HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

NGUYỄN QUANG TUẤN

MỘT SỐ THUẬT TOÁN HỌC MÁY TRONG PHÂN LOẠI HÀNH VI SỬ DỤNG GÓI CƯỚC DATA VIỄN THÔNG

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

Hà Nội - năm 2020

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

NGUYỄN QUANG TUẤN

MỘT SỐ THUẬT TOÁN HỌC MÁY TRONG PHÂN LOẠI HÀNH VI SỬ DỤNG GÓI CƯỚC DATA VIỄN THÔNG

CHUYÊN NGÀNH: KHOA HỌC MÁY TÍNH

MÃ SỐ: 8.48.01.01

LUẬN VĂN THẠC SĨ KỸ THUẬT
(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC

PGS.TS. TRẦN ĐÌNH QUẾ

Hà Nội - năm 2020

LỜI CAM ĐOAN

Tôi xin cam đoan: Khoá luận tốt nghiệp với đề tài "**MỘT SỐ THUẬT TOÁN HỌC MÁY TRONG PHÂN LOẠI HÀNH VI SỬ DỤNG GÓI CƯỚC DATA VIỄN THÔNG**" là công trình nghiên cứu của cá nhân tôi, các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác, không sao chép của bất cứ ai.

Tôi xin chịu mọi trách nhiệm về công trình nghiên cứu của riêng mình!

Hà Nội, ngày …….

Người cam đoan

Nguyễn Quang Tuấn

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
EDA	Exploratory Data Analysis	Phân tích, khai phá dữ liệu
ETL	Extract Transform Load	Quá trình trích xuất, biến đổi và tải
KPI	Key Performance Indicator	Chỉ số đo lường hiệu quả công việc
PTDL		Phân tích dữ liệu
PYC		Phiếu yêu cầu
RF	Random Forest	Thuật toán rừng ngẫu nhiên
ROC	Receiver operating characteristic	Biểu đồ hiệu năng phân loại
TNR	True Negative Rate	Tỉ lệ dự đoán đúng nhãn âm tính
TPR	True Positive Rate	Tỉ lệ dự đoán đúng nhãn dương tính
TUR	Take up rate	Tỉ lệ phản hồi của khách hàng

DANH MỤC CÁC BẢNG

DANH MỤC CÁC HÌNH

MỞ ĐẦU

Ngày nay, trong kỷ nguyên kỹ thuật số, với sự bùng nổ của thông tin, số lượng dữ liệu do con người tạo ra ngày càng khổng lồ. Số lượng điện thoại smartphone và thiết bị kết nối tăng nhanh chóng, ngành công nghiệp viễn thông tràn ngập với số lượng dữ liệu khổng lồ. Nguồn gốc của số lượng dữ liệu khổng lồ này bao gồm dữ liệu lưu lượng truy cập mạng, mô hình sử dụng dữ liệu của khách hàng, dữ liệu vị trí, ứng dụng đã tải về,… Ngành công nghiệp viễn thông đang ngày càng thay đổi và phát triển không ngừng. Điện thoại thông minh đã trở thành một nhu cầu cơ bản của mỗi người trong cuộc sống ngày nay. Mọi người có thể kết nối với nhau ở bất cứ nơi nào trên thế giới, xóa bỏ rào cản khoảng cách. Mọi thông tin đều có thể được thu thập và xử lý nhanh hơn bao giờ hết. Và phân tích dữ liệu lớn sẽ tạo điều kiện cho các ngành công nghiệp viễn thông phát triển mạnh mẽ trong thế giới kỹ thuật số. Các ứng dụng của phân tích số liệu trong lĩnh vực viễn thông, dữ liệu lớn là một cơ hội chuyển đổi ngành viễn thông sang hướng hoạt động hiệu quả hơn nhờ gia tăng mức độ hài lòng của khách hàng, tăng doanh thu nhờ tăng sản lượng và loại hình dịch vụ cung cấp, cắt giảm chi phí vận hành, giảm thiểu thiệt hại.

Trong khuôn khổ luận văn tập trung vào các kỹ thuật xử lý dữ liệu lớn và các thuật toán phân lớp dữ liệu bao gồm: Phân loại tuyến tính, Hồi quy logistic, Phân loại Naïve Bayes, Rừng ngẫu nhiên (RF). Ứng dụng thuật toán học máy trong lĩnh vực kinh doanh viễn thông sử dụng dữ liệu lịch sử của tập khách hàng để xây dựng các mô hình có khả năng phân loại, dự đoán nhu cầu sử dụng của khách hàng. Tập kết quả đó sẽ được dùng để hỗ trợ các đơn vị kinh doanh truyền thống đưa ra quyết định trong các chiến dịch kinh doanh của doanh nghiệp.

Cấu trúc của bài luận văn gồm 3 chương:

Chương 1: Tổng quan về bài toán phân loại hành vi sử dụng dịch vụ viễn thông: Trong chương này trình bày tổng quan quy trình phân tích dữ liệu, hệ thống xử lý dữ liệu phân tán và các phương pháp xử lý dữ liệu.

Chương 2: Mô hình hành vi và một số thuật toán học máy: Chương này sẽ đi sâu vào tìm hiểu 3 thuật toán là rừng ngẫu nhiên, phân loại Naïve Bayes, hồi quy Logistic.

Chương 3: Thử nghiệm và đánh giá: Chương này sẽ nêu mục tiêu thử nghiệm bài toán, ý nghĩa các chỉ số đo và thử nghiệm xây dựng mô hình dự đoán lần lượt với 3 thuật toán nêu trên và đánh giá kết quả.

CHƯƠNG 1 - TỔNG QUAN VỀ BÀI TOÁN PHÂN LOẠI HÀNH VI SỬ DỤNG DỊCH VỤ VIỄN THÔNG

1.1 Giới thiệu bài toán

Các ứng dụng của phân tích số liệu trong lĩnh vực viễn thông, dữ liệu lớn là một cơ hội chuyển đổi ngành viễn thông sang hướng hoạt động hiệu quả hơn nhờ gia tăng mức độ hài lòng của khách hàng, tăng doanh thu nhờ tăng sản lượng và loại hình dịch vụ cung cấp, cắt giảm chi phí vận hành, giảm thiểu thiệt hại. Trong khuôn khổ luận văn tập trung vào các kỹ thuật xử lý dữ liệu lớn và các thuật toán phân lớp dữ liệu bao gồm: Phân loại tuyến tính, Hồi quy logistic, Phân loại Naïve Bayes, Rừng ngẫu nhiên (RF). Ứng dụng thuật toán học máy trong lĩnh vực kinh doanh viễn thông sử dụng dữ liệu lịch sử của tập khách hàng để xây dựng các mô hình có khả năng phân loại, dự đoán nhu cầu sử dụng của khách hàng. Tập kết quả đó sẽ được dùng để hỗ trợ các đơn vị kinh doanh truyền thống đưa ra quyết định trong các chiến dịch kinh doanh của doanh nghiệp.

1.2 Tổng quan quy trình phân tích dữ liệu

1.2.1 Tổng quan

- Sự kiện bắt đầu: Kinh doanh gửi PYC thực hiện dự án.

- Sự kiện kết thúc: Triển khai theo dõi kết quả và hành vi sau tác động.

- Đầu vào: Tài liệu đánh giá phạm vi mục tiêu của chương trình ứng dụng kinh doanh dựa trên phân tích dữ liệu.

- Đầu ra:

- Bảng dữ liệu sau quá trình mô hình dự đoán
- Chương trình kinh doanh tác động đến khách hàng cuối dựa trên phân tích dữ liệu.
- Báo cáo kết quả đánh giá chương trình.
- Triển khai mở rộng và xây dựng các chiến dịch định kỳ

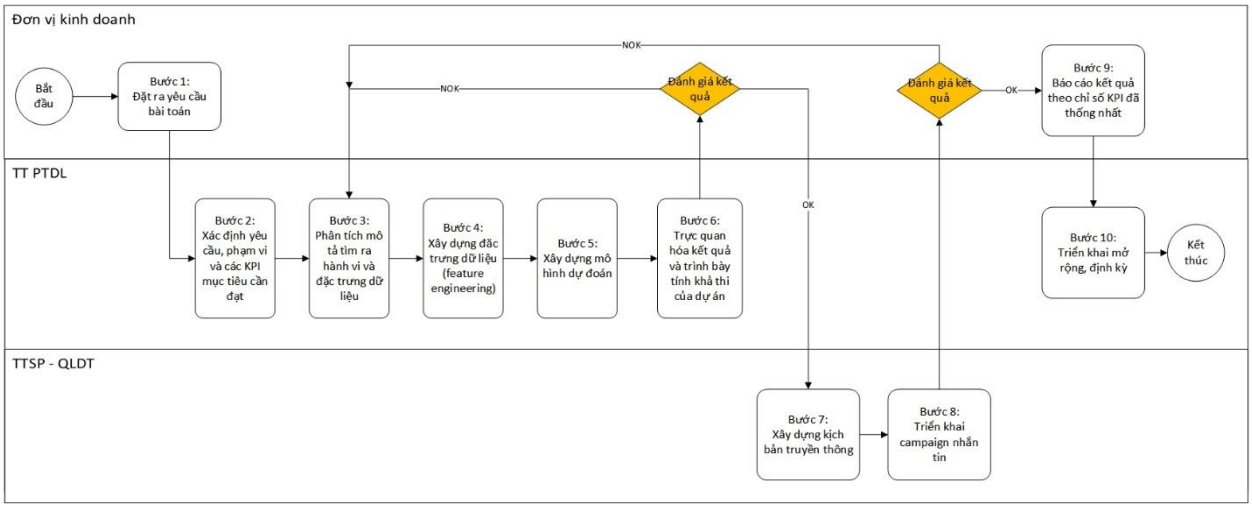
1.2.2 Quy trình triển khai bài toán phân tích dữ liệu

Bảng 1.1: Quy trình triển khai bài toán phân tích dữ liệu

Hoạt động chính	Các nội dung quan trọng
1. Đưa ra yêu cầu bài toán	Đơn vị kinh doanh: đưa ra yêu cầu bài toán, mô tả rõ hiện trạng và mục tiêu đầu ra mong muốn về cả doanh thu và tỷ lệ take up rate.
2. Xác định yêu cầu, phạm vi và các KPI mục tiêu cần đạt	Đơn vị kinh doanh: Đặt ra mục tiêu đầu ra mong muốn về cả doanh thu và tỷ lệ take up rate. Xác định các KPI và con số để đánh giá mô hình dự đoán trong bài toán phân tích. Xác định các KPI về kết quả triển khai của campaign ứng dụng phân tích dữ liệu.
3.Phân tích mô tả (Descriptive analytics)	Hypothesis testing Làm sạch dữ liệu, Khám phá dữ liệu, tìm hiểu và chuẩn bị dữ liệu Kế hoạch phân tích Phân tích và chứng minh giả thiết
4.Xây dựng đặc trưng dữ liệu (feature engineering)	TT PTDL đưa ra danh sách đặc trưng liên quan đến dữ liệu. TTSP sử dụng kết quả trực quan hóa và kinh nghiệm về mặt kinh doanh giúp đóng vai trò tư vấn
5.Xây dựng model dự đoán phù hợp với chương	TT PTDL xây dựng mô hình dự đoán theo các đặc trưng dữ liệu đã thống nhất.

