là một trong những phương pháp trí tuệ nhân tạo điển hình được sử dụng để phát triển các mô hình phân loại và hồi quy nhằm tối ưu hóa quá trình học của mô hình để giải quyết các vấn đề phi tuyến tính. Gradient Boosting được biết đến rộng rãi hơn với tên gọi cây quyết định hoặc cây hồi quy.

Gradient Boosting được đào tạo và xây dựng bằng cách bằng cách thêm người học mới theo cách tuần tự dần dần từ đó nhóm các mô hình dự đoán yếu, ví dụ, cây quyết định, thông qua các các nút và lá của cây quyết định, và kết quả dự đoán cuối cùng được xác định dựa trên các nút quyết định. Các cây quyết định riêng lẻ là những mô hình yếu, nhưng khi được xem như một tập hợp (Gradient Boosting), độ chính xác của chúng được cải thiện nhiều. Vì vậy, các quần thể được xây dựng dần dần theo cách tăng dần sao cho mọi quần thể sẽ sửa lỗi trong quần thể trước đó, từ đó nâng cao độ chính xác trong quá trình đào tạo mô hình[21].

$$min_{c_n=1:N, w_n=1:N} L(y, \sum_{n=1}^{N} c_n w_n))$$

Trong đó :

L : giá trị loss function

y : label

c_n: confidence score của weak learner thứ n (hay còn gọi là trọng số)

w_n : weak learner thứ n

Thay vì cố gằng quét tìm tất cả các giá trị c_n, w_n để tìm nghiệm tối ưu toàn cục (một công việc tốn nhiều thời gian và tài nguyên), nên cố gắng tìm các giá trị nghiệm cục bộ sau khi thêm mỗi một mô hình mới vào chuỗi mô hình với mong muốn dần đi đến nghiệm toàn cục.

b) XGBoost:

XGBoost là một công nghệ máy học tối ưu hóa cây có thể mở rộng, gần đây đã được sử dụng rộng rãi trong các ngành phân tích dữ liệu. Kỹ thuật XGBOOST đã được đề xuất như một máy tăng cường độ dốc được áp dụng có một không hai, đặc

biệt là trong cây phân loại và hồi quy. Khái niệm "tăng cường" là gốc rễ của XGBOOST, kết hợp dự đoán về những loại học yếu với các phương pháp đào tạo bổ sung để phát triển một loại học mạnh. Ngoài ra, quá trình này giúp tránh trang bị thừa và cải thiện khả năng toán học. Chức năng chung của dự báo được thiết lập ở bước p, ta có:

$$f_i^{(p)} = \sum_{k=1}^{p} f_k(x_i) = f_i^{(p-1)} + f_p(x_i)$$

Trong đó $f_p(x_i)$ biểu thị loại học ở bước p, $f_i^{(p)}$ biểu thị dự đoán tại p, $f_i^{(p-1)}$ biểu thị dự đoán tại p – 1 và x_i biểu thị đặc trưng đầu vào

Để làm cho overfitting hợp lý trong khi giảm tốc độ mô hình của toán học, công thức phân tích đã được tạo bởi XGBoost, để ước tính mức độ tốt của mô hình đối với chức năng ban đầu, ta có:

$$Obj^{(p)} = \sum_{k=1}^{n} l(\bar{y}_i, y_i) + \sum_{k=1}^{p} \sigma(f_i)$$

Trong đó l được thể hiện là hàm mất mát, n thể hiện số lượng quan sát được sử dụng, và σ trình bày thuật ngữ chính quy hóa như được biểu diễn:

$\sigma(f) = \gamma T + 0.5\lambda\omega^2$, trong đó ω thể hiện điểm vector trong lá, γ thể hiện sự mất mát tối thiểu cần thiết để chia nút lá hơn nữa, và λ biểu thị các tham số chuẩn hóa. [22]

2.3. Bài toán về phân tích thái độ khách hàng sử dụng ứng dụng di động

2.3.1 Khái niệm về khách hàng tham gia trên ứng dụng điện thoại.

Khách hàng tham gia trên ứng dụng điện thoại là quá trình tương tác giữa khách hàng và doanh nghiệp thông qua ứng dụng điện thoại. Quá trình này bao gồm các hoạt động như tải ứng dụng, đăng nhập, tương tác với sản phẩm hoặc dịch vụ, và chia sẻ thông tin.

Việc tham gia trên ứng dụng điện thoại cung cấp cho khách hàng một phương tiện thuận tiện để tương tác với sản phẩm hoặc dịch vụ của doanh nghiệp. Khách hàng

có thể tìm kiếm thông tin, mua sản phẩm, cập nhật tài khoản và thực hiện các hoạt động khác từ điện thoại của mình.

Đối với doanh nghiệp, việc tạo ra một ứng dụng điện thoại hấp dẫn và tương tác khách hàng thông qua ứng dụng có thể giúp tăng tương tác và gắn kết khách hàng. Một số hoạt động có thể được thực hiện để tăng cường khách hàng tham gia trên ứng dụng điện thoại, bao gồm:

- Cung cấp trải nghiệm người dùng tốt: Thiết kế giao diện và trải nghiệm người dùng tốt có thể giúp thu hút khách hàng sử dụng ứng dụng của doanh nghiệp.
- Cung cấp nội dung hấp dẫn: Cung cấp các nội dung hấp dẫn và giá trị cho khách hàng có thể giúp tăng tương tác và tạo sự gắn kết.
- Tích hợp tính năng xã hội: Tích hợp các tính năng xã hội có thể giúp khách hàng tương tác và chia sẻ thông tin với nhau.
- Tích hợp tính năng tính năng lưu trữ và quản lý: Tích hợp các tính năng quản lý và lưu trữ dữ liệu giúp khách hàng tiện lợi và dễ dàng trong việc sử dụng ứng dụng của doanh nghiệp.

Việc tham gia trên ứng dụng điện thoại là một cách hiệu quả để doanh nghiệp tăng tương tác và tạo sự gắn kết với khách hàng của mình. Nó giúp tạo ra một kênh tương tác thuận tiện và tiện lợi cho khách hàng

2.3.2. Khái niệm về phân tích hành vi người dùng rời đi trên ứng dụng di động

Người dùng : Là người sử dụng ứng dụng, dịch vụ hoặc sản phẩm.

Rời đi (Churn): Các hành động của người dùng để chấm dứt việc sử dụng sản phẩm hoặc dịch vụ của công ty. Điều này có thể bao gồm việc hủy đăng ký, chuyển sang sử dụng ứng dụng hoặc sản phẩm khác … trên ứng dụng di động.

Người dùng rời đi là tỷ lệ người sử dụng rời đi, không sử dụng sản phẩm hoặc dịch vụ của doanh nghiệp nữa trên ứng dụng di động trong một khoảng thời gian nhất định.

Phân tích hành vi người dùng trên ứng dụng di động: là phương pháp phân tích dữ liệu để hiểu rõ hơn về cách khách hàng sử dụng ứng dụng trên điện thoại di động. Các thông tin được thu thập từ phân tích hành vi người dùng trên ứng dụng di động bao gồm thời gian sử dụng, tần suất, tính năng được sử dụng nhiều nhất, thói quen sử

dụng và các tương tác với ứng dụng. Việc thu thập và phân tích dữ liệu phân tích hành vi người dùng trên ứng dụng di động cung cấp cho doanh nghiệp cái nhìn sâu sắc hơn về hành vi khách hàng, giúp cải thiện trải nghiệm sử dụng và tăng cường sự hài lòng của khách hàng.

CHƯƠNG 3: DỮ LIỆU

3.1. Enrollment

3.1.1 Giới thiệu Fintech

Công nghệ tài chính (*fintech*) là công nghệ mới và đổi mới nhằm cạnh tranh với các phương pháp tài chính truyền thống trong việc cung cấp các dịch vụ tài chính. Việc sử dụng điện thoại thông minh cho dịch vụ ngân hàng di động, dịch vụ đầu tư và tiền mã hóa là những ví dụ về công nghệ nhằm làm cho các dịch vụ tài chính dễ tiếp cận hơn với công chúng. Các công ty công nghệ tài chính bao gồm cả công ty mới thành lập và các công ty tài chính và công nghệ cố gắng thay thế hoặc tăng cường việc sử dụng các dịch vụ tài chính do các công ty tài chính hiện có cung cấp. FinTech là các ứng dụng mới, quy trình, sản phẩm hoặc mô hình kinh doanh trong ngành dịch vụ tài chính bao gồm một hoặc nhiều dịch vụ tài chính bổ sung và được cung cấp như là một quá trình kết thúc thông qua Internet.

3.1.2. Giới thiệu về bộ dữ liệu FinTech

Công ty Công nghệ Tài chính (Công ty Fin-Tech) ra mắt ứng dụng dành cho thiết bị di động. Ứng dụng này được sử dụng cho các mục đích tài chính như vay ngân hàng, tiết kiệm, v.v. ở một nơi. Nó có hai phiên bản miễn phí và cao cấp. Ứng dụng phiên bản miễn phí chứa các tính năng cơ bản và khách hàng muốn sử dụng tính năng cao cấp thì họ phải trả một số tiền để mở khóa.

Mục tiêu chính của công ty là bán ứng dụng phiên bản cao cấp với chi phí quảng cáo thấp nhưng họ không biết cách thực hiện. Đó là lý do họ được cung cấp tính năng cao cấp trong ứng dụng phiên bản miễn phí trong 24 giờ để thu thập hành vi của khách hàng. Sau đó, công ty đã thuê Kỹ sư máy học để tìm hiểu sâu sắc từ dữ liệu thu thập được (hành vi của khách hàng).

Công việc của kỹ sư ML là tìm hoặc dự đoán khách hàng mới có quan tâm mua sản phẩm hay không. Nếu khách hàng vẫn sẽ mua một sản phẩm thì không cần phải đưa ra lời đề nghị cho khách hàng đó và làm mất việc kinh doanh. Chỉ cung cấp ưu đãi cho những khách hàng muốn sử dụng ứng dụng phiên bản cao cấp nhưng họ không đủ

khả năng chi trả. Vì vậy, công ty sẽ cung cấp cho những khách hàng đó và kiếm được nhiều tiền hơn.

Bằng cách làm việc cho công ty. Chúng tôi có quyền truy cập vào dữ liệu hành vi ứng dụng của từng khách hàng. Dữ liệu này cho phép chúng tôi xem ngày và giờ cài đặt ứng dụng, cũng như các tính năng mà người dùng tương tác trong ứng dụng. Hành vi ứng dụng được đặc trưng bởi danh sách các màn hình ứng dụng mà người dùng đã xem và liệu người dùng có chơi các trò chơi nhỏ tài chính có sẵn hay không.

Dữ liệu sử dụng ứng dụng chỉ từ ngày đầu tiên của người dùng trong ứng dụng. Hạn chế này tồn tại bởi vì người dùng có thể tận hưởng bản dùng thử miễn phí 24 giờ đối với các tính năng cao cấp và công ty muốn nhắm mục tiêu họ bằng các ưu đãi mới ngay sau khi quá trình thử nghiệm kết thúc.

Dữ liệu cho dự án này là từ các lĩnh vực sản xuất dựa trên các xu hướng được tìm thấy trong các nghiên cứu điển hình trong thế giới thực. Các trường mô tả những gì công ty thường theo dõi từ người dùng của họ.

3.1.3 Mô tả dữ liệu

Trong bộ dữ liệu FineTech, có 50.000 dữ liệu người dùng với 12 tính năng khác nhau bao gồm:

- User: Đây là ID duy nhất của mỗi người dùng ứng dụng tham gia.
- First_open: đây là ngày/tháng/năm, thời điểm người dùng mở ứng dụng lần đầu tiên.
- Dayofweek: điều này cho thấy ngày trong số 7 ngày một tuần người dùng tham gia ứng dụng bao gồm: “0” :Chủ Nhật, “1” : Thứ Hai, “2” : Thứ Ba, “3” : Thứ Tư, “4” : Thứ Năm, “5”: Thứ Sáu, “6”: Thứ Bảy.
- Hour: Đây là ngoài 24 giờ trong ngày người dùng mở ứng dụng đầu tiên.
- Age: Đây là tuổi của người dùng.
- Screen_list: Phần này mô tả từng tên màn hình mà người dùng đã truy cập trong 24 giờ đầu tiên đó (tên màn hình được phân tách bằng dấu phẩy).
- Num_screen: Số lượng màn hình mà người dùng đã truy cập trong 24 giờ đầu tiên.

- Mini_game: Ứng dụng có tính năng trò chơi nhỏ, tính năng này cho biết người chơi có chơi bất kỳ trò chơi nhỏ nào không (“1”: Đã chơi, “0”: Chưa chơi).
- Liked: Có nút thích cho từng tính năng trong ứng dụng, cho biết người dùng có bấm vào nút thích của bất kỳ tính năng nào trong ứng dụng hay không (“1”: bấm vào nút thích, “0”: Không bấm vào).
- Used_premium_feature: Điều này cho biết liệu người dùng có sử dụng bất kỳ tính năng cao cấp nào (miễn phí trong 24 giờ đầu tiên) hay không trong 24 giờ đầu tiên (“1”: đã sử dụng, “0”: chưa sử dụng).
- Enroll: Đây là mục tiêu cho biết liệu người dùng có đăng ký trả phí sau khi dùng thử miễn phí hay không (“1”: đã đăng ký, “0”: chưa đăng ký).
- Enroll_date: ngày và thời gian đăng ký sản phẩm cao cấp nếu họ đăng ký cao cấp.

3.2. Churn

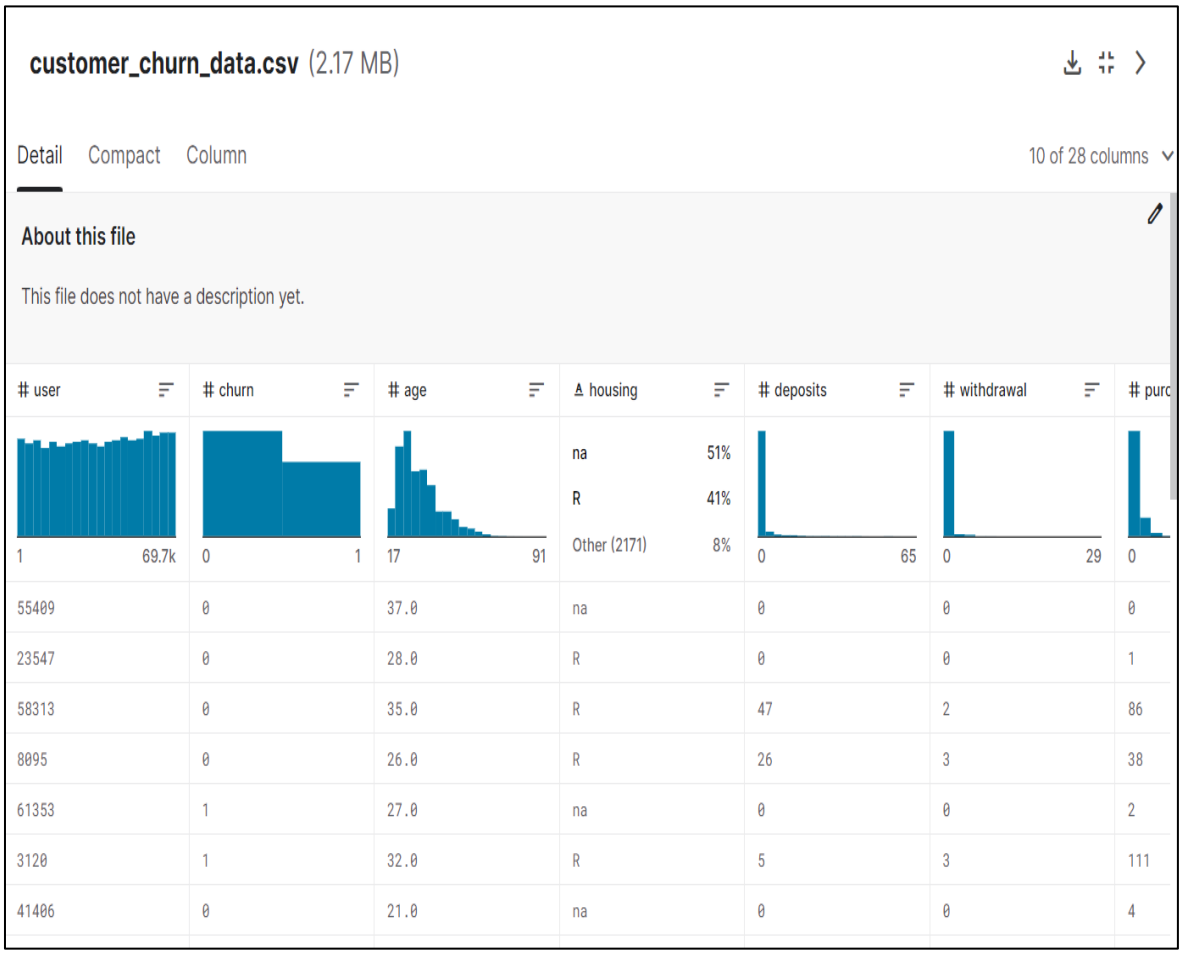
3.2.1. Dữ liệu Customer Churn

Bộ dữ liệu "customer-churn-data.csv" là một tập dữ liệu được sử dụng để nghiên cứu và phân tích hành vi của khách hàng trong lĩnh vực ứng dụng di động và các dịch vụ công nghệ thông tin liên quan. Bộ dữ liệu này thu thập thông tin về các khách hàng và các đặc trưng liên quan đến việc rời bỏ dịch vụ (churn), nhằm tìm hiểu và dự đoán các yếu tố ảnh hưởng đến việc khách hàng rời bỏ ứng dụng hoặc dịch vụ.

Bộ dữ liệu "customer-churn-data.csv" bao gồm các thông tin như ID khách hàng, độ tuổi, tình trạng nhà ở, số lần gửi/rút tiền, số lượng giao dịch, số lượng thẻ tín dụng, thông tin về việc sử dụng ứng dụng và các hệ điều hành (web, iOS, Android), số lượng điện thoại đăng ký, hình thức thanh toán, trạng thái chờ vay, hủy vay, nhận vay, từ chối vay, cung hoàng đạo, thời gian rời bỏ dịch vụ trong hai tháng liên tiếp và một tháng gần đây, tỷ lệ thưởng tích lũy và thông tin về việc được giới thiệu bởi người dùng khác.

Bộ dữ liệu này có thể được sử dụng để phân tích hành vi khách hàng, xác định các yếu tố ảnh hưởng đến việc rời bỏ dịch vụ, và xây dựng các mô hình dự đoán churn để giúp doanh nghiệp hiểu rõ hơn về khách hàng và đưa ra các biện pháp giữ

chân khách hàng hiệu quả (https://www.kaggle.com/datasets/canhlu/customer-churn-data) [28].



Hình 3.2.1: Ảnh chụp bộ dữ liệu Customer Churn

Mỗi hàng trong bảng đại diện cho một khách hàng khác nhau và mỗi cột trong bảng chứa thông tin về một số khía cạnh của khách hàng đó. Dữ liệu thô có tổng cộng 26996 hàng, đại diện cho khách hàng và 28 cột (đặc trưng). Cột "Churn" sẽ đóng vai trò là mục tiêu (Target).

3.2.2. Mô tả dữ liệu.

Có tất cả 28 đặc trưng.

Bảng 1: Mô tả dữ liệu Customer Churn

STT	Đặc trưng	Mô tả	Loại
1	User	Mã user duy nhất cho mỗi khách hàng	Integer
2	Churn	Khách hàng có rời đi hay không ("Có":1 hoặc "Không":0)	Integer
3	Age	Độ tuổi của khách hàng	Float
4	Housing	Tình trạng nhà ở của khách hàng (na: Không có thông tin, R: Thuê nhà, O: sở hữu nhà)	Object
5	Deposits	Số lần gửi tiền của khách hàng	Integer
6	Withdrawal	Số lần rút tiền của khách hàng	Integer
7	Purchases_partners	Số lượng giao dịch mà khách hàng đã thực hiện với các đối tác.	Integer
8	Purchases	Số lượng giao dịch mà khách hàng đã thực hiện.	Integer
9	CC_taken	Số lượng thẻ tín dụng mà khách hàng đã sử dụng.	Integer
10	CC_recommended	Số lượng thẻ tín dụng được khuyến nghị cho khách hàng.	Integer
11	CC_disliked	Số lượng thẻ tín dụng mà khách hàng không thích.	Integer
12	CC_liked	Số lượng thẻ tín dụng mà khách hàng thích	Integer
13	CC_application_begin	Số lượng đề nghị đăng ký thẻ tín dụng mà khách hàng đã bắt đầu.	Integer
14	App_downloaded	thể hiện liệu khách hàng đã tải xuống ứng dụng hay chưa. ("Có":1 hoặc "Không":0)	Integer
15	Web_user	Thể hiện liệu khách hàng đã truy cập trang web hay không. ("Có":1 hoặc "Không":0)	Integer
16	Ios_user	thể hiện liệu khách hàng sử dụng hệ điều hành iOS hay không ("Có":1 hoặc "Không":0)	Integer
17	Android_user	Thể hiện liệu khách hàng sử dụng hệ điều hành Android hay không. ("Có":1 hoặc "Không":0)	Integer

