HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

HOÀNG MẠNH HƯNG

ỨNG DỤNG MÁY HỌC ĐỂ DỰ ĐOÁN NGHỀ NGHIỆP CỦA THUÊ BAO DI DỘNG

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI - 2021

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

HOÀNG MẠNH HƯNG

ỨNG DỤNG MÁY HỌC ĐỂ DỰ ĐOÁN NGHỀ NGHIỆP CỦA THUÊ BAO DI DỘNG

CHUYÊN NGÀNH : **HỆ THỐNG THÔNG TIN**

MÃ SỐ: **8.48.01.04**

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC: PGS. TS. TRẦN QUANG ANH

HÀ NỘI - 2021

LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi. Nội dung của luận văn có tham khảo và sử dụng một số thông tin, tài liệu từ các nguồn sách, tạp chí được liệt kê trong danh mục các tài liệu tham khảo và được trích dẫn hợp pháp.

Tôi cũng cam đoan đã thực hiện việc kiểm tra mức độ tương đồng nội dung luận văn qua phần mềm DoIT một cách trung thực và đạt kết quả tương đồng là 8% toàn bộ nội dung luận văn. Bản luận văn kiểm tra qua phần mềm là bản cứng luận văn đã nộp để bảo vệ trước hội đồng. Nếu sai tôi xin chịu các hình thức kỷ luật theo quy định hiện hành của học viện.

Hà Nội, ngày 17 tháng 6 năm 2021

Học viên cao học

Hoàng Mạnh Hưng

LỜI CÁM ƠN

Em xin gửi lời cảm ơn và tri ân tới các thầy cô giáo, cán bộ của Học viện Công nghệ Bưu chính Viễn thông đã giúp đỡ, tạo điều kiện tốt cho em trong quá trình học tập và nghiên cứu chương trình Thạc sĩ.

Em xin gửi lời cảm ơn sâu sắc tới PGS. TS. Trần Quang Anh đã tận tình hướng dẫn, giúp đỡ và động viên em để hoàn thành tốt nhất Luận văn "ỨNG DỤNG MÁY HỌC ĐỂ DỰ ĐOÁN NGHỀ NGHIỆP CỦA THUÊ BAO DI DỘNG".

Do vốn kiến thức lý luận và kinh nghiệm thực tiễn còn chưa đủ sâu rộng nên luận văn không tránh khỏi những thiếu sót nhất định. Em xin trân trọng tiếp thu các ý kiến của các thầy, cô để luận văn được hoàn thiện

Trân trọng cám ơn.

Tác giả

Hoàng Mạnh Hưng

MỤC LỤC

DANH MỤC CÁC TỪ VIẾT TẮT

Kí hiệu	Nghĩa của kí hiệu	Nghĩa tiếng Việt
ANN	Artificial Neural Network	Mạng nơ-ron nhân tạo
DT	Decision Tree	Cây quyết định
RF	Random Forest	Rừng ngẫu nhiên
DRF	Distributed Random Forest	Rừng ngẫu nhiên phân tán
XRT	Extremely Randomized Trees	Cây cực kỳ ngẫu nhiên
GBM	Gradient Boosting Machines	Máy tăng cường độ dốc
GLM	Generalized Linear Model	Mô hình tuyến tính tổng quát
XGBoost	Extreme Gradient Boosting	Máy tăng cường độ dốc cao
AUC	Area Under The Curve	Diện tích dưới đường cong
ROC	Receiver Operating Characteristic	Đường cong đặc trưng
HDFS	Hadoop Distributed File System	Hệ thống tệp phân tán Hadoop
ETL	Extract, Transform and Load	Trích xuất biến đổi và tải dữ liệu vào kho tập trung
WWW	World wide web	Mạng toàn cầu
CDR	Call Detail Records	Các bản ghi chi tiết cuộc gọi
CRM	Customer Relationship Management	Hệ thống quản lý quan hệ khách hàng
SMS	Short Message Services	Dịch vụ tin nhắn
MMS	Multimedia Messaging Service	Dịch vụ tin nhắn đa phương tiện
VAS	Value Added Services	Dịch vụ giá trị gia tăng
ATTT		An toàn thông tin

DANH MỤC CÁC HÌNH

DANH MỤC CÁC BẢNG

CHƯƠNG 1 – TỔNG QUAN NGHIÊN CỨU

1.1. Mô tả chung về nghiên cứu

1.1.1 Giới thiệu

Việc xác định được khách hàng là ai hiện là mối quan tâm hàng đầu của các nhà cung cấp sản phẩm và dịch vụ bán hàng. Nhờ xác định được chân dung khách hàng mà các doanh nghiệp có thể đạt được hiệu quả tối đa khi thực hiện các chiến dịch quảng bá sản phẩm, nhắm vào những đối tượng cụ thể và có các cách tiếp cận hợp lý nhất. Các nhà quảng cáo cũng sẽ có cơ sở để quyết định quảng cáo của mình sẽ có nội dung như thế nào, đặt ở đâu, thời gian nào.. để có thể tiếp cận được tập khách hàng tối đa nhất.

Có rất nhiều biện pháp để xác định rõ ràng một bức tranh hoàn chỉnh về chân dung khách hàng, nhưng trong nghiên cứu này tôi sẽ tập trung vào việc xác định chân dung khách hàng từ tập thuê bao viễn thông. Đóng góp chính trong công việc của tôi là phát triển một mô hình dự đoán nghề nghiệp của thuê bao di động, giúp các nhà khai thác viễn thông dự đoán được đối tượng khách hàng của mình, từ đó có thể đưa ra các gói sản phẩm phù hợp cũng như cung cấp các dịch vụ giá trị gia tăng khác để thu hút cũng như giữ chân khách hàng, gia tăng lợi nhuận doanh nghiệp.

Mô hình được phát triển trong nghiên cứu này sử dụng các kĩ thuật học máy cho mục đích phân lớp nhị phân dựa trên bộ dữ liệu là các đặc trưng được xây dựng từ toàn bộ các bản ghi chi tiết về cuộc gọi, sử dụng dịch vụ để dự đoán một thuê bao có là sinh viên hay không. Để đo lường hiệu suất của mô hình, thước đo tiêu chuẩn AUC được sử

dụng và giá trị AUC đạt được là 94,6% dựa trên thuật toán XGBoost. Mô hình được chuẩn bị và thử nghiệm thông qua ứng dụng Spark và H20 và làm việc trên bộ dữ liệu lớn được cung cấp và mã hóa từ một trong các công ty viễn thông hàng đầu tại Việt Nam. Bộ dữ liệu chứa tất cả các thông tin CDR của khách hàng đã được mã hóa số thuê bao để đảm bảo tính bảo mật và ATTT của khách hàng được sử dụng để huấn luyện, thử nghiệm và đánh giá mô hình. Nghiên cứu cũng thử nghiệm 4 thuật toán: Rừng ngẫu nhiên, mô hình tuyến tính tổng quát (GLM), máy tăng cường Gradien "GBM" và tăng cường độ dốc cao "XGBoost". Tuy nhiên kết quả tốt nhất đã thu được bằng cách sử dụng thuật toán XGBoost, và thuật toán này sẽ được sử dụng để phân loại nghề nghiệp trong bài nghiên cứu này.

1.1.2 Đặt vấn đề và giải quyết bài toán

Viễn thông là một nhân tố quan trọng trong công cuộc phát triển công nghệ số và hội nhập kinh tế. Trong đó Dịch vụ viễn thông đã phát triển mạnh trong thập kỷ qua và thị trường Viễn thông truyền thống những năm gần đây đã trở nên bão hòa. Năm 2019 đạt 125,8 triệu thuê bao di động mặt đất, lớn hơn cả dân số hiện tại là hơn 96 triệu người, trong đó 3 công ty lớn VNPT, Viettel, Mobifone chiếm hơn 97% thị phần [2]. Để phát triển và tồn tại, các công ty viễn thông không chỉ tìm cách phát triển thuê bao mới, giữ chân thuê bao cũ mà còn cần phát triển mạnh mẽ các dịch vụ giá trị gia tăng, đánh trúng vào từng đối tượng cụ thể nhằm tăng trải nghiệm khách hàng và chất lượng dịch vụ. Do đó, nhu cầu định danh khách hàng là một mối quan tâm lớn của các công ty viễn thông. Việc xác định rõ chân dung khách hàng sẽ giúp các công ty viễn thông hiểu rõ nhu cầu, mong muốn của từng đối tượng cụ thể. Từ đó, họ có thể phục vụ để nâng cao trải nghiệm của khách hàng, cải thiện hiệu quả của các chiến lược tiếp thị của họ và phát triển các nguồn doanh thu mới. Với các thông tin về sở thích, hành vi và xu hướng của khách hàng, các công ty có thể tăng lợi nhuận và doanh thu trên toàn bộ chuỗi giá trị viễn thông - từ phát triển sản phẩm và vận hành mạng lưới, đến bán hàng, tiếp thị và dịch vụ khách hàng. Ngoài ra, việc định danh khách hàng còn giúp các công

ty viễn thông phát triển mạnh mẽ mảng quảng cáo, khi họ đã có sẵn hạ tầng và mạng lưới quảng bá, và mức độ tiếp cận gần như đạt tới mức tối đa vì hầu như ai cũng có ít nhất 1 thiết bị di động.

Vậy các công ty viễn thông, họ có trong tay một khối lượng dữ liệu vô cùng lớn thu thập được từ các hành vi sử dụng dịch vụ từ khách hàng của họ, họ lại càng khao khát khai phá thông tin một cách mãnh liệt hơn bao giờ hết. Vì thế, các nhà mạng không chỉ phát triển riêng các ứng dụng nội bộ, mà đồng thời còn thuê các đối tác bên ngoài tới để có thể khai thác một cách tối đa và hiệu quả nhất. Trong dự án này tôi cũng được cung cấp các dữ liệu liên quan tới các hành vi sử dụng dịch vụ viễn thông của các thuê bao di động, từ đó tôi có thể xây dựng các mô hình dự đoán chân dung khách hàng một cách toàn diện và chính xác.

Trong bài nghiên cứu này, tôi sẽ tập trung vào một phần nhỏ trong bức tranh tổng quát về chân dung khách hàng là nghề nghiệp của thuê bao di động, cụ thể là đánh giá xem thuê bao đó có phải là sinh viên hay không. Tôi chuyển về bài toán phân loại nhị phân để đơn giản hóa quá trình lựa chọn và đánh giá mô hình. Dữ liệu được sử dụng bao gồm tất cả các CDR của thuê bao trong suốt 6 tháng trước thời điểm lấy mẫu. Các dữ liệu đã được mã hóa số thuê bao để đảm bảo tính bảo mật về an toàn thông tin cho người dùng, phù hợp với các chính sách và yêu cầu bảo vệ dữ liệu cá nhân từ Bộ TT&TT. Khối lượng của tập dữ liệu này là hơn 50TB, được lưu trữ trên hệ thống tệp phân tán Hadoop (HDFS). Tôi sẽ sử dụng tập dữ liệu này để tổng hợp và trích xuất các đặc trưng cho từng khách hàng, sau đó sử dụng các mô hình học máy để dự đoán xem một thuê bao có là sinh viên hay không. Mở rộng ra, do các nhà mạng khác cũng có cùng bộ dữ liệu tương tự, nên tùy thuộc vào tập mẫu được lựa chọn, nghiên cứu này sẽ giúp cho các nhà phát triển có cách nhìn tổng quát về phương hướng xử lý các loại dữ liệu, cách thức tổng hợp và trích chọn đặc trưng phù hợp cũng như đưa ra thuật toán hiệu quả nhất để dự đoán về chân dung của một thuê bao di động.

1.2. Các đặc điểm về dữ liệu nhà mạng

1.2.1 Giới thiệu về dữ liệu của nhà mạng

Là một trong 3 công ty viễn thông lớn nhất Việt Nam [2], đối tác của tôi có rất nhiều loại dữ liệu bao gồm cả dữ liệu sử dụng dịch vụ của người dùng lẫn dữ liệu hoạt động của hệ thống. Các loại này được phân loại như sau :

- Dữ liệu khách hàng: Nó chứa tất cả các thông tin liên quan đến dịch vụ và hợp đồng của khách hàng. Bao gồm thông tin về các gói cước đã từng đăng kí, các khuyến mại và ưu đãi đã được nhận. Hơn nữa nó còn chứa các thông tin được tạo ra từ hệ thống CRM (thông tin nhân khẩu học của khách hàng như địa chỉ, ngày sinh, giới tính cũng như các giấy tờ pháp lý liên quan như hình ảnh, chứng minh thư, căn cước công dân …)

- Dữ liệu về vị trí: Thông tin về vị trí xảy ra các sự kiện của người dùng cũng được lưu lại dưới dạng mã vị trí. Từ mã vị trí này có thể ánh xạ ra kinh độ, vĩ độ cũng như địa chỉ cụ thể của người dùng tới các mức xã phường.

- Dữ liệu về khiếu nại, chăm sóc khách hàng: Bao gồm các thông tin phản ánh dịch vụ từ khách hàng, cũng như các khiếu nại liên quan tới cước, gói dịch vụ mà khách sở hữu, quan tâm.

- Dữ liệu nhật ký mạng: Chứa các thông tin về tình trạng hoạt động của hệ thống, các bản ghi lưu lại lịch sử hoạt động của ứng dụng, log chi tiết về hệ thống, cũng như lịch sử thay đổi của các phiên bản nâng cấp hệ thống.

- Dữ liệu chi tiết cuộc gọi: Chứa các thông tin chi tiết về cuộc gọi, sms, mms, truy cập internet, vasp.. Dữ liệu này được tạo ra dưới dạng văn bản

- Thông tin thiết bị di động : Nó chứa thông tin về thương hiệu, kiểu máy, loại điện thoại di động, dùng 1 sim hay 2 sim, hỗ trợ 4g hay không..