Hoạt động chính	Các nội dung quan trọng
trình (Predictive analystics)	
6.Trực quan hóa kết quả, thuyết phục với đơn vị ra yêu cầu	TT PTDL trực quan hóa đặc tính của các thuê bao được dự đoán. Thuyết phục đơn vị kinh doanh về kết quả đầu ra
7.Xây dựng kịch bản truyền thông	Lựa chọn sản phẩm, offer phù hợp với đặc tính từng nhóm thuê bao Xây dựng kịch bản tác động, nội dung tin nhắn, thời điểm, trigger tác động...
8.Triển khai và theo dõi kết quả	Phối hợp với các đơn vị P.QLDT, TTSP, TKCS để khai báo campaign tác động đến khách hàng cuối. Chia tập tác động thành 2 tập Target Group – để tác động và Control Group - để và theo dõi Xây dựng Dashboard để theo dõi các chỉ số KPI và diễn biến hành vi thuê bao sau tác động
9.Báo cáo kết quả	Báo cáo kết quả chương trình tới BTGĐ
10.Triển khai mở rộng, định kỳ	Nếu kết quả chương trình tốt, triển khai mở rộng và đựng thành luồng định kỳ hàng ngày/hàng tháng

1.2.3 Lưu đồ quy trình thực hiện dự án ứng dụng phân tích dữ liệu



Hình 1.1: Lưu đồ quy trình thực hiện dự án ứng dụng phân tích dữ liệu

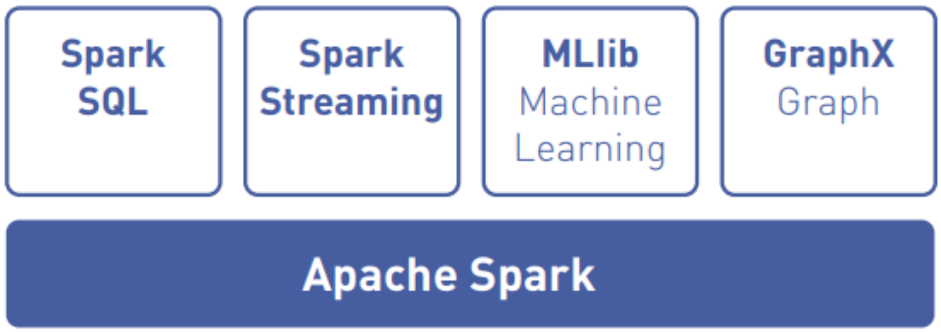
1.3 Xử lý dữ liệu phân tán với Spark

1.3.1 Giới thiệu

Apache Spark là một khung làm việc mã nguồn mở tính toán phân tán được phát triển sơ khởi vào năm 2009 bởi AMPLab tại đại học California. Sau này, Spark đã được trao cho Apache Software Foundation vào năm 2013 và được phát triển cho đến nay. Nó cho phép xây dựng các mô hình dự đoán nhanh chóng với việc tính toán được thực hiện trên một nhóm các máy tính, có có thể tính toán cùng lúc trên toàn bộ tập dữ liệu mà không cần phải trích xuất mẫu tính toán thử nghiệm. Tốc độ xử lý của Spark có được do việc tính toán được thực hiện cùng lúc trên nhiều máy khác nhau. Đồng thời việc tính toán được thực hiện ở bộ nhớ trong (in-memories) hay thực hiện hoàn toàn trên RAM.

Các ngôn ngữ lập trình được hỗ trợ bởi Spark bao gồm: Java, Python, Scala và R. Thông qua spark các lập trình viên và các nhà khoa học dữ liệu có thể truy vấn, phân tích, và chuyển đổi dữ liệu một cách nhanh chóng. Các task thường xuyên được sử dụng kết hợp với spark như ETL và SQL để thực hiện các câu lệnh truy vấn tuần

tự trên những tập dữ liệu lớn, xử lý dòng dữ liệu từ các cảm biến, hệ thống tài chính hay các task Machine learning.



Hình 1.2: Các thành phần chính của Spark

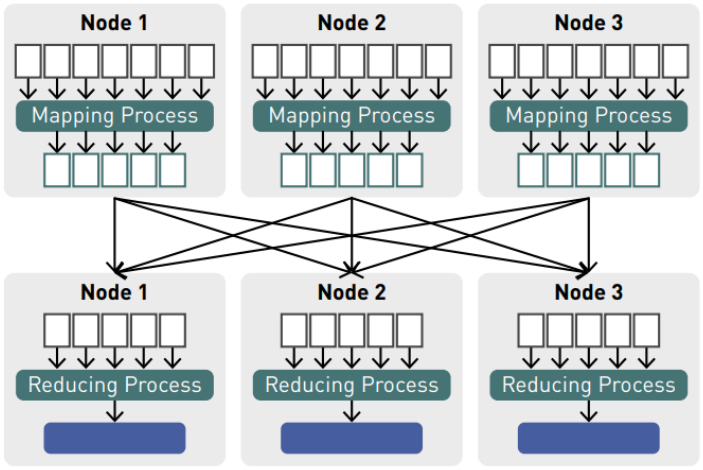
Thành phần chính của Spark là Spark Core: cung cấp những chức năng cơ bản nhất của Spark như lập lịch cho các tác vụ, quản lý bộ nhớ, khắc phục lỗi, tương tác với các hệ thống lưu trữ…Đặc biệt, Spark Core cung cấp API để định nghĩa RDD (Resilient Distributed DataSet) là tập hợp của các item được phân tán trên các nút của cụm và có thể được xử lý song song.

Spark có thể chạy trên nhiều loại quản lý cụm như Hadoop YARN, Apache Mesos hoặc trên chính quản lý cụm được cung cấp bởi Spark được gọi là Standalone Scheduler.

- Spark SQL cho phép truy vấn dữ liệu cấu trúc qua các câu lệnh SQL. Spark SQL có thể thao tác với nhiều nguồn dữ liệu như Hive tables, Parquet, và JSON.
- Spark Streaming cung cấp API để dễ dàng xử lý dữ liệu stream,
- MLlib cung cấp rất nhiều thuật toán của học máy như: phân loại nhãn, hồi quy, phân loại, lọc cộng tác…
- GraphX là thư viện để xử lý đồ thị.

1.3.2 Cơ chế hoạt động

Để tìm hiểu spark chúng ta sẽ bắt đầu với lịch sử hình thành và phát triển của nó. Trước Spark chúng ta đã từng biết tới MapReduce- một khung xử lý dữ liệu phân tán giúp Google thiết lập các chỉ mục trong sự bùng nổ của nội dung web, trên các cụm máy chủ lớn.



Hình 1.3: Cơ chế hoạt động của ứng dụng Spark

Có ba khái niệm cốt lõi trong chiến lược của Google:

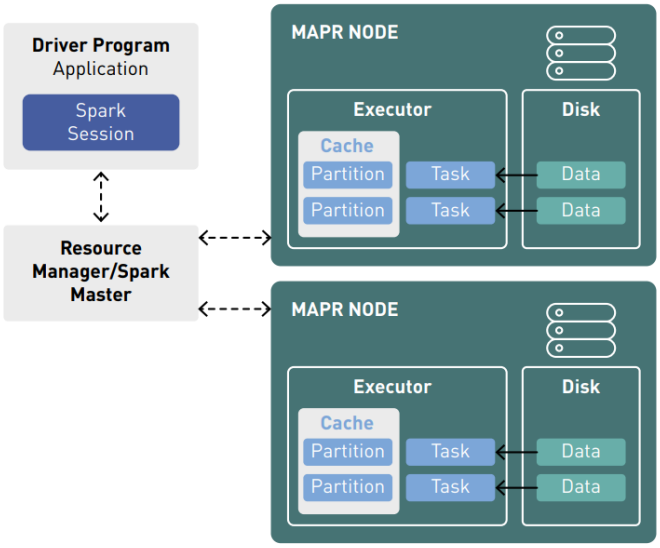
- *Dữ liệu phân tán*: Khi một tệp dữ liệu được tải lên cụm, nó sẽ được chia thành các phần được gọi là data block sau đó được phân phối chạy trên các data nodes và nhân rộng trên các cluster.
- *Tính toán phân tán:* người dùng chỉ định map function để xử lý dữ liệu dựa trên các cặp key/value. Để tạo ra một tập các cặp key/value và kết hợp chúng với reduce function thì tất cả các giá trị trung gian được liên kết với cùng một khóa. Một chương trình được viết theo cấu trúc này sẽ tự động chạy song song trên 1 cụm cluster lớn theo cách sau:
  - Quá trình mapping chạy trên mỗi node dữ liệu được chỉ định, chỉ hoạt động trên một block dữ liệu từ mỗi distribute file.
  - Các kết quả từ quá trình mapping được gửi tới Reducer trong một quy trình được gọi là "shuffle and sort": các cặp key/value từ quá trình mapping sẽ được sắp xếp theo key, được phân vùng theo số lượng reducer, sau đó được gửi qua hệ thống network và được danh sách key đã được sắp xếp sẽ được ghi lại trên reducer node.
  - Quá trình reducer thực hiện trên các node được chỉ định. Output của quá trình reducer sẽ được ghi vào 1 file input.

- *Khả năng chịu lỗi:* cả dữ liệu và tính toán có thể được chịu lỗi bằng cách chuyển sang node khác cho cả dữ liệu và tiến trình tính toán.

1.3.3 Spark application

Biểu đồ bên dưới biểu diễn luồng chạy của một ứng dụng Spark chạy trên một cụm cluster.

- Mỗi ứng dụng spark chạy dưới dạng các quy trình độc lập được điều phối bởi Spark Session.
- Trình quản lý tài nguyên hay quản lý cluster sẽ phân công nhiệm vụ cho các worker, một task cho một phân vùng.
- Mỗi task được giao cho 1 phần khối lượng của dataset trong partition của nó và output sẽ sẽ được xuất ra ở phân vùng dataset mới.
- Kết quả được gửi trở lại driver application hoặc có thể được lưu vào ổ đĩa.



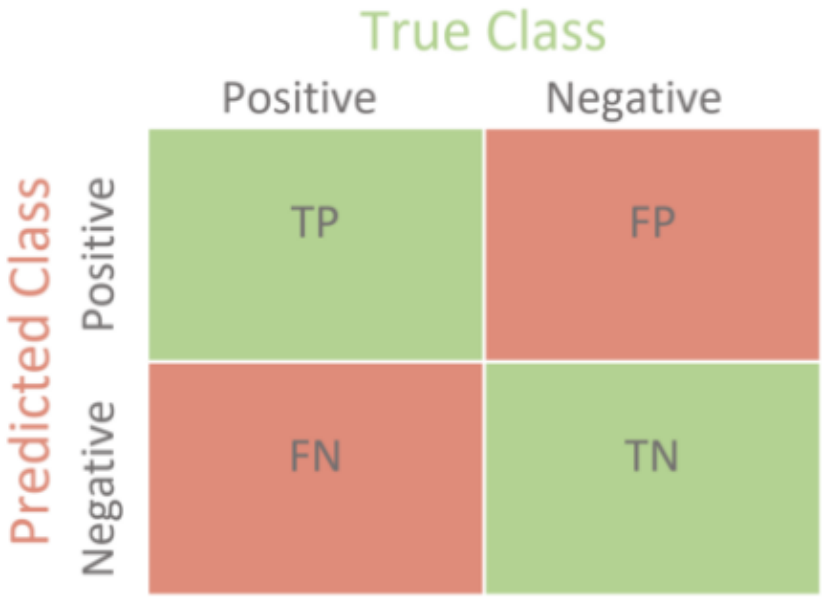
Hình 1.4: Luồng hoạt động của ứng dụng Spark

1.4 Các chỉ số đánh giá hiệu năng mô hình

1.4.1 Ma trận nhầm lẫn (Confusion matrix)

Ma trận nhầm lẫn (confusion matrix) là một chỉ số đo hiệu suất cơ bản để đánh giá hiệu năng dự đoán của một mô hình. Nó là một ma trận vuông kích thước 2x2

chứa bốn tổ hợp được tạo ra bởi 2 phân lớp nhị phân. Các chỉ số đo khác như độ chính xác, độ phủ hay các phương pháp đo như ROC cũng được xây dựng dựa trên ma trận nhầm lẫn. Từ yêu cầu bài toán là phân loại nhị phân với hai nhãn là 0 và 1 hoặc Yes/No. Các dự đoán đầu ra cho nhãn sẽ được chia thành hai loại là dự đoán "tích cực" và dự đoán "tiêu cực". Kết quả dự đoán của mô hình được chia thành 4 nhóm như hình bên dưới:



Hình 1.5: Ma trận nhầm lẫn

True Positive (TP): Số lượng dự đoán chính xác nhãn 1

True Negative (TN): Số lượng dự đoán chính xác nhãn 0

False Positive (FP): Số lượng dự đoán sai nhãn 1

True Negative (TN): Số lượng dự đoán sai nhãn 0

1.4.2 Các chỉ số Accuracy, Precision, Recall và F1 score

Accuracy: Chỉ số đánh giá độ chính xác tổng thể của mô hình. Giá trị của độ chính xác nằm trong khoảng 0 đến 1. Với 1 là giá trị độ chính xác tốt nhất và 0 là giá trị độ chính xác thấp nhất của một mô hình dự đoán. Độ chính xác (ACC) được tính bằng số tất cả các dự đoán đúng chia cho tổng số dự đoán của tập dữ liệu.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision: Chỉ số đánh giá tổng số dự đoán chính xác nhãn 1 chia cho tổng số dự đoán được dự đoán là nhãn 1. Giá trị lớn nhất của độ chính xác là 1 và nhỏ nhất là 0. Để tính Precision ta sử dụng công thức sau:

$$Prec = \frac{TP}{TP + FP}$$

Recall: Chỉ số thể hiện mô hình dự đoán đúng bao nhiêu phần nhãn 1 trong tổng số lượng nhãn 1 của cả tập. Nó còn có tên gọi là Tỉ lệ dương tính thực (TPR). Để tính recall ta sử dụng công thức sau:

$$Recall = \frac{TP}{TP + FN}$$

F1-score: Chỉ số kết hợp giữa 2 chỉ số Precision và Recall. Để tính F1-score ta sử dụng công thức sau:

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

1.4.3 Đường cong ROC

Đường cong ROC (receiver operating characteristic) là biểu đồ thể hiện hiệu năng phân loại nhãn của mô hình trên tất cả các ngưỡng điểm phân loại. Biểu đồ được tạo nên từ hai trục chứa giá trị True Positive Rate và False Positive Rate.