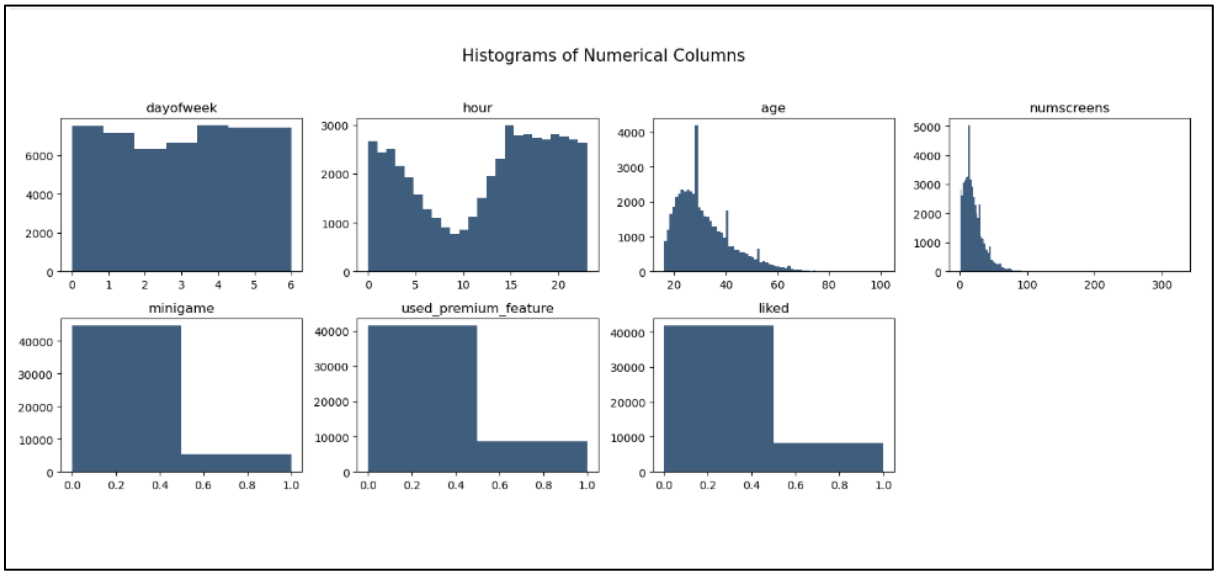
18	Registered_phones	Số lượng điện thoại đã đăng ký của khách hàng	Integer
19	Payment_type	Loại hình thanh toán được sử dụng bởi khách hàng	Object
20	Waiting_4_loan	thể hiện liệu khách hàng đang chờ vay hay không. ("Có":1 hoặc "Không":0)	Integer
21	Cancelled_loan	Thể hiện liệu khách hàng đã hủy vay hay không. ("Có":1 hoặc "Không":0)	Integer
22	Received_loan	Thể hiện liệu khách hàng đã nhận được vay hay không. ("Có":1 hoặc "Không":0)	Integer
23	Rejected_loan	Thể hiện liệu khách hàng đã bị từ chối vay	Integer
24	Zodiac_sign	Cung hoàng đạo của khách hàng (đây là một biến định tính có 13 giá trị khác nhau).	Object
25	Left_for_two_month_plus	Thể hiện liệu khách hàng đã rời bỏ dịch vụ trong ít nhất hai tháng liên tiếp hay không. ("Có":1 hoặc "Không":0)	Integer
26	Left_for_one_month	Thể hiện liệu khách hàng đã rời bỏ dịch vụ trong một tháng gần đây hay không. ("Có":1 hoặc "Không":0)	Integer
27	Reward_rate	Tỷ lệ thưởng tích lũy mà khách hàng đã nhận được từ các giao dịch hoặc hoạt động khác.	Float
28	Is_referred	Thể hiện liệu khách hàng đã được giới thiệu bởi người dùng khác hay không. ("Có":1 hoặc "Không":0)	Integer

CHƯƠNG 4: THỰC NGHIỆM VÀ KẾT QUẢ

4.1. Enrollment

4.1.1. Khám phá dữ liệu

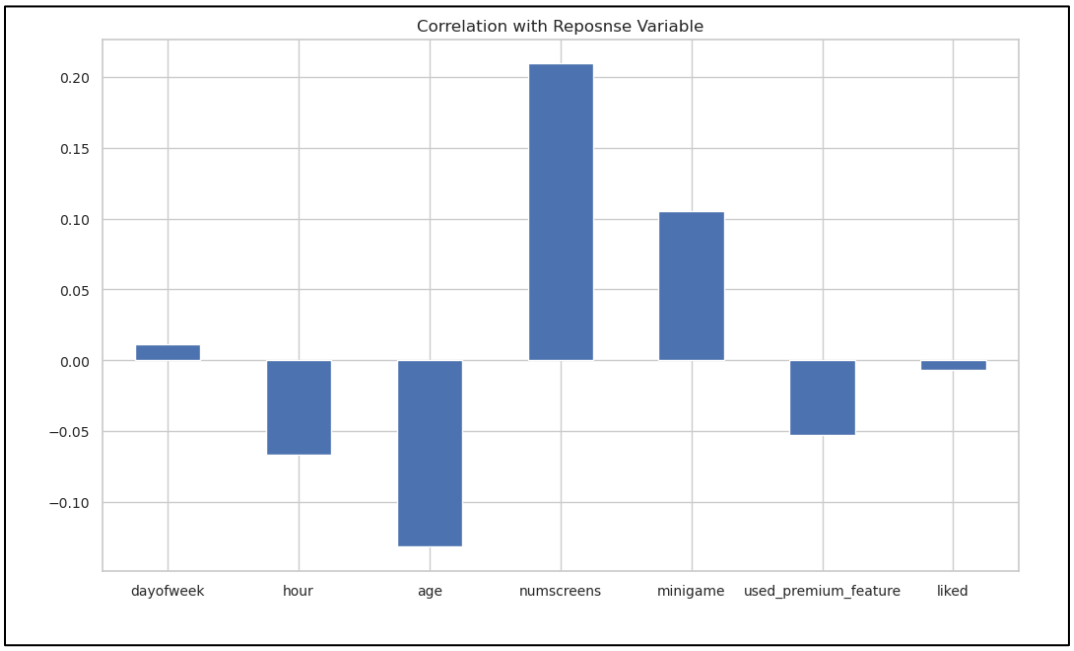
Trong bước khám phá trên bộ dữ liệu đã được thực hiện đã tìm kiếm các giá trị null, loại bỏ và vẽ đồ thị trực quan. Vẽ đồ thị để xem mối quan hệ giữa các đặc trưng phân loại khác nhau và khách hàng đăng kí trên cơ sở các đặc trưng đó.



Hình 4.1.1: Đồ thị histgram của các cột Numerical

Qua Hình 4.1.1. cho thấy được:

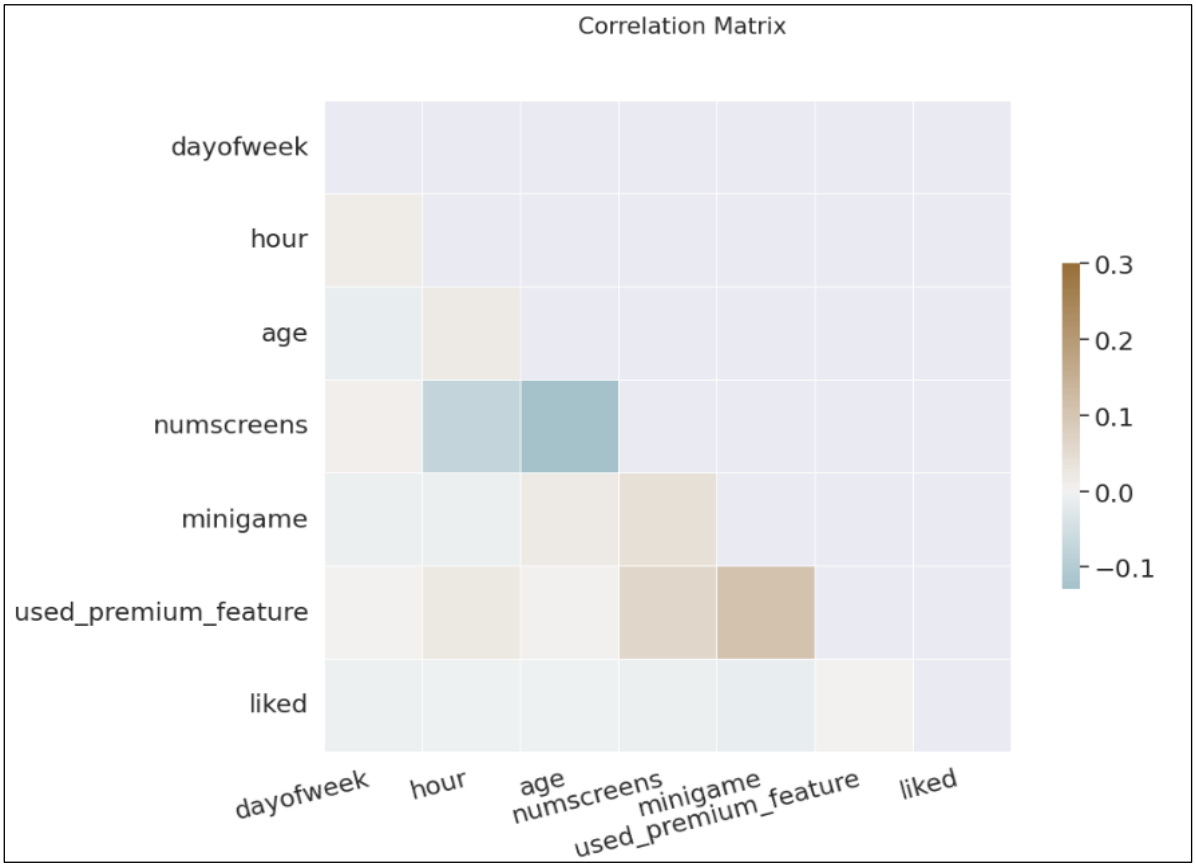
- Hầu hết người dùng tham gia ứng dụng vào cuối tuần
- Hầu hết người dùng tham gia ứng dụng vào cuối tuần
- Hầu hết người dùng lần đầu tiên mở ứng dụng vào khoảng 15 giờ
- Hầu hết người dùng ở độ tuổi khoảng 30 năm
- Hầu hết người dùng đã truy cập khoảng 20 màn hình của ứng dụng
- Không có nhiều người dùng chơi bất kỳ minigame nào
- Không nhiều người dùng nhấn nút thích
- Không có nhiều người dùng sử dụng tính năng cao cấp trong 24 giờ đầu tiên



Hình 4.1.2: Tương quan với biến phản hồi.

Hình 4.1.2 cho thấy được:

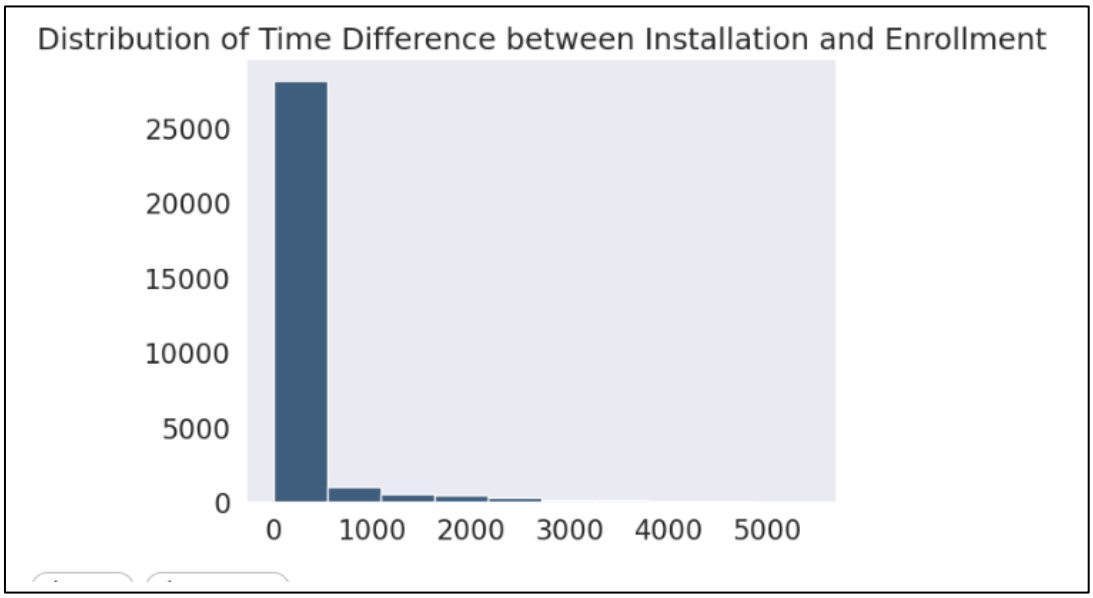
- Ngày trong tuần ít tương quan tích cực nhất nói rằng nếu bạn tham gia ứng dụng vào ngày 0 (chủ nhật) thì có nhiều khả năng được đăng ký các tính năng cao cấp.
- Giờ có tương quan nghịch với biến mục tiêu cho thấy giờ càng sớm (vào ban đêm) càng có nhiều khả năng được đăng ký.
- Độ tuổi cũng có mối tương quan nghịch phản ánh rằng những người dùng trẻ tuổi có nhiều khả năng đăng ký nhất.
- Numscreen tương quan thuận với mục tiêu cho thấy càng nhiều thì càng không. lượt truy cập của người dùng trên màn hình nhiều cơ hội được đăng ký hơn.
- Minigame cũng cho thấy rằng càng nhiều người dùng minigame chơi thì càng có nhiều cơ hội được ghi danh.
- Độ yêu thích là rất ít tiêu cực nhất mà không có nhiều tác động trong mục tiêu.
- Used_premium_feature có tương quan nghịch với phản hồi, nghĩa là nếu người dùng đã sử dụng tính năng cao cấp trong 24 giờ đầu tiên thì họ có thể không đăng ký sau phiên bản dùng thử của các tính năng cao cấp.



Hình 4.1.3: Đồ thị heatmap của ma trận tương quan giữa các biến.

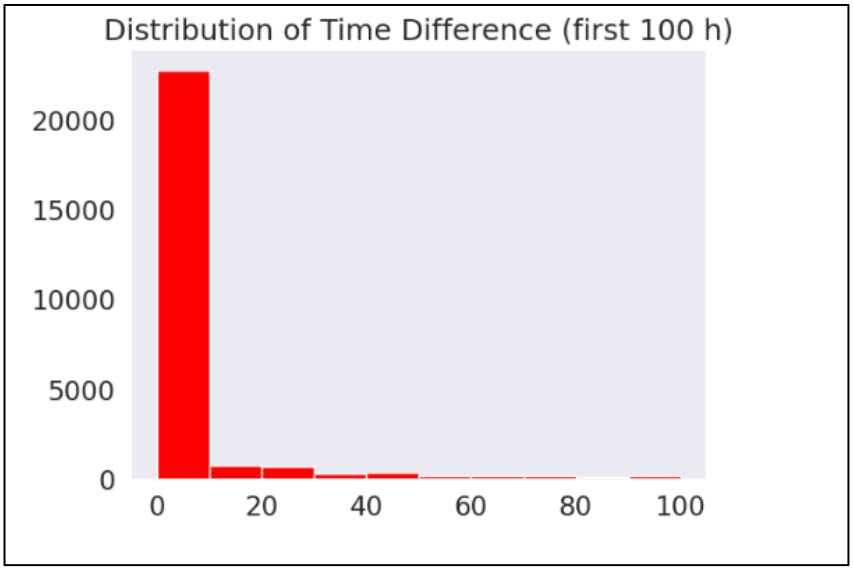
Không có mối tương quan chặt chẽ giữa bất kỳ tính năng. Có rất ít mối tương quan giữa ‘numscreen' và 'enrolled'. Điều đó có nghĩa là những khách hàng đó đã nhìn thấy nhiều màn hình hơn mà họ đang sử dụng ứng dụng cao cấp. Có một mối tương quan nhỏ giữa 'minigame' với 'enrolled' và 'used_premium_feature'. Mối tương quan hơi tiêu cực giữa 'age' với 'enrolled' và numscreens. Điều đó có nghĩa là những khách hàng lớn tuổi không sử dụng ứng dụng cao cấp và họ không nhìn thấy nhiều màn hình.

Tìm kiếm xem khách hàng mất bao nhiêu thời gian để đăng ký ứng dụng tính năng cao cấp sau khi đăng ký



Hình 4.1.4: Phân phối chênh lệch thời gian giữa cài đặt và đăng ký

Tìm kiếm xem khách hàng mất bao nhiêu thời gian để đăng ký ứng dụng tính năng cao cấp sau khi đăng ký trong 100 giờ đầu tiên



Hình 4.1.5: Phân phối chênh lệch thời gian giữa cài đặt và đăng ký trong 100 giờ đầu tiên

Như vậy, số lượng khách hàng tối đa đã đăng ký ứng dụng trong 10 giờ kể từ khi đăng ký.

"Screen_list" sẽ giúp xác định hành vi của người dùng là gì và họ dành bao nhiêu thời gian trên màn hình thông qua một bộ dữ liệu là “top_screen.csv”. Sau đó

kiểm tra xem tên màn hình này có sẵn trong ‘screen_list’ nếu có rồi thêm giá trị 1 khác 0 vào cột được nối thêm.

```
array(['Loan2', 'location', 'Institutions', 'Credit3Container',
       'VerifyPhone', 'BankVerification', 'VerifyDateOfBirth',
       'ProfilePage', 'VerifyCountry', 'Cycle', 'idscreen',
       'Credit3Dashboard', 'Loan3', 'CC1Category', 'Splash', 'Loan',
       'CC1', 'RewardsContainer', 'Credit3', 'Credit1', 'EditProfile',
       'Credit2', 'Finances', 'CC3', 'Saving9', 'Saving1', 'Alerts',
       'Saving8', 'Saving10', 'Leaderboard', 'Saving4', 'VerifyMobile',
       'VerifyHousing', 'RewardDetail', 'VerifyHousingAmount',
       'ProfileMaritalStatus', 'ProfileChildren ', 'ProfileEducation',
       'Saving7', 'ProfileEducationMajor', 'Rewards', 'AccountView',
       'VerifyAnnualIncome', 'VerifyIncomeType', 'Saving2', 'Saving6',
       'Saving2Amount', 'Saving5', 'ProfileJobTitle', 'Login',
       'ProfileEmploymentLength', 'WebView', 'SecurityModal', 'Loan4',
       'ResendToken', 'TransactionList', 'NetworkFailure', 'ListPicker'],
      dtype=object)
```

Hình 4.1.6:Các giá trị chuỗi trong Screen_list

Như vậy, bộ dữ liệu hiện tại có tổng cộng là 68 cột, bao gồm bộ dữ liệu gốc là FinTech_appdata.csv và top_screen.csv

	user	dayofweek	hour	age	screen_list	numscreens	...	SecurityModal	Loan4	R
0	235136	3	2	23	joinscreen,product_review,ScanPreview,VerifyTo...	15	...	0	0	0
1	333588	6	1	24	joinscreen,product_review,product_review2,Scan...	13	...	0	0	0
2	254414	1	19	23		3	...	0	0	0
3	234192	4	16	28	product_review,Home,product_review,ReferralCon...	40	...	0	0	0
4	51549	1	18	31	joinscreen,ScanPreview,VerifySSN,Home,SelectIn...	32	...	0	0	0

Hình 4.1.7: Bộ dữ liệu mới bao gồm 68 cột

Tất cả các saving_screen tương quan với nhau, nghĩa là lấy tổng của tất cả các saving_screen trong mỗi hàng và lưu trữ trong một hàng cho tất cả khách hàng. Tương tự với credit_screens và loan_screens

Sau khi lọc 3 cột “saving_screen”,”credit_screen” và “loan_screen” có tổng cộng là 52 cột

4.1.2 Huấn luyện mô hình

Giải thích về AUC và ROC Curve

- Điểm AUC (Diện tích dưới đường cong) là số liệu được sử dụng để đánh giá hiệu suất của mô hình phân loại nhị phân. Nó biểu thị khu vực bên dưới đường cong Đặc tính hoạt động của máy thu (ROC), là biểu đồ của Tỷ lệ dương thực (TPR) so với Tỷ lệ dương sai (FPR) ở các cài đặt ngưỡng khác nhau.

- Đường cong ROC (Đặc tính hoạt động của máy thu) là biểu diễn đồ họa về hiệu suất của mô hình phân loại nhị phân. Nó là đồ thị của Tỷ lệ dương thực (TPR) so với Tỷ lệ dương sai (FPR) ở các cài đặt ngưỡng khác nhau. Trong một vấn đề phân loại nhị phân, mô hình đưa ra dự đoán cho từng đầu vào và đầu ra là dương hoặc âm. Bằng cách thay đổi ngưỡng mà tại đó chúng tôi phân loại đầu vào là dương hoặc âm, thu được các giá trị TPR và FPR khác nhau. TPR biểu thị tỷ lệ mẫu dương tính thực sự được phân loại chính xác là dương tính, trong khi FPR biểu thị tỷ lệ mẫu âm tính được phân loại không chính xác là dương tính.

a) Tiền xử lý dữ liệu

Dữ liệu được chia thành 2 tập là training set và testing set với tỷ lệ train | test là 0.8 | 0.2

Sử dụng StandardScaler để làm chuẩn các tính năng để tránh ưu thế của tính năng cụ thể trong toàn bộ mô hình

	dayofweek	hour	age	numscreens	minigame	used_premium_feature	...	NetworkFailure	ListPicker	Other	SavingCount	CMCount	LoansCount
20330	2	20	32	5	0	1	...	0	0	2	0	0	2
17532	1	21	22	42	1	0	...	0	0	13	0	1	1
45819	0	4	26	85	1	0	...	0	0	10	0	2	2
34807	4	13	25	24	1	0	...	0	0	10	0	0	1
31888	3	16	50	11	0	0	...	0	0	3	0	2	0

Hình 4.1.8: Sử dụng StandardScaler để làm chuẩn tính năng

b) Huấn luyện mô hình

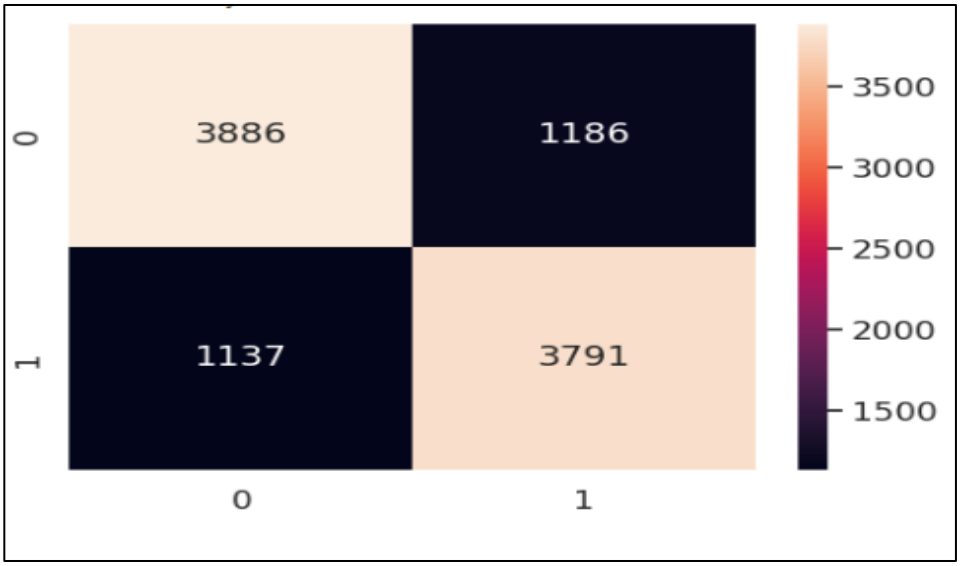
- Mô hình hồi quy Logistic

Sử dụng hệ số phạt 'L2' vì có thể có các tính năng tương quan như màn hình và 'L2' xử phạt bất kỳ trường nào tương quan chặt chẽ với biến phản hồi

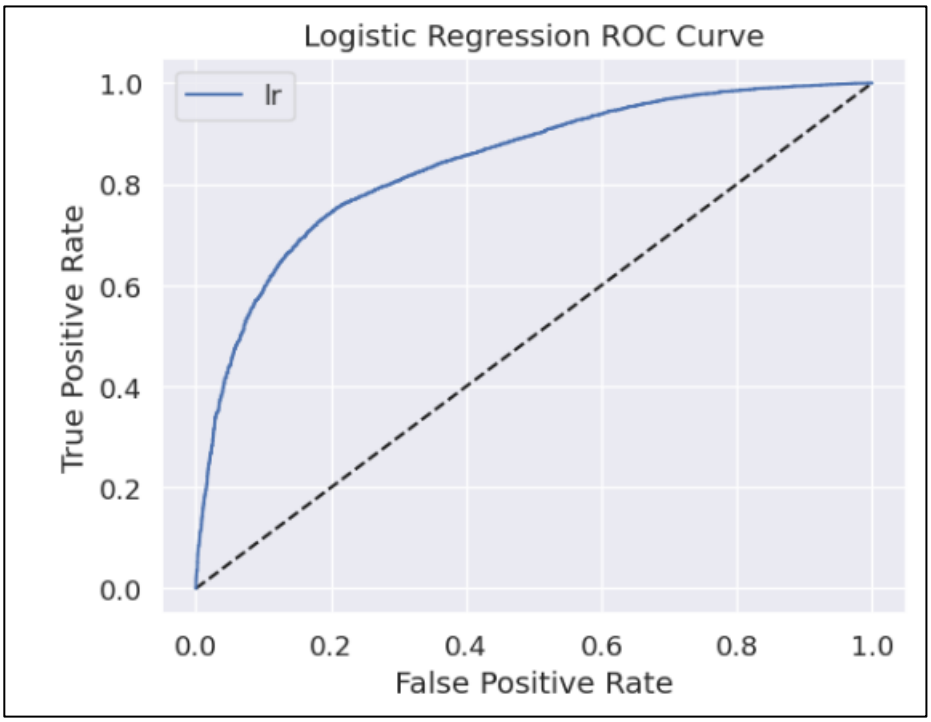
```
Classification Report:
              precision    recall  f1-score   support

           0       0.77      0.77      0.77      5072
           1       0.76      0.77      0.77      4928

    accuracy                           0.77     10000
   macro avg       0.77      0.77      0.77     10000
weighted avg       0.77      0.77      0.77     10000

accuracy_socre:  0.7677
precision_score:  0.7617038376532047
recall_score:  0.7692775974025974
f1_score:  0.7654719838465421
```

Hình 4.1.9: Chỉ số đánh giá phân loại của Hồi quy Logistic

- AUC: 0.84, cho thấy khả năng phân loại của mô hình là tương đối tốt.
- Precision (độ chính xác): 0.73, cho biết trong các khách hàng được dự đoán là không tham gia
- Recall (độ phủ): 0.77, cho biết mô hình có thể xác định được 77% khách hàng thực sự bỏ đi.
- F1-score: 0.76, cho biết mô hình có độ chính xác và độ phủ cân đối.
- Accuracy: 0.765, cho biết tỉ lệ dự đoán đúng của mô hình là 76,5%.