Các dữ liệu này có kích thước rất lớn và chứa rất nhiều thông tin về nó. Tuy nhiên, do các vấn đề bảo mật và an toàn dữ liệu cho người dùng, cũng như các vấn đề về an toàn thông tin cho chính doanh nghiệp viễn thông, tôi chỉ có thể tiếp cận một số loại dữ liệu như dữ liệu về vị trí, dữ liệu về thiết bị di động và dữ liệu về chi tiết cuộc

gọi. Nhưng đó vẫn chứa một khối lượng rất lớn thông tin về các hành vi và thói quen sử dụng dịch vụ của khách hàng. Tôi đã phải dành rất nhiều thời gian để tìm hiểu và phân tích về cách thức lưu trữ, tần suất xuất dữ liệu cũng như ý nghĩa của từng loại dữ liệu. Đó cũng là một thách thức lớn vì lượng dữ liệu này được thu thập từ nhiều nguồn khác nhau và thiếu các tài liệu mô tả liên quan.

1.2.2 Khối lượng dữ liệu lớn và phức tạp

Vì tôi không biết thông tin nào có thể hữu ích cho quá trình xây dựng mô hình, vì thế tôi phải xử lý tất cả các dữ liệu phản ánh hành vi và hoạt động của tất cả các khách hàng. Các bản ghi về cuộc gọi thoại, SMS, MMS, Internet, VAS, giao dịch cước.. đều phải được tìm hiểu và xử lý. Các trường thông tin phải được hiểu chính xác và cặn kẽ. Điều này thực sự là một khó khăn và mất thời gian, vì có những bảng chứa tới vài trăm cột, và tài liệu mô tả thì không đầy đủ. Ngoài ra, đôi khi do việc chuyển giao giữa các nhân viên nhà mạng không được đầy đủ, nên có một số trường không thể tìm được tài liệu liên quan.

Bộ dữ liệu tôi sử dụng được lấy trong vòng 6 tháng đến thời điểm lấy mẫu. Tuy nhiên, đây vẫn là một khối lượng khổng lồ, với trung bình mỗi ngày xấp xỉ 300GB dạng text, tương đương với việc tôi phải xử lý tất cả hơn 50TB dữ liệu thô. Vì vậy tôi đã sử dụng hệ thống quán trị của Cloudera trên nền tảng tệp dữ liệu phân tán Hadoop (HDFS) để quản lý và nâng cao hiệu suất xử lý, lưu trữ dữ liệu cũng như cung cấp nhiều lợi ích cho các quá trình khai phá dữ liệu sau này.

1.2.3 Dữ liệu đa dạng và trùng lặp

Dữ liệu CDR được đến từ nhiều nguồn khác nhau, do cách lấy, cấu trúc, cũng như cách lưu log khác nhau từ nguồn cung cấp dữ liệu. Do đó, dữ liệu bao gồm rất nhiều thông tin trùng lặp và phân tán của cùng một loại dữ liệu. Ví dụ như cùng là một dữ liệu tin nhắn SMS, có nguồn dữ liệu chỉ lưu các bản tin phát sinh cước, có nguồn dữ

liệu chỉ ghi các bản tin roaming, có nguồn dữ liệu lưu đầy đủ các bản ghi nhưng lại thiếu vài trường dữ liệu so với các nguồn thông tin khác. Vì vậy, tôi phải xử lý tất các các nguồn dữ liệu, hiểu nó và sau đó so sánh và chọn một hoặc kết hợp nhiều nguồn dữ liệu về một bảng thống nhất. Giai đoạn thực hiện ETL dữ liệu này thực sự rất vất vả và mất thời gian, nhưng kết quả đạt được là đã rút gọn đi được 1 nửa số bảng và trường dữ liệu trùng lặp và không cần thiết.

1.2.4 Tập dữ liệu không cân bằng

Tập dữ liệu được tạo không cân bằng vì nó là một trường hợp đặc biệt của bài toán phân loại trong đó sự phân bố của một lớp thường không đồng nhất với một lớp khác. Tập dữ liệu là không cân bằng nếu một trong các danh mục của nó nhỏ hơn hoặc bằng 10% so với tập còn lại.

Mặc dù các thuật toán học máy thường được thiết kế để cải thiện độ chính xác bằng cách giảm sai số, nhưng không phải tất cả chúng đều tính đến việc có các tập không cân bằng và điều đó có thể cho kết quả không tốt. Nói chung, các lớp cần được cần được coi là cân bằng để có tầm quan trọng như nhau trong huấn luyện dữ liệu.

1.2.5 Giá trị bị mất

Các khách hàng khác nhau có thể có các gói dịch vụ khác nhau. Vì thế, có thể có khách hàng có những gói cước, dịch vụ hoặc sản phẩm mà khách hàng khác không có, trong khi họ lại có một số thứ khác.

Ngoài việc giá trị bị mất, còn có thể xảy ra trường hợp dữ liệu bị mất. Đó có thể do lý do khách quan như lỗi hệ thống, mất log, thiếu log hoặc lỗi chủ quan như xử lý sai dữ liệu, đứt đường truyền khiến một số bản ghi bị mất mà không thể khôi phục lại.

Những trường hợp này đều gây khó khăn và gây ảnh hưởng đến chất lượng của mô hình.

1.2.6 Giá trị cố định

Sau khi khai phá dữ liệu, tôi nhận thấy rằng khoảng 50% biến số chứa một hoặc hai giá trị rời rạc và khoảng 80% tất cả các biến phân loại có ít hơn 10 danh mục, 15% biến số và biến phân loại chỉ có một giá trị. Có những biến mà hầu hết giá trị của chúng là 0 hoặc là hằng số. Tôi thấy rằng có khoảng 77% các biến số có hơn 97% giá trị của chúng là 0, là hằng số hoặc rỗng. Những kết quả này chỉ ra rằng, một lượng lớn các biến có thể loại bỏ do chúng không có giá trị hoặc giá trị cố định.

1.3. Phân nhóm nghề nghiệp và dữ liệu mẫu

1.3.1 Lý thuyết chọn mẫu

Tổng thể là tập hợp tất cả các đối tượng khảo sát. Mẫu là một tập hợp nhỏ những phần tử được lấy ra từ một tổng thể lớn, người ta sẽ nghiên cứu những mẫu đó để tìm ra đặc trưng của mẫu. Các đặc trưng của mẫu được sử dụng để suy rộng ra các đặc trưng của tổng thể do nó đại diện. Khi thực hiện nghiên cứu, chúng ta không thể điều tra toàn bộ cái tổng thể do tốn kém rất nhiều chi phí và thời gian, công sức. Vì vậy, chúng ta sẽ thực hiện chọn mẫu, nó cho phép ta vẫn đạt được độ chính xác tương đối mà lại tiết kiệm thời gian và tăng tốc độ nghiên cứu. Tuy nhiên, vì chỉ là một phần nhỏ của tổng thế nên hạn chế của việc lấy mẫu là tồn tại một sai số chọn mẫu.

Có hai cách chọn mẫu : Chọn mẫu theo xác suất và chọn mẫu phi xác suất

- Chọn mẫu theo xác suất là phương pháp chọn mẫu mà khả năng được chọn vào tổng thể của tất cả các đơn vị tổng thế là như nhau. Có 4 cách chọn mẫu theo xác suất là : Chọn mẫu ngẫu nhiên đơn giản, chọn mẫu hệ thống, chọn mẫu phân tầng và chọn mẫu tập trung.

- Chọn mẫu phi xác suất là phương pháp chọn mẫu các đơn vị trong tổng thể chung không có khả năng ngang nhau để lựa chọn vào mẫu nghiên cứu. Cũng có 4 cách chọn mẫu phi xác suất là: Chọn mẫu thuận tiện, chọn mẫu theo phán đoán, chọn mẫu tích lũy nhanh, và chọn mẫu theo định mức[5].

1.3.2 Mẫu nghiên cứu

Trong nghiên cứu này, với định hướng là sẽ dự đoán nghề nghiệp cho toàn bộ thuê bao của một nhà mạng viễn thông cụ thể. Tuy nhiên, do điều kiện thời gian và khó khăn trong lấy mẫu để nghiên cứu. Tôi sẽ thu nhỏ lại bài toán là dự đoán một thuê bao có là sinh viên hay không.

Năm học 2018-2019 cả nước có 237 trường đại học, 58 trường cao đẳng với tổng số 1,526,111 sinh viên đại học và 33,239 sinh viên cao đẳng trải dài trên khắp các miền của đất nước [1]. Vì vậy, để thu thập được số mẫu đại diện cho tất cả là rất khó khăn, tôi đã sử dụng phương pháp chọn mẫu phi xác suất – chọn mẫu thuận tiện để lựa chọn mẫu. . Cụ thể, tôi lựa chọn ra 7,388 sinh viên thuộc nhà mạng mà tôi xem xét từ các trường Đại học Hà Nội, Đại học Y Hà Nội và Đại học Đà Nẵng. Sau đó tôi lựa chọn thêm hơn 8000 mẫu đối lập để tạo thành hơn 15000 nhãn. Nhưng sau khi thực hiện lấy đặc trưng, tôi chỉ còn 6438 mẫu sinh viên và 6990 mẫu đối lập cho tổng thể 13428 mẫu. Một số mẫu do không tìm được thông tin trong nhà mạng, hoặc đã bị ngưng hoạt động, chuyển sang nhà mạng khác trong thời gian thống kê nên tôi đã loại bỏ. Cách lấy mẫu này có ưu điểm là chọn mẫu một cách thuận tiện, dễ tiếp cận và lấy thông tin. Nhưng có nhược điểm là không xác định được sai số lấy mẫu và không kết luận được tổng thể từ kết quả mẫu. Tôi sẽ phải cải thiện bằng cách thu thập thêm nhiều mẫu ngẫu nhiên hơn, loại bỏ các đặc trưng có thể bị phân lập vào nhãn để đảm bảo mô hình có thể đạt được kết quả tốt nhất.

1.4. Kết luận

Như vậy, dữ liệu của nhà mạng là rất phức tạp và đồ sộ. Việc chuẩn hóa dữ liệu này hết sức mất thời gian và công sức. Để có thể tổng hợp ra những đặc trưng mạnh mẽ có tính phân loại cao, tôi phải thử đi thử lại nhiều thuật toán với các tham số khác nhau để đạt được một mô hình tốt nhất. Bằng việc sử dụng phần mềm hỗ trợ H2O, tôi có thể đẩy nhanh quá trình huấn luyện và kiểm tra. Chế độ mạnh mẽ nhất của H2O là AutoML, chế độ này sẽ sử dụng 4 mô hình cơ bản là eXtreme Gradient

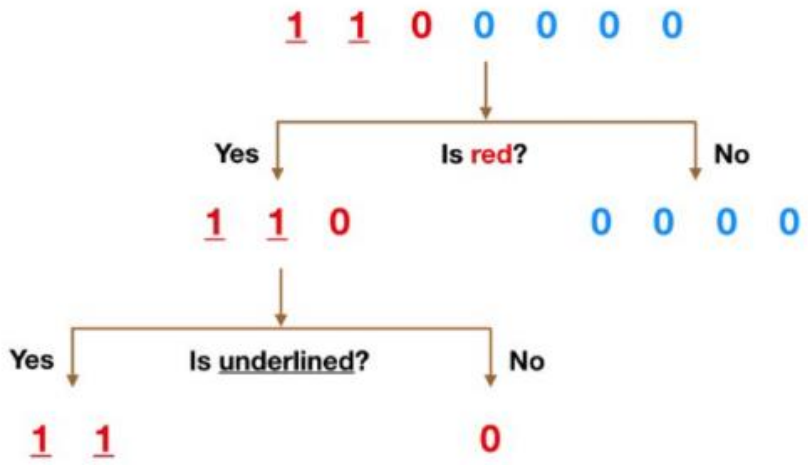
Boosting(XGBoost), Gradient Boosting Machine (GBM), General Linear Model (GLM), Distributed Random Forest (DRF) để thực hiện huấn luyện. Tôi sẽ giới thiệu mô hình lý thuyết và cách thức thực hiện huấn luyện và kiểm tra mô hình ứng với các thuật toán này trong các chương sau.

CHƯƠNG 2 – MỘT SỐ THUẬT TOÁN HỌC MÁY LIÊN QUAN

Trong chương này chúng ta sẽ tiếp cận một số thuật toán về học máy, từ các thuật toán đơn giản như cây quyết định đến thuật toán phức tạp hơn như XGBoost.

2.1 Cây quyết định

Cây quyết định (Decision Tree) là một mô hình thuộc nhóm thuật toán Học có giám sát (Supervised Learning). Cây quyết định là một cây phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào dãy các luật. Các thuộc tính của đối tượngncó thể thuộc các kiểu dữ liệu khác nhau như Nhị phân (Binary) , Định danh (Nominal), Thứ tự (Ordinal), Số lượng (Quantitative) trong khi đó thuộc tính phân lớp phải có kiểu dữ liệu là Binary hoặc Ordinal. Tóm lại, cho dữ liệu về các đối tượng gồm các thuộc tính cùng với lớp (classes) của nó, cây quyết định sẽ sinh ra các luật để dự đoán lớp của các dữ liệu chưa biết.