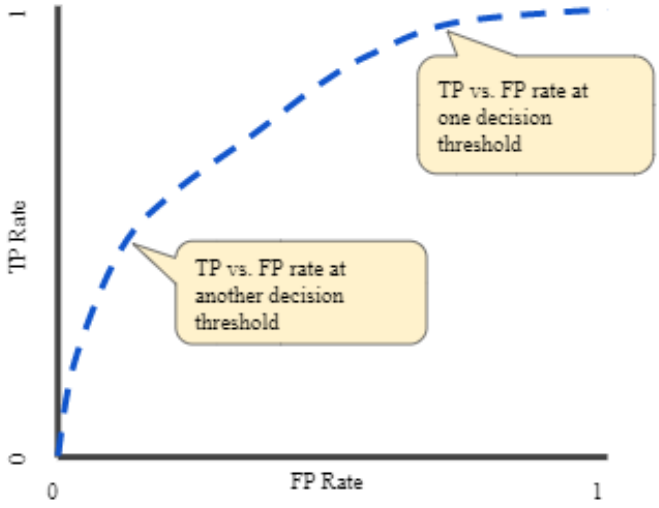
True Positive Rate (TPR) hay chính là Recall đã được trình bày ở phần trên. Công thức tính TPR:

$$TPR = \frac{TP}{TP + FN}$$

False Positive Rate (FPR) được tính bởi công thức:

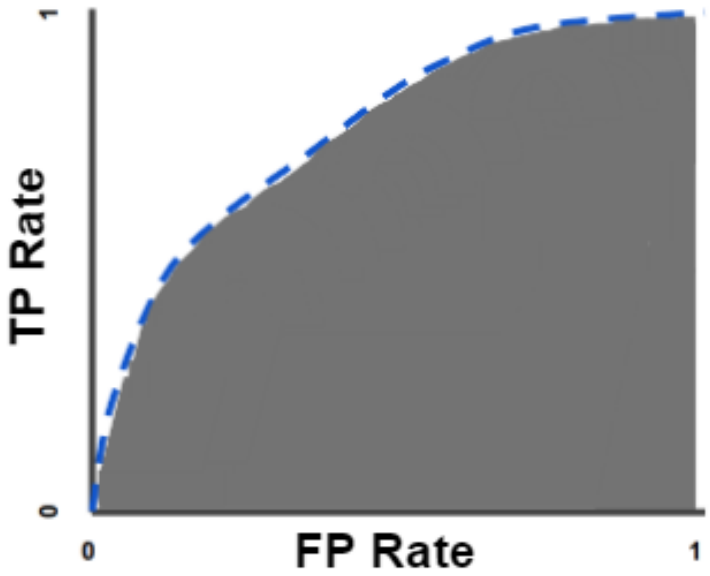
$$FPR = \frac{FP}{FP + TN}$$

Biểu đồ đường cong ROC được vẽ bởi các giá trị khác nhau của TPR và FPR trên mỗi ngưỡng cắt khác nhau của phân lớp. Việc hạ thấp ngưỡng phân loại sẽ phân loại nhiều được nhiều nhãn dương tính song cũng làm tăng cả đúng nhãn dương tính và sai nhãn dương tính.



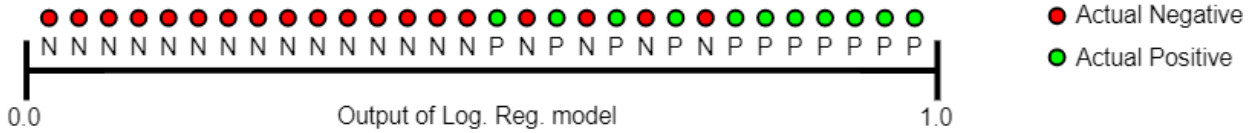
Hình 1.6: Đường cong ROC

Để đánh giá một mô hình người ta sử dụng AUC: Area Under the ROC Curve. AUC được tính bằng diện tích phần hình nằm bên dưới đường cong. Giá trị diện tích đó nằm trong khoảng [0,1].



Hình 1.7: Diện tích bên dưới đường cong ROC

AUC là độ đo để đánh giá hiệu suất dự đoán trên tất cả các ngưỡng phân loại có thể có của mô hình dự đoán. Hay nói một cách khác thì AUC là xác suất mà mô hình xếp hạng một mẫu dương tính ngẫu nhiên cao hơn một mẫu âm tính ngẫu nhiên.



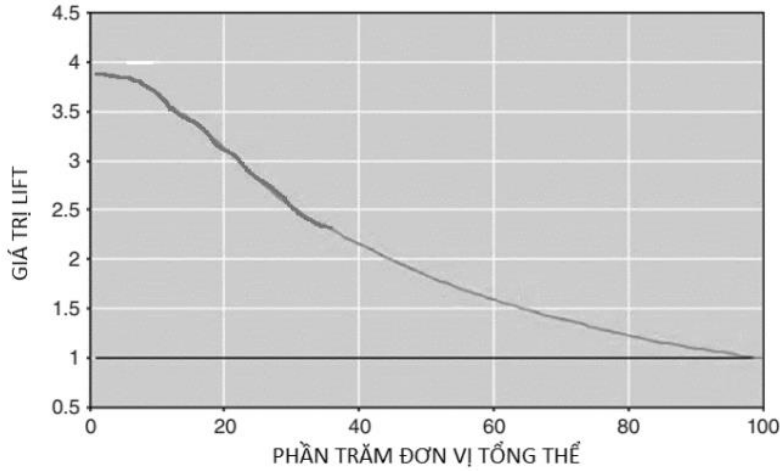
Hình 1.8: Xác suất phần loại nhãn

Hình trên mô phỏng một tập bản ghi được sắp xếp theo thứ tự tăng dần về điểm số xác suất phân loại nhãn. AUC có giá trị từ 0 đến 1. Một mô hình dự đoán sai 100% sẽ có AUC = 0 và dự đoán đúng 100% sẽ có AUC = 1. AUC có thể cho chúng ta thấy hiệu suất dự đoán của mô hình trên toàn bộ ngưỡng điểm do mô hình trả ra nhưng lại không cho ta biết ngưỡng điểm mô hình dự đoán chính xác nhất.

1.4.4 Biểu đồ Lift

Biểu đồ lift là phương pháp đánh giá hiệu quả của mô hình phân loại dựa trên việc đánh giá tỉ lệ phản hồi, hay so sánh các kết quả phân loại có được từ việc sử dụng mô hình so với không sử dụng mô hình. Khác với những phương pháp đo hiệu quả

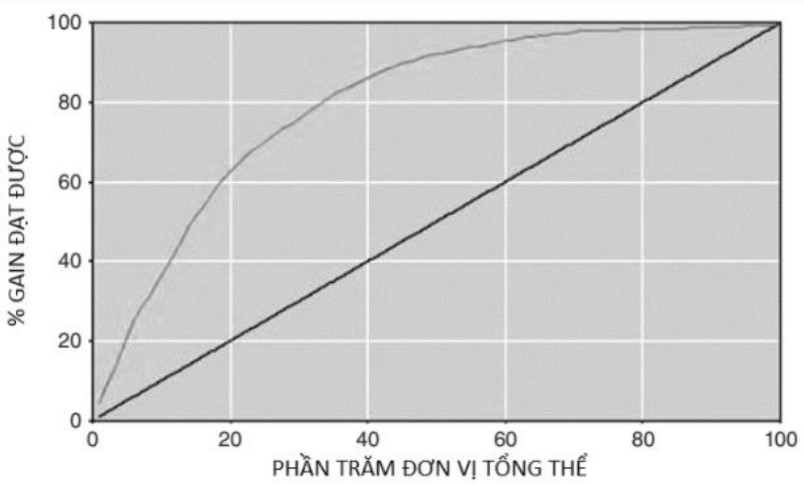
mô hình khác thì lift tính toán, định lượng độ đo hiệu quả theo % của tập dữ liệu tổng thể và kết hợp trực quan hóa qua đồ thị. Mỗi điểm trên trên biểu đồ lift được tính bằng cách tính xác suất mà mỗi đơn vị dữ liệu được phân loại Positive sau đó sắp xếp các đơn vị dữ liệu này theo thứ tự giảm dần ứng với giá trị tỉ lệ vừa tính sau đó tính lift cho từng mẫu. Biểu đồ lift được xây dựng từ việc tính giá trị lũy kế theo quy mô mẫu dữ liệu tăng dần. Do đó chúng ta sẽ có một số đơn vị dữ liệu trong tổng thể được phân loại theo phân lớp ấy và khi tiến gần đến 100% thì lift giảm dần đến 1. Lúc này mô hình không còn phân loại dữ liệu theo nhãn dương tính do đã phân loại hết các đơn vị dữ liệu cần phân loại. Để so sánh giữa các mô hình phức tạp để tìm ra mô hình hiệu quả nhất thì biểu đồ lift là một hướng tiếp cận thích hợp với cơ sở là cùng xét tại vị trí phân vị bất kỳ thì đồ thị nào có lift cao hơn sẽ hiệu quả hơn.



Hình 1.9: Biểu đồ lift

1.4.5 Biểu đồ Gain

Biểu đồ Gain kết hợp với biểu đồ Lift để thể hiện rõ hơn độ hiệu quả của mô hình phân loại. Biểu đồ Gain cung cấp cho chúng ta thông tin là trong % số đơn vị tổng thể chúng ta có thể đạt được bao nhiêu % đơn vị dữ liệu được phân loại chính xác. Dựa trên kết quả tính toán ở bước xây dựng biểu đồ Lift chúng ta sẽ xây dựng được biểu đồ Gain tương ứng.



Hình 1.10: Biểu đồ Gain

Ví dụ như trong hình vẽ trên thì tại khoảng 40% dữ liệu tổng thể thì có khoảng 85% số đơn vị được phân loại chính xác.