Hình 4.1.10: Chỉ số ma trận của Hồi quy Logistic



Hình 4.1.11: Đường cong ROC của mô hình hồi quy Logistic

- Precision của phân lớp 0 là cao hơn phân lớp 1, ngược lại recall phân lớp 1 cao hơn phân lớp 0 .Tuy nhiên giá trị f1-score của hai lớp này gần như bằng nhau. Điều này cho thấy mô hình có độ chính xác cao trong việc phân loại các khách hàng ở lớp 0, nhưng lại có thể bỏ sót một số khách hàng quan trọng ở phân lớp 1.

- AUC của mô hình là 0.86, cho thấy mô hình có khả năng phân loại các điểm dữ liệu đúng lớp là class 0 hay class 1 khá tốt. Nên mô hình có khả năng phân loại tốt hơn so với việc dự đoán ngẫu nhiên.

- Mô hình KNN:

Sử dụng n_neighbors là 15 và hệ số đo mặc định là “minkowski”

	precision	recall	f1-score	support
0	0.72	0.72	0.72	5072
1	0.71	0.71	0.71	4928
accuracy			0.72	10000
macro avg	0.72	0.72	0.72	10000
weighted avg	0.72	0.72	0.72	10000

Hình 4.12: Chỉ số đánh giá phân loại của KNN

- Precision (độ chính xác): KNN đạt được độ chính xác khoảng 72% cho phân lớp 0 và 71% cho phân lớp 1.
- Recall (độ phủ): KNN đạt được độ phủ khoảng 72% cho phân lớp 0 và 71% cho phân lớp 1.
- F1-score (trung bình điều hòa của Precision và Recall): KNN đạt được F1-score khoảng 72% cho phân lớp 0 và 71% cho phân lớp 1.
- Accuracy (độ chính xác toàn bộ mô hình): KNN đạt được độ chính xác khoảng 72%.
- AUC của mô hình KNN là 0.78

Mô hình Naïve Bayes

	precision	recall	f1-score	support
0	0.70	0.74	0.72	5072
1	0.72	0.67	0.69	4928
accuracy			0.71	10000
macro avg	0.71	0.71	0.71	10000
weighted avg	0.71	0.71	0.71	10000

Hình 4.1.13 Chỉ số đánh giá phân loại của Naïve Bayes

- Precision: Naïve Bayes đạt được độ chính xác khoảng 70% cho phân lớp 0 và 72% cho phân lớp 1.
- Recall: Naïve Bayes đạt được độ phủ khoảng 74% cho phân lớp 0 và 67% cho phân lớp 1.
- F1-score: Naïve Bayes đạt được F1-score khoảng 72% cho phân lớp 0 và 69% cho phân lớp 1.
- Accuracy: Naïve Bayes đạt được độ chính xác khoảng 71%.
- AUC của mô hình Naïve Bayes là 0.75

Mô hình SVM

	precision	recall	f1-score	support
0	0.77	0.75	0.76	5072
1	0.75	0.77	0.76	4928
accuracy			0.76	10000
macro avg	0.76	0.76	0.76	10000
weighted avg	0.76	0.76	0.76	10000

- Precision: SVM đạt được độ chính xác khoảng 77% cho phân lớp 0 và 75% cho phân lớp 1.
- Recall: SVM đạt được độ phủ khoảng 75% cho phân lớp 0 và 77% cho phân lớp 1.
- F1-score: SVM đạt được F1-score của phân lớp 0 và phân lớp 1 đều bằng 76%
- Accuracy: SVM đạt được độ chính xác khoảng 76%.

Mô hình Rừng ngẫu nhiên (RF)

Mô hình rừng ngẫu nhiên sử dụng số lượng cây mặc định là 100 và tiêu chuẩn là entropy

	precision	recall	f1-score	support
0	0.77	0.78	0.78	5072
1	0.77	0.76	0.77	4928
accuracy			0.77	10000
macro avg	0.77	0.77	0.77	10000
weighted avg	0.77	0.77	0.77	10000

Hình 4.1.14: Chỉ số đánh giá phân loại của Random Forest

- Precision: RF đạt được độ chính xác của phân lớp 0 và phân lớp 1 đều bằng 77%
- Recall: RF đạt được độ phủ khoảng 78% cho phân lớp 0 và 76% cho phân lớp 1.

- F1-score: RF đạt được F1-score khoảng 78% cho phân lớp 0 và 67% cho phân lớp 1.
- Accuracy: RF đạt được độ chính xác khoảng 77%.
- AUC của mô hình RF là 0.857

Mô hình XGBoost

Mô hình XGBoost sử dụng những tham số bao gồm:

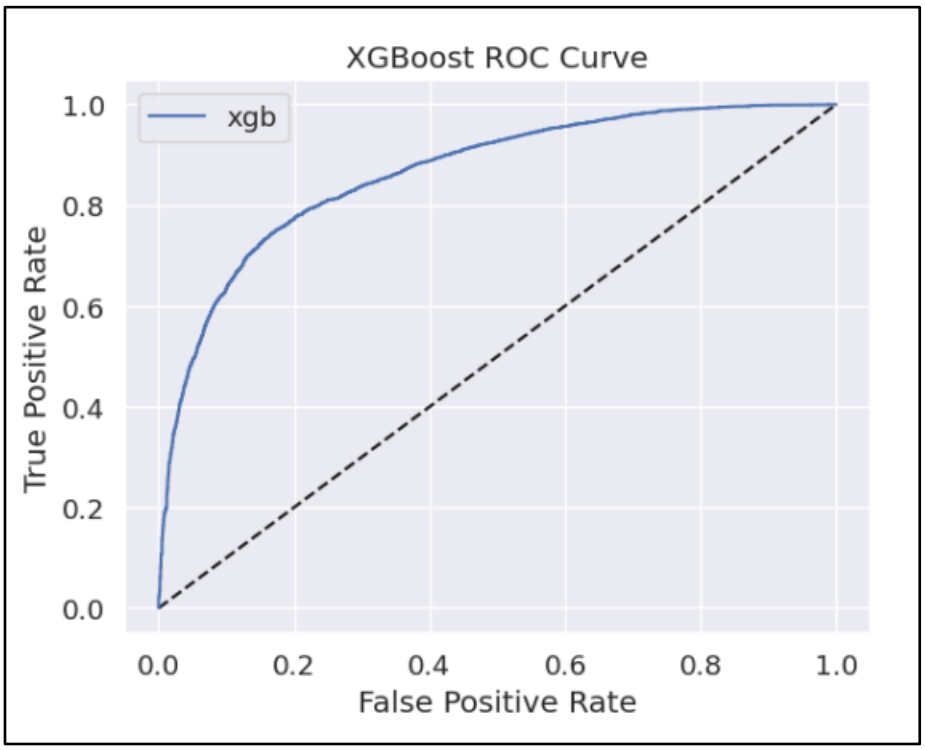
- learning_rate =0.01,
- n_estimators=5000,
- max_depth=4,
- min_child_weight=6,
- gamma=0,
- subsample=0.8,
- colsample_bytree=0.8,
- reg_alpha=0.005,
- objective= 'binary:logistic',
- nthread=4,
- scale_pos_weight=1,
- seed=27

	precision	recall	f1-score	support
0	0.77	0.82	0.80	5072
1	0.80	0.75	0.78	4928
accuracy			0.79	10000
macro avg	0.79	0.79	0.79	10000
weighted avg	0.79	0.79	0.79	10000

Hình 4.1.15: Chỉ số đánh giá phân loại XGBoost

- Precision: XGBoost đạt được độ chính xác của phân lớp 0 là 77% và phân lớp 1 là 80%

- Recall: XGBoost đạt được độ phủ khoảng 82% cho phân lớp 0 và 75% cho phân lớp 1.
- F1-score: XGBoost đạt được F1-score khoảng 80% cho phân lớp 0 và 78% cho phân lớp 1.
- Accuracy: XGBoost đạt được độ chính xác khoảng 79%.



Hình 4.1.16: Đường cong ROC của XGBoost

Mô hình XGBoostin có AUC tương đối cao là 0.86

- Kết quả**:**

	user	enrolled	predicted_results
0	239786	1	1
1	279644	1	1
2	98290	0	0
3	170150	1	1
4	237568	1	0
...	...	...	...
9995	143036	1	1
9996	91158	1	1
9997	248318	0	0
9998	142418	1	1
9999	279355	1	0

Hình 4.1.17: Kết quả dự đoán của mô hình phân loại XGBoost

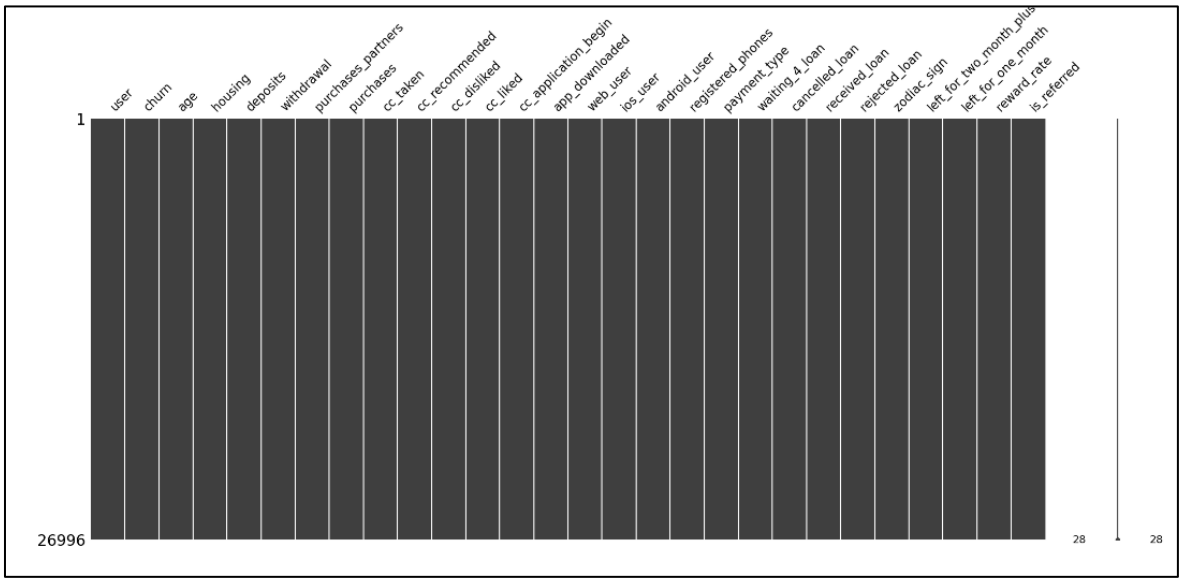
	user	enrolled	predicted_results
0	239786	1	1
1	279644	1	1
2	98290	0	0
3	170150	1	1
4	237568	1	1
...	...	...	...
9995	143036	1	0
9996	91158	1	1
9997	248318	0	0
9998	142418	1	1
9999	279355	1	0

Hình 4.1.18: Kết quả dự đoán của mô hình phân loại hồi quy Logistic

4.2. Churn

4.2.1. Khám phá và phân tích dữ liệu (EDA)

Trong bước khám phá trên bộ dữ liệu đã được thực hiện đã tìm kiếm các giá trị null, loại bỏ và vẽ đồ thị trực quan. Vẽ đồ thị để xem mối quan hệ giữa các đặc trưng phân loại khác nhau và khách hàng rời đi trên cơ sở các đặc trưng đó.



Hình 4.2.4: Sử dụng thư viện missingno để phát hiện dữ liệu bị thiếu hoặc null.

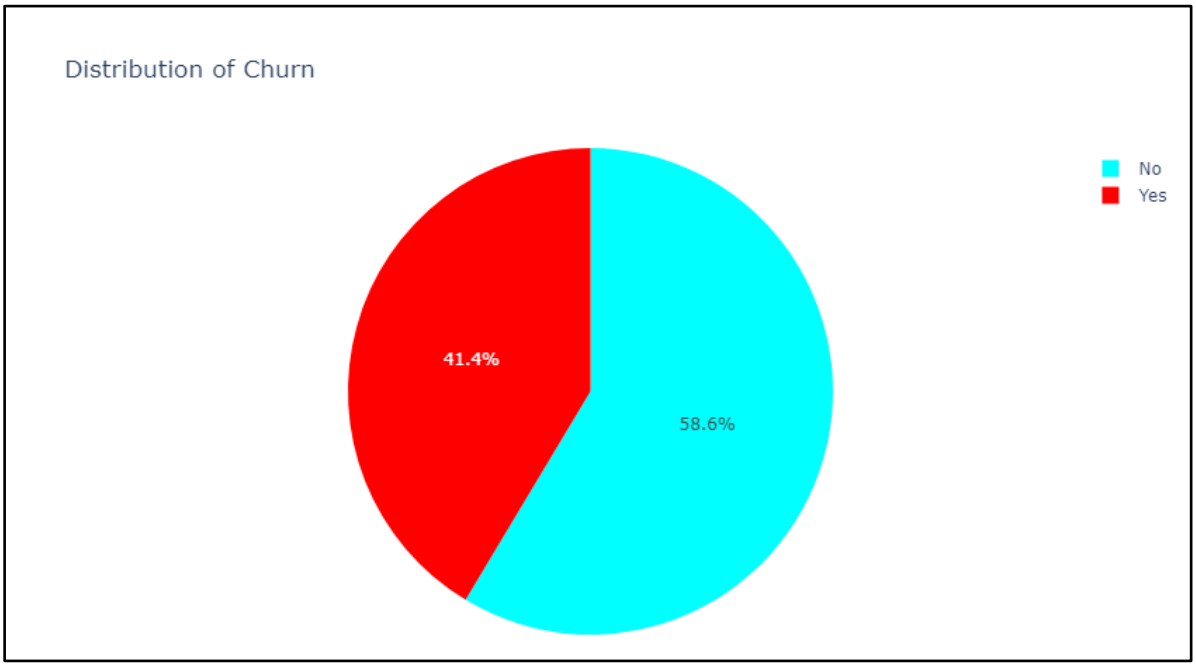
Hình 4.2.4 cho thấy các dữ liệu trong bộ dữ liệu đã được xử lý dữ liệu thiếu và null.

	Variable	Cutoff	N	Events	% of Events	Non-Events	% of Non-Events	WoE	IV
0	user	(0.999, 7001.0]	2700	1067	0.095490	1633	0.103211	0.077756	0.000600
1	user	(14183.0, 21438.5]	2699	988	0.088420	1711	0.108141	0.201339	0.003971
2	user	(21438.5, 28544.0]	2700	997	0.089225	1703	0.107635	0.187584	0.003453
3	user	(28544.0, 35745.5]	2699	1061	0.094953	1638	0.103527	0.086452	0.000741
4	user	(35745.5, 42878.0]	2700	1126	0.100770	1574	0.099482	-0.012863	0.000017
..	...	...	...	...	...	...	...	...	...
6	reward_rate	(1.33, 1.7]	2633	872	0.078038	1761	0.111301	0.355036	0.011809
7	reward_rate	(1.7, 2.03]	2696	807	0.072221	1889	0.119391	0.502667	0.023711
8	reward_rate	(2.03, 4.0]	2679	749	0.067031	1930	0.121982	0.598725	0.032901
0	is_referred	0	18411	8136	0.728119	10275	0.649412	-0.114397	0.009004
1	is_referred	1	8585	3038	0.271881	5547	0.350588	0.254246	0.020011

[122 rows x 9 columns]

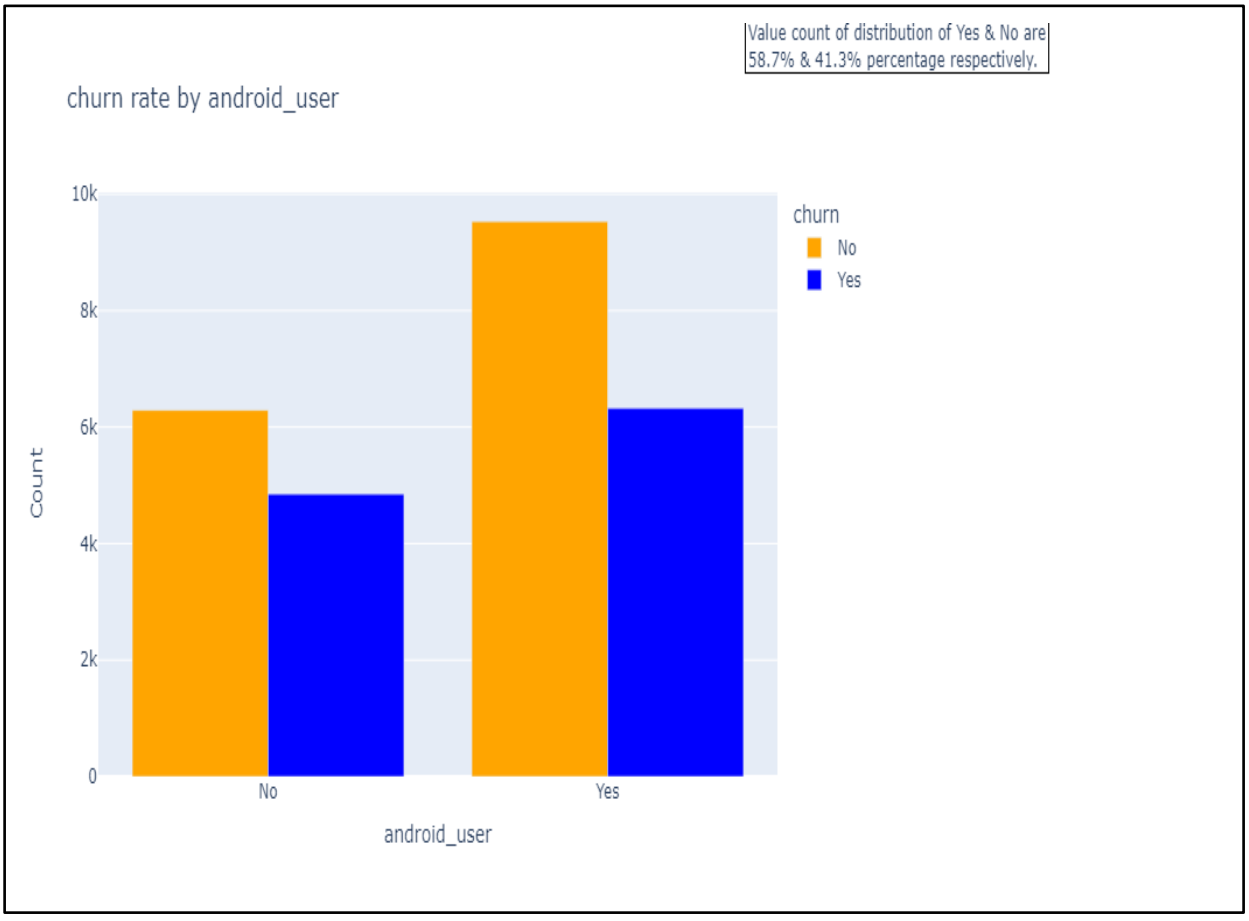
Hình 4.2.5: Tính WoE và IV cho bộ dữ liệu.

Hình 4.2.5 cho thấy các giá trị WOE cao thì đặc trưng có ảnh hưởng tích cực (hoặc tiêu cực) đến biến mục tiêu (churn) và có quan trọng cao trong mô hình. Gía trị IV cao (trên 0.5) cho thấy một đặc trưng quan trọng trong việc phân loại, dự đoán biến mục tiêu(churn) và là quan trọng trong mô hình.