Hình 2.1 Mô hình diễu diễn cây quyết định

Có nhiều thuật toán khác nhau được đề xuất và sử dụng để hình thành cây quyết định. Trong phần này sẽ giới thiệu về thuật toán ID3 (Iterative Dichotomiser 3) do Ross Quinlan phát triển và có một phiên bản cải tiến gọi là C4.5. Nhiệm vụ của thuật toán học là xây dựng cây quyết định phù hợp với tập dữ liệu huấn luyện, tức là cây quyết định có đầu ra giống (nhiều nhất) với nhãn phân loại cho trong tập mẫu. Để bắt đầu, thuật toán học lựa chọn thuộc tính cho nút gốc. Thuộc tính được lựa chọn là thuộc tính cho phép phân chia tốt nhất các đối tượng thành những tập con, sao cho mỗi tập con càng đồng nhất càng tốt. Thuật toán ID3 sử dụng *entropy* làm mức đo độ đồng nhất của tập dữ liệu.

Entropy là thuật ngữ thuộc Nhiệt động lực học, là thước đo của sự biến đổi, hỗn loạn hoặc ngẫu nhiên. Năm 1948, Shannon [3] đã mở rộng khái niệm Entropy sang lĩnh vực nghiên cứu, thống kê với công thức như sau:

Với một phân phối xác suất của một biến rời rạc x có thể nhận n giá trị khác nhau $x_1, x_2, .., x_n$. Giả sử rằng xác suất để x nhận các giá trị này là p_i = p(x=x_i).

Ký hiệu phân phối này là p = ($p_1, p_2, \ldots, p_n$). Entropy của phân phối này được định nghĩa là: $H(p) = -\sum_{n=1}^{n} p_i \log(p_i)$

Sử dụng entropy như độ đo mức đồng nhất của tập mẫu, ta có thể đánh giá độ tốt của thuộc tính bằng cách so sánh entropy trước và sau khi tập mẫu được phân chia thành tập con theo giá trị của thuộc tính. Sự chênh lệch entropy trước và sau khi phân chia một tập dữ liệu bằng một thuộc tính nào đó được gọi là độ tăng thông tin. Độ tăng thông tin (Information Gain) kí hiệu là IG là chỉ số đánh giá độ tốt của thuộc tính trong việc phân chia tập dữ liệu thành những tập con đồng nhất. Để xây dựng một cây quyết định, ta phải tìm tất cả thuộc tính trả về IG cao nhất.

Để xác định các nút trong mô hình cây quyết định, ta thực hiện tính IG tại mỗi nút theo trình tự sau:

Bước 1: Tính toán hệ số Entropy của biến mục tiêu S có N phần tử với N_c phần tử thuộc lớp c cho trước:

$$H(S)= -\sum_{c=1}^{c} (N_c/N) \log(N_c/N)$$

Bước 2: Tính hàm số Entropy tại mỗi thuộc tính: với thuộc tính x, các điểm dữ liệu trong S được chia ra K child node S_1, S_2, …, S_K với số điểm trong mỗi node con lần lượt là m_1, m_2 ,…, m_K , ta có:

$$H(x, S) = \sum_{k=1}^{K} (m_k / N) * H(S_k)$$

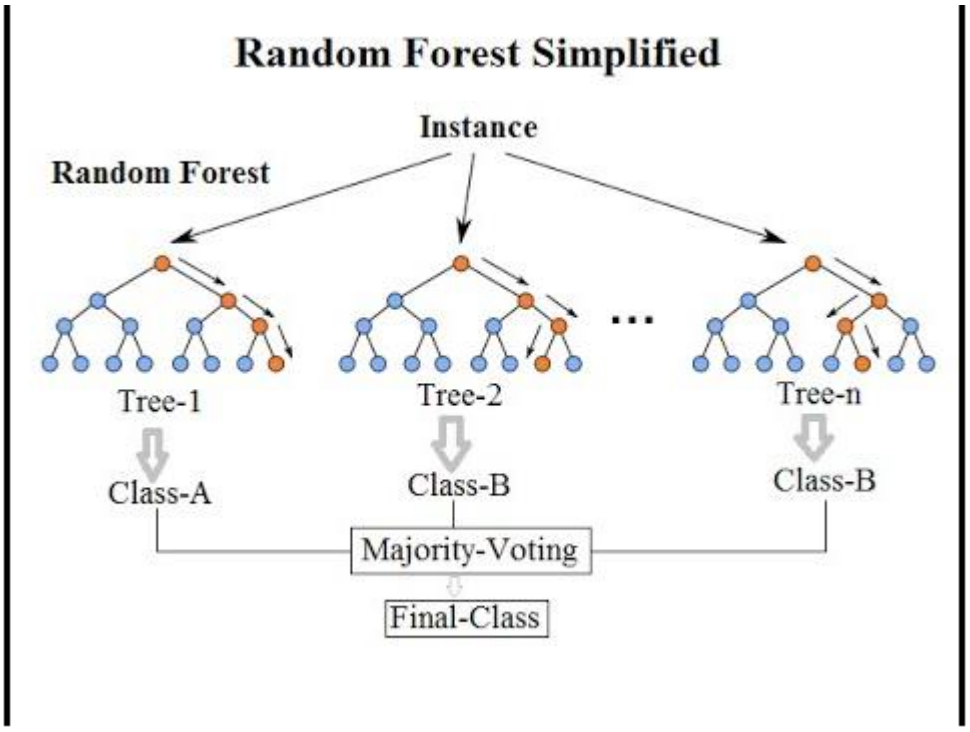
Bước 3: Chỉ số IG được tính bằng:

$$IG(x, S) = H(S) – H(x,S)$$

Giá trị của IG được sử dụng làm tiêu chí để lựa chọn thuộc tính tốt nhất tại mỗi nút. Theo thuật toán ID3, thuộc tính được lựa chọn là thuộc tính có giá trị IG lớn nhất [11].

2.2 Rừng ngẫu nhiên

Rừng ngẫu nhiên là phương pháp xây dựng một tập hợp rất nhiều cây quyết định và sử dụng phương pháp bầu chọn để đưa ra quyết định về biến mục tiêu cần được dự báo. Random Forest (RF) là một tập hợp của hàng trăm cây quyết định (DF), trong đó mỗi cây quyết định được tạo nên ngẫu nhiên từ việc tái chọn mẫu (chọn ngẫu nhiên một phần của dữ liệu để xây dựng) và ngẫu nhiên các biến từ toàn bộ các biến trong trong dữ liệu.



Hình 2.2. Mô hình biểu diễn rừng ngẫu nhiên

Với một cơ chế như vậy, Random Forest cho ta một kết quả chính xác rất cao nhưng đánh đổi bằng việc ta không thể hiểu cơ chế hoạt động của thuật toán này do cấu trúc quá phức tạp của mô hình này - do vậy thuật toán này là một trong những phương thức Black Box - tức ta sẽ bỏ tay vào bên trong và rút ra được kết quả chứ không thể giải thích được cơ chế hoạt động của mô hình [8].

2.3 Mô hình tuyến tính tổng quát

Mô hình tuyến tính tổng quát (GLM) là một sự tổng quát hóa linh hoạt của hồi quy tuyến tính thông thường cho phép các biến đáp ứng có mô hình phân phối lỗi khác với phân phối chuẩn . GLM tổng quát hóa hồi quy tuyến tính bằng cách cho phép mô hình tuyến tính có liên quan đến biến phản hồi thông qua một hàm liên kết và bằng cách cho phép độ lớn của phương sai của mỗi phép đo là một hàm của giá trị dự đoán của nó.

Mô hình tuyến tính tổng quát được xây dựng bởi John Nelder và Robert Wedderburn như một cách thống nhất các mô hình thống kê khác nhau, bao gồm hồi quy tuyến tính , hồi quy logistic và hồi quy Poisson [3]. Họ đề xuất một phương pháp bình phương nhỏ nhất được tái trọng số lặp đi lặp lại để ước tính khả năng tối đa của các tham số mô hình. Ước tính khả năng tối đa vẫn phổ biến và là phương pháp mặc định trên nhiều gói tính toán thống kê. Các cách tiếp cận khác, bao gồm cách tiếp cận Bayes và bình phương nhỏ nhất phù hợp với phương sai ổn định phản hồi, đã được phát triển.

Trong mô hình tuyến tính tổng quát (GLM), mỗi kết quả **Y** của các biến phụ thuộc được giả định được tạo ra từ một phân phối cụ thể theo một họ hàm mũ , một lớp các phân phối xác suất như phân phối chuẩn, nhị thức, phân phối Poisson và gamma... Giá trị trung bình (μ) của phân phối phụ thuộc vào các biến độc lập **X** là:

$$E(Y|X) = \mu = g^{-1}(X\beta)$$

trong đó $E(Y|X)$ là giá trị kỳ vọng của Y có điều kiện trên X ; Xβ là công cụ dự báo tuyến tính, một tổ hợp tuyến tính của các tham số chưa biết β ; g là hàm liên kết.

Khi đó, hàm phương sai V được định nghĩa :

$$Var(Y|X) = V(\mu) = V(g^{-1}(X\beta))$$

Sẽ rất thuận tiện nếu V là một họ phân phối theo cấp số nhân, nhưng nó có thể đơn giản là một hàm của giá trị dự đoán.

Một số đặc điểm của mô hình tuyến tính tổng quát hóa:

- Ước lượng Maximum Likelihood: Khác với mô hình tuyến tính bình thường, mô hình tuyến tính tổng quát hoá không sử dụng ước lượng bình phương tối thiểu (least square estimation) để ước lượng giá trị của hệ số beta. Thay vào đó, nó sử dụng ước lượng Maximum Likelihood (Maximum likelihood estimation: MLE).
- Hệ số xác định (Coefficients of Determination): Trong mô hình tuyến tính bình thường, hệ số xác định (Coefficient of Determination), ký hiệu là R^2, được sử dụng để giải thích và phân tích ý nghĩa của mô hình. Tuy nhiên, trong mô hình tuyến tính tổng

quát hoá, không thể sử dụng R^2 để phân tích và lý giải cho mô hình. Thay vào đó, người ta sử dụng độ lệch (deviance). Một trong những độ lệch hay được sử dụng trong hồi quy logistic là chỉ số Nagelkerke.

Như vậy, việc chuyển các mô hình thông thường về mô hình tuyến tính sử dụng các hàm nối gọi là mô hình tuyến tính tổng quát hóa. Hai mô hình thường gặp nhất là mô hình hồi quy Logistic và mô hình hồi quy Poisson (phân tích tuyến tính log: log linear analytics).

2.3 Các thuật toán Boosting

Có hai thuật toán boosting được giới thiệu trong chương này là Extreme Gradient Boosting (XGBoost) và Gradient Boosting Machine (GBM). Cả hai thuật toán đều phát triển từ thuật toán máy tăng cường độ dốc (gradient boosting). Tuy nhiên XGBoost thường cho kết quả tốt hơn, do nó sử dụng phương thức chính thức hóa mô hình một cách chính quy hơn để kiểm soát việc quá vừa dữ liệu (over-fitting). XGBoost là thuật toán state-of-the-art nhằm giải quyết bài toán học có giám sát (supervised learning) cho độ chính xác khá cao. XGBoost nhận đầu vào là bảng dữ liệu với mọi kích thước và dạng dữ liệu bao gồm cả categorical mà dạng dữ liệu này thường được sử dụng nhiều hơn trong thực tế. Bên cạnh đó, XGboost có tốc độ huấn luyện nhanh, có khả năng scale để tính toán song song trên nhiều server, có thể tăng tốc bằng cách sử dụng GPU, nhờ vậy mà Big Data không phải là vấn đề của mô hình này.

2.3.1 Phát biểu bài toán

y là biến ngẫu nhiên cho đầu ra: nhãn, phân loại, phân nhóm.

$x = \{x_1, x_2, .., x_1\}$ là các vecto đặc trưng của dữ liệu đầu vào.

$\{y_i, x_i\}$ là mẫu dữ liệu dùng để huấn luyện mô hình.

$F^*(x)$ là hàm mục tiêu ánh xạ x sang y.

$L(y, F(x))$ là hàm mất mát:

- Sai số toàn phương : $(y - F)^2$.

- Sai số tuyệt đối : $|y - F|, y \in R^1$ (hồi quy).
- Negative binomial log-likelihood: $\log(1 + e^{-2yF}), y \in \{-1,1\}$ (phân loại).

Mục tiêu của chúng ta tìm được hàm mục tiêu F^* sao cho cực tiểu hoá kỳ vọng của hàm lỗi.

2.3.2 Boosting

Ý tưởng: thay vì xây dựng một mô hình dự đoán (chẳng hạn descision tree) có độ chính xác tương đối, ta đi xây dựng nhiều mô hình dự đoán có độ chính xác kém hơn (weak learner) khi đi riêng lẻ nhưng lại cho độ chính xác cao khi kết hợp lại.Ta có thể hình dung mỗi weak learner gồm học sinh yếu, khá, giỏi và thầy giáo. Trong đó, trọng số uy tín về kiến thức của thầy giáo sẽ là cao nhất và học sinh yếu sẽ là thấp nhất. Khi bạn đặt câu hỏi nào đó và cần những người này đưa ra kết luận, nếu nhiều người cùng có chung kết luận hoặc uy tín của những người đưa ra kết luận cao hơn tập thể thì ta có thể tin kết luận này là đúng.