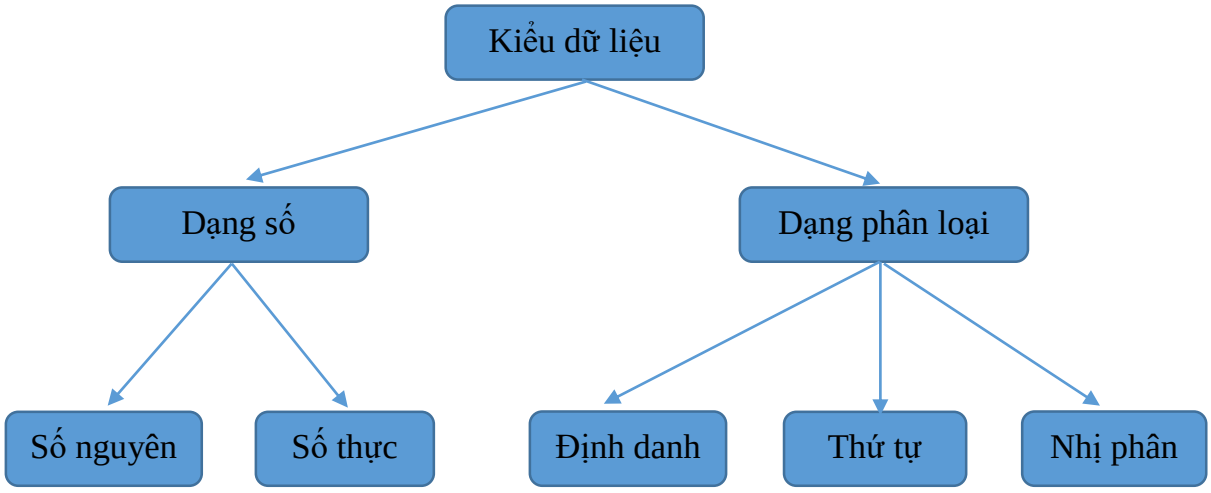
1.5 Các phương pháp xây dựng đặc trưng dữ liệu

Xây dựng đặc trưng dữ liệu là tiến trình lựa chọn các đặc tính của tập dữ liệu hay giảm số lượng các trường dữ liệu trong quá trình xây dựng các mô hình dự đoán. Với mục đích giảm thời gian tính toán, chi phí và cải thiện hiệu năng dự đoán của mô hình. Có nhiều phương pháp để lựa chọn đặc trưng dữ liệu nhưng có thể chia chúng thành ba nhóm chính:

- Phương pháp lọc: Xác định một số chỉ số nhất định và dựa trên các chỉ số đó để lựa chọn đặc trưng. Ví dụ như dựa vào chỉ số tương quan hoặc chi bình phương.
- Phương pháp đóng gói: Phương pháp này xem xét việc lựa chọn một tập các đặc trưng như một vấn đề tìm kiếm. Ví dụ như thuật toán đệ quy loại bỏ tính năng.
- Phương pháp nhúng: Phương pháp nhúng sử dụng các thuật toán có các phương pháp lựa chọn đặc trưng được tích hợp sẵn. Ví dụ như Lasso và RF có các phương pháp lựa chọn đặc trưng riêng của nó.

1.5.1 Các phương pháp thống kê lựa chọn đặc trưng dữ liệu với phương pháp lọc

Các phương pháp thống kê lựa chọn đặc trưng dữ liệu với phương pháp lọc thường sử dụng các chỉ số thể hiện mức độ tương quan giữa các biến đầu vào và biến đầu ra để làm cơ sở cho việc lựa chọn đặc trưng. Do đó việc lựa chọn các phương pháp thống kê phụ thuộc nhiều vào kiểu dữ liệu của các biến. Các kiểu dữ liệu phổ biến bao gồm dữ liệu dạng số và dữ liệu dạng phân loại, mỗi loại có thể chia thành nhiều kiểu dữ liệu như dạng số nguyên, dạng số thập phân cho dữ liệu dạng số và dạng nhị phân, thứ tự và định danh cho dữ liệu dạng phân loại.



1.5.1.1 Hệ số tương quan Pearson’s

Hệ số tương quan là một chỉ số thống kê đo mối liên hệ tương quan giữa hai biến số. Giá trị của hệ số tương quan r ($-1 \leq r \leq 1$). Hệ số tương quan càng gần 0 hoặc bằng 0 có nghĩa là hai biến đang xét không có mối liên hệ gì với nhau; ngược lại nếu giá trị của hệ số tương quan càng gần 1 hoặc -1 nghĩa là hai biến có mối quan hệ tuyệt đối. Nếu hệ số tương quan có giá trị âm thì đó là hai biến nghịch biến và hệ số tương quan dương thì đó là hai biến đồng biến. Hiện nay có nhiều công thức để tính hệ số tương quan giữa hai biến nhưng thông dụng nhất là công thức tính hệ số tương quan Pearson. Tương quan Person sẽ xác định một đường thẳng phù hợp nhất

với mối quan hệ tuyến tính của hai biến. Xét hai biến số x và y được lấy từ n mẫu, hệ số tương quan Pearson sẽ được tính bằng công thức sau:

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

Ví dụ: Tính hệ số tương quan giữa hai trường dữ liệu dân số và thu nhập

```
##            Population Income
## Alabama          3615   3624
## Alaska            365   6315
## Arizona          2212   4530
## Arkansas         2110   3378
## California      21198   5114
## Colorado         2541   4884
```

Cú pháp tính hệ số tương quan pearson trên python:

```
r1 = Correlation.corr(df, "features", "spearman")
```

Kết quả:

```
##            Population    Income
## Population  1.0000000 0.2082276
## Income      0.2082276 1.0000000
```

1.5.1.2 Hệ số tương quan hạng Spearman

Hệ số tương quan hạng Spearman được sử dụng thay thế hệ số tương quan Pearson để kiểm tra mối quan hệ giữa hai biến được xếp hạng hoặc một biến được xếp hạng và một biến đo lường. Sử dụng khi phân phối của tổng thể được giả sử không phải là phân phối chuẩn hoặc trong trường hợp có các giá trị quan sát bất thường (lớn quá hoặc nhỏ quá).

$$spearman_{cor} = 1 - \frac{6\sum_i^n d_i^2}{n(n^2-1)}$$

Trong đó d_i là hiệu hạng của 2 biến được tính bằng:

$$d_i = rankX_i - rankY_i$$

Ví dụ: Tính hệ số tương quan giữa hai trường dữ liệu dân số và thu nhập

```
##            Population Income
## Alabama          3615   3624
## Alaska            365   6315
## Arizona          2212   4530
## Arkansas         2110   3378
```

```
## California      21198   5114
## Colorado         2541   4884
```

Bảng tính tương quan giữa hai trường dân số và thu nhập

Bảng 1.2: Tương quan giữa hai trường dân số và thu nhập

	Population	Income	rgX	rgY	d	d2	Spearman_cor
2	365	6315	1	50	-49	2401	0.12461
50	376	4566	2	29	-27	729	0.12461
45	472	3907	3	12	-9	81	0.12461
8	579	4809	4	37	-33	1089	0.12461
28	590	5149	5	46	-41	1681	0.12461

1.5.1.3 Kiểm định chi bình phương (Chi squared)

Là phương pháp tính hệ số tương quan giữa các biến độc lập và biến phụ thuộc. Các biến được chọn làm đặc trưng của tập dữ liệu là các biến có hệ số Chi bình phương lớn. Công thức tính Chi bình phương:

$$X^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

Trong đó: O_i là các giá trị quan sát

E_i là các giá trị kỳ vọng

Ví dụ: Tính giá trị chi bình phương cho hai biến là giới tính và bằng cấp.

Bảng 1.3: Bảng tính giá trị chi bình phương

		Bằng cấp		
		Cao đẳng	Đại học	Sau đại học
Giới tính	Nam	6	35	15
	Nữ	4	34	6

Bảng tính xác suất cho từng sự kiện:

Bảng 1.4: Bảng tính xác suất cho từng sự kiện

		Bằng cấp			
		Cao đẳng	Đại học	Sau đại học	Xác suất
Giới tính	Nam	6	35	15	56%
	Nữ	4	34	6	44%
	Xác suất	10%	69%	21%	100%

Bảng tính giá trị kỳ vọng cho từng sự kiện:

Bảng 1.5: Bảng tính giá trị kỳ vọng

		Bằng cấp			
		Cao đẳng	Đại học	Sau đại học	Xác suất
Giới tính	Nam	5.6	38.64	11.76	56%
	Nữ	4.4	30.36	9.24	44%
	Xác suất	10%	69%	21%	100%

Áp dụng công thức tính đã nêu ở trên ta tính được hệ số Chi bình phương = 2.873

1.5.2 Các phương pháp thống kê lựa chọn đặc trưng dữ liệu với phương pháp đóng gói

Đệ quy loại bỏ đặc tính (Recursive Feature Elimination-RFE) là một trong những phương pháp lựa chọn đặc trưng dữ liệu phổ biến nhất hiện nay. RFE sẽ loại bỏ các trường dữ liệu có tương quan yếu đối với biến phụ thuộc cho tới khi đạt tới số lượng trường dữ liệu cần thiết do người dùng xác định từ trước. Với số lượng trường dữ liệu ít hơn mô hình dự đoán sẽ chạy hiệu quả hơn, giảm tài nguyên, thời gian chạy và đôi khi là nâng cao hiệu năng dự đoán. RFE hoạt động bằng cách tìm kiếm một tập con các trường dữ liệu bắt đầu bằng việc sử dụng tất cả các trường dữ liệu. Sau mỗi lần huấn luyện mô hình, các trường dữ liệu sẽ được sắp xếp theo thứ tự giảm dần của mức độ quan trọng. Sau đó các trường dữ liệu mức độ quan trọng thấp sẽ được bỏ ra và lặp lại quá trình huấn luyện.

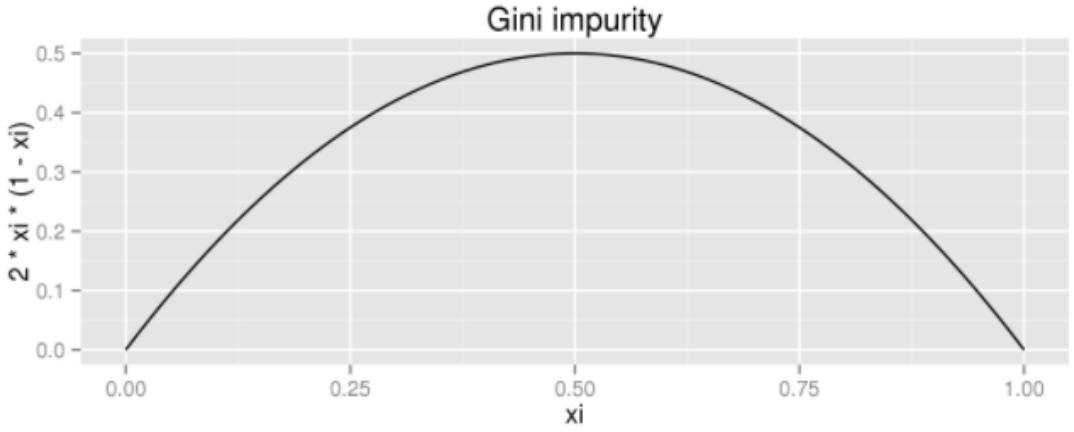
Trong Python ta có thể sử dụng hàm make_classification() với các tham số truyền vào như n_samples: số lượng bản ghi, n_features: số lượng thuộc tính đầu vào, n_informative: số lượng thuộc tính lựa chọn, n_redundant: số lượng thuộc tính loại trừ, random_state: giá trị khởi tạo cho việc lấy mẫu ngẫu nhiên.

1.5.3 Các phương pháp thống kê lựa chọn đặc trưng dữ liệu với phương pháp nhúng

Sử dụng thuật toán Rừng ngẫu nhiên để tính mức độ quan trọng của các thuộc tính. Đối với thuật toán rừng ngẫu nhiên mỗi lần thực hiện phân chia tại nốt cha sẽ tạo ra hai lớp con có chỉ số độ thuần khiết GINI nhỏ hơn nốt cha.