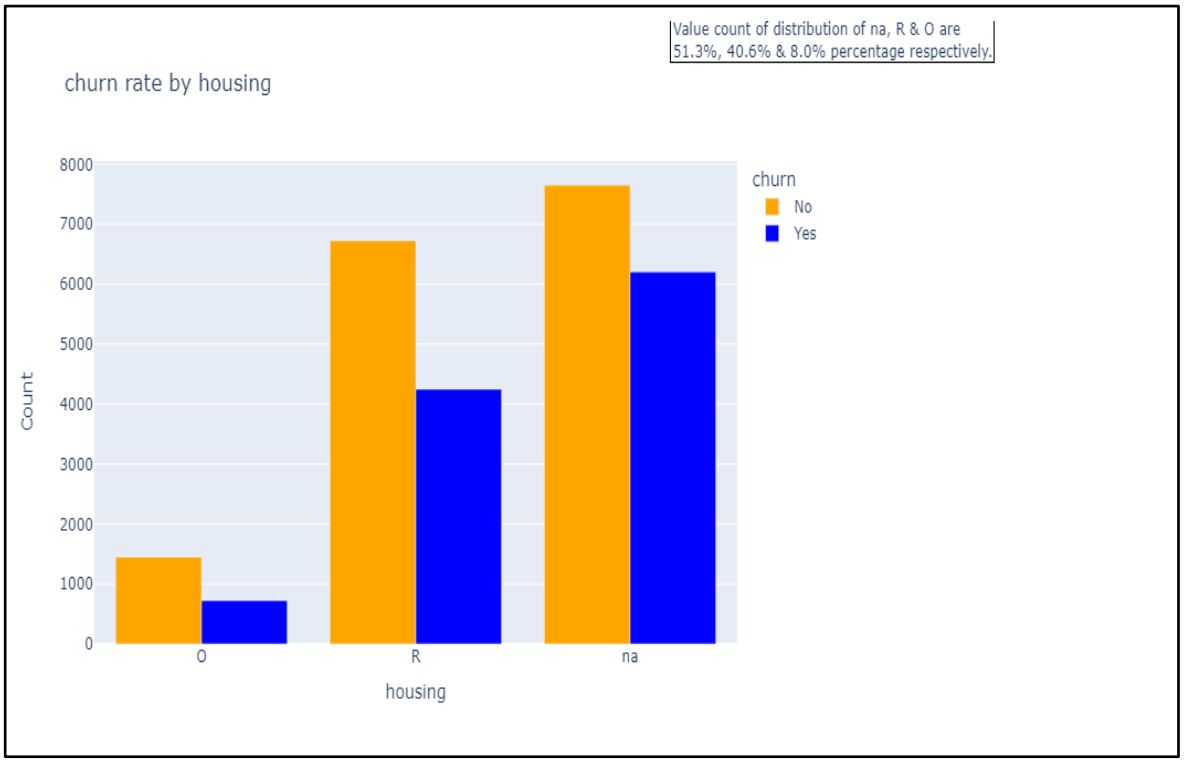


Hình 4.2.6: Phân phối của đặc trưng “Churn”

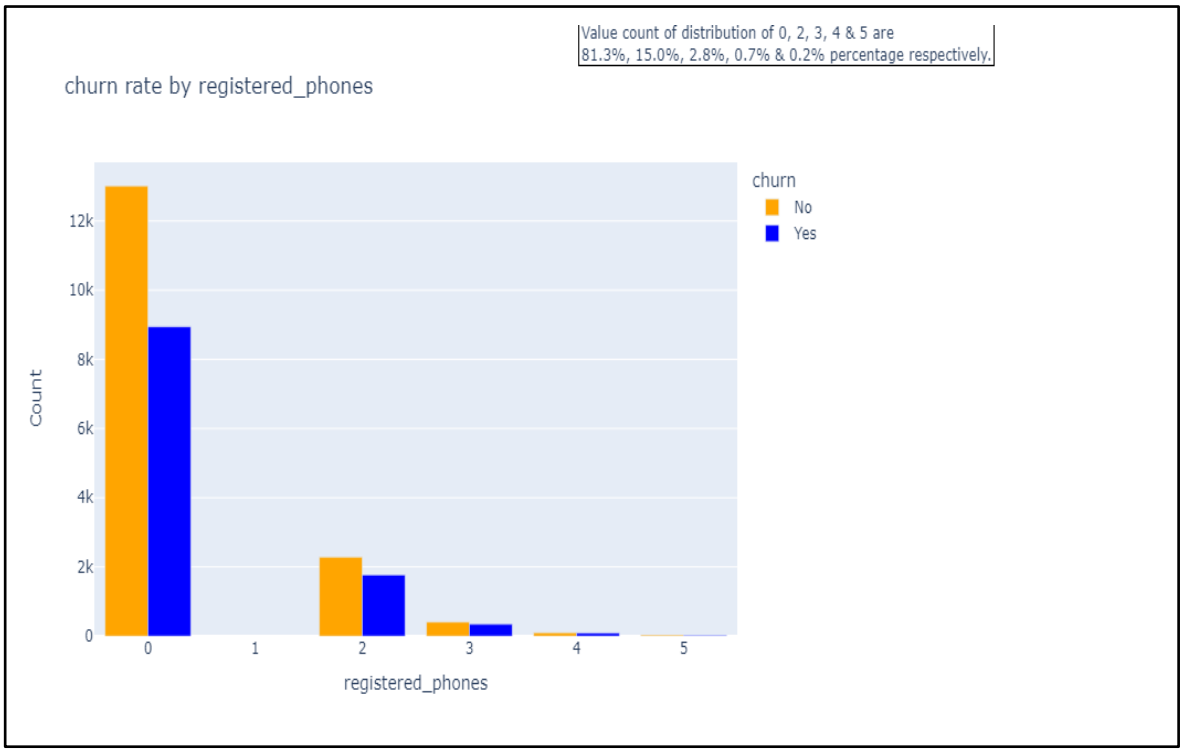
Hình 4.2.6 cho thấy tỉ lệ churn và không churn lần lượt chiếm 41,1% và 58,6%.



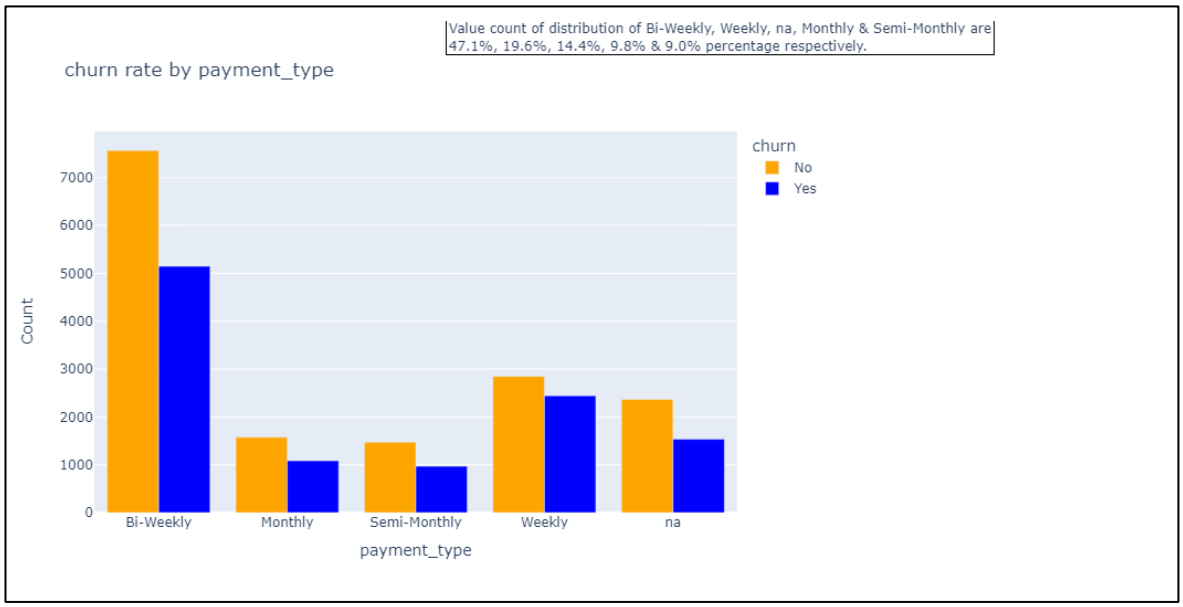
Hình 4.2.7: Tỉ lệ Churn với android_user



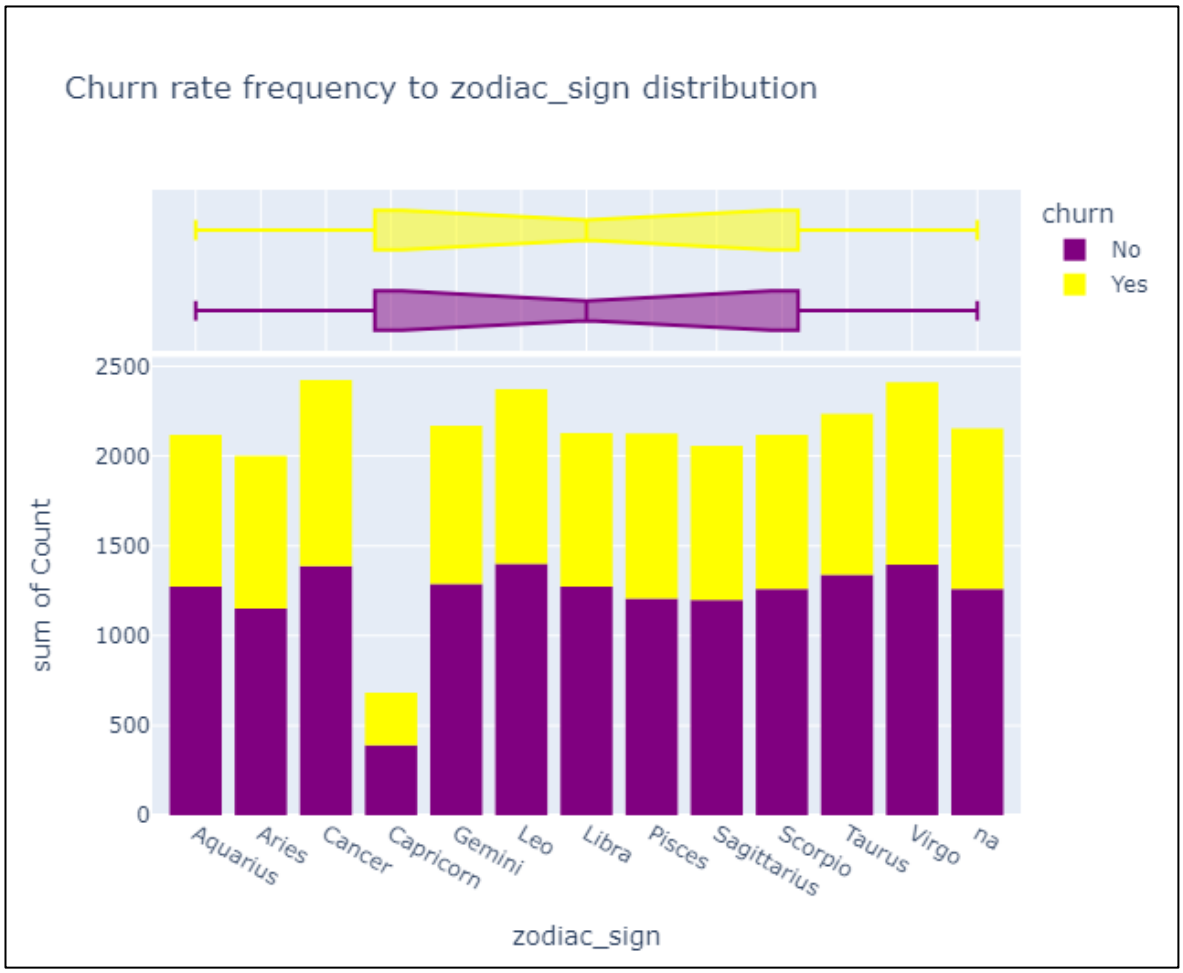
Hình 4.2.8: Tỉ lệ Churn với housing



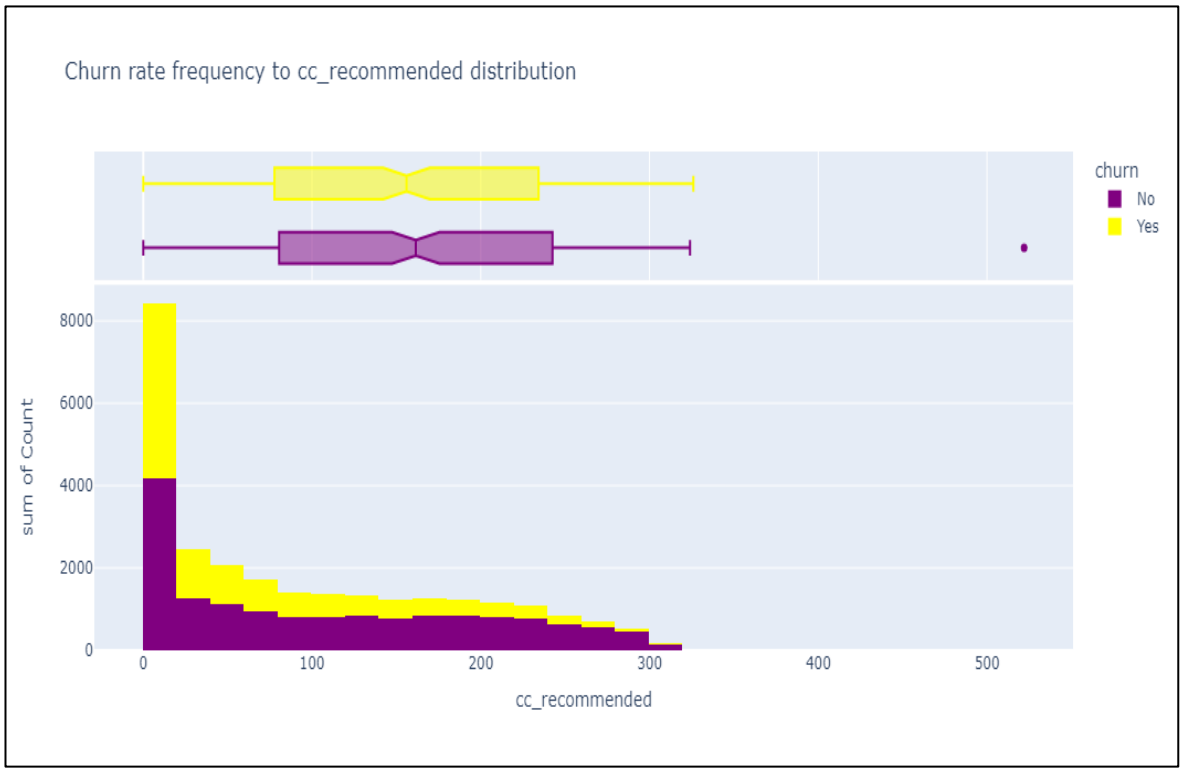
Hình 4.2.9: Tỉ lệ Churn với registered_phones



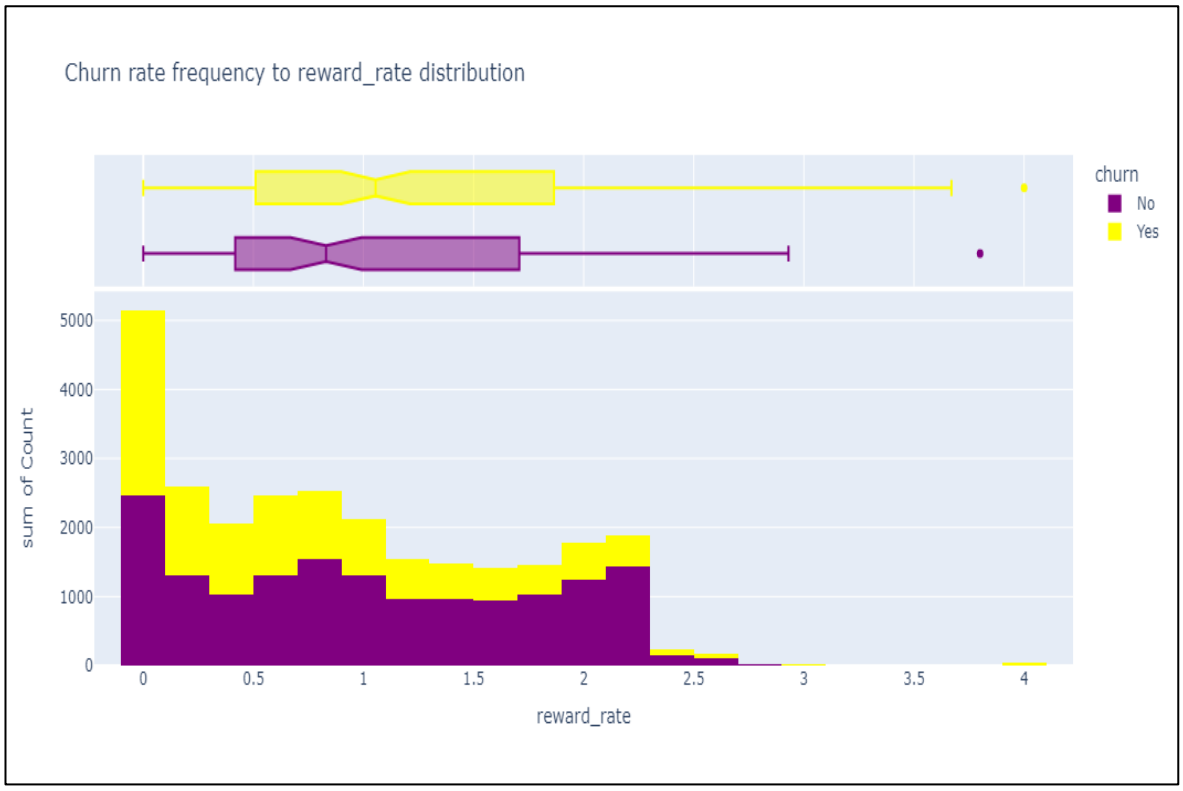
Hình 4.2.10: Tỉ lệ Churn với payment_type



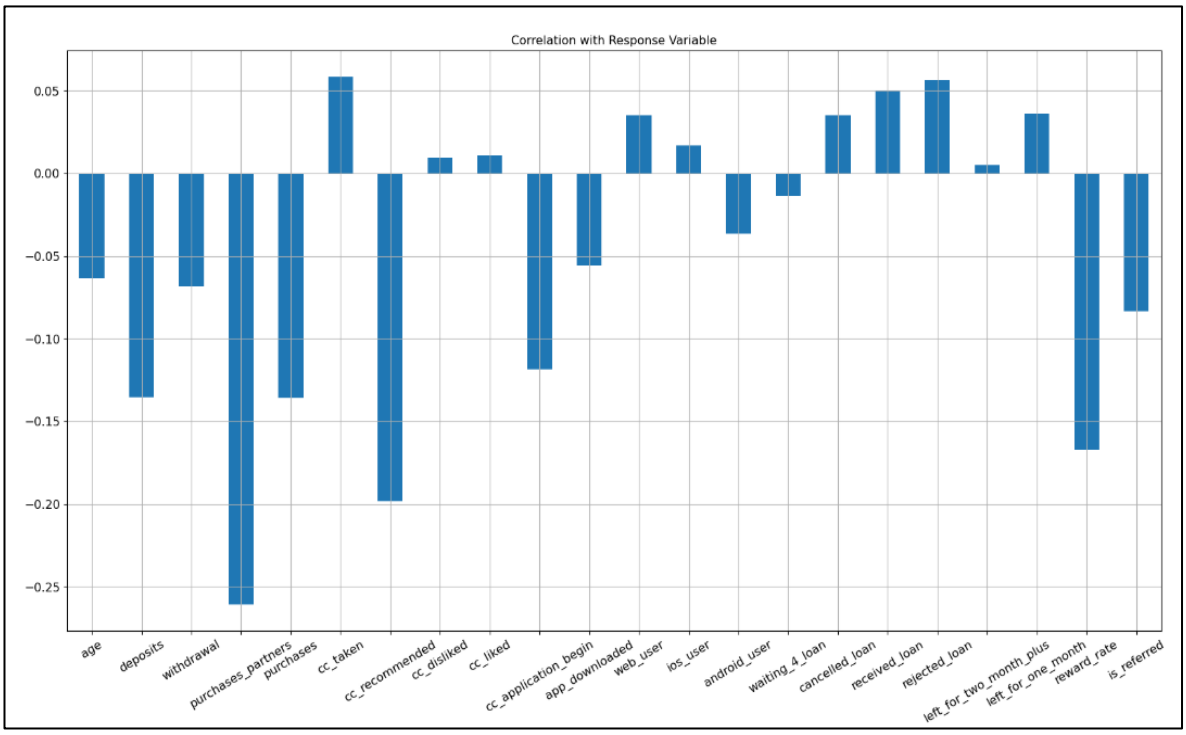
Hình 4.2.11: Phân phối tần suất của tỷ lệ churn đến zodiac_sign