Ví dụ trong thuật toán AdaBoost, mỗi lần huấn luyện trong tập có độ chính xác kém hơn, mô hình sẽ tính lại trọng số cho các điểm dữ liệu đã bị phân lớp sai, để những lượt huấn luyện tiếp theo những điểm dữ liệu này sẽ có cơ hội nhiều hơn được phân lớp đúng. Dưới đây là mô hình dự đoán tổng quát:

$$H(x) = sign(\propto_1 h_1(x) + \propto_2 h_2(x) + .. + \propto_k h_k(x))$$

2.3.3 Gradient descent

Mục tiêu là đi tìm vector các tham số sao cho tối ưu hoá hàm mục tiêu cụ thể nào đó:

$$\boldsymbol{P}^* = argmin_{\boldsymbol{P}} \Phi(P)$$

Phương pháp Gradient descent:

- Gradient: $\boldsymbol{g}_m = \{g_{jm}\} = \{[\frac{\partial \Phi(\boldsymbol{P})}{\partial P_j}]\}_{\boldsymbol{P}=\boldsymbol{P}_{m-1}}\}$
- Parameters: $\boldsymbol{P}_m = -\rho_m \boldsymbol{g}_m$
- Learning rate: $\rho_m = argmin_\rho \Phi(\boldsymbol{P}_{m-1} - \rho \boldsymbol{g}_m)$
- Target parameters: $\boldsymbol{P}^* = \sum_{m=0}^{M} \boldsymbol{P}_m$

Như vậy, kết quả của gradient descent là kết hợp các trọng số của các gradient.

2.3.4 Kết hợp hai hướng tiếp cận

Như vậy, mục tiêu của chúng ta là đi xây dựng một mô hình phụ gia (additive model):

$$F(x; \{\beta_m, a_m\}_1^M) = \sum_{m=1}^{M} \beta_m h(x; a_m))$$

Nhưng sẽ rất khó nếu ta huấn luyện trực tiếp để tìm tập parameters trong không gian tham số.Vì vậy, ta sẽ dùng greedy-stagewise trong không gian hàm số để giải:

$$(\beta_m, a_m) = argmin_{\beta,\alpha} \sum_{i=1}^{N} L\big(y_i, F_{m-1}(x_i) + \beta h(x_i; a)\big)$$

khi đó : $$F_m(x) = F_{m-1}(x) + \beta_m h(x; a_m)$$

2.3.5 Thuật toán Gradient boosting (GBM)

Thuật toán này nhằm xấp xỉ gradient thông qua một hàm tham số hoá $h(x;\ a_m)$.

Tại mỗi vòng lặp, ta tính gradient $\tilde{y}_m$. Ta xem $\{-\tilde{y}_i, \mathbf{x}_i\}_1^N$ là tập dùng để huấn luyện hàm $h(x;\ a_m)$.. Từ đó, ta có thể dự đoán $-\tilde{y}_m$ từ x.

Thuật toán Gradient_Boost được biểu diễn như sau:

Gradient_Boost()

$F_0(x) = argmin_\rho \sum_{i=1}^{N} L(y_i, \rho)$

For m = 1 to M do:

gradient step

$\tilde{y} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}, i = 1, N$

$a_m = argmin_{\alpha,\beta} \sum_{i=1}^{N} [\tilde{y} - \beta h(x_i; a)]$

boosting step

$\rho_m = argmin_\rho \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + \rho h(x_i; a_m)$

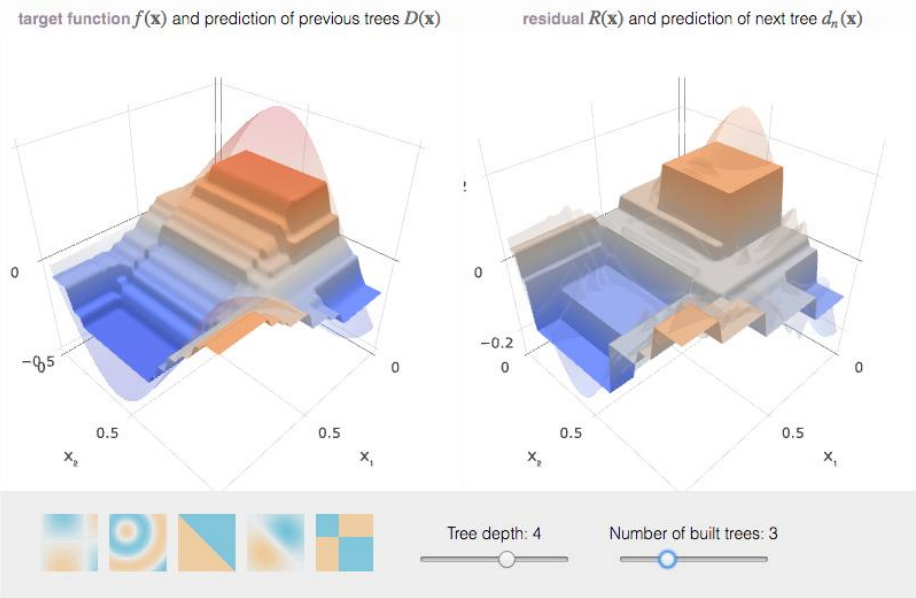
$F_m(x) = F_{m-1}(x) + \rho_m h(x; a_m)$

endFor

EndAlgorithm

Như vậy, $\tilde{y}_i$ vừa là gradient trong không gian hàm số vừa là nhãn trong không gian tham số. β là tốc độ học máy (leanring rate) để tìm tham số a_m và ρ_m là tốc độ học máy để boosting mô hình phụ gia $F_m(x)$. Từ thuật toán cơ sở này, ta có thể mở rộng cho các mô hình khác thông qua các hàm mất mát được định nghĩa trước [9].

2.3.6 Triển khai thuật toán XGBoost



Hình 2.3 Mô hình XGBoost

Đặt:

- n : số lượng mẫu huấn luyện.
- m : số lượng đặc trưng (features).
- $\mathrm{D} = \{(x_i, y_i)\}$ là tập dữ liệu với $|\mathrm{D}| = n, x_i \in \mathbb{R}^m, y_i \in \mathbb{R}$
- q : cấu trúc của một cây, ánh xạ mẫu dữ liệu vào nút lá tương ứng.
- T : số lượng nút lá trên cây.
- f_k : cấu trúc các cây k độc lập của mô hình.
- ω_i : trọng số của nút lá thứ i.

- $\hat{y}_i^{(t)}$: giá trị dự đoán của instance thứ i tại vòng lặp thứ t.
- $f_t^2(x_i)$: đạo hàm bậc 2 của hàm f.
- $I_j = \{i|q(x_i) = j\}$: tập các giá trị tại nút lá j
- I_L : tập giá trị nút lá bên trái.
- I_R : tập giá trị nút lá bên phải.
- $I = I_L \cup I_R$.

Khi đó :

- $I = I_L \cup I_R$.
- Mô hình học: $\hat{y}_i = \emptyset(x_i) = \sum_{k=1}^{K} f_k(x_i),\ f_k \in \mathcal{F}$

  Trong đó, $\mathcal{F} = \{f(x) = \omega_{q(x)}\}(q: \mathbb{R}^m \to T, \omega \in \mathbb{R}^T$
- Hàm học: $\mathcal{L}(\emptyset) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$

  Trong đó, $\Omega(f) = \gamma T + \frac{1}{2}\lambda||\omega||^2$
- Tiến trình học:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t)$$

$$\mathcal{L}^{(t)} \sim \sum_{i=1}^{n}[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)] + \Omega(f_t)$$

với, $g_i = \partial_{\hat{y}(t-1)} l\left(y_i, \hat{y}^{(t-1)}\right)$ và $h_i = \partial^2_{\hat{y}(t-1)} l(y_i, \hat{y}^{(t-1)})$

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n}[g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)] + \lambda T + \frac{1}{2}\lambda\sum_{j=1}^{T}\omega_j^2$$

$$\mathcal{L}^{(t)} = \sum_{j=1}^{T}[\left(\sum_{i \in I_j} g_i\right)\omega_j + \frac{1}{2}(h_j + \lambda)\omega_j^2] + \lambda T$$

- Trọng số tối ưu tại mỗi nút lá: $\omega_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}$
- Hàm lỗi tính trên toàn bộ cây: $\tilde{\mathcal{L}}^{(t)}(q) = -\frac{1}{2}\sum_{j=1}^{T}\frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \lambda T$
- Điều kiện rẽ nhánh: $\mathcal{L}_{split} = \frac{1}{2}\left[\frac{\left(\sum_{i \in I_L} g_i\right)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{\left(\sum_{i \in I_R} g_i\right)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{\left(\sum_{i \in I_j} g_i\right)^2}{\sum_{i \in I_j} h_i + \lambda}\right] - \lambda$

XGBoost và Gradient boosting đều dựa trên cùng ý tưởng đó là boosting thông qua gradient descent trong không gian hàm số. Tuy nhiên, điều làm nên hiệu suất ấn tượng và khả năng tính toán của XGBoost nằm ở ba yếu tố:

- Engineering để tránh quá vừa dữ liệu (overfiting) như: lấy mẫu phụ theo hàng, theo cột, và cột trên mỗi cấp độ phân chia, áp dụng tăng cường chính quy với cả L1 và L2.
- Khả năng tận dụng tài nguyên hệ thống: tính toán song song trên CPU/GPU, tính toán phân tán trên nhiều server, tính toán khi tài nguyên bị giới hạn, tối ưu bộ nhớ đệm để tăng tốc huấn luyện.
- Và cuối cùng là khả năng xử lý các giá trị dữ liệu bị thiếu, tiếp tục huấn luyện bằng mô hình đã được xây dựng trước đó để tiết kiệm thời gian.

Tuy nhiên, xgboost thực sự đề cập đến mục tiêu kỹ thuật để đẩy giới hạn tài nguyên tính toán cho các thuật toán cây được tăng cường. Đó là lý do tại sao nhiều người sử dụng xgboost [12].

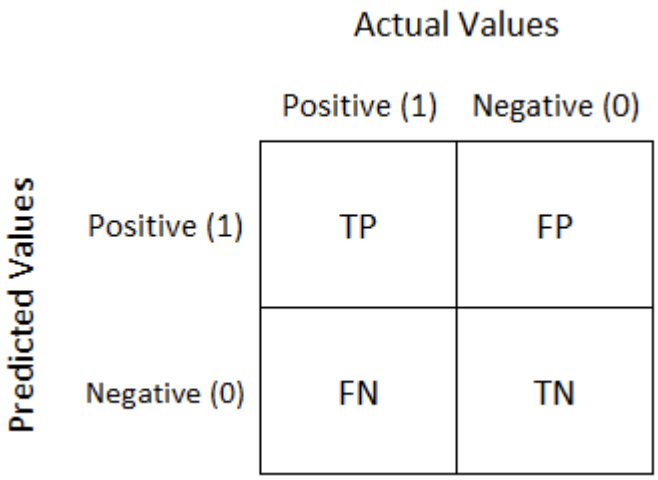
2.4 Đánh giá mô hình

Để đánh giá hiệu quả của mô hình, ta cần có các tiêu chí hay các độ đo sự hiệu quả. Có nhiều độ đo khác nhau có thể sử dụng, tùy vào ứng dụng cụ thể của thuật toán phân loại hoặc hồi quy trong từng trường hợp. Phần này sẽ giới thiệu một số độ đo thông dụng nhất.

2.4.1 Độ đo dùng trong phân loại

Khi xây dựng một mô hình Machine Learning, chúng ta cần một phép đánh giá để xem mô hình sử dụng có hiệu quả không và để so sánh khả năng của các mô hình. Trước tiên, xét trường hợp phân loại hai lớp, trong đó mỗi ví dụ có thể nhận nhãn dương hoặc âm. Với mỗi trường hợp ví dụ mà mô hình dự đoán nhãn, có bốn khả năng xảy ra như liệt kê trên Bảng 2.1, trong đó nhãn thật là nhãn của dữ liệu và nhãn dự đoán là do mô hình tính toán ra:

Bảng 2.1. Ma trận nhầm lẫn



Theo Bảng 2.1, nếu một ví dụ loại dương được mô hình dự đoán là dương thì được gọi là dương đúng (true positive: TP), nếu được dự đoán là âm thì gọi là âm sai (false negative: FN). Một ví dụ loại âm nếu được mô hình dự đoán là dương thì gọi là dương sai (false positive: FP), nếu được dự đoán là âm thì gọi là âm đúng (true negative: TN). Sử dụng các khái niệm TP, TN, FP, FN như trên, có thể định nghĩa một số độ đo hiệu quả phân loại như sau (lưu ý: ta sẽ sử dụng TP, TN, FP, FN để ký hiệu số ví dụ dương đúng, âm đúng, dương sai, âm sai, N là tổng số mẫu):

- N = TP + FP + FN + TN
- Tỷ lệ lỗi : $error = \frac{FP+FN}{N}$
- Độ chính xác : $\text{accuracy} = \frac{TP+TN}{N} = 1 - error$
- Tỉ lệ dương đúng: $tpr = \frac{TP}{TP+FN}$
- Tỉ lệ dương sai: $fpr = \frac{FP}{TP+FN}$
- Độ chính xác precision: $precision = \frac{TP}{TP+FP}$
- Độ thu hồi: $recall = \frac{TP}{TP+FN} = tpr$
- Độ đo F : $F - measure = \frac{precision+recall}{2}$
- Độ nhậy: $sensitivity = \frac{TP}{TP+FN} = tpr$