Công thức tính độ thuần khiết GINI:

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



Hình 1.11: Đồ thị biểu diễn độ thuần khiết GINI

Tại mỗi nốt chỉ số đánh giá mức độ quan trọng của thuộc tính sẽ được tính bằng công thức:

$$I = G_{parent} - G_{split1} - G_{split2}$$

Trong đó: G_{parent} là độ thuần khiết của nốt cha

G_{split1} là độ thuần khiết của nốt con thứ nhất

G_{split2} là độ thuần khiết của nốt con thứ hai

Chỉ số mức độ quan trọng của thuộc tính trong một cây được xác định bởi công thức:

$$f_{i_i} = \frac{\sum n_{i_j}}{\sum n_{i_k}}$$

Trong đó f_{i_i} là chỉ số mức độ quan trọng của thuộc tính i

n_{i_j} là chỉ số mức độ quan trọng của nốt chứa thuộc tính i

n_{i_k} là chỉ số mức độ quan trọng của tất cả các nốt chứa thuộc tính i

Công thức chuẩn hóa chỉ số mức độ quan trọng của thuộc tính trong một cây:

$$norm f_{i_i} = \frac{f_{i_i}}{\sum f_{i_j}}$$

Trong đó $norm f_{i_i}$ là giá trị chuẩn hóa mức độ quan trọng của thuộc tính i

f_{i_i} là chỉ số mức độ quan trọng của thuộc tính i

f_{i_j} là chỉ số mức độ quan trọng của tất cả các nốt thuộc tính

Chỉ số mức độ quan trọng của thuộc tính trong thuật toán Rừng ngẫu nhiên sẽ được tính bằng trung bình cộng của các chỉ số mức độ quan trọng trên từng cây.

$$RF f_{i_i} = \frac{\sum norm f_{i_i}}{T}$$

Trong đó $RF f_{i_i}$ là giá trị mức độ quan trọng của thuộc tính i trong mô hình

$norm f_{i_i}$là giá trị chuẩn hóa mức độ quan trọng của thuộc tính i trong các cây

T là tổng số lượng cây

1.6 Kỹ thuật tiền xử lý dữ liệu

Kỹ thuật tiền xử lý dữ liệu là một trong những kỹ thuật tối quan trọng trong quá trình xây dựng các mô hình dự đoán với các thuật toán học máy. Chúng ta đều biết rằng các thuật toán học máy sẽ dựa vào tập dữ liệu đầu vào để đưa ra kết quả dự

đoán. Nhưng vấn đề lớn nhất mà các mô hình này gặp phải là chất lượng dữ liệu đầu vào không đủ tốt. Đó chính là lý do chúng ta dành phần lớn thời gian trong quá trình xây dựng mô hình dự đoán cho tiến trình tiền xử lý dữ liệu. Các kỹ thuật tiền xử lý dữ liệu là điểm khác biệt lớn giữa mô hình dự đoán tốt và mô hình dự đoán không tốt.

Các kỹ thuật tiền xử lý dữ liệu được ra đời với hai mục tiêu chính:

- Chuẩn bị dữ liệu đầu vào thích hợp, tương thích với yêu cầu của các thuật toán học máy.
- Nâng cao hiệu năng dự đoán của mô hình

1.6.1 Xử lý thiếu giá trị

Các trường dữ liệu bị thiếu giá trị là một trong những vấn đề chúng ta sẽ thường xuyên đối mặt trong quá trình xử lý dữ liệu đầu vào cho mô hình học máy. Nguyên nhân của việc thiếu giá trị trên các trường dữ liệu có thể do lỗi của người nhập dữ liệu, lỗi luồng tổng hợp dữ liệu, các nguyên nhân đến từ quyền riêng tư của người dùng. Cho dù là với lý do gì thì việc thiếu dữ liệu cũng sẽ ảnh hưởng đến hiệu năng dự đoán của các mô hình học máy. Một vài thuật toán học máy sẽ tự động bỏ các bản ghi thiếu giá trị trong quá trình huấn luyện dẫn tới giảm hiệu năng do số lượng mẫu huấn luyện giảm. Đa số các thuật toán học máy không chấp nhận những tập dữ liệu đầu vào bị thiếu giá trị.

1.6.1.1 Loại bỏ các bản ghi thiếu dữ liệu

Giải pháp đơn giản nhất để xử lý tập dữ liệu thiếu giá trị đó là bỏ đi những bản ghi hoặc thậm chí là cả trường dữ liệu thiếu giá trị đó. Có thể thiết lập giá trị ngưỡng cho việc quyết định có loại bỏ bản ghi hay trường dữ liệu thiếu ra khỏi tập dữ liệu huấn luyện hay không. Sẽ loại bỏ các bản ghi hay trường dữ liệu có tỉ lệ thiếu dữ liệu lớn hơn ngưỡng mà ta thiết lập.

Ví dụ: Thiết lập giá trị ngưỡng = 0.7 và loại bỏ các trường hay bản ghi có tỉ lệ thiếu dữ liệu lớn ngưỡng thiết lập. Cú pháp thực hiện với ngôn ngữ lập trình python.

```
threshold = 0.7

#Xóa các cột với tỉ lệ thiếu dữ liệu > 70%

data = data[data.columns[data.isnull().mean() < threshold]]


# Xóa các cột với tỉ lệ thiếu dữ liệu > 70%

data = data.loc[data.isnull().mean(axis=1) < threshold]
```

1.6.1.2 Thay thế các trường dữ liệu dạng số

Thay thế các trường dữ liệu bị thiếu bằng một giá trị là phương pháp được ưa thích hơn phương pháp loại bỏ dữ liệu trong tiến trình tiền xử lý. Tuy nhiên, việc lựa chọn giá trị thay thế cho dữ liệu bị thiếu là điều tối quan trọng. Giả sử chúng ta có một trường dữ liệu chỉ gồm giá trị 1 và NA thì chúng ta sẽ tiến hành thay thế giá trị NA bằng giá trị 0. Ngoài ra có thể thay thế các giá trị thiếu bằng giá trị trung bình, trung vị, tần số của trường dữ liệu. Hoặc cũng có thể là kết hợp thay thế giá trị thiếu bằng nhiều giá trị khác nhau.

Thay thế hồi quy là phương pháp thay thế dựa vào mối liên hệ giữa trường dữ liệu đang bị thiếu và các trường dữ liệu khác. Hay nói cách khác là sử dụng giá trị ở các trường khác để hồi quy tìm ra giá trị cần thay thế. Không giống như những phương pháp thay thế dữ liệu thiếu bằng trung bình hay trung vị là dựa phân bố giá trị tại chính trường dữ liệu đang xét.

Tổng hợp một số phương pháp thay thế ứng với từng kiểu dữ liệu được trình bày ở bảng bên dưới:

Bảng 1.6: Các phương pháp thay thế

	Dữ liệu dạng số	**Dữ liệu dạng danh mục**
Giá trị đã có	Giá trị nhỏ nhất/ Giá trị lớn nhất	Giá trị liền trước, sau
Giá trị thống kê	Giá trị trung bình, trung vị	Giá trị tần số
Giá trị dự đoán	Thuật toán hồi quy	Thuật toán phân loại

1.6.2 Xử lý giá trị ngoại lai

Trước khi bắt đầu với việc xử lý các giá trị ngoại lai như thế nào thì phải nói rằng cách tốt nhất để có thể nhận diện ra các giá trị ngoại lai đó là trực quan dữ liệu. Việc nhận biết các giá trị ngoại lai từ trực quan dữ liệu mang lại độ chính xác lớn hơn so với các phương pháp thống kê thông thường. Tuy nhiên trong khuôn khổ bài luận này tôi sẽ nói sâu hơn về các phương pháp thống kê để xác định giá trị ngoại lai. Cụ thể là sử dụng độ lệch chuẩn và giá trị phần trăm.

1.6.2.1 Xác định giá trị ngoại lai với độ lệch chuẩn

Nếu các giá trị của một trường dữ liệu có khoảng cách tới giá trị trung bình lớn hơn x lần giá trị của độ lệch chuẩn thì ta có thể coi đó là giá trị ngoại lai. Không có cách chọn chính xác cho giá trị x nhưng thông thường ta thường chọn giá trị $2 \leq x \leq 3$. Công thức tính độ lệch chuẩn:

$$\sigma = \sqrt{\frac{\sum(x_i - \overline{X})^2}{N}}$$

Trong đó: x_i là giá trị của điểm dữ liệu

$\overline{X}$ là giá trị trung bình của trường dữ liệu đang xét

N là số điểm dữ liệu

Hoặc có thể sử dụng z-score thay cho công thức trên để quy về dạng phân phối chuẩn. Công thức tính z-score:

$$z - score = \frac{x - \mu}{\sigma}$$

Trong đó: x là giá trị của điểm dữ liệu

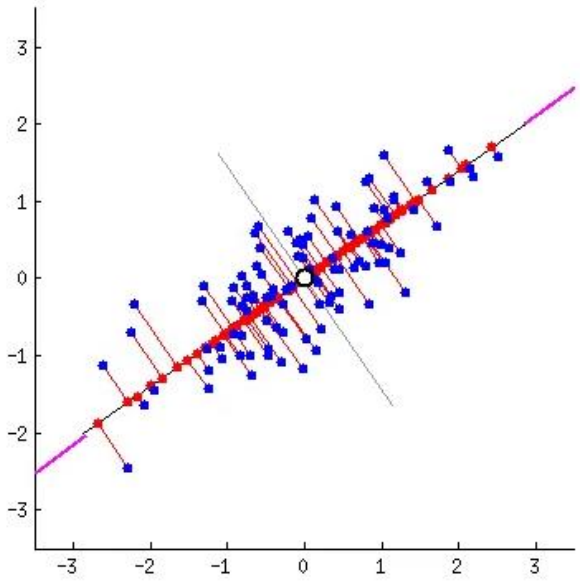
μ là giá trị trung bình của trường dữ liệu đang xét

σ là độ lệch chuẩn của tập dữ liệu

1.7 Thuật toán giảm chiều dữ liệu (PCA)

Thuật toán giảm chiều dữ liệu PCA (Principal Components Analysis) là kỹ thuật chuyển đổi các trường dữ liệu trong tập dữ liệu thành các trường dữ liệu mới

gọi là các thành phần chính (Principal Component). Mục tiêu chính là số trường dữ liệu mới giảm tối thiểu nhất có thể so với số lượng trường dữ liệu ban đầu mà vẫn chứa đủ những thông tin đại diện cho cả tập dữ liệu. Hay nói cách khác PCA là kỹ thuật gộp các trường dữ liệu hiện hành. Mỗi trường dữ liệu mới là tổ hợp có trọng số của các trường dữ liệu gốc. Các PC được hình thành theo cách gán trọng số lớn hơn cho các PC thành phần có tính đại diện lớn hơn cho dữ liệu gốc.



Hình 1.12: Mô phỏng thuật toán PCA

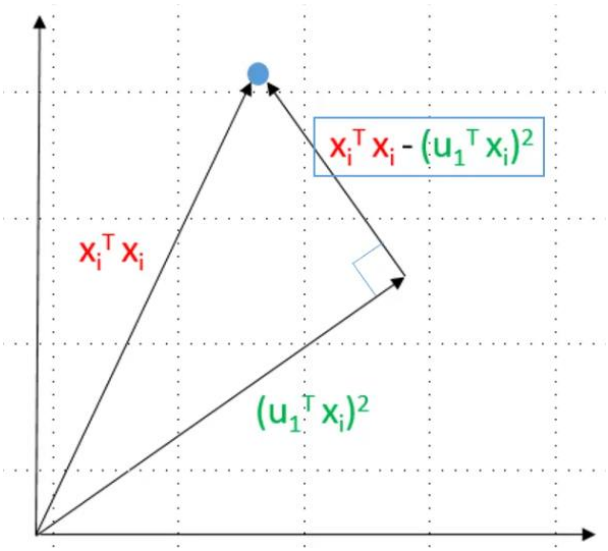
Hình vẽ trên mô phỏng tập dữ liệu chỉ gồm hai trường dữ liệu, nhiệm vụ của thuật toán PCA là tìm ra trường dữ liệu mới có dạng biểu diễn như một đường thẳng đi qua nhiều điểm biểu diễn trường dữ liệu ban đầu nhất có thể. Để thỏa mãn điều kiện đó thì tổng khoảng cách từ các điểm dữ liệu đến đường thẳng phải là nhỏ nhất.

Giả sử u1 là vectơ cần tìm có khả năng đại diện cho hai trường dữ liệu. Chúng ta cần tìm khoảng cách nhỏ nhất từ các điểm biểu diễn hai trường dữ liệu ban đầu đến vectơ u1. Hướng của vectơ u1 chính là hướng của PC1 thay thế cho 2 trường dữ liệu gốc. x_i là tọa độ của các điểm dữ liệu trong hệ trục. Áp dụng định lý Pitago để tính khoảng cách từ điểm biểu diễn dữ liệu tới u1.

Hàm mục tiêu để tính khoảng cách nhỏ nhất:

$$\min(\frac{1}{n}\sum_{i}^{n}(x_i^T x_i - (u_1^T x_i)^2))$$

Giá trị nhỏ nhất đạt được khi giá trị của u_1 bằng giá trị vectơ riêng của ma trận hiệp phương sai của X.