Hình 4.2.12: Phân phối tần suất của tỷ lệ churn đến cc_recommended

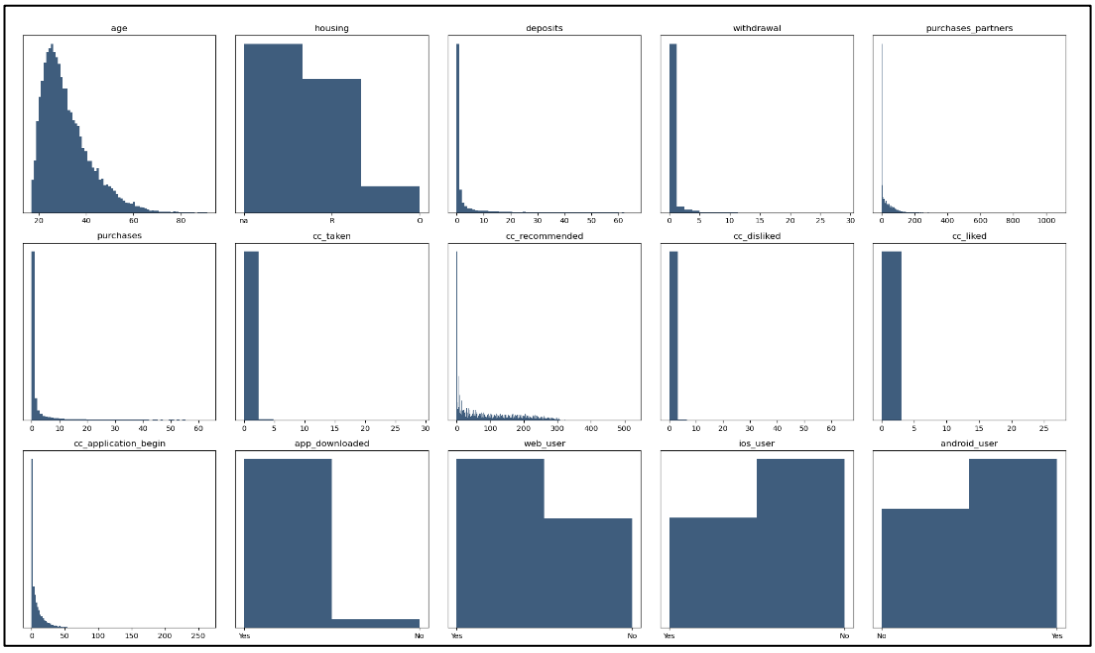


Hình 4.2.13: Phân phối tần suất của tỷ lệ churn đến reward_rate

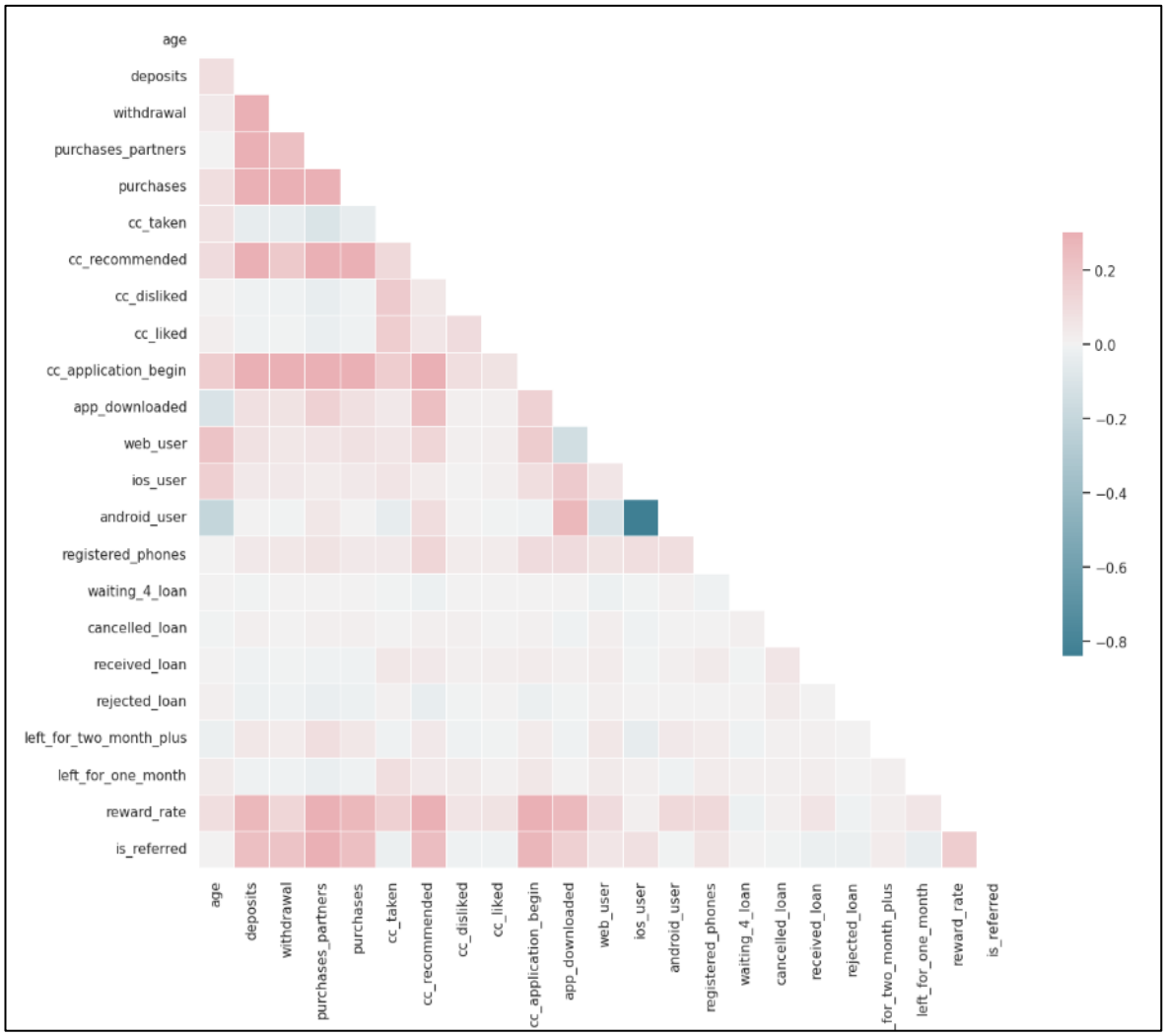


Hình 4.2.14: Tương quan với biến phản hồi.

Hình 4.2.14 cho thấy biến 'purchases_partners', 'cc_recomended', 'reward_rate' thể hiện sự tương quan nghịch lớn nhất, khi các biến trên càng tăng thì tỉ lệ 'churn' càng giảm. Ngược lại, các biến 'cc_taken', 'rejected_loan', 'received_loan' thể hiện mối tương quan thuận lớn nhất, các biến càng tăng thì tỉ lệ churn càng tăng. Các biến tương quan âm với biến phản hồi cho thấy mối tương quan nghịch, khi một biến tăng thì tỉ lệ biến 'churn' giảm và ngược lại.



Hình 4.2.15: Đồ thị histogram của các cột dữ liệu số



Hình 4.2.16: Đồ thị heatmap của ma trận tương quan giữa các biến.

Hình 4.2.16 cho thấy mối tương quan dương và âm: Các ô màu sáng trong heatmap thể hiện mối tương quan dương (tăng cùng nhau) giữa các cặp biến, trong khi các ô màu tối (xám) thể hiện mối tương quan âm (tăng ngược nhau). Sự khác biệt màu sắc giữa các ô cho thấy mức độ mạnh yếu của tương quan. Các cặp biến ‘android_user’ và ‘ios_user’ thể hiện các cặp biến không có tương quan tuyến tính đáng kể.

Generalized Linear Model Regression Results

Dep. Variable:	churn	No. Observations:	26996
Model:	GLM	Df Residuals:	26955
Model Family:	Binomial	Df Model:	40
Link Function:	Logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-16787.
Date:	Thu, 11 May 2023	Deviance:	33574.
Time:	19:18:16	Pearson chi2:	2.93e+04
No. Iterations:	5	Pseudo R-squ. (CS):	0.1067
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	0.3765	0.105	3.578	0.000	0.170	0.583
housing[T.R]	0.2201	0.053	4.123	0.000	0.115	0.325
housing[T.na]	0.1214	0.054	2.268	0.023	0.016	0.226
payment_type[T.Monthly]	0.0384	0.046	0.829	0.407	-0.052	0.129
payment_type[T.Semi-Monthly]	-0.0005	0.047	-0.010	0.992	-0.094	0.093
payment_type[T.Weekly]	0.2493	0.035	7.140	0.000	0.181	0.318
payment_type[T.na]	0.0919	0.040	2.288	0.022	0.013	0.171
zodiac_sign[T.Aries]	0.1347	0.067	2.016	0.044	0.004	0.266
zodiac_sign[T.Cancer]	0.1451	0.064	2.273	0.023	0.020	0.270
zodiac_sign[T.Capricorn]	0.1942	0.094	2.060	0.039	0.009	0.379
zodiac_sign[T.Gemini]	0.0283	0.066	0.430	0.667	-0.101	0.157
zodiac_sign[T.Leo]	0.0784	0.064	1.218	0.223	-0.048	0.205
zodiac_sign[T.Libra]	0.0323	0.066	0.489	0.625	-0.097	0.162
zodiac_sign[T.Pisces]	0.1884	0.066	2.859	0.004	0.059	0.318
zodiac_sign[T.Sagittarius]	0.0952	0.066	1.432	0.152	-0.035	0.226
zodiac_sign[T.Scorpio]	0.0593	0.066	0.895	0.371	-0.070	0.189
zodiac_sign[T.Taurus]	0.0336	0.065	0.515	0.607	-0.094	0.161
zodiac_sign[T.Virgo]	0.1069	0.064	1.674	0.094	-0.018	0.232
zodiac_sign[T.na]	0.0785	0.066	1.195	0.232	-0.050	0.207
age	-0.0150	0.001	-10.634	0.000	-0.018	-0.012
deposits	0.0418	0.028	1.504	0.133	-0.013	0.096
withdrawal	0.0469	0.016	2.947	0.003	0.016	0.078
purchases_partners	-0.0179	0.001	-30.126	0.000	-0.019	-0.017
purchases	-0.0566	0.028	-1.992	0.046	-0.112	-0.001

Hình 4.2.17: Mô hình GLM đánh giá kết quả của một mô hình hồi quy tuyến tính.

Hình 4.2.17 cho thấy giá trị cột (P>|z|): Nếu giá trị p tuyệt đối nhỏ hơn 0.05, điều đó có nghĩa là tính năng này ảnh hưởng đến tỷ lệ churn theo cách có ý nghĩa thống kê.

```
Intercept                          1.457242
housing[T.R]                       1.246144
housing[T.na]                      1.129112
payment_type[T.Monthly]            1.039157
payment_type[T.Semi-Monthly]       0.999543
payment_type[T.Weekly]             1.283141
payment_type[T.na]                 1.096240
zodiac_sign[T.Aries]               1.144236
zodiac_sign[T.Cancer]              1.156140
zodiac_sign[T.Capricorn]           1.214396
zodiac_sign[T.Gemini]              1.028667
zodiac_sign[T.Leo]                 1.081607
zodiac_sign[T.Libra]               1.032840
zodiac_sign[T.Pisces]              1.207373
zodiac_sign[T.Sagittarius]         1.099872
zodiac_sign[T.Scorpio]             1.061847
zodiac_sign[T.Taurus]              1.034176
zodiac_sign[T.Virgo]               1.112785
zodiac_sign[T.na]                  1.081700
age                                0.985142
deposits                           1.042660
withdrawal                         1.048001
purchases_partners                 0.982249
purchases                          0.944992
cc_taken                           1.177580
cc_recommended                     1.000664
cc_disliked                        0.979899
cc_liked                           0.966854
cc_application_begin               1.003305
app_downloaded                     0.839714
ios_user                           1.151551
android_user                       1.003173
registered_phones                  1.124622
waiting_4_loan                     0.407040
cancelled_loan                     1.813179
received_loan                      2.088432
rejected_loan                      5.028056
left_for_two_month_plus            1.177423
left_for_one_month                 1.494797
reward_rate                        0.782696
is_referred                        1.050905
dtype: float64
```

Hình 4.2.18: Tính exp của các tham số (parameters) hồi quy trong mô hình

Hình 4.2.18 cho thấy các tỷ lệ lẻ, các giá trị lớn hơn 1 cho biết tỷ lệ churn tăng lên. Các giá trị nhỏ hơn 1 cho biết tình trạng churn đang diễn ra ít hơn.

4.2.2 Huấn luyện mô hình

a) Tiền xử lý dữ liệu.

Encoding là quá trình biến đổi dữ liệu từ một định dạng sang một định dạng khác để cho phù hợp với thuật toán hoặc mô hình. Các định dạng khác nhau của dữ liệu có thể là các kiểu dữ liệu khác nhau hoặc các biến số có thể có các giá trị khác nhau. Encoding là một phần quan trọng trong tiền xử lý dữ liệu trước khi đưa vào mô hình học máy để giúp cải thiện độ chính xác và hiệu suất của mô hình.

Data split là quá trình chia tập dữ liệu thành 2 phần, một phần được sử dụng để huấn luyện mô hình và phần còn lại được sử dụng để kiểm định mô hình. Quá trình này giúp đánh giá hiệu quả của mô hình trên dữ liệu không được sử dụng trong quá trình huấn luyện, từ đó giúp giảm thiểu hiện tượng overfitting và tăng tính khả thi của mô hình. Thông thường, dữ liệu được chia thành 2 tập là training set và testing set, tỷ lệ chia thường là 70/30 hoặc 80/20.

Data Balance (cân bằng dữ liệu) là quá trình điều chỉnh tỷ lệ của các lớp trong tập dữ liệu để đảm bảo mỗi lớp đóng góp tương đương trong quá trình huấn luyện và kiểm tra mô hình. Khi một lớp trong tập dữ liệu chiếm đa số, việc huấn luyện và kiểm tra mô hình sẽ bị thiên vị và không đảm bảo tính chính xác của mô hình. Vì vậy, data balance là một bước quan trọng để đảm bảo tính công bằng và chính xác của mô hình.

StandardScaler là một phương pháp chuẩn hóa dữ liệu trong machine learning. Phương pháp này giúp đưa các giá trị của các biến về cùng một phạm vi, giúp cho các thuật toán machine learning hoạt động hiệu quả hơn. Cụ thể, phương pháp này chuyển đổi dữ liệu sao cho trung bình bằng 0 và độ lệch chuẩn bằng 1. Bằng cách đó, dữ liệu sẽ được chuyển đổi thành dạng chuẩn (standardized form) và các thuật toán có thể dễ dàng xử lý các giá trị này mà không bị ảnh hưởng bởi các đơn vị đo lường khác nhau hoặc phạm vi giá trị khác nhau của các biến.

Sau khi tiền xử lý xong, chuyển sang huấn luyện mô hình.

b) Huấn luyện và đánh giá mô hình

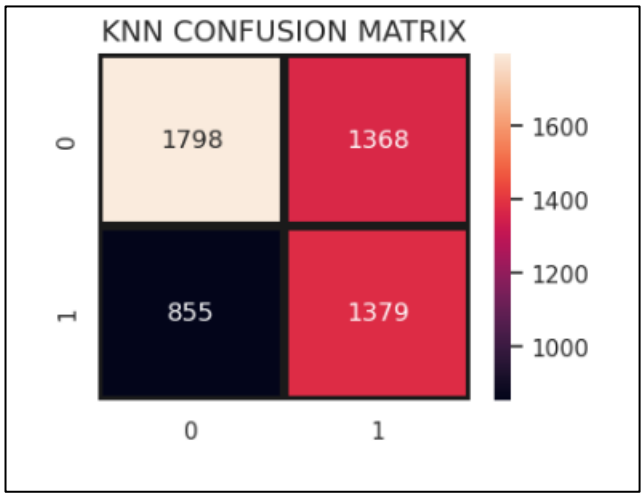
Các mô hình Học máy gồm các thuật toán phân lớp nhị phân sau:

- KNN:

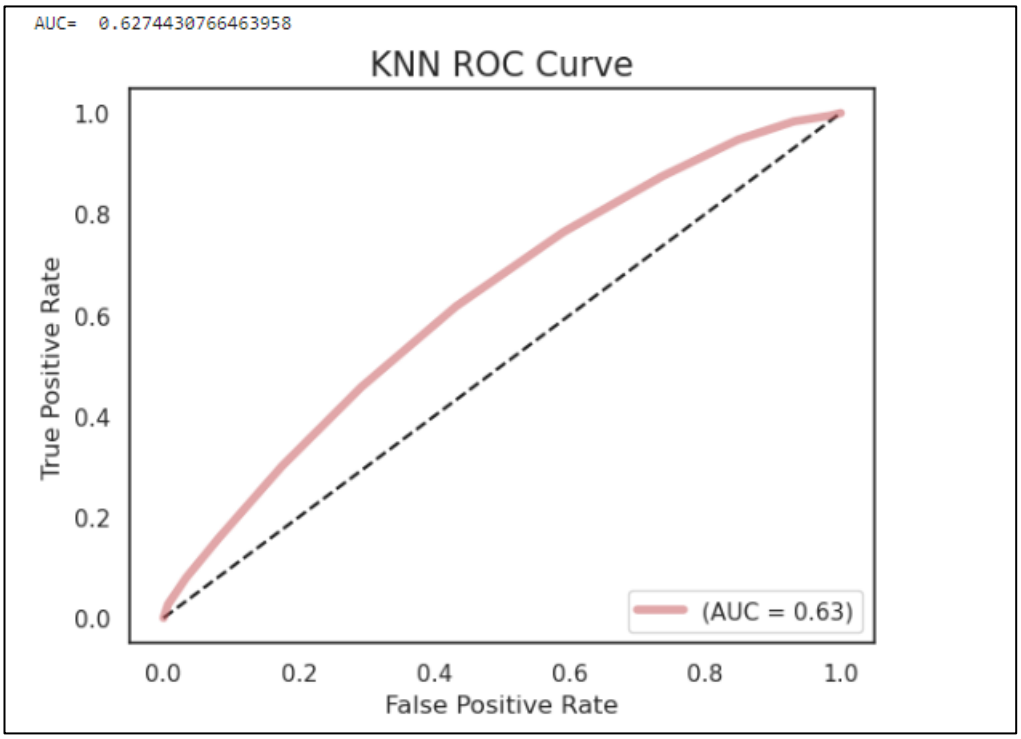
KNN accuracy: 0.5883333333333334

	precision	recall	f1-score	support
0	0.68	0.57	0.62	3166
1	0.50	0.62	0.55	2234
accuracy			0.59	5400
macro avg	0.59	0.59	0.59	5400
weighted avg	0.61	0.59	0.59	5400

Hình 4.2.19: Chỉ số đánh giá mô hình phân loại của mô hình KNN



Hình 4.2.20: Chỉ số ma trận nhầm lẫn của mô hình KNN



Hình 4.2.21: Chỉ số AUC và đường cong ROC của mô hình KNN

Nhận xét:

- Precision (độ chính xác): KNN đạt được độ chính xác khoảng 68% cho phân lớp 0 và 50% cho phân lớp 1.
- Recall (độ phủ): KNN đạt được độ phủ khoảng 57% cho phân lớp 0 và 62% cho phân lớp 1.
- F1-score : KNN đạt được F1-score khoảng 62% cho phân lớp 0 và 57% cho phân lớp 1.

- Accuracy (độ chính xác toàn bộ mô hình): KNN đạt được độ chính xác khoảng 59%.
- AUC của mô hình KNN đạt 63%.

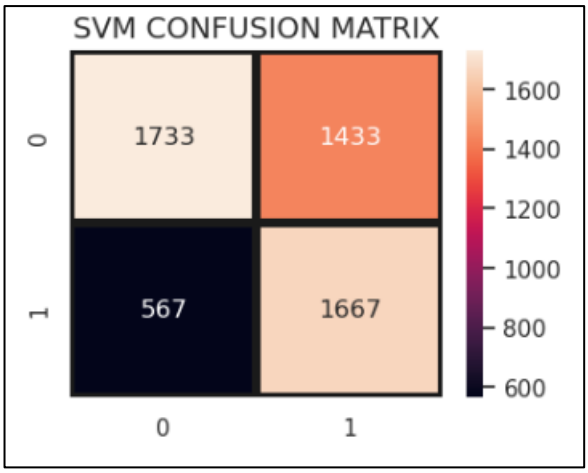
Từ các thông số trên, ta thấy mô hình KNN cho kết quả không tốt, đặc biệt là trong việc dự đoán phân lớp 1, vì cả Precision và F1-score của phân lớp 1 đều thấp.

SVM (SVC: Support Vector Classifier):

SVM accuracy : 0.6296296296296297

	precision	recall	f1-score	support
0	0.75	0.55	0.63	3166
1	0.54	0.75	0.63	2234
accuracy			0.63	5400
macro avg	0.65	0.65	0.63	5400
weighted avg	0.66	0.63	0.63	5400

Hình 4.2.22: Chỉ số đánh giá mô hình phân loại của mô hình SVM



Hình 4.2.23: Chỉ số ma trận nhầm lẫn của mô hình SVM

Nhận xét:

- Mô hình SVC (Support Vector Classifier) cho customer churn đạt được độ chính xác trung bình, accuracy là 63%.
- Tuy nhiên, giá trị precision và recall cho phân lớp 1 (churn) là khá thấp, chỉ đạt 54% và 75% tương ứng
- Kết quả đánh giá precision và recall cho cả hai phân lớp đều thấp, đặc biệt là recall cho phân lớp 0 chỉ đạt được 55%.

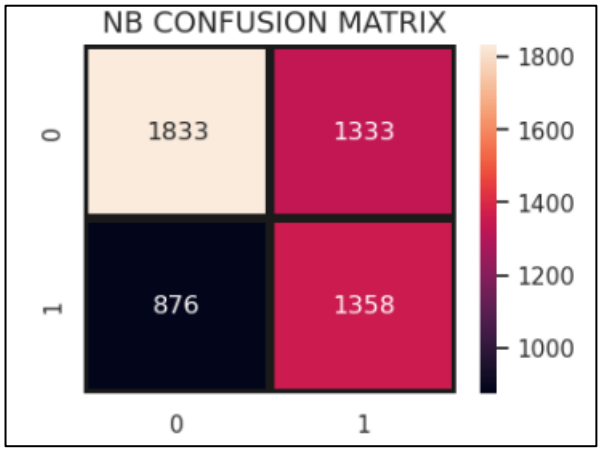
- F1-score cho cả hai lớp đều xấp xỉ 63%.
- Tổng thể, mô hình này chưa tốt, chưa đạt hiệu quả cao trong việc phân loại khách hàng churn.

GaussianNB:

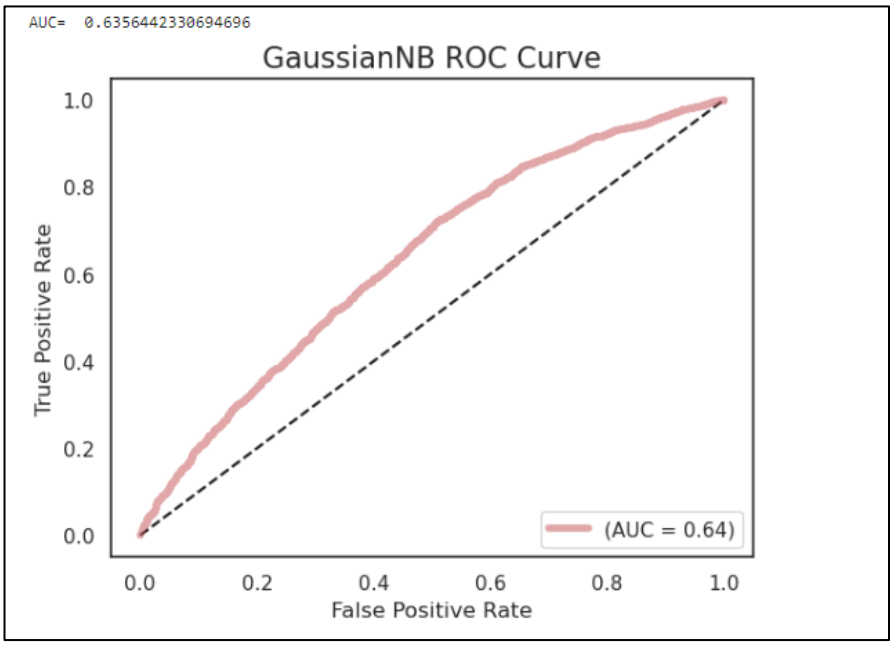
GaussianNB accuracy : 0.590925925925926

	precision	recall	f1-score	support
0	0.68	0.58	0.62	3166
1	0.50	0.61	0.55	2234
accuracy			0.59	5400
macro avg	0.59	0.59	0.59	5400
weighted avg	0.61	0.59	0.59	5400

Hình 4.2.24: Chỉ số đánh giá mô hình phân loại của mô hình GaussianNB



Hình 4.2.25: Chỉ số ma trận nhầm lẫn của mô hình GaussianNB



Hình 4.2.26: Chỉ số AUC và đường cong ROC của mô hình GaussianNB

Nhận xét:

- Mô hình GaussianNB cho kết quả khá thấp, với accuracy chỉ đạt 59% độ chính xác không cao và AUC đạt 64%.

- Precision cho phân lớp 1 chỉ đạt khoảng 50%, cho thấy mô hình dự đoán sai rất nhiều khách hàng churn. Recall đạt khoảng 61%, cho thấy mô hình dự đoán đúng khá nhiều khách hàng churn.

- F1-score phân lớp 1 và 0 lần lượt đạt 58% và 61%.