- Độ cụ thể: $specificity = \frac{TN}{FP+TN}$

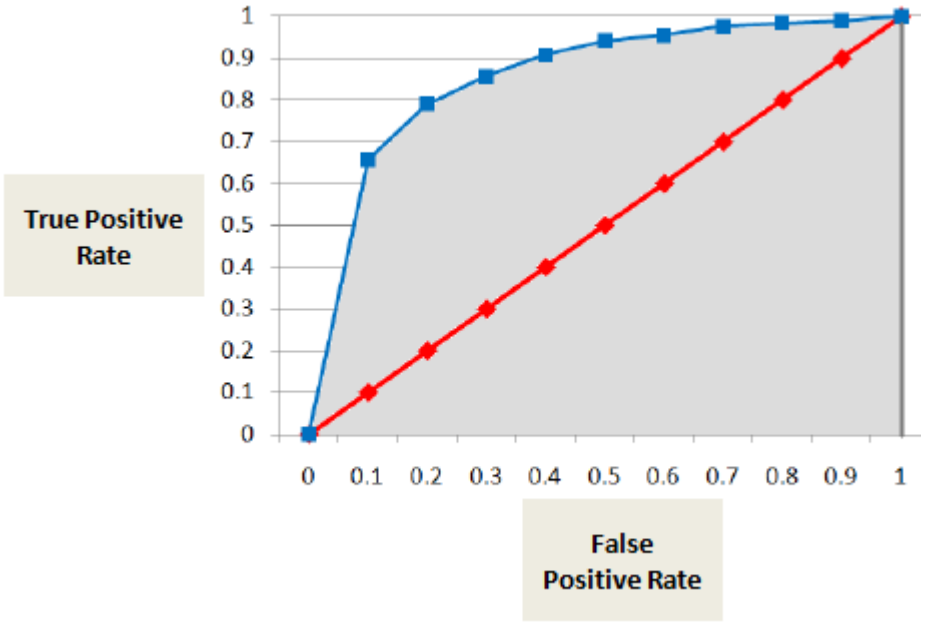
Các độ đo nói trên đều có giá trị nằm trong khoảng [0, 1]. Trong các độ đo nói trên, các độ đo accuracy, recall và precision thường được sử dụng nhất. Độ đo accuracy được dùng khi ta chỉ quan tâm tới độ chính xác nói chung. Độ đo precision và recall được dùng khi ta quan tâm tới hiệu suất phân loại cho một lớp cụ thể. Ví dụ, khi phân loại email thành "thư rác" và "thư bình thường", ta cần quan tâm tới tỷ lệ thư rác phát hiện được, tức là độ đo recall, và tỷ lệ thư rác phát hiện đúng trong số thư rác được dự đoán, tức là độ đo precision. Cần chú ý rằng, khi recall tăng thì precision thường giảm và ngược lại. Ví dụ, trong trường hợp lọc thư rác, ta có thể dự đoán tất cả thư là thư rác, khi đó recall đạt giá trị cực đại bằng 1. Tuy nhiên, khi đó, giá trị p' cũng tăng lên và do vậy precision sẽ giảm đi. Độ đo F cho phép tính tới vấn đề này bằng cách lấy trung bình của hai giá trị precision và recall [10].

Trong trường hợp phân loại nhiều lớp, các độ đo recall, precision, sensitivity, và specificity cho mỗi lớp được tính bằng cách coi đó là lớp dương và tất cả các lớp còn lại được gộp chung thành lớp âm.

2.4.2 ROC và AUC

Đường cong ROC (Receiver operating characteristic) và AUC (Area under Curve) được sử dụng để ước lượng và tính toán hiệu năng của mô hình phân loại. Nó đo lường khả năng phân biệt (discrimination power) của mô hình phân loại. Nói một cách đơn giản, nó kiểm tra khả năng phân biệt các sự kiện trong mô hình phân lớp.

Đường cong ROC biểu diễn tỷ lệ dương tính đúng (tpr) so với tỉ lệ dương tính sai (fpr). AUC được tính là diện tính phía dưới đường cong ROC. AUC cung cấp một thước đo tổng hợp về hiệu suất trên tất cả các ngưỡng phân loại có thể có. Như hình bên dưới, AUC được tính là phần diện tích màu xám. Trong nhiều mô hình học máy, AUC được sử dụng làm thước đo chính để đánh giá mô hình.



Hình 2.4. Đường cong ROC và AUC

2.4.3 Đánh giá mô hình bằng kiểm tra chéo

Khi sử dụng các thuật toán phân loại và hồi quy, một công đoạn quan trọng là đánh giá độ chính xác của mô hình. Việc đánh giá mô hình là cần thiết do hai lý do. Thứ nhất, cần biết mô hình được xây dựng và huấn luyện có độ chính xác ra sao, có đáp ứng yêu cầu của bài toán đặt ra không, trước khi sử dụng mô hình để giải quyết bài toán. Thứ hai, thông thường ta có thể có nhiều mô hình phân loại hoặc hồi quy và cần lựa chọn mô hình tốt nhất trong số đó cho bài toán cần giải quyết.

Một trong những cách đánh giá mô hình là huấn luyện mô hình trên toàn bộ tập dữ liệu huấn luyện, sau đó thử nghiệm độ chính xác trên cùng tập dữ liệu đó bằng cách dùng mô hình đã huấn luyện để dự đoán giá trị đầu ra cho các ví dụ huấn luyện và so sánh giá trị dự đoán với giá trị thực của đầu ra cho các ví dụ. Tuy nhiên, cách đánh giá này là không hợp lý và không nên sử dụng. Lý do là cách đánh giá này không cho kết quả khách quan nếu mô hình bị quá vừa dữ liệu, tức là cho độ chính xác cao trên dữ liệu huấn luyện nhưng lại cho kết quả kém chính xác trên dữ liệu mới.

Thay vì đánh giá mô hình trên cùng bộ dữ liệu đã dùng huấn luyện mô hình, cách đánh giá khách quan hơn là kiểm tra chéo (cross-validation). Có 2 phương pháp kiểm tra chéo thường được sử dụng.

a. Kiểm tra chéo với tập kiểm tra tách riêng (hold-out cross validation)

Phương pháp này có thể gọi là kiểm tra chéo đơn giản, là phương pháp kiểm tra chéo đơn giản nhất và được thực hiện như sau:

- Chia tập dữ liệu huấn luyện S ban đầu một cách ngẫu nhiên thành hai tập con: tập thứ nhất S_{hl} được gọi là tập huấn luyện, và tập thứ hai (phần còn lại) S_{kt} gọi là tập kiểm tra. Thông thường, S_{hl} gồm 70% tập dữ liệu ban đầu và S_{kt} gồm 30% còn lại.
- Huấn luyện mô hình cần đánh giá trên tập S_{hl}
- Đánh giá độ chính xác của mô hình hi trên tập kiểm tra S_{kt}
- Chọn mô hình có độ chính xác cao nhất trên tập kiểm tra để sử dụng (nếu mục đích là lựa chọn mô hình).

b. Kiểm tra chéo k-fold

Một nhược điểm của phương pháp sử dụng bộ dữ liệu kiểm tra riêng là phần dùng để huấn luyện (tập S_{hl}) chỉ còn khoảng 70% tập ban đầu và do vậy bỏ phí quá nhiều dữ liệu để kiểm tra. Do đó người ta thường một phương pháp kiểm tra chéo khác cho phép sử dụng ít dữ liệu kiểm tra hơn. Các bước thực hiện như sau:

- Chia ngẫu nhiên tập dữ liệu ban đầu *S* thành k tập dữ liệu có kích thước (gần) bằng nhau $S_1, S_2, .., S_k$.
- Lặp lại thủ tục sau k lần với $i = 1, .., k$:
    - Dùng tập S_i làm tập kiểm tra. Gộp k-1 tập còn lại thành tập huấn luyện.
    - Huấn luyện mô hình cần đánh giá trên tập huấn luyện.
    - Đánh giá độ chính xác của mô hình trên tập kiểm tra.
- Độ chính xác của mô hình được tính bằng trung bình cộng độ chính xác trên k lần kiểm tra ở bước trên.

- Chọn mô hình có độ chính xác trung bình lớn nhất.

Ưu điểm chính của kiểm tra chéo k-fold là nhiều dữ liệu hơn được sử dụng cho huấn luyện. Mỗi ví dụ được sử dụng để kiểm tra đúng 1 lần, trong khi được sử dụng trong tập huấn luyện k - 1 lần [7]. Nhược điểm của phương pháp này là cần huấn luyện và đánh giá mô hình k lần, do vậy đòi hỏi nhiều thời gian.

Thông thường, phương pháp này được sử dụng với $k = 10$. Giá trị này vừa cho kết quả đánh giá khách quan vừa không đòi hỏi huấn luyện mô hình quá nhiều.

2.5 Kết luận

Trong chương này tôi đã giới thiệu các thuật toán từ cơ bản đến nâng cao sẽ được áp dụng để huấn luyện mô hình. Đồng thời tôi cũng đưa ra các chỉ số để đánh giá chất lượng mô hình, từ đó tôi có thể quyết định lựa chọn sử dụng mô hình nào cho bước cuối cùng là huấn luyện và dự đoán toàn tập thuê bao. Khi đã chuẩn bị được dữ liệu, lựa chọn được các thuật toán sẽ sử dụng, tôi bắt đầu tiến vào bước cuối cùng, bước thực nghiệm. Từ dữ liệu đã có, tôi phải trích xuất ra các đặc trưng, sau đó lựa chọn các đặc trưng mạnh mẽ, cuối cùng áp dụng các thuật toán đã giới thiệu trong chương này để tiến hành huấn luyện, kiểm tra và rút ra kết luận cuối cùng.

CHƯƠNG 3 - ỨNG DỤNG HỌC MÁY ĐỂ PHÂN NHÓM NGHỀ NGHIỆP

3.1 Mô hình đề xuất

Có rất nhiều thuật toán có thể được sử dụng cho bài toán phân loại, nhất là bài toán phân loại nhị phân. Để có thể đánh giá xem thuật toán nào đạt chất lượng tốt nhất, tôi sử dụng phần mềm H2O trên nền tảng của R. H2O cho phép hệ thống tự động lựa chọn các thuật toán phổ biến nhất trong danh sách bao gồm XGBoost, GBM, GLM, DRF (bao gồm DF và XFT), chạy lặp đi lặp lại nhiều lần và sắp xếp chúng theo độ chính xác AUC giảm dần. Sử dụng H2O có thể giúp tôi đẩy nhanh quá trình training và testing, từ đó có thời gian tập trung vào việc xử lý dữ liệu và xây dựng đặc trưng phù hợp, nâng cao chất lượng mô hình.

3.2 Xử lý dữ liệu

3.2.1 Các bảng dữ liệu chính

Dữ liệu telco mà tôi cần xử lý được chia làm rất nhiều bảng với các trường thông tin khác nhau, chúng ta sẽ đi khảo sát các bảng chính sau:

3.2.1.1 Dữ liệu voice, sms

Đây là các dữ liệu phát sinh do hành động gọi thoại, nhắn tin sms, mms của thuê bao. Dữ liệu gốc bao gồm rất nhiều bảng phân ra các hành động riêng biệt như thoại, thoại roaming, thoại trả sau, nhắn tin thường, nhắn tin mất phí, nhắn tin roaming … Vì vậy, trong quá trình ETL (xử lý và làm sạch dữ liệu),tôi đã phải lọc các dữ liệu trùng lặp, các trường dư thừa có giá trị rỗng, bỏ đi các trường mô tả thông tin hệ thống, các trường có giá trị hằng số.. để chỉ giữ lại các thông tin thể hiện sự khác biệt về hành vi người dùng.Thông tin các trường cơ bản của bảng call và sms như trên Bảng 3.1.

Bảng 3.1. Bảng call và sms

Tên cột	Kiểu dữ liệu	Mô tả	Ví dụ

from_isdn_key	int	Số thuê bao thực hiện cuộc gọi	123456789
to_isdn_key	int	Số thuê bao nhận cuộc gọi	156273643
start_time	timestamp	thời gian bắt đầu cuộc gọi	2020-02-01 12:23:34
call_duration	int	Thời gian gọi (giây)	36
imei	string	imei của thiết bị	35932006097721 36
calling_prefix_number	string	đầu số thuê bao gọi	97
called_prefix_number	string	đầu số thuê bao nhận	83
price_plan_id	int	mã gói cước chính	430
roaming_numbe r	int	thông tin roaming	11
chage_basic_acc t	int	số tiền trừ trên tài khoản chính	0
pre_basic_bal	int	số tiền trước cuộc gọi	375605174
basic_bal	int	số tiền sau cuộc gọi	375605174
cell_id	int	vị trí phát sinh cuộc gọi	125367
call_type	int	hướng cuộc gọi	1

log_type	string	loại cuộc gọi	pre_call_out
data_date_key	int	Ngày xử lý dữ liệu	20200201

3.2.1.2 Dữ liệu về số dư hàng ngày

Đây là bảng lưu trữ về số dư hàng ngày của từng thuê bao dữ liệu được xuất ra với tần suất mỗi ngày một lần. Các trường cơ bản của bảng balance như sau:

Bảng 3.2 Bảng balance

Tên cột	Kiểu dữ liệu	Mô tả	Ví dụ
isdn_key	int	Số thuê bao dạng mã hóa	123456789
active_date	timestamp	thời gian kích hoạt	2019-02-01 12:23:34
price_sub_id	int	Gói cước chính	430
sub_state	string	Trạng thái thuê bao	A
bal_1	int	tài khoản chính	13700
bal_10	int	tài khoản khuyến mại	0
data_date_key	int	ngày xuất dữ liệu	20200201

3.2.1.3 Dữ liệu về tiêu dùng hàng tháng

Đây là bảng tổng hợp các thông tin sử dụng cước của thuê bao trong một tháng. Dữ liệu được xuất hàng tháng. Các trường cơ bản của bảng consume như sau:

Bảng 3.3 Bảng consume

Tên cột	Kiểu dữ liệu	Mô tả	Ví dụ
isdn_key	int	Số thuê bao dạng mã hóa	123456789
t_org_cost	int	Tổng tiêu dùng trên tài khoản chính	49142
t_prom_cost	int	Tổng tiêu dùng trên tài khoản khuyến mại (KM)	2491
v_org_cost	int	Cước thoại tài khoản chính	49142
v_prom_cost	int	Cước thoại tài khoản KM	1491
v_int_org_cost	int	Cước thoại nội mạng	43102
v_ext_org_cost	int	Cước thoại ngoại mạng	16040
v_intn_org_cost	int	Cước thoại quốc tế	0
s_org_cost	int	Cước sms tài khoản chính	0
s_prom_cost		Cước sms tài khoản KM	1000
s_int_org_cost	int	Cước nhắn tin nội mạng	1000
s_ext_org_cost	int	Cước nhắn tin ngoại mạng	0
s_intn_org_cost	int	Cước nhắn tin quốc tế	0
data_date_key	int	Ngày xuất dữ liệu	

3.2.1.4 Dữ liệu về sử dụng Internet

Đây là dữ liệu chi tiết về thời gian truy cập và lưu lượng sử dụng mạng internet của thuê bao. Các trường cơ bản của bảng g22 như sau:

Bảng 3.4 Bảng g22

Tên cột	Kiểu dữ liệu	Mô tả	Ví dụ
isdn_key	int	Số thuê bao dạng mã hóa	123456789
begin_time	timestamp	thời gian bắt đầu phiên	2020-02-01 12:23:34
imei	string	Imei thiết bị di động	354530090319292
cell_a	int	vị trí bts của thuê bao	14367
up_data	int	dung lượng kb tải lên	667642
down_data	int	dung lượng kb tải xuống	224297
total_bytes	int	tổng dung lượng trao đổi	891939
charge_basic	int	cước sử dụng	0
price_plan_id	int	gói cước chính	430
ip	string	ip nội bộ	10.136.97.20
data_date_key	int	ngày xuất dữ liệu	20200201

3.2.1.5 Dữ liệu nạp thẻ

Đây là bảng mô tả chi tiết về các lần nạp thẻ của thuê bao. Các trường cơ bản của bảng recharge là:

Bảng 3.5 Bảng recharge

Tên cột	Kiểu dữ liệu	Mô tả	Ví dụ
isdn_key	int	Số thuê bao dạng mã hóa	123456789
sta_datetime	timestamp	thời gian nạp	2020-02-01 12:23:34
charge	int	số tiền nạp	20000
party_code	string	nguồn nạp	MMLSERVER
data_date_key	int	ngày xuất dữ liệu	20200201

3.2.1.6 Dữ liệu ứng tiền

Khi một thuê bao trả trước có số tiền trong tài khoản chính nhỏ hơn 1000 đồng thì có thể ứng trước một khoản tiền của nhà mạng để có thể tiếp tục sử dụng dịch vụ mà không bị gián đoạn. Các trường cơ bản của bảng loan như sau:

Bảng 3.6 Bảng loan

Tên cột	Kiểu dữ liệu	Mô tả	Ví dụ
isdn_key	int	Số thuê bao dạng mã hóa	123456789
loan_time	timestamp	thời gian ứng tiền	2020-02-01 12:23:34
pay_time	timestamp	thời gian trả	2020-02-03

			21:32:19
loan_type	string	0: ứng tiền | 1: trả tiền	0
loan_money	int	tiền ứng	-10000
prebal	int	tài khoản trước ứng	34
bal	int	tài khoản sau ứng	10034
data_date_key	int	ngày xuất dữ liệu	20200201

3.2.1.5 Dữ liệu sử dụng dịch vụ vasp

Đây là bảng thể hiện chi tiết các hành vi sử dụng dịch vụ giá trị gia tăng của thuê bao, như đăng kí, hủy, gia hạn dịch vụ.. Thông tin các trường cơ bản của bảng vas_2g3g và vas_mps như sau:

Bảng 3.7 Bảng vas_2g3g và vas_mps

Tên cột	Kiểu dữ liệu	Mô tả	Ví dụ
isdn_key	int	Số thuê bao dạng mã hóa	123456789
request_time	timestamp	thời gian đăng kí	2020-02-01 12:23:34
response_time	timestamp	thời gian phản hồi kết quả	2020-02-01 12:23:34
response_code	int	mã lỗi khi đăng kí	0
service_name	string	mã dịch vụ	GAME9029
sub_service_name	string	chi tiết dịch vụ	GAME_VNG1

cmd	string	hành động	REGISTER
price	int	cước sử dụng	10000
provider_name	string	nhà cung cấp	VAS_GAME
data_date_key	int	ngày xuất dữ liệu	20200201

Ngoài ra, từ các bảng trên, tôi xây dựng thêm 2 bảng dẫn xuất về thông tin IMEI của thiết bị mà thuê bao sử dụng như số tac, tên thương hiệu, tên thiết bị, hệ điều hành, phiên bản hệ điều hành, thời gian sử dụng.. và thông tin vị trí của người dùng như tên tỉnh, huyện, thời gian xuất hiện, số lần xuất hiện…

3.2.2 Xây dựng đặc trưng

Sau quá trình khảo sát và đánh giá dữ liệu, các bản ghi log cần được xử lý để chuyển từ trạng thái thô thành các đặc trưng để có thể sử dụng trong các thuật toán học máy. Quá trình mất nhiều thời gian nhất do số lượng bảng và cột là rất lớn. Các đặc trưng này được tạo ra từ tất cả các loại CDR, chẳng hạn như mức trung bình của các cuộc gọi khách hàng thực hiện mỗi tháng, mức trung bình của truy cập internet tải lên / tải xuống, số lượng gói đã đăng ký, số lượng dịch vụ vasp sử dụng, số lượng các khoản vay, nạp tiền, tỷ lệ cuộc gọi tính trên số SMS và nhiều tính năng được tạo ra từ dữ liệu tổng hợp của các CDR. Ngoài ra, tôi còn chia nhỏ các dữ liệu ra thành nhiều khung thời gian nhỏ hơn như ngày, giờ, phút và tổng hợp tất cả các thông tin lại một lần nữa.

Không phải tất cả các thông tin đều được sử dụng để xây dựng đặc trưng, tôi nhận thấy có hơn 77% các trường có hơn 97% các giá trị của nó bằng 0, rỗng hoặc là một hằng số cố định. Các thông tin này không có giá trị hữu ích cho quá trình học máy, nên tôi xóa các trường này ra khỏi danh sách xây dựng đặc trưng. Đối với các trường có giá trị thiếu trên 60%, tôi loại bỏ nó ra khỏi mô hình. Đối với các trường hợp còn lại tôi cũng loại bỏ các dòng tương ứng với giá trị thiếu đó.

Sau khi đã khảo sát dữ liệu, tôi bắt đầu xây dựng các đặc trưng và đặt tên chúng theo cú pháp sau:

{ten_bang}_{tổ hợp}_{giá trị đo}

Ngoài ra, có một số đặc trưng khác cấu trúc cũng được bổ sung vào sau. Cụ thể như trên Bảng 3.8.

Bảng 3.8 Tổ hợp các đặc trưng

Tên bảng	Tổ hợp	Giá trị đo	Ví dụ
- call - sms	- outgoing: số cuộc gọi đi - incoming: số cuộc gọi đến - h{xx}: tổng hợp theo khung giờ xx - h{xx}-h{yy}: tổng hợp từ khung giờ xx-yy - ratio: tỉ lệ so với tổ hợp lớn nhất - weekend: đo vào cuối tuần - weekday : đo vào trong tuần	- contacts: số thuê bao - times: số cuộc gọi - duration: thời gian gọi - hours: số giờ ghi nhận - dates: số ngày ghi nhận - months: số tháng ghi nhận	call_outgoing_contacts call_h00_incoming_times call_h20_h24_dates …
Balance	- le_{x}k : số dư ít hơn x ngàn đồng - ge_{x}k : số dư trên x ngàn. - weekend: chỉ lấy cuối tuần - weekday: chỉ lấy trong tuần - ratio: tỉ lệ	- dates: số ngày ghi nhận - avg_balance: trung bình số dư	balance_le_050k_dates balance_weekend_le_020k_dates balance_ratio_le_005k_dates …
consume	- min	- cost: tổng tiêu	consume_origin_cost

	- max - avg - origin: tk chính - internal: nội mạng - external: ngoại mạng - international: quốc tế	dùng - voice_cost: cước thoại - sms_cost: cước sms	consume_min_origin_cost consume_min_origin_external _voice_cost …
g22	- ips : số ip - up_data: tổng up - download_data: tổng down - total_data: tổng sử dụng - charge: số tiền - zero_charges: số lần charge =0 - non_zero_charge: số lần charge > 0	- hours: số giờ ghi nhận - dates: số ngày ghi nhận - months: số tháng ghi nhận	g22_dates g22_h00_down_data g22_h22_total_data
recharge	- {party_code}: nguồn nạp thẻ - bổ sung thêm các khung thời gian	- times: số lần nạp - charge: số tiền nạp - hours, dates, months: thời gian	recharge_times recharge_hours recharge_mmlserver_hours
loan	- loan: tiền vay - pay: tiền trả - diff: chênh lệch vay và trả - bổ sung các khung thời gian	times, dates, hours	loan_loan_months loan_pay_money loan_pay_times loan_h07_diff_money
vas_2g3g vas_mps	- charge: tiền sử dụng dịch vụ - services: dịch vụ -	times, dates, months	vas_2g3g_charge vas_2g3g_weekday_c

	providers: nhà cung cấp - bổ sung các khung thời gian		harge vas_2g3g_times vas_mps_months vas_mps_dates
imei tac	- bổ sung các khung thời gian	- tacs: số mã tac times, dates, months, hours	imei_dates imei_weekday_hours tac_weekday_tacs tac_weekday_dates

Trên đây là các tổ hợp cơ bản nhất để tạo ra các đặc trưng cho từng bảng. Ngoài ra, tôi còn kết hợp thêm một số bảng trích dẫn khác được tạo ra từ các bảng cơ bản để bổ sung thêm mối liên kết giữa các thông tin. Các đặc trưng được phân nhỏ theo từng khung thời gian để phát hiện các thói quen sử dụng của người dùng, từ đó có thể có các đặc trưng mạnh mẽ cho từng đối tượng cụ thể. Kết quả, tôi đã xây dựng được một bộ hơn 5000 đặc trưng để phục vụ cho bước tiếp theo của bài toán.

3.2.2 Trích chọn đặc trưng

Trong bước trước, tôi đã tập hợp được hơn 5000 đặc trưng tất cả. Tuy nhiên trong học máy, không phải cứ càng nhiều đặc trưng thì độ chính xác càng cao, mà ngược lại nó còn gây nhiễu và khiến quá trình học máy tốn thời gian hơn và thiếu chính xác hơn. Vì vậy, tôi đã phải rút gọn bớt các đặc trưng mà vẫn đảm bảo được tính hiệu quả của mô hình.

Tôi sử dụng phương pháp trọng số dấu hiệu (WOE - weight of evidence) để trích chọn các đặc trưng. Phương pháp này sẽ xếp hạng các đặc trưng thành mạnh, trung bình, yếu, không tác động,… dựa trên khả năng, sức mạnh dự đoán. Tiêu chuẩn xếp hạng sẽ là chỉ số giá trị thông tin IV (information value) được tính toán từ phương pháp WOE. Đồng thời mô hình cũng tạo ra các giá trị features cho mỗi biến. Giá trị

này sẽ đo lường sự khác biệt trong phân phối giữa good và bad. Phương pháp WOE sẽ có các kĩ thuật xử lý khác biệt đối với biến liên tục và biến phân loại:

- Trường hợp biến liên tục, WOE sẽ gán nhãn cho mỗi một quan sát theo nhãn giá trị bins mà nó thuộc về. Các bins sẽ là các khoảng liên tiếp được xác định từ biến liên tục sao cho số lượng quan sát ở mỗi bin là bằng nhau. Để xác định các bins thì ta cần xác định số lượng bins. Chúng ta có thể hình dung đầu mút của các khoảng bins chính là các quantile.

- Trường hợp biến phân loại, WOE có thể cân nhắc mỗi một class là một bin hoặc có thể nhóm vài nhóm có số lượng quan sát ít vào một bin. Ngoài ra mức độ chênh lệch giữa phân phối good/bad được đo lường thông qua chỉ số WOE cũng có thể được sử dụng để nhận diện các nhóm có cùng tính chất phân loại. Nếu giá trị WOE của chúng càng gần nhau thì có thể chúng sẽ được nhóm vào một nhóm. Ngoài ra, trường hợp Null cũng có thể được coi là một nhóm riêng biệt nếu số lượng của nó là đáng kể hoặc nhóm vào các nhóm khác nếu nó là thiểu số.

Ta có thể tính WOE = $\ln \frac{\%Good}{\%Bad}$

Trong nghiên cứu ngày, chúng ta có thể coi Good là nhãn của giá trị là sinh viên, và bad là nhãn của giá trị không phải là sinh viên.

Giá trị thông tin (IV – Information Value) là một trong những kỹ thuật hữu ích nhất để chọn các đặc trưng quan trọng trong mô hình dự đoán. Nó giúp xếp hạng các đặc trưng trên cơ sở tầm quan trọng của chúng. IV được tính theo công thức sau:

$$\text{IV} = \sum_{i=1}^{n}(\%Good_i - \%Bad_i) * WOE_i$$

Ta nhận thấy IV luôn nhận giá trị dương vì WOE và (%Good-%Bad) luôn đồng biến. Gía trị IV sẽ cho ta biết mức độ chênh lệch của %Good và %Bad ở mỗi bin là nhiều hay ít. Nếu IV cao thì sự khác biệt trong phân phối giữa %Good và %Bad sẽ lớn và đặc trưng sẽ hữu ích hơn trong việc phân loại mô hình và trái lại IV nhỏ thì đặc trưng ít hữu ích trong việc phân loại mô hình. Một số tài liệu cũng đưa ra tiêu chuẩn phân loại sức mạnh của biến theo giá trị IV như bên dưới:

<= 0.02: Biến không có tác dụng trong việc phân loại

0.02 - 0.1: yếu

0.1 - 0.3: trung bình

0.3 - 0.5: mạnh

=> 0.5: Biến rất mạnh, tuy nhiên trường hợp này cần được điều tra lại để tránh trường hợp biến có mối quan hệ trực tiếp quyết định tính phân loại.