Hình 1.13: Mô phỏng cách tính khoảng cách nhỏ nhất trong thuật toán PCA

KẾT CHƯƠNG

Chương này tập trung chủ yếu trình bày các nội dung liên quan tới:

- Tổng quan quy trình phân tích dữ liệu
- Các chỉ số đánh giá dữ liệu và hiệu năng mô hình
- Kỹ thuật tiền xử lý dữ liệu
- Các phương pháp xây dựng đặc trưng dữ liệu
- Công nghệ Hadoop cho xử lý dữ liệu phân tán
- Thuật toán giảm chiều dữ liệu (PCA)

CHƯƠNG 2 - MÔ HÌNH HÀNH VI VÀ MỘT SỐ THUẬT TOÁN HỌC MÁY

2.1 Thuật toán rừng ngẫu nhiên (Random Forest)

2.1.1 Cây quyết định

2.1.1.1 Khái niệm

Cây quyết định (Decision tree) là một mô hình học có giám sát (supervised learning), có thể được áp dụng vào cả hai bài toán phân loại nhãn và hồi quy. Việc xây dựng một cây quyết định trên dữ liệu huấn luyện cho trước là việc đi xác định các câu hỏi và thứ tự của chúng. Cây quyết định có thể làm việc được với tập dữ liệu có đặc trưng dạng danh mục và dạng số. Cây quyết định là thuật toán có cấu trúc dạng cây, trong đó mỗi nút thể hiện cho một thuộc tính dữ liệu, mỗi nhánh con của nút biểu diễn giá trị của thuộc tính và mỗi nốt lá sẽ chứa nhãn.

2.1.1.2 Ý tưởng thuật toán

Bước 1: Bắt đầu với việc set tập dữ liệu S ở nút gốc

Bước 2: Lặp lại việc tính toán Entropy(H) và Information Gain(IG) với từng thuộc tính

Bước 3: Lựa chọn thuộc tính có Entropy nhỏ nhất hoặc Information Gain lớn nhất làm nút gốc

Bước 4: Chia tập S theo từng thuộc tính đã được lựa chọn để tạo ra các tập con dữ liệu

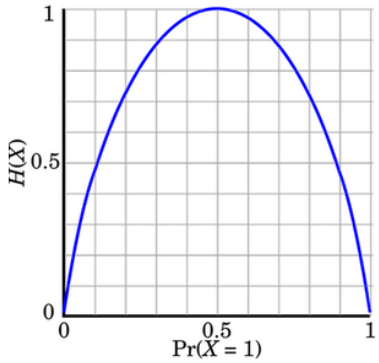
Bước 5: Thuật toán lặp lại trên mỗi tập con và chỉ xem xét các thuộc tính chưa được lựa chọn làm nút gốc trước đó.

2.1.1.3 Cơ sở lý thuyết

a. Hàm số Entropy

Cho một phân phối xác suất của một biến rời rạc *x* có thể nhận *n* giá trị khác nhau *x*1,*x*2,…,*xn*. Giả sử rằng xác suất để *x* nhận các giá trị này là *pi*=*p*(*x*=*xi*). Ký hiệu phân phối này là **p**=(*p*1,*p*2,…,*pn*). Entropy của phân phối này là:

$$H_{(p)} = -\sum_{i=1}^{n} p_i \, log_2^{p_i}$$



Hình 2.1: Đồ thị của hàm Entropy

b. Information Gain

Information Gain được tính dựa trên sự giảm của hàm Entropy khi tập dữ liệu được phân chia trên một thuộc tính. Để xây dựng một cây quyết định, ta phải tìm tất cả thuộc tính trả về Infomation gain cao nhất. Do H(S) là không đổi với mỗi tầng, ta chọn thuộc tính f có Entropy nhỏ nhất để thu được Gain(x,S) lớn nhất.

$$G_{(x,S)} = H_{(s)} - H_{(x,S)}$$

Trong đó: H(S) là Entropy tổng của toàn bộ tập data set S.

H(x,S) là Entropy được tính trên thuộc tính x.

2.1.2 Thuật toán rừng ngẫu nhiên (Random Forest)

2.1.2.1 Khái niệm

Rừng ngẫu nhiên là một tập hợp các mô hình (ensemble) gồm nhiều cây quyết định (decision tree). Mô hình rừng ngẫu nhiên rất hiệu quả cho các bài toán phân loại vì nó huy động cùng lúc hàng trăm mô hình nhỏ hơn bên trong với quy luật khác nhau để đưa ra quyết định cuối cùng. Mỗi mô hình con có thể mạnh yếu khác nhau, nhưng

theo nguyên tắc “wisdom of the crowd”, ta sẽ có cơ hội phân loại chính xác hơn so với khi sử dụng bất kì một mô hình đơn lẻ nào.

Như tên gọi của nó, rừng ngẫu nhiên dựa trên cơ sở :

- Random = Tính ngẫu nhiên

- Forest = nhiều cây quyết định (decision tree)

Đơn vị của RF là thuật toán cây quyết định, với số lượng hàng trăm. Mỗi cây quyết định được tạo ra một cách ngẫu nhiên từ việc: Tái chọn mẫu (bootstrap, random sampling) và chỉ dùng một phần nhỏ tập biến ngẫu nhiên (random features) từ toàn bộ các biến trong dữ liệu. Ở trạng thái sau cùng, mô hình RF thường hoạt động rất chính xác, nhưng đổi lại, rất khó để có thể hiểu được cơ chế hoạt động bên trong mô hình vì cấu trúc quá phức tạp.

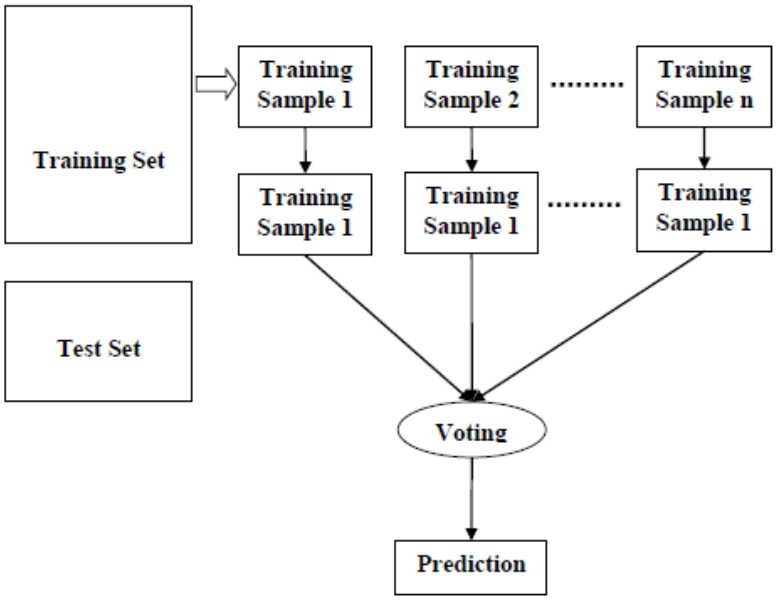
2.1.2.2 Ý tưởng thuật toán

Bước 1: Lựa chọn ngẫu nhiên một một tập dữ liệu con từ tập dữ liệu mẫu

Bước 2: Thuật toán sẽ tạo cây quyết định cho từng tập dữ liệu con. Sau đó nhận lại kết quả từ mỗi cây quyết định

Bước 3: Thực hiện voting cho các kết quả dự đoán

Bước 4: Cuối cùng chọn kết quả được dự đoán nhiều nhất làm kết quả cuối cùng.



Hình 2.2: Ý tưởng thuật toán Rừng ngẫu nhiên

2.1.2.3 Ưu điểm, nhược điểm

Ưu điểm:

- Giảm thiểu rủi ro quá khớp (overfitting) vì thuật toán dựa trên voting của tất cả các cây quyết định
- Rừng ngẫu nhiên có thể được sử dụng trong cả hai bài toán phân loại và hồi quy.
- Rừng ngẫu nhiên cũng có thể xử lý các giá trị còn thiếu.
- Thuật toán có độ chính xác cao trên tập dữ liệu lớn
- Các cây có thể được xây dựng song song

Nhược điểm:

- Tốc độ dự đoán chậm do có nhiều cây quyết định, mỗi khi dự đoán, tất cả các cây trong rừng phải đưa ra dự đoán cho cùng một đầu vào cho trước và sau đó thực hiện bỏ phiếu trên đó (tuy nhiên có thể khắc phục nếu chạy trên môi trường song song)

2.2 Thuật toán Naïve Bayes

2.2.1 Suy diễn Bayes

Suy diễn Bayes là một phương pháp suy diễn thống kê, trong đó định lý Bayes được sử dụng để cập nhật xác suất/khả năng xảy ra của một giả thuyết khi càng nhiều dữ liệu/thông tin về giả thuyết đó được cung cấp đầy đủ. Suy diễn Bayes được hình thành dựa trên xác suất có điều kiện. Biết rằng A và B là hai sự kiện xảy ra, khi đó xác xuất xảy ra A với điều kiện B biết trước được tính bằng công thức:

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

Trong đó: P(A|B): là xác suất hậu nghiệm (posterior probability)

P(B|A): là xác suất hợp lý (likelihood probability)

P(A): là xác suất tiên nghiệm (prior probability)

P(B): là thực chứng (evidence)

Ví dụ:

Với dữ liệu viễn thông sẵn của một nhà mạng, ước lượng xác suất khả năng dùng gói data đối với đối tượng học sinh sinh viên là khách hàng. Khi đó sự kiện A là học sinh sinh viên, sự kiện B dùng gói data. Cần tìm P(A|B) khi biết các giả định sau đây:

+ Xác suất dùng gói data P(A) = 0.2186

+ Xác suất là học sinh sinh viên P(B) = 0.4077

+ Xác suất là học sinh sinh viên trong những khách hàng sử dụng data P(B|A) = 0.4678

Xác suất sử dụng data trong nhóm đối tượng là học sinh sinh viên P(A|B) = 0.251

Tuy nhiên trên thực tế, liệu toàn bộ dữ liệu đều có đầy đủ nhãn phân loại, khách hàng nào học sinh sinh viên, khách hàng nào có đăng ký dùng data trong tháng. Khi đấy suy diễn Bayes trở nên hữu ích trong việc suy diễn thống kê trên toàn bộ tập quần thể.

2.2.2 Cơ sở lý thuyết

Đối với các bài toán phân loại trong học máy, phương pháp Naïve-Bayes được dùng tương đối phổ biến và đem lại kết quả khả quan. Trong thuật toán này, xác suất có điều kiện được ứng dụng để xác định xác suất xảy ra tại từng nhãn và chọn ra nhãn có xác suất cao nhất với điều kiện là các trường dữ liệu thuộc tính của một điểm dữ liệu. Giả sử thuật toán phân loại Naïve-bayes chỉ ra nhãn Y cho bởi các điểm dữ liệu, x1, x2, …xn và xác suất hậu nghiệm trong suy diễn Bayes (coi theta Θ là Y, và data là x1,x2,..xn) với xác suất xảy ra như sau:

$$P(Y|x_1, x_2, \ldots, x_n) = \frac{P(x_1, x_2, \ldots, x_n|Y)P(Y)}{P(x_1, x_2, \ldots, x_n)}$$

$$P(\mathbf{\Theta}|data) = \frac{P(data|\mathbf{\Theta}) \times P(\mathbf{\Theta})}{P(data)}$$

Tuy nhiên để đơn giản hóa quá trình tính toán xác suất (do dữ liệu gồm nhiều chiều), xác suất P(Y|x1,x2,...,xn) – tức phân phối của các điểm dữ liệu trong nhãn, được giả sử các thành phần (các chiều) trong điểm dữ liệu (biến ngẫu nhiên) là độc lập với nhau và được cho bởi nhãn Y cho trước. Khi đấy:

$$P(x_1, x_2, \ldots|Y) = P(x_1|Y)P(x_2|Y)\ldots$$

Do đó, xác suất thuật toán chỉ ra nhãn Y dựa trên các chiều của điểm dữ liệu cho trước được viết lại như sau:

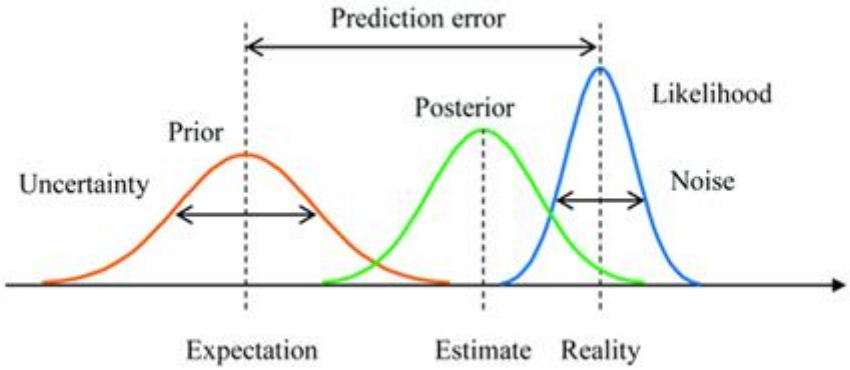
$$P(Y|x_1, x_2, \ldots, x_n) = \frac{P(x_1|Y)P(x_2|Y)\ldots P(x_n|Y)P(Y)}{P(x_1, x_2, \ldots, x_n)}$$

$$= \frac{P(Y)\prod_{i=1}^{n} P(x_i|Y)}{P(x_1, x_2, \ldots, x_n)}$$

$$\propto P(Y)\prod_{i=1}^{n} P(x_i|Y)$$

Bản chất của Suy diễn có thể được hiểu như sau. Với những nhận định giả sử được cho trước về xác suất xảy ra của một sự kiện (prior probability), sau đó đánh

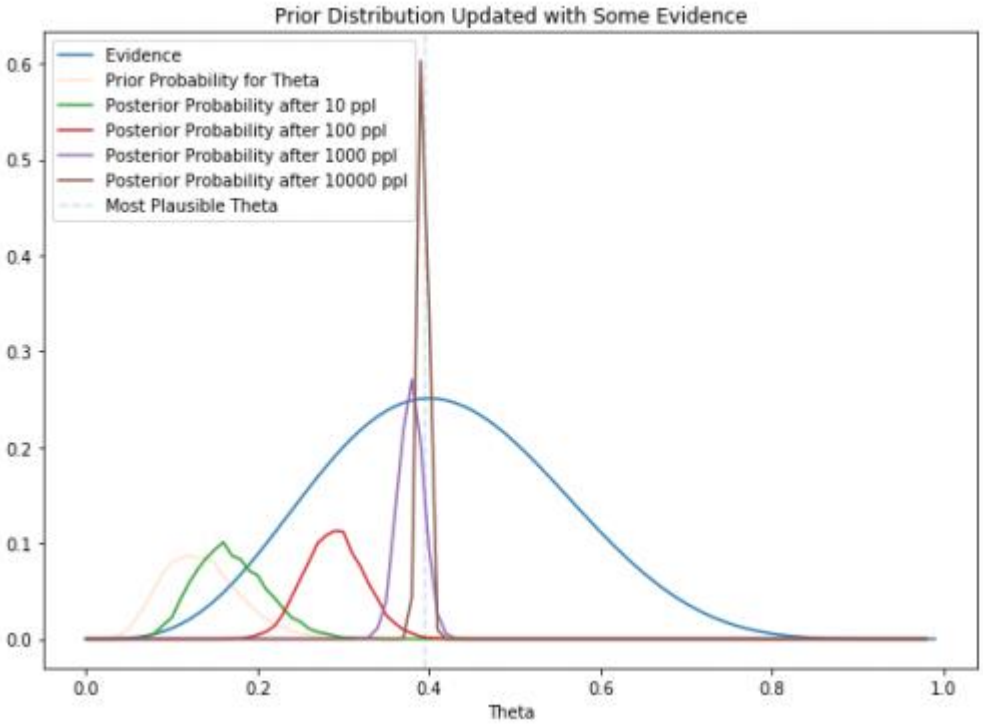
giá nhận định đó với dữ liệu thu thập được (likelihood). Dựa trên những gì quan sát được, nhận định ban đầu được cập nhật (khi đó prior probability trở thành posterior probability).



Hình 2.3: Mô tả suy diễn Bayes

Việc cập nhật này có thể thực hiện một lần hay nhiều lần, tuy thuộc vào lượng data có sẵn ban đầu. Trong trường hợp dữ liệu mới được bổ sung, xác suất hậu nghiệm tìm ra lại trở thành tiền nghiệm và xác suất hậu nghiệm mới sẽ được cập nhật lại. Xác suất hậu nghiệm được xem như là sự thỏa hiệp (compromise) giữa tiền nghiệm và khả năng xảy ra (likelihood).

- Khi lượng dữ liệu ít, phân phối của hậu nghiệm sẽ lệch về gần với phân phối của tiền nghiệm.
- Khi lượng dữ liệu nhiều, phân phối của hậu nghiệm sẽ lệch về gần với phân phối của khả năng xảy ra (likelihood).



Hình 2.4: Phân bố Bayes

Như hình trên, ta có thể thấy được xác suất xảy ra của tiền nghiệm là 0.15 (đường màu vàng). Tuy nhiên với các lượng mẫu lớn hơn (từ 10, 100, 1000 đến 10000) thì xác suất này được cập nhật thành (hậu nghiệm) và có xu hướng co về với dữ liệu thực tế.

2.2.3 Ứng dụng của Bayes trong phân tích dữ liệu

Trong các bài toàn phân tích dữ liệu, trường phân loại (nhãn) trong tập dữ liệu gốc thường không bao gồm đầy đủ cho các điểm dữ liệu. Do vậy, việc phân tích và ước lượng xác suất trên một tập mẫu (tập có đầy đủ nhãn) và suy đoán trên toàn tập lớn (population) là hoàn toàn cần thiết.

Thực vậy, đối với các bài toán ứng dụng phân tích dữ liệu hiện tại, có nhiều nguyên nhân dẫn đến việc tập dữ liệu không có đầy đủ thông tin. Có thể kể đến trường hợp, tập khách hàng đa dạng và nằm ngoài vùng dữ liệu viễn thông. Do đó khi phân tích hành vi viễn thông của khách hàng thì chỉ mới chỉ ra được đặc điểm khách hàng và sẽ suy đoán trên nhóm khách hàng sử dụng các nhà mạng khác. Mặt khác, đối với các bài toán phân khúc khách hàng, sau khi đã phân loại ra được từng phân khúc thì

mục tiêu tiếp theo là nhặt ra được nhóm khách hàng tiềm năng trong các phân khúc có tỷ lệ cao nhằm tăng hiệu quả tác động. Khi đó suy diễn Bayes có thể thực hiện để dự đoán xác suất tỷ lệ khách hàng tiềm năng để nhặt ra những nhóm đối tượng tác động chỉ định.

2.3 Thuật toán Logistic Regression

2.3.1 Khái niệm

Logistic Regression (Hồi quy logistic) là một mô hình hồi quy nhằm dự đoán phân lớp giá trị đầu ra ứng với một vector đầu vào. Nói cách khác, mục tiêu phương pháp nhằm phân loại các đối tượng vào các lớp tương ứng. Đầu vào của mô hình là một tập dữ liệu với các biến phụ thuộc và biến độc lập. Mô hình sẽ sử dụng giá trị của các biến phụ thuộc để dự đoán giá trị của biến độc lập. Đối với bài toán Hồi quy logistic thì đầu ra của bài toán là xác suất dự đoán ứng với từng giá trị của biến độc lập.

Ví dụ, nhận xét xem một khách hàng có khả năng ngừng sử dụng dịch vụ hay không. Ở đây ta coi đầu ra là y=1 nếu khách hang ngừng sử dụng dịch vụ của công ty, y=0 là khách hàng vẫn duy trì dịch vụ của công ty. Đầu vào x là các dữ liệu về tiêu dùng của khách hàng, chất lượng dịch vụ, các yếu tố hành vi…

2.3.2 Cơ sở lý thuyết

Sử dụng phương pháp thống kê ta có thể cho rằng khả năng một đối tượng có các thuộc tính x nằm vào một nhóm y0 là xác suất của nhóm y0 khi biết x: $p(y_0|x)$

Dựa vào công thức xác suất có điều kiện ta có:

$$p(y_0|x) = \frac{p(x\,|y_0)\,p(y_0)}{p(x)} = \frac{p(x\,|y_0)\,p(y_0)}{p(x\,|y_0)\,p(y_0) + \,p(x\,|y_1)\,p(y_1)}$$

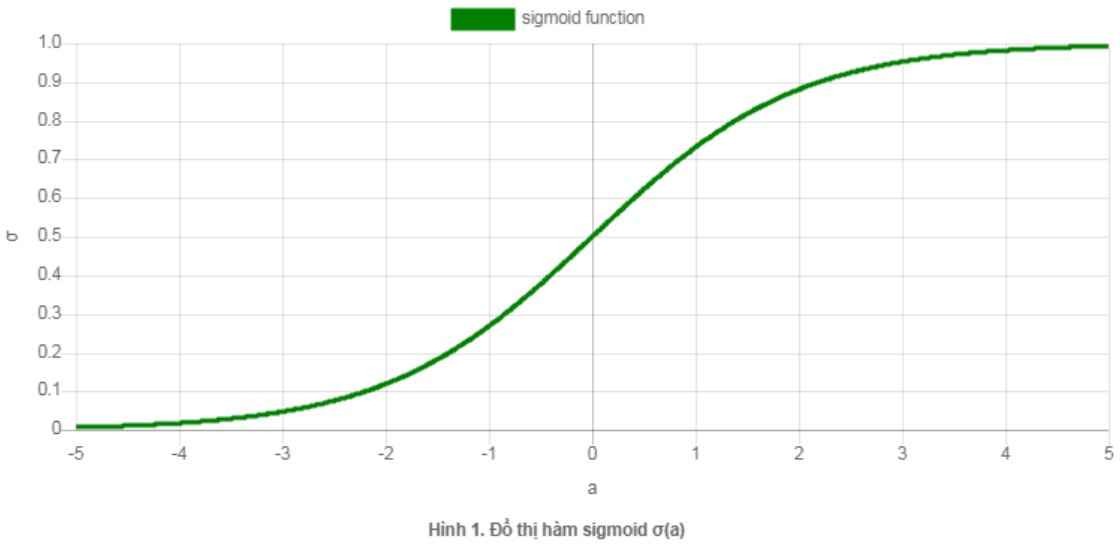
Nếu ta đặt:

$$a = \ln\frac{p(x\,|y_0)\,p(y_0)}{p(x\,|y_1)\,p(y_1)}$$

Ta có:

$$p(y_0|x) = \frac{1}{1+e^{-a}} = \sigma(a)$$

Hàm ở trên được gọi là hàm sigmoid của biến a, khi vẽ phân phối của a và hàm sigmoid, ta có:



Hình 1. Đồ thị hàm sigmoid σ(a)

Hình 2.5: Đồ thị hàm sigmoid

Có thể thấy dải xác suất đầu ra luôn nằm trong khoảng [0,1]. Dựa vào từng bài toán, chúng ta sẽ xác định 1 điểm cắt lớp k, với xác suất p < k, chúng ta coi với bộ số đầu vào x tương ứng, đầu ra (biến y_0) sẽ được gán nhãn là 0, trường hợp còn lại y_0 được gán nhãn 1.

KẾT CHƯƠNG

Chương này tập trung chủ yếu trình bày các nội dung liên quan tới:

- Phát biểu bài toán
- Giới thiệu, mô tả dữ liệu
- Mô hình hành vi người dùng
- Thuật toán rừng ngẫu nhiên
- Thuật toán phân loại Naive Bayes
- Thuật toán hồi quy Logistic

CHƯƠNG 3 - THỬ NGHIỆM VÀ ĐÁNH GIÁ

3.1 Đặt vấn đề

Ứng dụng thuật toán học máy trong lĩnh vực kinh doanh viễn thông sử dụng dữ liệu lịch sử của tập khách hàng để xây dựng các mô hình có khả năng phân loại, dự đoán nhu cầu sử dụng của khách hàng. Tập kết quả đó sẽ được dùng để hỗ trợ các đơn vị kinh doanh truyền thống đưa ra quyết định trong các chiến dịch kinh doanh của doanh nghiệp.