- Điểm AUC của mô hình cũng chỉ đạt 64%, cho thấy khả năng phân loại của mô hình chỉ hơn mức đoán ngẫu nhiên một chút.

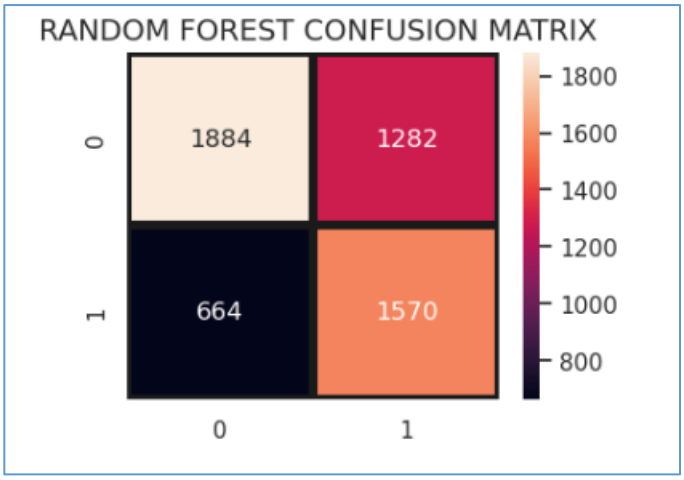
Điều này cho thấy mô hình này không thực sự phù hợp để dự đoán khách hàng churn.

- Random Forest:

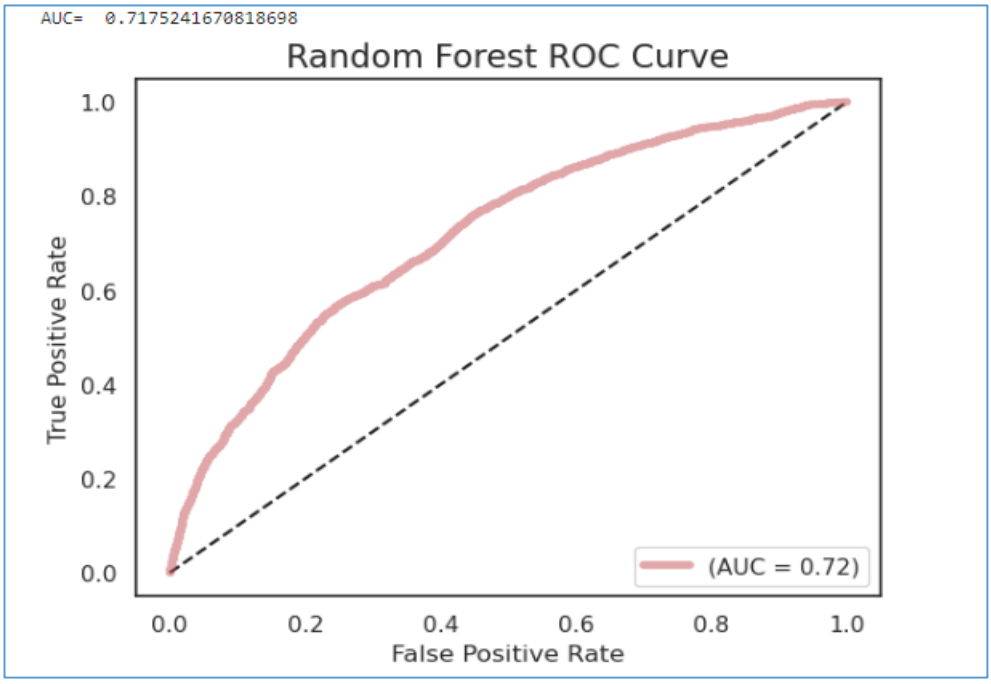
Random Forest accuracy: 0.6396296296296297

	precision	recall	f1-score	support
0	0.74	0.60	0.66	3166
1	0.55	0.70	0.62	2234
accuracy			0.64	5400
macro avg	0.64	0.65	0.64	5400
weighted avg	0.66	0.64	0.64	5400

Hình 4.2.27: Chỉ số đánh giá mô hình phân loại của mô hình Random Forest



Hình 4.2.28: Chỉ số ma trận nhầm lẫn của mô hình Random Forest



Hình 4.2.29: Chỉ số AUC và đường cong ROC của mô hình Random Forest

Nhận xét:

- AUC: 0.72, cho thấy khả năng phân loại của mô hình là tương đối tốt.
- Precision (độ chính xác): 0.74, cho biết trong các khách hàng được dự đoán là không rời đi, có 74% thực sự không rời đi. Và 55% thực sự rời đi.
- Recall (độ phủ): cho phân lớp 1 là 0.7, cho biết mô hình có thể xác định được 70% khách hàng thực sự bỏ đi.
- F1-score cho phân lớp 1 và 0 lần lượt là 60% và 70%.
- Accuracy: 0.64, cho biết tỉ lệ dự đoán đúng của mô hình là 64%.

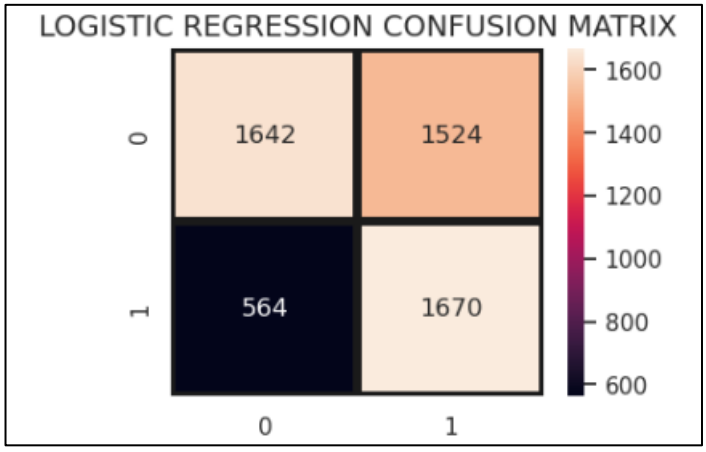
Tổng quan, mô hình Random Forest cho kết quả tương đối tốt trong việc dự đoán khách hàng rời đi.

- Logistic Regression

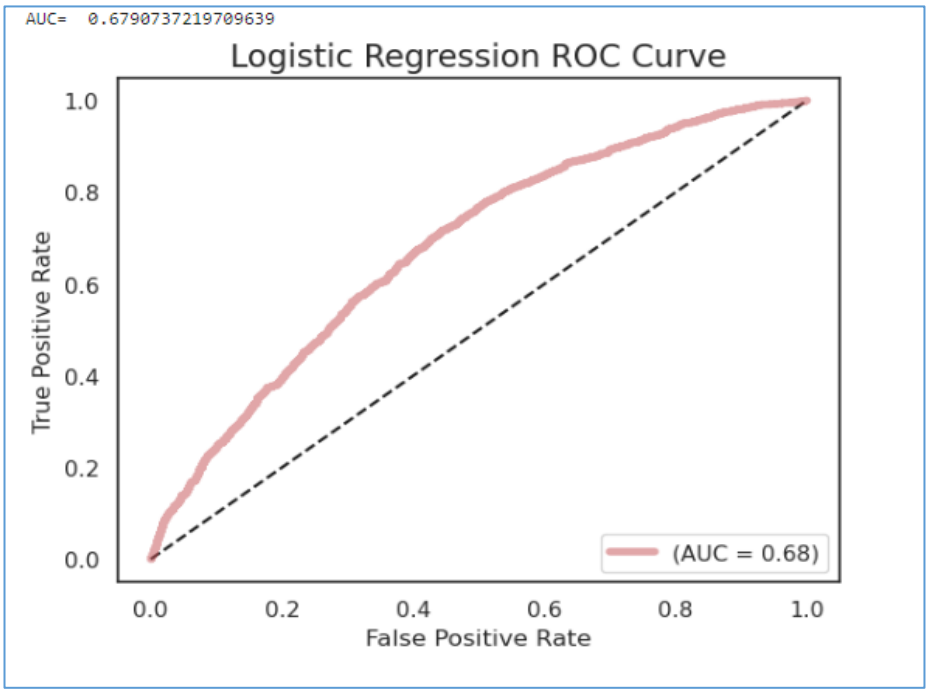
Logistic Regression accuracy : 0.6133333333333333

	precision	recall	f1-score	support
0	0.74	0.52	0.61	3166
1	0.52	0.75	0.62	2234
accuracy			0.61	5400
macro avg	0.63	0.63	0.61	5400
weighted avg	0.65	0.61	0.61	5400

Hình 4.2.30: Chỉ số đánh giá mô hình phân loại của mô hình Logistic Regression



Hình 4.2.31: Chỉ số ma trận nhầm lẫn của mô hình Logistic Regression



Hình 4.2.32: Chỉ số AUC và đường cong ROC của mô hình Logistic Regression

Nhận xét:

- Precision của phân lớp 0 là cao hơn phân lớp 1, ngược lại recall phân lớp 1 cao hơn phân lớp 0 .Tuy nhiên giá trị f1-score của hai lớp này gần như bằng nhau. Điều này cho thấy mô hình có độ chính xác cao trong việc phân loại các khách hàng ở lớp 0, nhưng lại có thể bỏ sót một số khách hàng quan trọng ở phân lớp 1.
- Giá trị AUC của mô hình là 0.68, cho thấy mô hình có khả năng phân loại các điểm dữ liệu đúng lớp là class 0 hay class 1 khá tốt. Nên mô hình có khả năng phân loại tốt hơn so với việc dự đoán ngẫu nhiên.

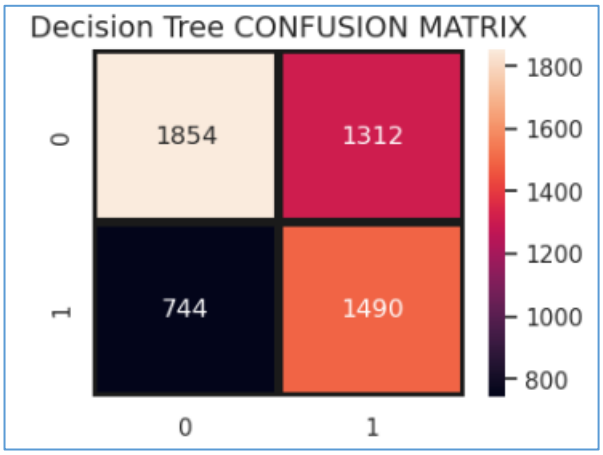
Tổng thể, mô hình Logistic Regression cho kết quả tương đối khá tốt.

- Decision Tree Classifier:

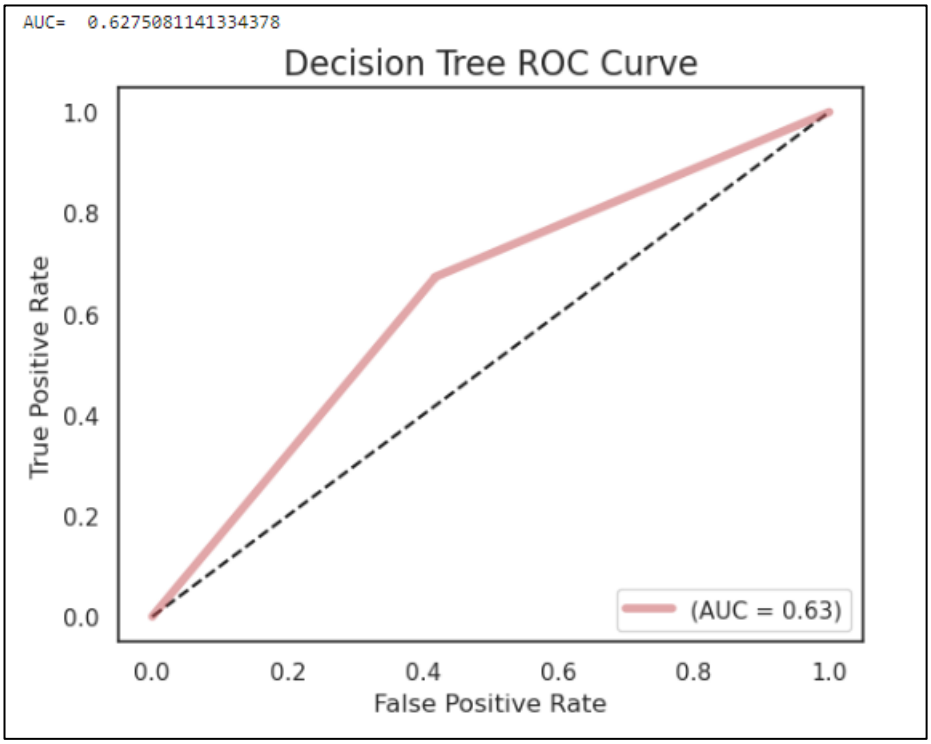
Decision Tree accuracy : 0.6192592592592593

	precision	recall	f1-score	support
0	0.71	0.59	0.64	3166
1	0.53	0.67	0.59	2234
accuracy			0.62	5400
macro avg	0.62	0.63	0.62	5400
weighted avg	0.64	0.62	0.62	5400

Hình 4.2.33: Chỉ số đánh giá mô hình phân loại của mô hình Decision Tree Classifier



Hình 4.2.34: Chỉ số ma trận nhầm lẫn của mô hình Decision Tree Classifier



Hình 4.2.35: Chỉ số AUC và đường cong ROC của mô hình Decision Tree Classifier

Nhận xét:

- AUC đạt 63%, cho thấy mô hình có khả năng phân loại dữ liệu tốt hơn so với việc dự đoán ngẫu nhiên, nhưng vẫn chưa tốt lắm.

- Precision cho các phân lớp 0 và 1 lần lượt là 71% và 53%, cho biết khả năng mô hình đưa ra dự đoán chính xác đối với các điểm dữ liệu được dự đoán là không churn là tương đối cao hơn so với dự đoán churn .

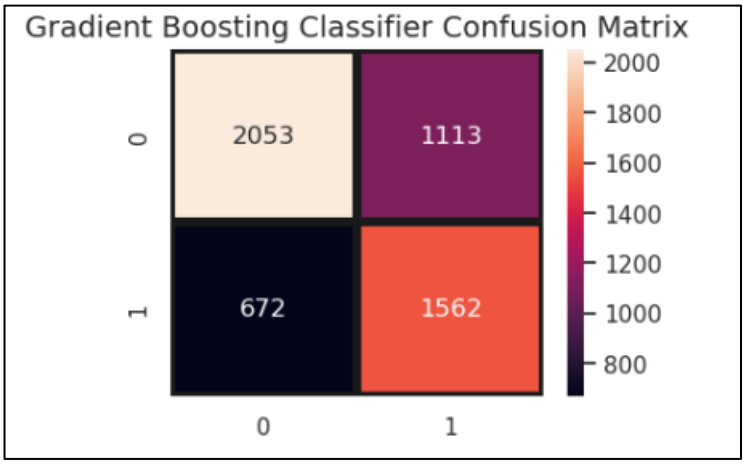
- Recall cho các phân lớp 0 và 1 lần lượt là 59% và 67%, cho biết khả năng mô hình tìm ra tất cả các điểm dữ liệu churn là tương đối cao, nhưng lại thấp đối với điểm dữ liệu không churn.
- F1-score trung bình cho các lớp 0 và 1 lần lượt là 64% và 59%, cho thấy sự cân bằng giữa precision và recall của mô hình còn chưa tốt.
- Tổng quan, mô hình Decision Tree Classifier cho thấy hiệu suất dự đoán chưa tốt.

- Gradient Boosting Classifier

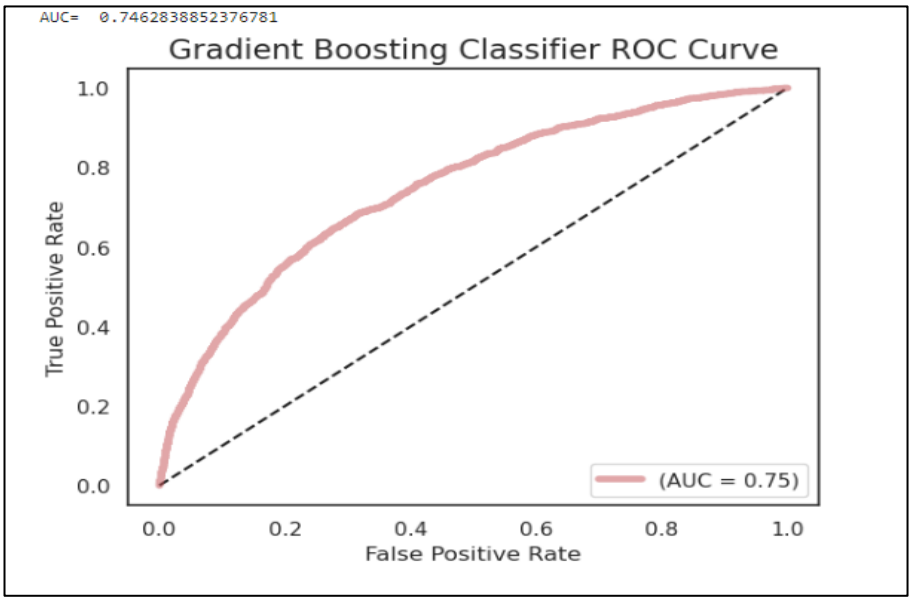
Gradient Boosting Classifier: 0.6694444444444444

	precision	recall	f1-score	support
0	0.75	0.65	0.70	3166
1	0.58	0.70	0.64	2234
accuracy			0.67	5400
macro avg	0.67	0.67	0.67	5400
weighted avg	0.68	0.67	0.67	5400

Hình 4.2.36: Chỉ số đánh giá mô hình phân loại của mô hình Gradient Boosting Classifier



Hình 4.2.37: Chỉ số ma trận nhầm lẫn của mô hình Gradient Boosting Classifier



Hình 4.2.38: Chỉ số AUC và đường cong ROC của mô hình Gradient Boosting Classifier

Nhận xét:

- Mô hình Gradient Boosting Classifier có AUC và độ chính xác (accuracy) khá tốt, đạt tới 75% và 67% tương ứng. Precision, recall và f1-score của mô hình cũng khá cân bằng, với giá trị f1-score cho cả hai lớp đều trên 0.6.
- Mô hình này có khả năng dự đoán chính xác khách hàng có khả năng churn và không churn khá tốt.

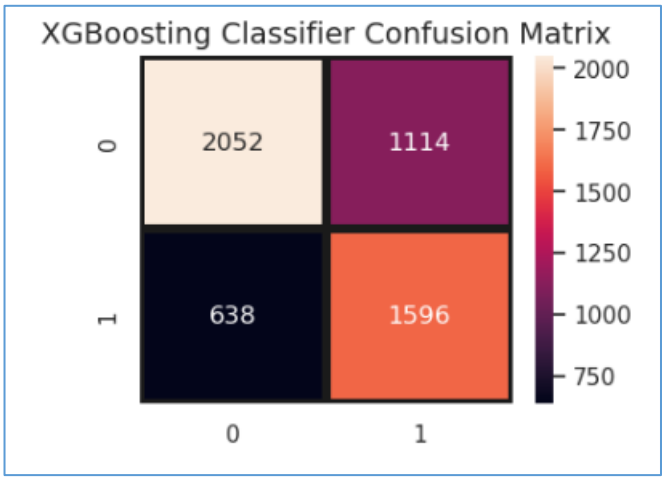
Tổng quan, Gradient Boosting Classifier là một mô hình khá tốt trong việc dự đoán khả năng churn, đánh giá trên các chỉ số đánh giá đồng thời, tuy nhiên mô hình vẫn còn cần phải khả năng cải thiện được kết quả Precision và F1-score của phân lớp 1.