Bằng cách này tôi sẽ rút gọn và lấy ra được các đặc trưng mạnh mẽ nhất để xây dựng mô hình. Thực hiện công việc với mỗi bảng và lấy ra top 100 đặc trưng tốt nhất, tôi rút gọn còn 811 đặc trưng để phục vụ cho giai đoạn tiếp theo.

Bảng 3.9 Bảng mô tả đặc trưng

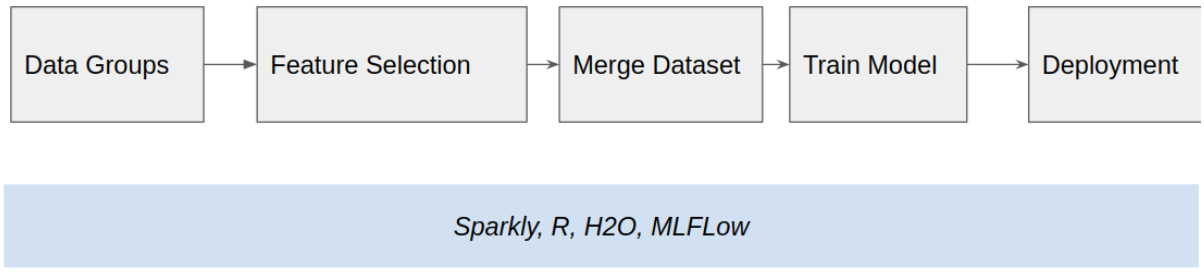
isdn_key <int>	label <chr>	sim_age_months <int>	sim_age_days <int>	balance_weekday_ge_010k_dates <int>	balance_dates <int>
140210475	N	127	3865	17	92
139428761	Y	56	1709	37	92
175807197	N	NA	NA	-1	-1
160554310	N	61	1859	44	92
152913189	Y	45	1374	49	92
116201750	Y	66	2036	7	92
124637675	Y	68	2087	0	92
130703688	Y	96	2950	61	92
136241880	N	133	4064	52	92
122153068	Y	63	1920	0	92

1-10 of 12,102 rows | 1-6 of 813 columns

Previous 1 2 3 4 5 6 ... 100 Next

3.3 Thực nghiệm và kết quả

Sau khi xử lý dữ liệu và trích chọn xong các đặc trưng, tôi tiến hành bước cuối cùng là huấn luyện và đánh giá mô hình.



Hình 3.1 Quy trình đánh giá mô hình

Để thực hiện nhanh việc huấn luyện và kiểm tra mô hình, tôi sử dụng phần mềm H2O trên nền tảng R. Đây là những công cụ mạnh mẽ giúp các nhà phát triển dễ dàng thử nghiệm các mô hình một cách đơn giản và nhanh chóng.

AutoML của H2O có thể được sử dụng để tự động hóa các quy trình học máy, bao gồm đào tạo tự động và điều chỉnh nhiều mô hình trong giới hạn thời gian do người dùng chỉ định. AutoML có thể thực hiện một loạt các hành động huấn luyện và kiểm tra dựa trên rất nhiều thuật toán như Ngoài ra, AutoML còn cung cấp 2 mô hình Stacked Ensemble – một bộ dựa trên tất cả các mô hình đã được đào tạo trước đó, một bộ khác dựa trên các mô hình tốt nhất của mỗi loại thuật toán và thông thường, chúng sẽ cho ra kết quả tốt nhất trong các mô hình được chạy. Các thuật toán sẽ được sử dụng bao gồm eXtreme Gradient Boosting(XGBoost), Gradient Boosting Machine (GBM), General Linear Model (GLM), Distributed Random Forest (DRF) .. Tùy vào thời gian thiết lập cho phép mà AutoML sẽ chạy được số các thuật toán khác nhau, sau đó nó sẽ xếp hạng chúng theo tiêu chí tốt nhất ở trên đầu bảng.

Tôi chia dữ liệu thành hai nhóm: nhóm đào tạo và nhóm thử nghiệm. Nhóm đào tạo gồm 90% tập dữ liệu nhằm mục đích đào tạo các thuật toán, nhóm kiểm tra chứa 10% tập dữ liệu sử dụng để kiểm tra các thuật toán. Cụ thể, dữ liệu thực tế của tôi bao gồm 12102 bản ghi dành cho việc huấn luyện và 1286 bản ghi dành cho việc kiểm thử. Các tham số của thuật toán được tối ưu hóa bằng cách sử dụng xác thực chéo K-lần (K=9). Tôi sử dụng R để thực thi H20. Dữ liệu để huấn luyện của tôi bao gồm 811 đặc trưng.

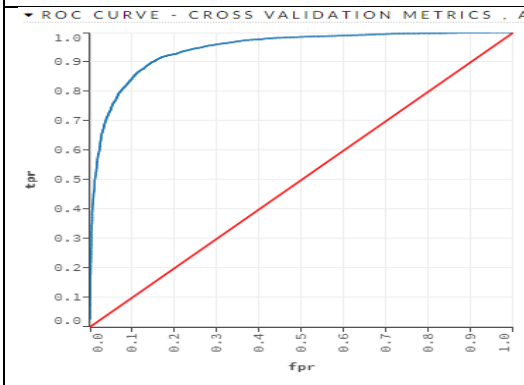
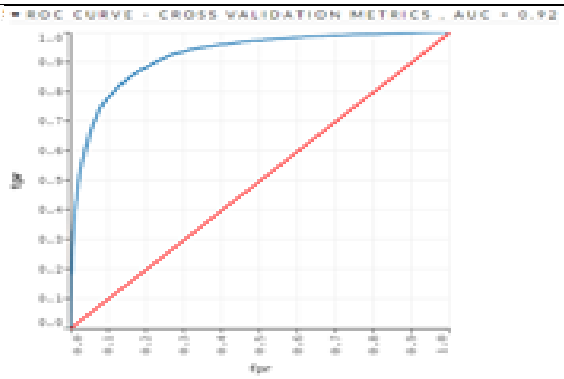
Sau khi cho H2O chạy AutoML, tôi thu được một bảng xếp hạng theo độ chính xác AUC của các thuật toán XGBoost, GBM, GLM, DRF với các tham số khởi chạy ngẫu nhiên. Bảng 3.10 là danh sách 11 thuật toán có độ chính xác cao nhất.

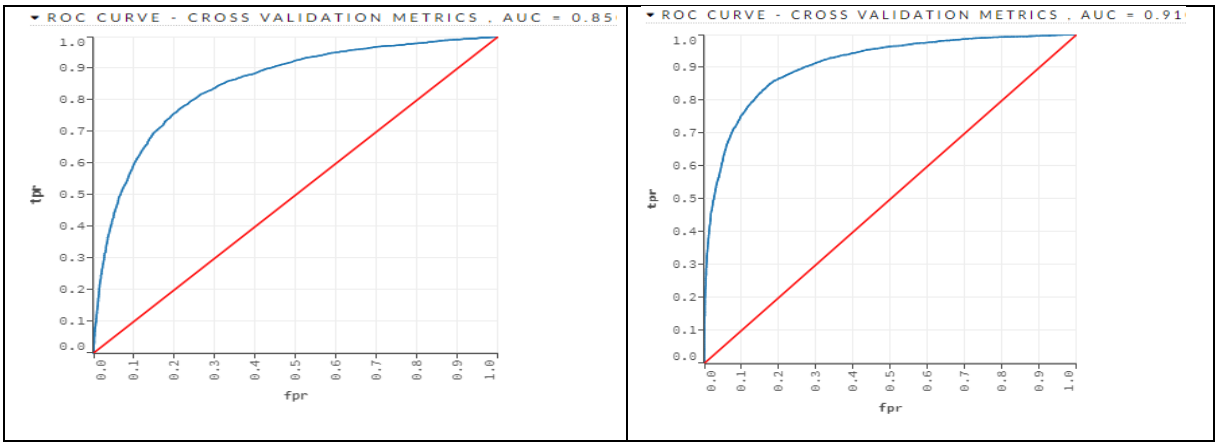
Bảng 3.10 Danh sách các thuật toán triển khai

	model_id	auc	logloss	aucpr	mean_per_class_
0	StackedEnsemble_AllModels_AutoML_20210609_002811	0.9497677153738137	0.28606582052759394	0.9469150884401346	0.1170641982526
1	StackedEnsemble_BestOfFamily_AutoML_20210609_002811	0.9478811899311	0.2914783971570033	0.9439908759675965	0.1212370710144
2	XGBoost_grid__1_AutoML_20210609_002811_model_2	0.9457839076817162	0.2976819808622501	0.9418718025765045	0.1233685027237
3	GBM_grid__1_AutoML_20210609_002811_model_1	0.9442256678268377	0.30215738936474423	0.940926021236419	0.1249659324988
4	XGBoost_grid__1_AutoML_20210609_002811_model_3	0.9435203995213612	0.30302015911450586	0.9396156570251059	0.1273919511089
5	XGBoost_grid__1_AutoML_20210609_002811_model_5	0.9426119282697425	0.3066418306252005	0.9389862440830209	0.1275566744436
6	GBM_grid__1_AutoML_20210609_002811_model_4	0.9424627907485016	0.30877330100129674	0.9390695254959427	0.1283730377720
7	XGBoost_grid__1_AutoML_20210609_002811_model_1	0.9418519252099509	0.31972009333174195	0.9381840493688117	0.1296144799173
8	GBM_5_AutoML_20210609_002811	0.9418080226643136	0.3111425804265272	0.9391739941025811	0.1305564859386
9	GBM_1_AutoML_20210609_002811	0.9417687371256536	0.3080610315331768	0.9388137867177171	0.1298012091640
10	GBM_grid__1_AutoML_20210609_002811_model_2	0.9415701648462729	0.3090118191897934	0.9392709830171573	0.1301226512001

Có thể nhận thấy, ngoài mô hình StackedEnsemble là mô hình tổng hợp các họ đặc trưng tốt nhất dựa trên H2O thì các mô hình có độ chính xác cao nhất là sử dụng thuật toán GBM và XGBoost. Trong quá trình huấn luyện, mô hình XGBoost đang cho kết quả tốt nhất với giá trị AUC đạt 94.6%, GBM đạt giá trị AUC tốt nhất là 94,4%. Tôi sẽ chọn mô hình có kết quả tốt nhất của 4 thuật toán XGBoost, GBM, GLM và DRF để tiến hành xem xét và đánh giá chi tiết.

Trước hết chúng ta hãy xem xét đường cong ROC validation sau khi thực hiện xác thực 10-fold, có thể nhận thấy cả ba mô hình đều có khả năng phân loại rất tốt. Trong đó, ở quá trình kiểm thử, XGBoost là tốt nhất với mức AUC = 94.5%, GBM đạt 92,3%, DRF đạt 91,5%, còn GLM thì tệ nhất nhưng vẫn đạt 89,4%.

XGBoost	GBM
	
GLM	DRF



Hình 3.2 Đường cong ROC validation

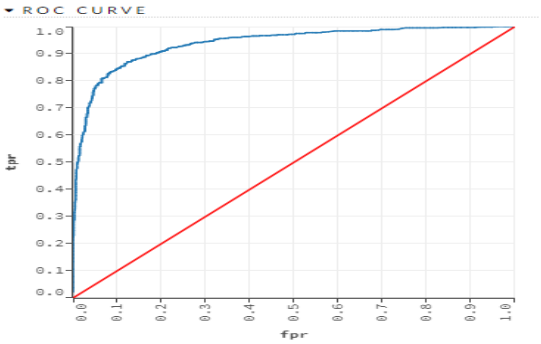
Như vậy ta có thể thấy, thuật toán XGBoost đang đạt hiệu quả cao nhất. Sử dụng mô hình này chúng ta kiểm tra lại kết quả trên tập dùng để thử nghiệm đã được tách ra từ trước cho kết quả như trên Bảng 3.11.

Bảng 3.11 Ma trận nhầm lẫn khi thực hiện dự đoán trên mẫu kiểm thử

PREDICTION - CONFUSION MATRIX ROW LABELS:

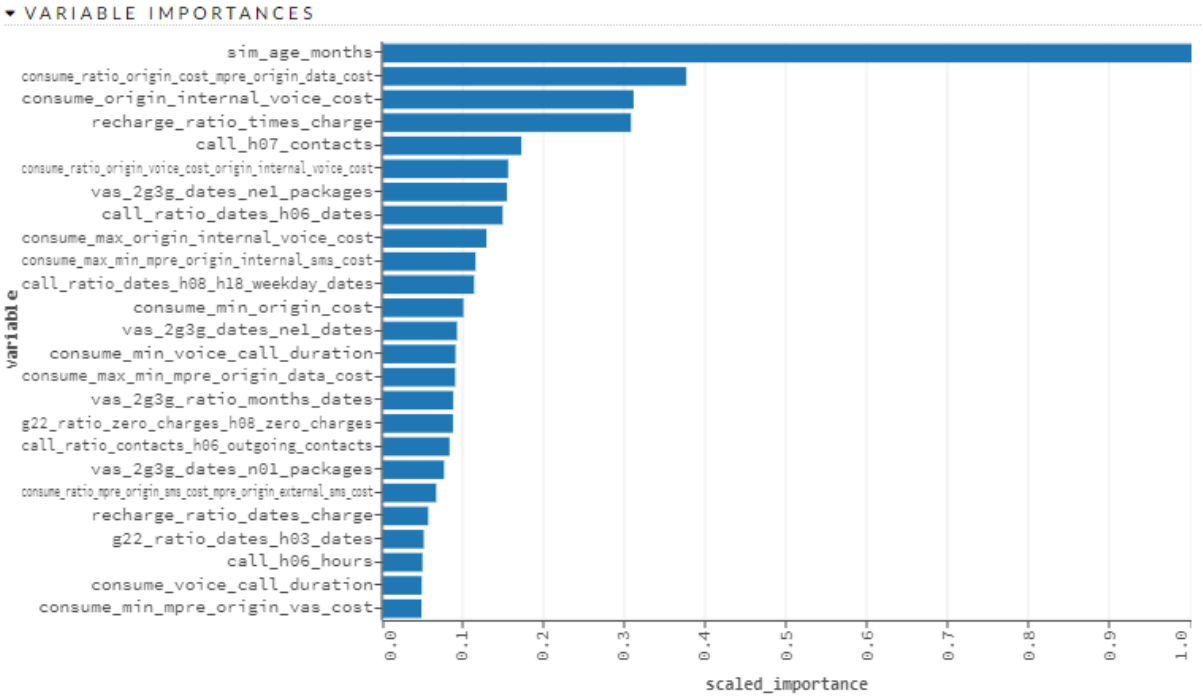
0	580	82	0.1239	82 / 662	0.88
1	82	551	0.1295	82 / 633	0.87
Total	662	633	0.1266	164 / 1,295	
Recall	0.88	0.87			

Chúng ta có thể thấy, mô hình XGBoost đã đoán trúng được 1131 mẫu trong tổng số 1295 quan sát, độ chính xác mà mô hình đạt được là 87,3%. Sử dụng các giá trị của bảng ma trận nhầm lẫn, chúng ta dựng nên biểu đồ đường cong ROC và tính ra được AUC = 93.8%.



Hình 3.3 Đường cong ROC cho mẫu kiểm thử

Như vậy, thuật toán XGBoost đã dựng nên một mô hình phân loại tốt. Bước tiếp theo tôi sẽ xem xét các đặc trưng quan trọng nhất mà mô hình đã sử dụng.



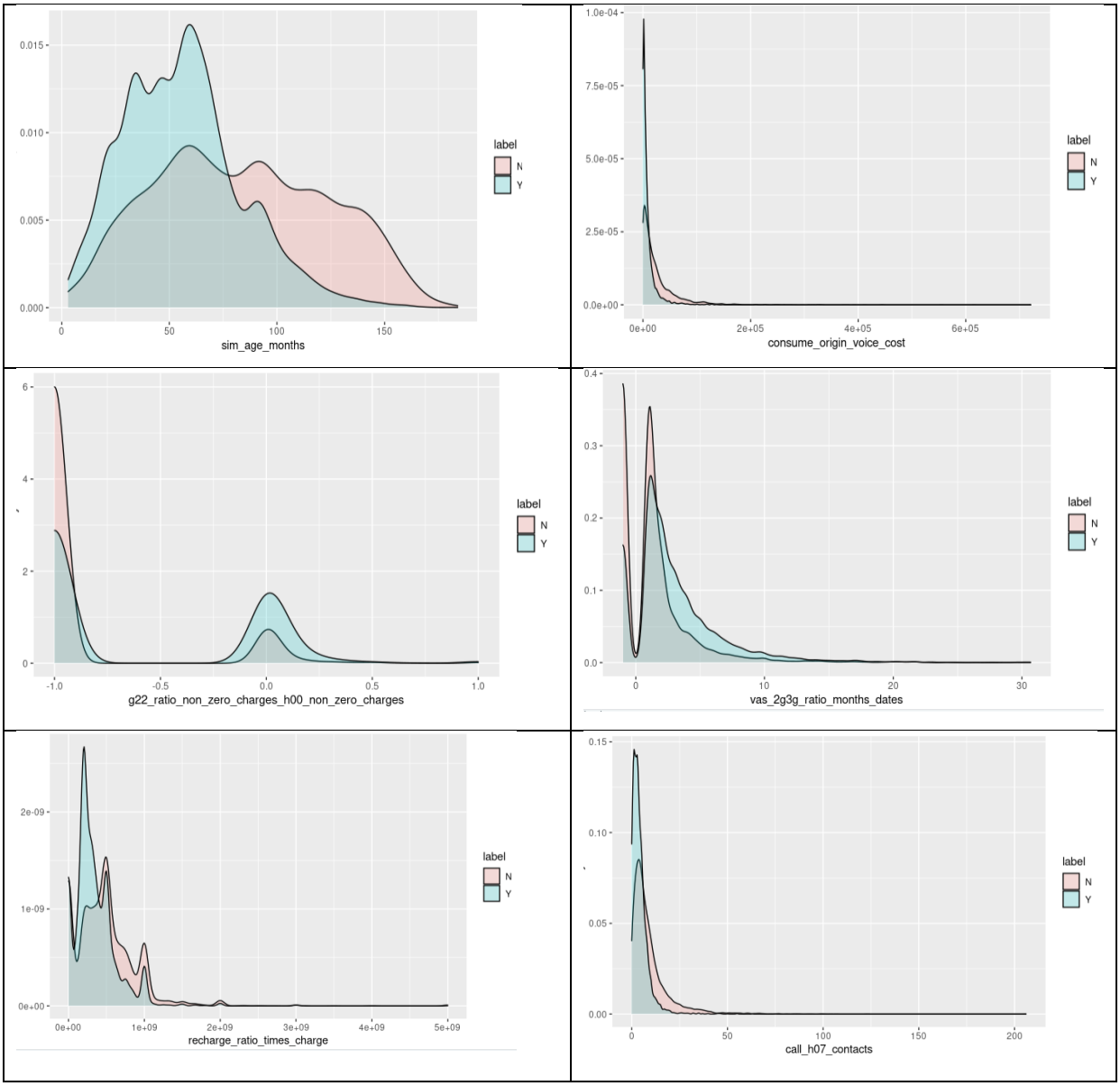
Hình 3.4 Top các đặc trưng theo độ quan trọng

Có thể thấy, tuổi sim (sim_age_months) đang có trọng số cao nhất trong mô hình, điều này có lẽ đúng khi đa phần sinh viên đều là người mới sử dụng điện thoại, hoặc có thói quen thay đổi sim liên tục, không giống như lớp người đi làm, thường sẽ sử dụng cố định một số di động. Tiếp theo là tỉ lệ sử cước dụng dịch vụ giá trị gia tăng (vas) và cước dùng 3g (consume_ratio_origin_cost_mpre_origin_data_cost), tỉ lệ cước gọi nội mạng (consume_origin_internal_voice_cost) trên tổng cước thoại, tỉ lệ số lần nạp tiền trên tổng số tiền nạp (recharge_ratio_times_charge), số người gọi trong khung giờ 7h (call_h07_contacts) .. Các chỉ số này khá phù hợp với lịch trình và điều kiện sinh hoạt chung của sinh viên.

Sau khi xác định được các đặc trưng, chúng ta phải quay lại bước tổng hợp đặc trưng, xem các đặc trưng có phân bố hợp lý hay không, có đặc trưng nào bị thiên lệch.

Sau khi loại bỏ các đặc trưng thiên lệch, chúng ta lại quay lại huấn luyện mô hình. Làm đi làm lại các bước nhiều lần, chúng ta sẽ đạt được mô hình tốt nhất để áp dụng dự đoán cho toàn bộ tập thuê bao.

Phân bố của một số đặc trưng có trọng số cao nhất :



Hình 3.5 Biểu đồ histogram

Như vậy, ta có thể thấy phân bố các giá trị của đặc trưng là tương tự nhau giữa các nhãn. Điều này thể hiện các đặc trưng trên có thể đại diện cho sự phân hóa của

nhãn, hay có thể nói thuật toán đã hoạt động khá tốt, các đặc trưng lựa chọn đã khá chính xác, và chúng ta có thể lựa chọn thuật toán XGBoost để thực hiện công việc dự đoán trên toàn bộ tập dữ liệu.

3.4 Kết luận

Như vậy, sau quá trình thực nghiệm, tôi đã chọn ra được mô hình XGBoost là mô hình có chất lượng tốt nhất. Sử dụng mô hình này để áp dụng cho toàn bộ dữ liệu của nhà mạng, quá trình này cũng mất rất nhiều thời gian và công sức, vì khối lượng dữ liệu là vô cùng lớn, vì vậy hệ thống chạy rất lâu mới có kết quả. Sau khi đã phân loại được thuê bao, tôi đã thực hiện gọi điện để kiểm tra bằng tay. Kết quả cũng khá khả quan, khi tỉ lệ đạt là 80%. Vì vậy, trong thời gian tới, để nâng cao chất lượng sản phẩm, tôi sẽ tiếp tục tối ưu phần xử lý dữ liệu và xây dựng đặc trưng. Ngoài ra, dựa vào nền tảng có sẵn từ dự án này, tôi có thể phát triển sang các ngành nghề khác, các bài toán khác trong dự án bài toán dữ liệu lớn của nhà mạng.

KẾT LUẬN

Các nhà mạng viễn thông có một khối lượng dữ liệu lớn và đa dạng về cách hành vi sử dụng di động của khách hàng. Bằng các công cụ học máy hiện đại, chúng ta có thể trích xuất ra rất nhiều thông tin hữu ích từ đó, như chân dung khách hàng, thói quen, sở thích hoặc xu hướng của họ.

Đóng góp của báo cáo này là tôi đã cho thấy sử dụng thuật toán XGBoost dựa trên một bộ các đặc trưng có tính phân lập mạnh mẽ từ các bản ghi thô của dữ liệu viễn thông phức tạp để phân loại các thuê bao có là sinh viên hay không. Tôi đã đánh giá 811 đặc trưng này, và thấy rằng chúng có khả năng bao phủ cho khá nhiều mẫu đánh giá khác nhau. Từ đó, không chỉ là dự đoán rằng một thuê bao có là sinh viên hay không, tôi có thể áp dụng cho việc dự đoán các nghề nghiệp khác tùy thuộc vào mẫu thu thập, hoặc tôi còn có thể áp dụng các bài toán khác như đánh giá sở thích, điểm tín dụng cá nhân, điểm tích cực của một thuê bao…

Việc xác định các đặc trưng là tùy thuộc vào từng cá nhân, từng nhiệm vụ cụ thể, vì vậy ngoài các đặc trưng đã có, tôi vẫn phải tiếp tục nghiên cứu dữ liệu và tìm hiểu thêm các đặc trưng mới. Bởi dữ liệu là vô cùng, và cách kết hợp dữ liệu là vô tận, nên định hướng tiếp theo của nghiên cứu tôi vẫn là xây dựng đặc trưng mới, đánh giá hiệu quả mô hình trên thực tiễn và hiệu chỉnh mô hình khi cần thiết.

DANH MỤC CÁC TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] https://moet.gov.vn/thong-ke/Pages/thong-ke.aspx - Truy cập ngày 20/05/2021

[2] https://vietnamcredit.com.vn/products/vietnam-industries/bao-cao-nganh-vien-thong-viet-nam-2020-54 - Truy cập ngày 20/05/2021

Tiếng Anh

[3] CE Shannon (1948), "A Mathematical Theory of Communication", *Bell System Technical Journal* 27(3), 379-423.

[4] Chawla N (2005), "Data mining for imbalanced datasets: an overview", *Data mining and knowledge discovery handbook*, Berlin: Springer, Berlin, 853-867

[5] Yoav Ben-Shlomo, Sara Brookes, Matthew Hickman (2013). *Lecture Notes: Epidemiology, Evidence-based Medicine and Public Health*, 6th Edition, Wiley-Blackwell, Oxford.

[6] Fawcett, Tom (2006). "An Introduction to ROC Analysis", *Pattern Recognition Letters* 27 (8), 861-874

[7] Kuhn, Max; Johnson, Kjell (2013), *Applied Predictive Modeling*, NY: Springer, New York

[8] Ho, Tin Kam (1995), "Random Decision Forests", *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, pp. 278-282.

[9] Jerome H. Friedman (2001), "Greedy function approximation: A gradient boosting machine.." *Ann. Statist* 29(5), 1189-1232.

[10] Powers, David M W (2011), "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation", *Journal of Machine Learning Technologies* 2 (1), 37-63.

[11] Quinlan, J. R. (1986). "Induction of decision trees", *Machine Learning* 1(1), 81-106

[12] Tianqi Chen, Carlos Guestrin (2016), “XGBoost: A Scalable Tree Boosting System”, “*Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*”, ACM, 785–794.

Kết quả kiểm tra DoIT

DoIT

ĐHQGHN VNU ĐẠI HỌC QUỐC GIA HÀ NỘI Vietnam National University, Hanoi

Hệ thống hỗ trợ nâng cao chất lượng tài liệu

KẾT QUẢ KIỂM TRA TRÙNG LẶP TÀI LIỆU

THÔNG TIN TÀI LIỆU

Tác giả	Hoàng Mạnh Hưng
Tên tài liệu	HoangManhHung_LV_TS_CNTT_2021
Thời gian kiểm tra	17-06-2021, 09:12:03
Thời gian tạo báo cáo	17-06-2021, 09:15:19

KẾT QUẢ KIỂM TRA TRÙNG LẶP

Điểm	8
Nguồn trùng lặp tiêu biểu	[text.123doc.org, machinelearningcoban.com, techtalk.vn]

Học viên	Người hướng dẫn khoa học
Hoàng Mạnh Hưng	**PGS. TS. Trần Quang Anh**