- XGBoosting Classifier

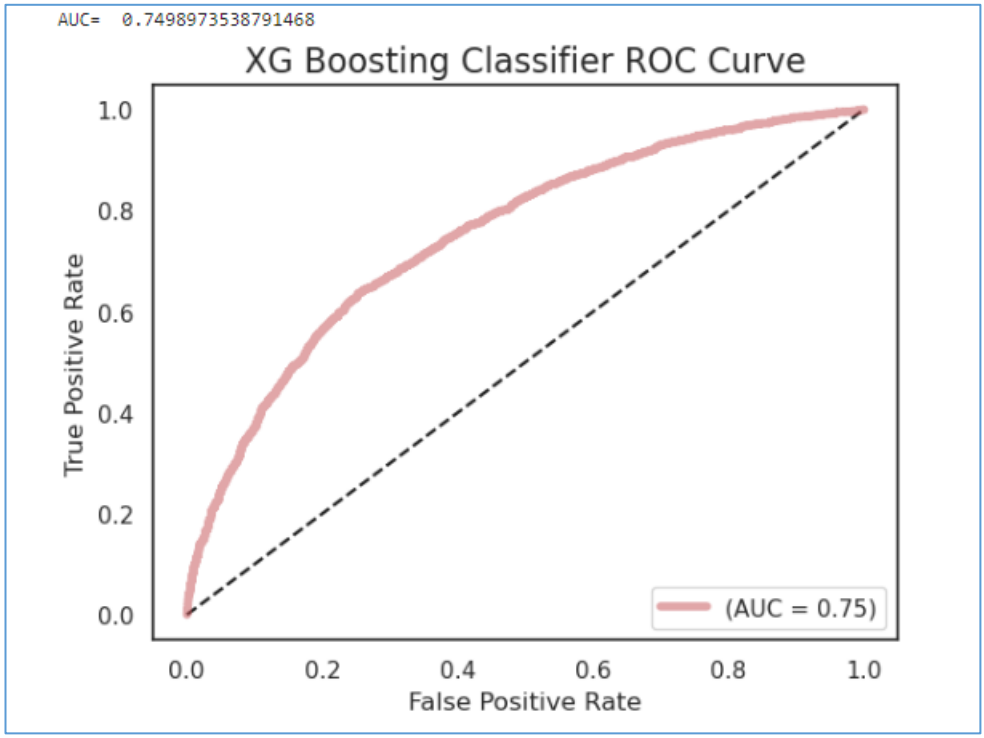
XGBoost Classifier: 0.6755555555555556

	precision	recall	f1-score	support
0	0.76	0.65	0.70	3166
1	0.59	0.71	0.65	2234
accuracy			0.68	5400
macro avg	0.68	0.68	0.67	5400
weighted avg	0.69	0.68	0.68	5400

Hình 4.2.39: Chỉ số đánh giá mô hình phân loại của mô hình XGBoosting Classifier



Hình 4.2.40: Chỉ số ma trận nhầm lẫn của mô hình XGBoosting Classifier



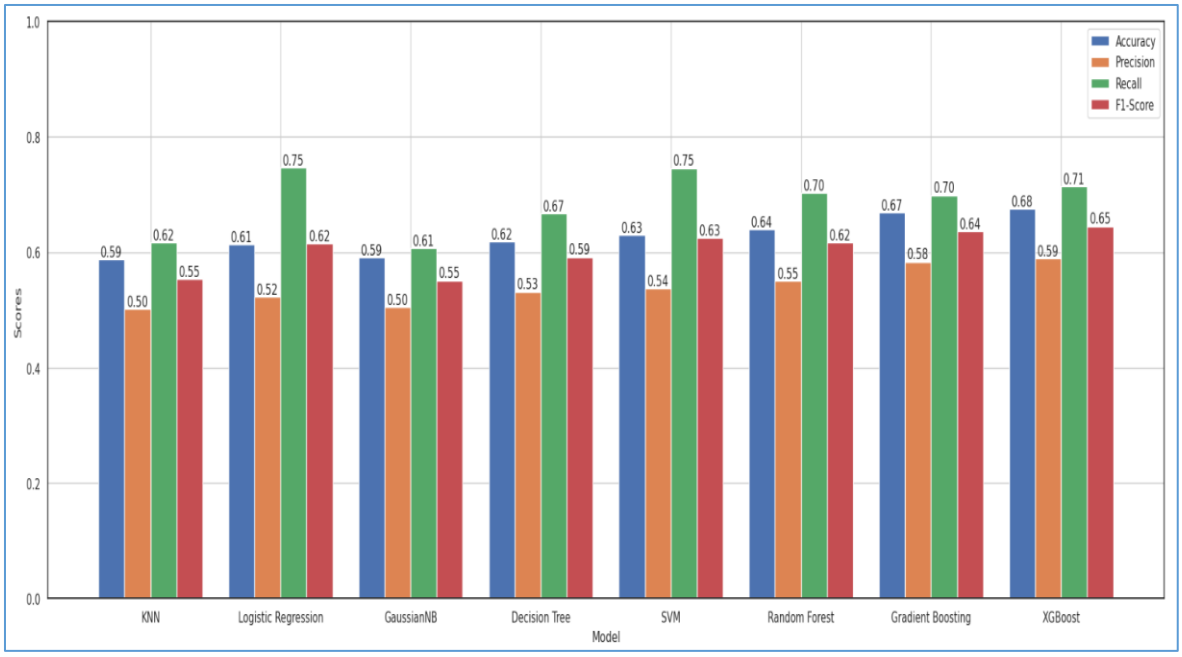
Hình 4.2.41: Chỉ số AUC và đường cong ROC của mô hình XGBoosting Classifier

Nhận xét:

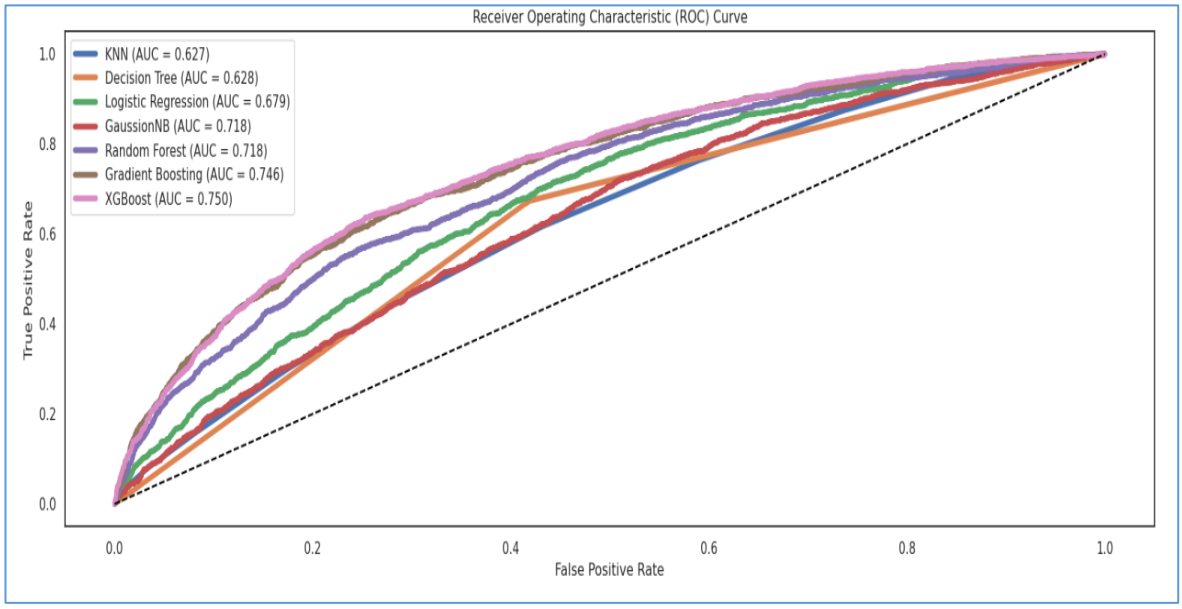
- Mô hình XGBoosting Classifier cho kết quả AUC tương đối cao là 75%, tức là khả năng phân loại tốt giữa hai nhóm khách hàng churn và không churn.
- Kết quả đánh giá của các chỉ số precision, recall và f1-score cũng tương đối tốt, khi cho thấy độ chính xác, độ phủ và chỉ số giữa precision và recall đều cao.

Mô hình XGBoosting Classifier cũng là một lựa chọn rất tốt cho bài toán phân loại churn trong bộ dữ liệu này.

⇒ **Tổng kết:**



Hình 4.2.42: So sánh độ đo chỉ số của các mô hình trong việc dự đoán churn.



Hình 4.2.43: So sánh độ đo AUC và đường cong ROC của các mô hình trong việc dự đoán phân loại churn.

⇒ **Nhận xét chung:**

- XGBoosting là mô hình có kết quả tốt nhất. Có độ chính xác và độ đo AUC cao nhất trong số tất cả các mô hình được đánh giá. Mô hình này được đề xuất để sử dụng để dự đoán phân loại customer churn tốt nhất.
- Các mô hình Gradient Boosting Classifier và Random Forest Classifier cũng đạt được AUC và độ chính xác tương đối cao, với chỉ số giữa precision và recall phù hợp, do đó chúng cũng có thể được sử dụng để dự đoán customer churn.
- Logistic Regression có kết quả khá tốt và có thể được sử dụng trong một số trường hợp.
- Mô hình GaussionNB, Decision Tree, SVM Classifier cho kết quả khá thấp.
- Mô hình KNN là các mô hình có kết quả thấp nhất trong số tất cả các mô hình được đánh giá.

Trong số các mô hình này, XGBoost có hiệu suất tốt nhất. Vì vậy, mô hình XGBoosting được đánh giá tốt nhất trong số các mô hình đã huấn luyện, vì nó có AUC tốt và độ chính xác cao, phù hợp để dự đoán phân loại customer churn.

c) Cải thiện mô hình.

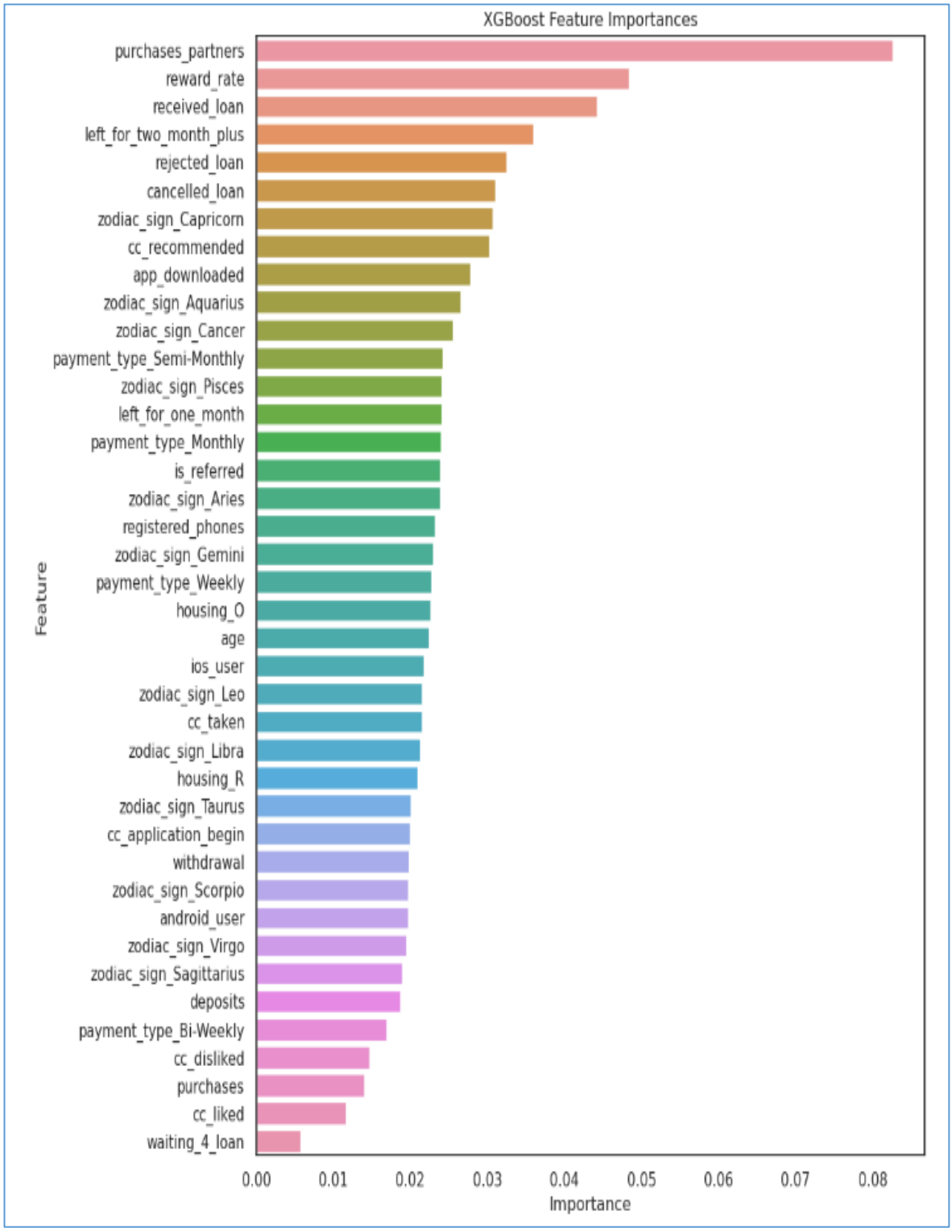
- Cross Validation (Chia dữ liệu kiểm tra chéo):

Là một phương pháp kiểm tra hiệu quả của mô hình. Nó thực hiện bằng cách chia dữ liệu thành nhiều tập con và thực hiện huấn luyện và kiểm tra mô hình trên những tập con này.

Cross validation giúp đánh giá hiệu suất của mô hình một cách chính xác hơn và tránh được tình trạng overfitting hoặc underfitting. Ngoài ra, nó cũng giúp tăng khả năng tái sử dụng dữ liệu và giảm thiểu việc chọn sai dữ liệu.

- Feature Importances:

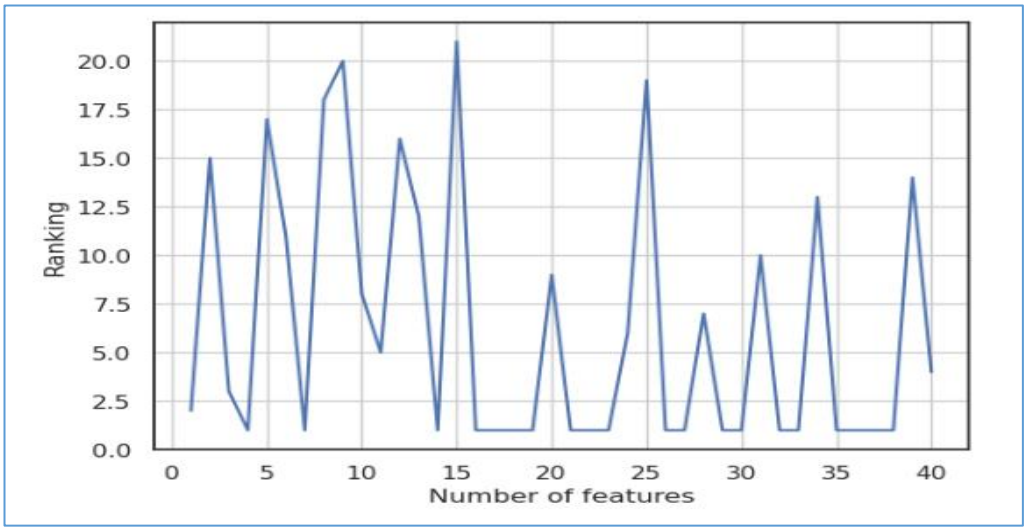
Feature Importances được sử dụng để đánh giá mức độ quan trọng của các đặc trưng (Churn) đối với kết quả của mô hình. Nó cho phép chúng ta xác định những đặc trưng nào ảnh hưởng mạnh đến dự đoán và giúp cho việc giải thích mô hình trở nên dễ dàng hơn.



Hình 4.2.44: Đặc trưng quan trọng của mô hình XGBoost

- Feature Selection:

RFE: số lượng các features được chọn(RFE) là 20.

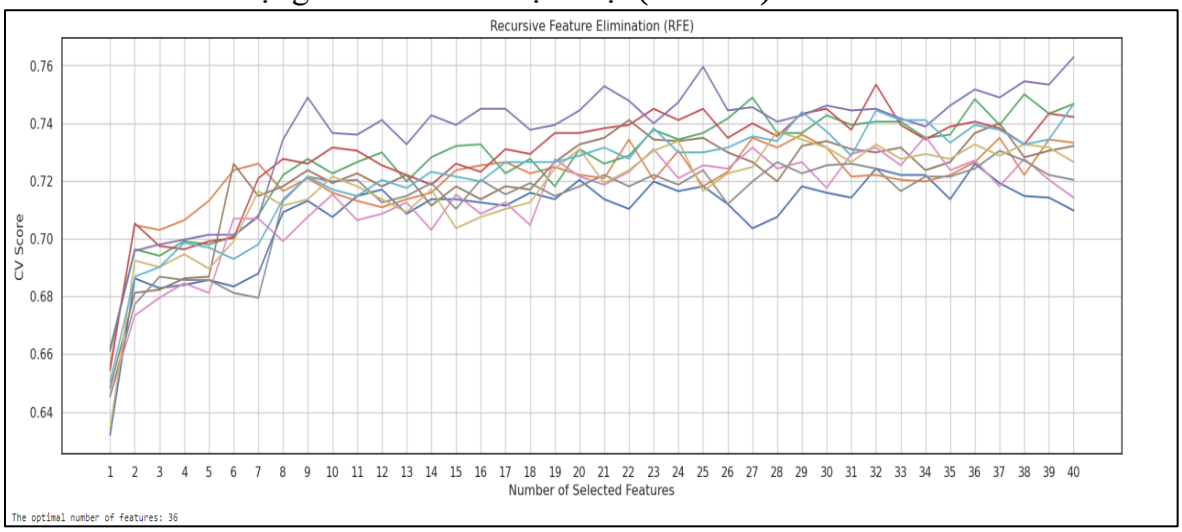


Hình 4.2.45: Biểu đồ đường thể hiện ranking của các đặc trưng sau khi sử dụng Recursive Feature Elimination (RFE)

```
Index(['purchases_partners', 'cc_recommended', 'registered_phones',
       'cancelled_loan', 'received_loan', 'rejected_loan',
       'left_for_two_month_plus', 'reward_rate', 'is_referred', 'housing_O',
       'payment_type_Monthly', 'payment_type_Semi-Monthly',
       'zodiac_sign_Aquarius', 'zodiac_sign_Aries', 'zodiac_sign_Capricorn',
       'zodiac_sign_Gemini', 'zodiac_sign_Libra', 'zodiac_sign_Pisces',
       'zodiac_sign_Sagittarius', 'zodiac_sign_Scorpio'],
      dtype='object')
```

Hình 4.2.46: Danh sách đặc trưng được chọn bởi RFE

RFECV: số lượng các features được chọn(RFECV) là 36.



Hình 4.2.47: Đồ thị biểu diễn giá trị của cross-validation scores theo số lượng features được lựa chọn bởi Recursive Feature Elimination with Cross-Validation (RFECV).

```
X_rfe dimension: (17880, 36)
X_rfe column list: ['age', 'deposits', 'withdrawal', 'purchases_partners', 'purchases', 'cc_taken', 'cc_re
commended', 'cc_application_begin', 'app_downloaded', 'ios_user', 'android_user', 'registered_phones', 'ca
ncelled_loan', 'received_loan', 'rejected_loan', 'left_for_two_month_plus', 'left_for_one_month', 'reward_
rate', 'is_referred', 'housing_O', 'housing_R', 'payment_type_Monthly', 'payment_type_Semi-Monthly', 'paym
ent_type_Weekly', 'zodiac_sign_Aquarius', 'zodiac_sign_Aries', 'zodiac_sign_Cancer', 'zodiac_sign_Capricor
n', 'zodiac_sign_Gemini', 'zodiac_sign_Leo', 'zodiac_sign_Libra', 'zodiac_sign_Pisces', 'zodiac_sign_Sagit
tarius', 'zodiac_sign_Scorpio', 'zodiac_sign_Taurus', 'zodiac_sign_Virgo']
```

Hình 4.2.48: Danh sách đặc trưng được chọn bởi RFECV

- Huấn luyện lại mô hình sử dụng Feature Selection (RFE)

Kiến trúc mô hình sau khi được cải thiện:

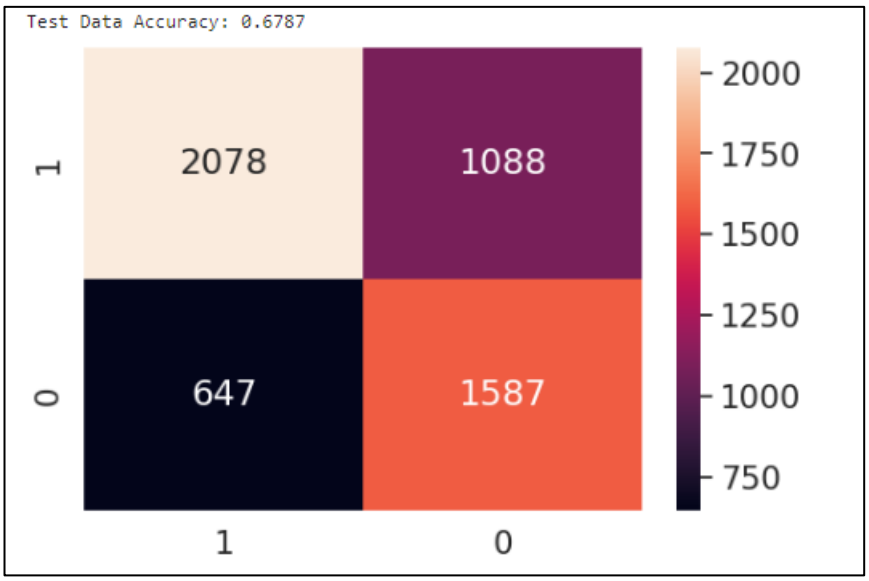
```
XGBClassifier(base_score=0.5, booster='gbtree', callbacks=None,
              colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
              early_stopping_rounds=None, enable_categorical=False,
              eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
              importance_type=None, interaction_constraints='',
              learning_rate=0.300000012, max_bin=256, max_cat_to_onehot=4,
              max_delta_step=0, max_depth=6, max_leaves=0, min_child_weight=1,
              missing=nan, monotone_constraints='()', n_estimators=100,
              n_jobs=0, num_parallel_tree=1, predictor='auto', random_state=0,
              reg_alpha=0, reg_lambda=1, ...)
```

Hình 4.2.49: Kiến trúc mô hình sau khi được cải thiện bởi Feature Selection (RFE).

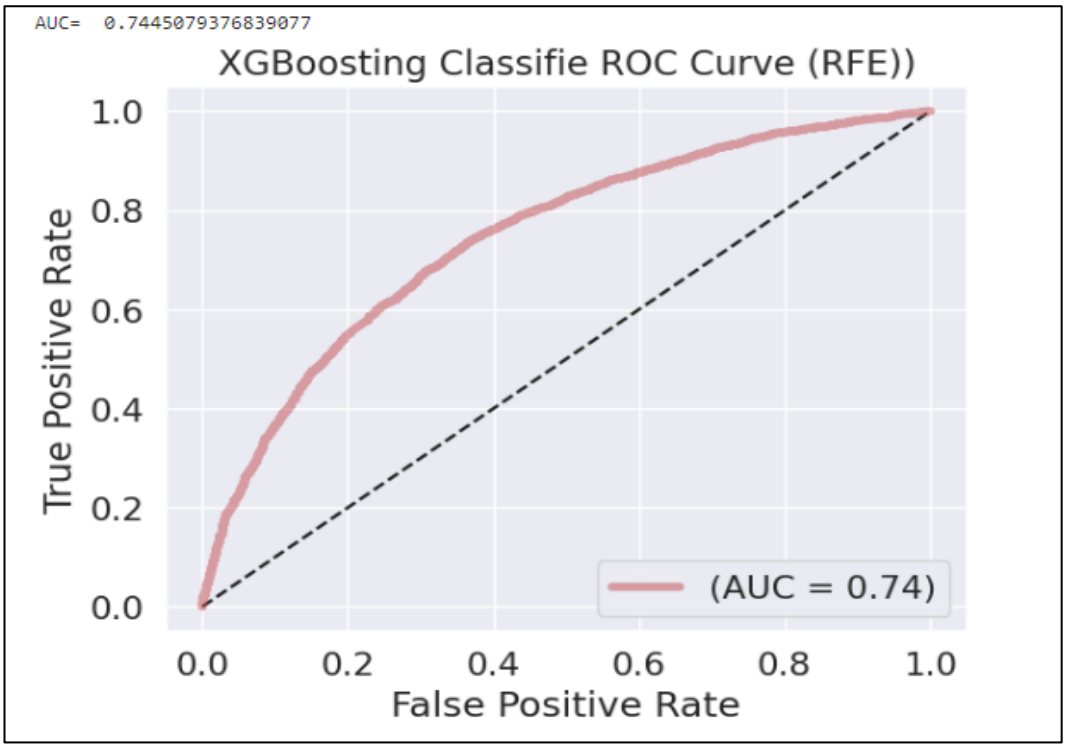
- XGBoosting Classifier (RFE) Acurracy = 0.6787037037037037

	precision	recall	f1-score	support
0	0.76	0.66	0.71	3166
1	0.59	0.71	0.65	2234
accuracy			0.68	5400
macro avg	0.68	0.68	0.68	5400
weighted avg	0.69	0.68	0.68	5400

Hình 4.2.50: Chỉ số đánh giá mô hình phân loại của mô hình XGBoosting Classifier sau khi được cải thiện bởi Feature Selection (RFE).



Hình 4.2.51: Chỉ số ma trận nhầm lẫn của mô hình XGBoosting Classifier sau khi được cải thiện bởi Feature Selection (RFE).



Hình 4.2.52: Chỉ số AUC và đường cong ROC của mô hình XGBoosting Classifier sau khi được cải thiện bởi Feature Selection (RFE).

- Huấn luyện lại mô hình sử dụng Feature Selection (RFECV)

Kiến trúc mô hình sau khi được cải thiện:

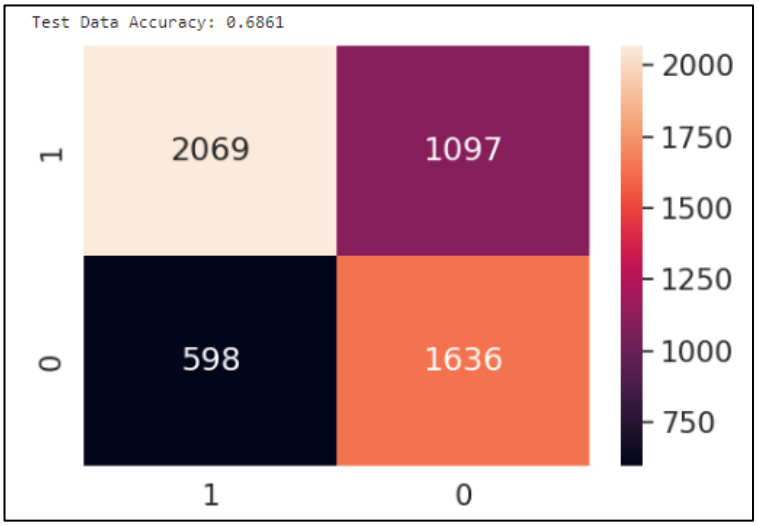
```
XGBClassifier(base_score=0.5, booster='gbtree', callbacks=None,
              colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
              early_stopping_rounds=None, enable_categorical=False,
              eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
              importance_type=None, interaction_constraints='',
              learning_rate=0.300000012, max_bin=256, max_cat_to_onehot=4,
              max_delta_step=0, max_depth=6, max_leaves=0, min_child_weight=1,
              missing=nan, monotone_constraints='()', n_estimators=100,
              n_jobs=0, num_parallel_tree=1, predictor='auto', random_state=0,
              reg_alpha=0, reg_lambda=1, ...)
```

Hình 4.2.53: Kiến trúc mô hình sau khi được cải thiện bởi Feature Selection (RFECV).

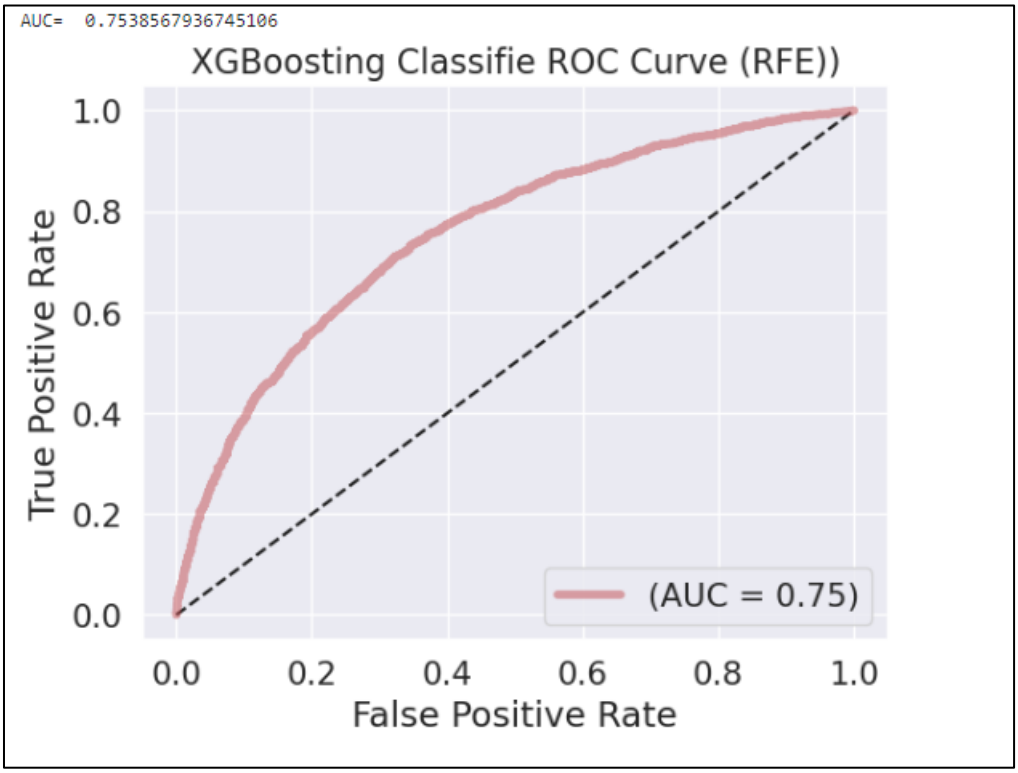
- XGBoosting Classifier (RFECV) Acurracy = 0.6744444444444444

	precision	recall	f1-score	support
0	0.78	0.65	0.71	3166
1	0.60	0.73	0.66	2234
accuracy			0.69	5400
macro avg	0.69	0.69	0.68	5400
weighted avg	0.70	0.69	0.69	5400

Hình 4.2.54: Chỉ số đánh giá mô hình phân loại của mô hình XGBoosting Classifier sau khi được cải thiện bởi Feature Selection (RFECV).

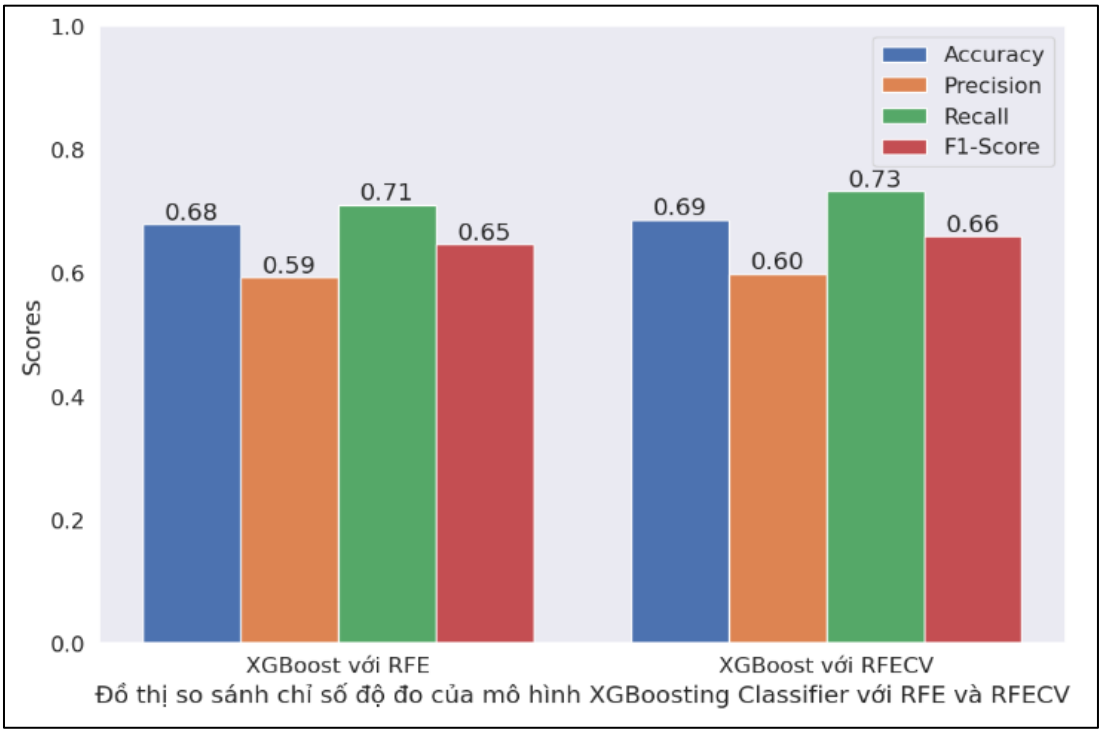


Hình 4.2.55: Chỉ số ma trận nhầm lẫn của mô hình XGBoosting Classifier sau khi được cải thiện bởi Feature Selection (RFECV).

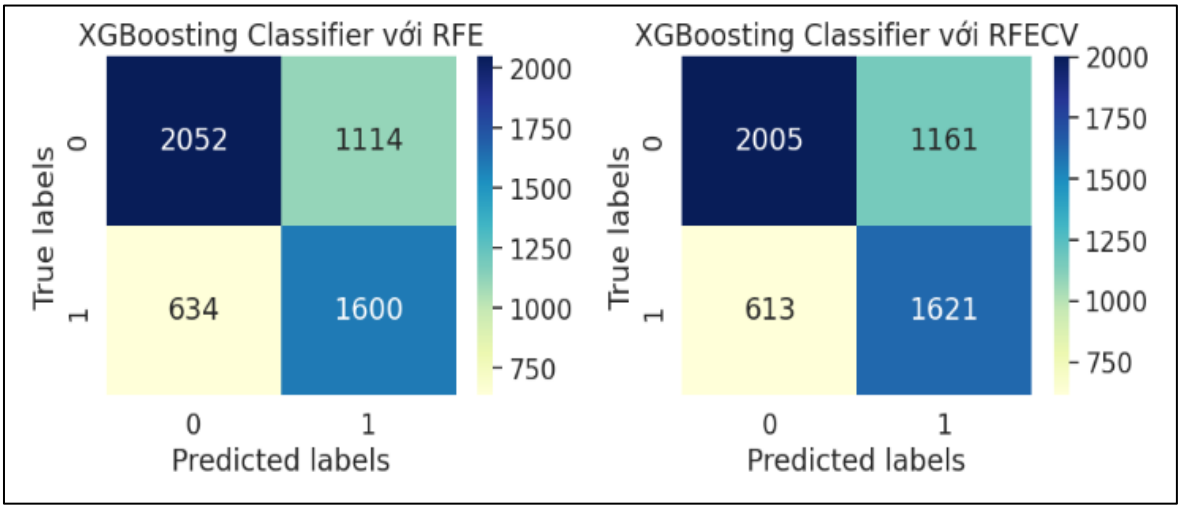


Hình 4.2.56: Chỉ số AUC và đường cong ROC của mô hình XGBoosting Classifier sau khi được cải thiện bởi Feature Selection (RFECV).

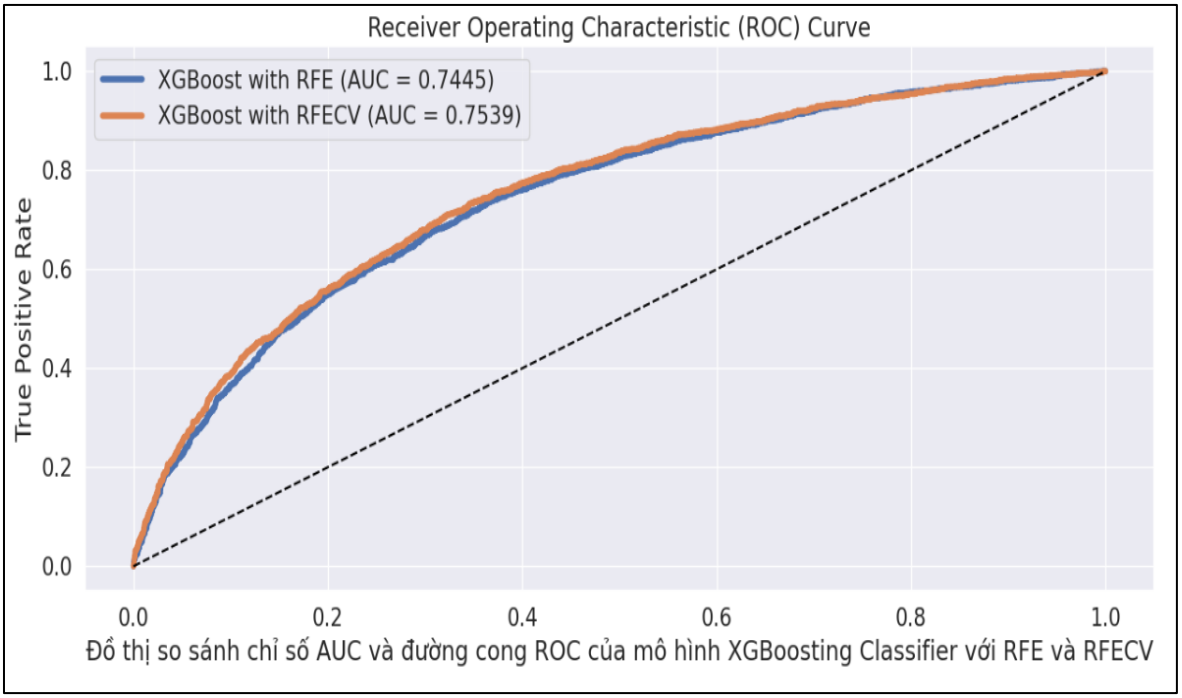
⇒ **So sánh:**



Hình 4.2.57: Đồ thị so sánh chỉ số độ đo của mô hình XGBoosting Classifier với RFE và RFECV



Hình 4.2.58: Đồ thị so sánh ma trận nhầm lẫn của mô hình XGBoosting Classifier với RFE và RFECV



Hình 4.2.59: Đồ thị so sánh chỉ số AUC và đường cong ROC của mô hình XGBoosting Classifier với RFE và RFECV

⇒ **Nhận xét:**

Mô hình XGBoosting Classifier được cải thiện bởi RFE và RFECV cũng cải thiện được các chỉ số so với mô hình XGBoosting Classifier ban đầu. Trong đó,

XGBoosting Classifier với RFECV đạt được AUC và kết quả chỉ số cao hơn XGBoosting Classifier với RFE, với chỉ số giữa precision và recall phù hợp.

Vì vậy, mô hình XGBoosting Classifier (RFECV) được đánh giá tốt nhất phù hợp để dự đoán phân loại churn.

⇒ **Kết quả**

Bảng 2: Chỉ số độ đo của dự đoán churn của khách hàng.

Model	**Precision**	**Recall**	**f1-score**	**Accuracy**	**AUC**
KNN	0.50	0.62	0.55	0.5883	0.627
SVM	0.54	0.75	0.63	0.6296	-
GaussianNB	0.50	0.61	0.55	0.5909	0.636
Random Forest	0.55	0.70	0.62	0.6396	0.718
Logistic Regression	0.52	0.75	0.62	0.6133	0.679
Decision Tree Classifier	0.53	0.67	0.59	0.6192	0.628
Gradient Boosting Classifier	0.58	0.70	0.64	0.6694	0.746
XGBoosting Classifier	0.59	0.71	0.65	0.6755	0.75

Bảng 3: Chỉ số độ đo mô hình XGBoosting Classifier khi sử dụng Recursive Feature Elimination (RFE) và Recursive Feature Elimination with Cross-Validation (RFECV) để dự đoán churn

XGBoosting Classifier	**Precision**	**Recall**	**f1-score**	**Accuracy**	**AUC**
RFE	0.59	0.71	0.65	0.6787	0.745
RFECV	0.60	0.73	0.66	0.6861	0.754

	user	churn	predicted_churn
0	61353	1.0	1
1	67679	0.0	0
2	21269	0.0	0
3	69531	0.0	0
4	25997	0.0	0
...	...	...	...
5395	22377	0.0	1
5396	24291	1.0	1
5397	23740	0.0	1
5398	47663	1.0	0
5399	52752	1.0	0

5400 rows × 3 columns

Hình 4.2.60: Kết quả dự đoán của mô hình phân loại XGBoosting Classifier (RFECV)

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận:

5.1.1 Kết quả bài toán

Sau khi huấn luyện nhều mô hình Máy học, chúng em đã lựa chọn và áp dụng mô hình XGBoosti có khả năng dự đoán phân lớp tốt nhất so với các mô hình còn lại. Điều này có thể phát triển phân tích hành vi khách hàng tham gia trong ứng dụng di động để dự đoán chính xác và có thể phát triển sang hướng khác như trong doanh nghiệp,…

5.1.2. Kiến thức

Qua khóa luận, chúng em đã biết và học được thêm nhiều kiến thức về phân tích dữ liệu, máy học mà cụ thể là cách xử lý dữ liệu thô và các mô hình phân lớp nhị phân để dự đoán khách hàng tham gia.

5.1.3. Hạn chế

Các phương pháp và mô hình: Các phương pháp và mô hình mà chúng em sử dụng trong đề tài có thể không phù hợp hoặc không đủ tốt để giải quyết vấn đề. Có thể có các phương pháp và mô hình khác tốt hơn nhưng không được sử dụng.

Không thể dự đoán chính xác 100%: Mặc dù các mô hình có độ chính xác cao, tuy nhiên không thể dự đoán chính xác 100% và có thể có những trường hợp bị nhầm lẫn hoặc dự đoán sai. Và cải thiện mô hình nhưng kết quả độ chính xác không cải thiện đáng kể.

5.1.4. Kinh nghiệm

Sau khi hoàn thành đề tài khóa luận, chúng em có thể học được nhiều kỹ năng quan trọng như: Kỹ năng tìm kiếm, đọc hiểu và áp dụng các công trình nghiên cứu và bài báo trong lĩnh vực Machine Learning. Và cách trình bày kết quả nghiên cứu, còn giúp chúng em lựa chọn, phân tích và áp dụng các phương pháp xử lý dữ liệu và mô hình Machine Learning phù hợp với bài toán cụ thể để dự đoán hành vi của khách hàng.

5.2. Hướng phát triển:

Trong tương lai chúng em sẽ tiếp tục nghiên cứu, áp dụng các thuật toán phân loại này vào các lĩnh vực khác để tiếp tục khai thác tiềm năng dữ liệu của nhiều ngành khác nhau với phương pháp học máy này.

Nghiên cứu sâu hơn về các yếu tố ảnh hưởng đến sự chuyển đổi khách hàng trong các ứng dụng di động, bao gồm các yếu tố liên quan đến sản phẩm, dịch vụ, chính sách, giá cả, kinh nghiệm người dùng, v.v.

Tìm hiểu và áp dụng các phương pháp phân tích dữ liệu mới và nâng cao để cải thiện độ chính xác của các mô hình dự đoán hành vi của khách hàng, bao gồm các phương pháp học sâu và học tăng cường.

TÀI LIỆU THAM KHẢO

[1] Những người đóng góp vào các dự án Wikimedia. (2023). https://vi.wikipedia.org/wiki/H%E1%BB%8Dc_m%C3%A1y

[2] Sara Brown (May 11, 2023), Machine learning, explained, Receive from https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained

[3] Speech and Language Processing. Daniel Jurafsky & James H. Martin. Copyright 2023. All rights reserved. Draft of January 7, 2023.

[4] CS245 Project Report 2 KNN Classification with Different Distance Metrics Litao Zhou, Yuxiang Lu, Ren Zhou May 6, 2021

[5] Kittipong Chomboon*, Pasapitch Chujai, Pongsakorn Teerarassamee, Kittisak Kerdprasop, Nittaya Kerdprasop (2015), School of Computer Engineering, Institute of Engineering, Suranaree University of Technology, Nakhorn Ratchasima 3000, Thailand.

[6] Smith, Karl (2013), *Precalculus: A Functional Approach to Graphing and Problem Solving*, Jones & Bartlett Publishers, p. 8, ISBN 978-0-7637-5177-7

[7] Cohen, David (2004), *Precalculus: A Problems-Oriented Approach* (6th ed.), Cengage Learning, p. 698, ISBN 978-0-534-40212-9

[8] Sarang Anil Gokte, Praxis Business School (2020) , Most Popular Distance Metrics Used in KNN and When to Use Them , Receive from https://www.kdnuggets.com/2020/11/most-popular-distance-metrics-knn.html

[9] Tristan Fletcher (2008), Support Vector Machines Explained, pp 2 - 4

[10] Xiaojin Zhu (2010), Support Vector Machines, CS769 Spring 2010 Advanced Natural Language Processing, pp 1-3

[11] Nikolay Stanevski, Dimiter Tsvetkov (2005), Usìng Support Vector Machine as a Binary Classifier, International Conference on Computer Systems and Technologies – CompSysTech' 2005,

[12] Mikhail Kanevski, Aleksey Pozdnukhov, Stephane Canu, Michel Maignan, Patrick Wong, Syed Shibli (2001), SUPPORT VECTOR MACHINES FOR CLASSIFICATION AND MAPPING OF RESERVOIR DATA, IDIAP RR-01-04, pp 7-9

[13] Tristan Fletcher (2008), Support Vector Machines Explained, pp 7 – 8

[14] Kaitlin Kirasich, Trace Smith, Bivin Sadler (2018), Random Forest vs Logistic Regression: Binary Classification for Heterogeneous Datasets, vol. 1, num. 3, article 9.

[15] Vijaykumar B (B091956), Vikramkumar (B092633), Trilochan (B092654), Bayes and Naive-Bayes Classifier, Rajiv Gandhi University of Knowledge Technologies Andhra Pradesh, India.

[16] Wikipedia contributors. (2023). Naive Bayes classifier. Wikipedia. Retrieved from. https://en.wikipedia.org/w/index.php?title=Naive_Bayes_classifier&oldid=1154199322

[17] Arun D. Kulkarni, University of Texas at Tyler, Barrett Lowe (2016), Random Forest Algorithm for Land Cover Classification, International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 4 Issue: 3 58 - 63

[18] Cuong Nguyen, Yong Wang, Ha Nam Nguyen (2013), Random forest classifier combined with feature selection for breast cancer diagnosis and prognostic, J. Biomedical Science and Engineering, 2013, 6, 551-560

[19] A. Aylin Tokuç (2022), Gradient Descent Equation in Logistic Regression, Receive from https://www.baeldung.com/cs/gradient-descent-logistic-regression

[20] Kathrin Melcher (2018), Understanding Regularization for Logistic Regression, Receive from https://www.knime.com/blog/regularization-for-logistic-regression-l1-l2-gauss-or-laplace

[21] Đạt, V. C, Đảm, N. Đ., & Bình, P. T. Xây dựng bản đồ phân vùng nguy cơ sạt lở đất tại huyện Mường Chà, tỉnh Điện Biên sử dụng các kỹ thuật phân loại K-Nearest-Neighbor và Gradient Boosting. Received from http://tapchikttv.vn/data/article/3517/8.%20Proofreading%201.pdf

[22] Odey Alshboul, Ali Shehadeh, Ghassan Almasabha, Ali Saeed Almuflih (2022), Extreme Gradient Boosting-Based Machine Learning Approach for Green Building Cost Prediction, pp 8-9

[23] *Trí Tuệ Nhân Tạo là gì? Hé LỘ top 10 ứng Dụng Của Trí Tuệ Nhân Tạo*. TMA activities. (n.d.). Retrieved December 6, 2022, from https://www.tma.vn/Hoi-dap/Cam-nang-nghe-nghiep/Tri-Tue-Nhan-Tao-La-Gi-He-Lo-Top-10-Ung-Dung-Cua-Tri-Tue-Nhan-Tao/66349

[24] customer_churn_data. (2023, March 24). Retrieved from

https://www.kaggle.com/datasets/canhlu/customer-churn-data