Xuất phát từ các yêu cầu của đơn vị kinh doanh sau quá trình phân tích số liệu và hành vi tập khách hàng thì bài toán sẽ cơ bản được định hình với những tiêu chí như tập khách hàng tác động là ai, danh sách sản phẩm truyền thông là gì, thời điểm truyền thông là bao giờ. Thay vì việc phải truyền thông toàn bộ tập khách hàng lớn với chi phí lớn và hiệu quả không cao thì việc xây dựng các mô hình dự đoán trước nhu cầu sử dụng dịch vụ viễn thông, sở thích của khách hàng sẽ giúp nâng cao hiệu quả của các chiến dịch kinh doanh. Nếu như các đơn vị kinh doanh chỉ đánh giá hành vi của khách hàng bằng các phương pháp thống kê cơ bản trên một vài trường dữ liệu thì mô hình học máy có khả năng đánh giá dựa trên hàng trăm, hàng nghìn trường dữ liệu. Từ đó nhận diện hành vi của khách hàng một cách rõ ràng hơn giúp các chiến dịch kinh doanh tác đông đúng người cần, đúng sản phẩm khách hàng mong muốn sử dụng, đem lại hiệu quả cao hơn.

3.2 Xác định bài toán

Mục tiêu bài toán: Xây dựng mô hình dự đoán tập khách hàng có nhu cầu sử dụng gia tăng về lưu lượng, tiêu dùng dịch vụ. Song song với đó là xây dựng mô hình đề xuất sản phẩm viễn thông phù hợp với nhu cầu gia tăng tiêu dùng của khách hàng. Thử nghiệm xây dựng mô hình dự đoán lần lượt với 3 thuật toán là Hồi quy tuyến tính, Phân loại Naïve Bayes và Rừng ngẫu nhiên (RF). Từ đó so sánh hiệu năng để tìm ra thuật toán phù hợp nhất với bộ dữ liệu đang xét. Sau đó ứng dụng kết quả dự đoán của mô hình vào thực tế so sánh hiệu quả dựa trên các chỉ số và tỉ lệ dự đoán đúng tự nhiên.

Điều kiện lọc dữ liệu:

TB di động trả trước hoạt động 2 chiều, tuổi TB > 6 tháng, thuê bao sử dụng thiết bị smart phone;

Loại bỏ nhóm thuê bao đặc biệt:

- Thuê bao nghi ngờ sử dụng multisim
- TB có sử dụng bất kỳ gói Data/Combo/Thoại/SMS nào trong cả 2 tháng (tn-1, tn-2)

Nguồn dữ liệu: Dữ liệu được sử dụng từ bảng tổng hợp dữ liệu lịch sử xử dụng dịch vụ viễn thông trong tháng của khách hàng. Bao gồm các nhóm dữ liệu liên quan tới sử dụng dịch vụ data, thoại, nhắn tin, tiêu dùng. Các dữ liệu có trong dataset:

- Dữ liệu về các thuê bao: trạng thái thuê bao.

- Dữ liệu hành vi sử dụng: lưu lượng thoại, tin nhắn, data, vas trong quá khứ.

- Dữ liệu tiêu dùng: doanh thu thoại, tin nhắn, data, vas trong quá khứ.

- Dữ liệu nạp thẻ: số lần nạp thẻ, mệnh giá, tổng doanh thu nạp thẻ trong quá khứ

Đầu ra: Bảng dữ liệu chứa thông tin khách hàng bao gồm: số thuê bao, xác suất sự kiện có nhu cầu dùng tăng lưu lượng, xác suất từng sản phẩm mục tiêu tương ứng với mỗi khách hàng.

3.3 Quy trình xây dựng mô hình học máy

Quy trình xây dựng một mô hình học máy cơ bản sẽ gồm các bước sau:

Bước 1: Tiền xử lý dữ liệu

Bước 2: Phân tích, khai phá dữ liệu và lựa chọn thuộc tính dữ liệu

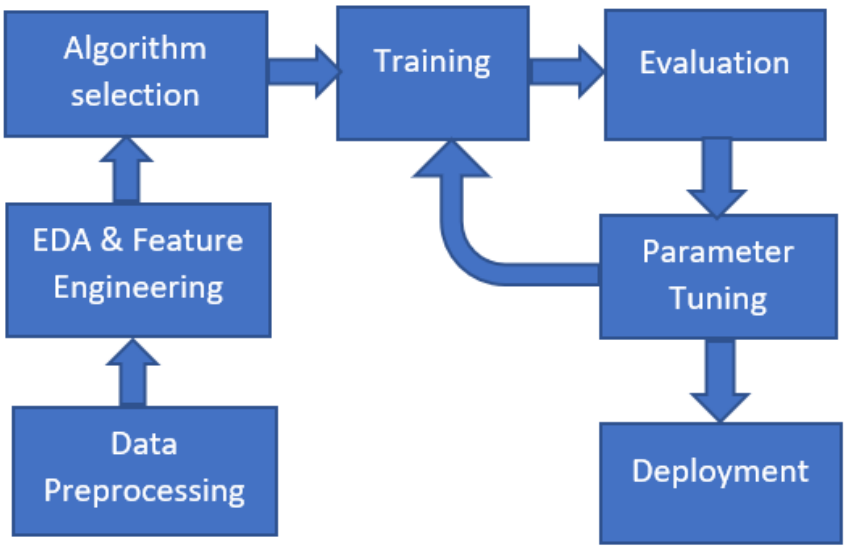
Bước 3: Lựa chọn thuật toán học máy phù hợp với tính chất dữ liệu và yêu cầu bài toán

Bước 4: Xây dựng và huấn luyện mô hình học máy.

Bước 5: Đánh giá hiệu năng dự đoán của mô hình

Bước 6: Điều chỉnh tham số mô hình.

Bước 7: Ứng dụng triển khai kết quả dự đoán từ mô hình.



Hình 3.1: Các bước xây dựng mô hình học máy

3.4 Thực nghiệm

3.4.1 Nhập vào các thư viện

```
%pyspark
from pyspark.sql import SparkSession
import pyspark.sql.functions as F
from pyspark.sql import SQLContext
from pyspark.ml import Pipeline
from pyspark.ml.regression import
RandomForestRegressor,LinearRegression,GBTRegressor
from pyspark.ml.classification import RandomForestClassifier
from pyspark.ml.feature import VectorIndexer
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.evaluation import BinaryClassificationEvaluator
from pyspark.ml.feature import OneHotEncoder, StringIndexer, VectorAssembler
from pyspark.ml.tuning import CrossValidator, ParamGridBuilder
```

```
from sklearn.metrics import precision_score, recall_score, confusion_matrix,
precision_recall_curve
from sklearn.metrics import roc_curve, auc
from matplotlib import pyplot as plt
from pyspark.sql.types import IntegerType, FloatType
from pyspark.sql.functions import *
spark = SparkSession.builder.appName("tuannq_model").getOrCreate()
sc =spark.sparkContext
import seaborn as sns
import pandas as pd
import math
import builtins
from namnt54.DAC_utils import DAC_utils
import dac_automl
from dac_automl.autotuning.classification_model_selector import
ClassificationModelSelector
from dac_automl.autotuning.regression_model_selector import
RegressionModelSelector
from dac_automl.autotuning.base_model_selector import BaseModelSelector
from dac_automl.feature_selection.feature_selector import FeatureSelector
```

3.4.2 Khai báo biến ngày tháng

```
%pyspark
PARALLEL_LEVEL_TRAIN = 128
PARALLEL_LEVEL_TEST = 128

spark.conf.set("spark.sql.shuffle.partitions", PARALLEL_LEVEL_TEST)
spark.conf.set("spark.default.parallelism", PARALLEL_LEVEL_TEST)

df_month_n = sqlContext.sql("""Select
date_Format(date_add(last_day(add_months(current_date, -2)),1),'yyyyMM01') as
month_n""")
df_month_n_1 = sqlContext.sql("""Select
date_Format(date_add(last_day(add_months(current_date, -3)),1),'yyyyMM01') as
month_n_1""")
df_month_n_2 = sqlContext.sql("""Select
date_Format(date_add(last_day(add_months(current_date, -4)),1),'yyyyMM01')as
month_n_2""")
```

```
df_month_n_3 = sqlContext.sql("""Select
date_Format(date_add(last_day(add_months(current_date, -5)),1),'yyyyMM01')as
month_n_3""")
df_month_n_4 = sqlContext.sql("""Select
date_Format(date_add(last_day(add_months(current_date, -6)),1),'yyyyMM01')as
month_n_4""")

partition_n = df_month_n.collect()[0][0]
partition_n_1 = df_month_n_1.collect()[0][0]
partition_n_2 = df_month_n_2.collect()[0][0]
partition_n_3 = df_month_n_3.collect()[0][0]
partition_n_4 = df_month_n_4.collect()[0][0]

month_n = df_month_n.select(df_month_n.month_n.substr(5,2)).collect()[0][0]
month_n_1 =
df_month_n_1.select(df_month_n_1.month_n_1.substr(5,2)).collect()[0][0]
month_n_2 =
df_month_n_2.select(df_month_n_2.month_n_2.substr(5,2)).collect()[0][0]
month_n_3 =
df_month_n_3.select(df_month_n_3.month_n_3.substr(5,2)).collect()[0][0]
month_n_4 =
df_month_n_4.select(df_month_n_4.month_n_4.substr(5,2)).collect()[0][0]

year_n =  df_month_n.select(df_month_n.month_n.substr(1,4)).collect()[0][0]
year_n_4 =
df_month_n_4.select(df_month_n_4.month_n_4.substr(1,4)).collect()[0][0]
```

```
%pyspark
MODEL_NAME = 'upgrade_sim_4g'
MODEL_VERSION = partition_n

CLASS_COL = '4g_next_month'
CLASS_NAME = '4g_next_month'

# update parameters every month
TRAIN_VALID_MONTH = [int(month_n_3),int(month_n_2)]
```

```
TEST_MONTH = [int(month_n_1)]
INFER_MONTH = [int(month_n)]

DATASET_MONTH = TRAIN_VALID_MONTH + TEST_MONTH + INFER_MONTH
DATASET_YEAR = [year_n, year_n_4]
DATASET_PAR =
[int(partition_n_3),int(partition_n_2),int(partition_n_1),int(partition_n)]

model_path =
"/work_zone/upsell_core/data_all/adp_model_{}_{}".format(MODEL_NAME,
MODEL_VERSION)
model_n_1 =
"/work_zone/upsell_core/data_all/adp_model_{}_{}".format(MODEL_NAME,
partition_n_1)
```

3.4.3 Import cơ sở dữ liệu

```
%pyspark
#base 2 dành cho c360_v1_9, month từ tháng 8 trở đi
query = '''
select a.*, b.telecom_service_id, b.status, b.age_tb, b.is_dcom, b.post,
b.register_infra_2g, b.register_infra_3g, b.register_infra_4g,
b.type_infra_home,
case when b.os_type = "SmartPhone" then 1 else 0 end as os_type, b.act_status,
b.is_3k3d, b.is_4g_thuc, b.is_attach, b.is_khm, b.sub_segment, case when
b.sex_id = "nam" then 1 else 0 end as sex_id
,b.is_sim_4g, b.is_mocha, b.change_sim4g, b.change_device, b.is_multisim
,case when b.lv_area = "Thành thị" then 1 else 0 end as lv_area
from
(select * from test_db.c360_v1_9 where year = 2020) a
inner join
(select *
from f_sub_info_d
where (sub_id is not null) and (trim(sub_id)!='') and (isdn is not null) and
(trim(isdn) !='')
and (act_status not like '%3%') and status ='2' and (is_dcom == 0) and
is_multisim == 0
and status = 2
and device_type like '%4G%'
```

```
and partition in ('20200731','20200831'))b
on a.p_isdn = b.isdn and a.month = substring(b.partition,5,2)
'''


print(query)

base2 = sqlContext.sql(query)
```

3.4.4 Tiền xử lý dữ liệu

Khai báo hàm tìm kiếm các trường dữ liệu chỉ gồm một giá trị duy nhất hoặc chứa nhiều giá trị NULL.

```
%pyspark
def drop_null_columns(df, no_drop):
    """
    This function drops all columns which contain 1/3 null values.
    param df: A PySpark DataFrame
    """
    null_counts = df.select([F.count(F.when(F.col(c).isNull(), c)).alias(c) for
c in df.columns]).collect()[0].asDict()
    to_drop = [k for k, v in null_counts.items() if v > no_drop]
    return to_drop


def drop_only_1_value_cols(df):
    count_distinct_df =
df.select([approx_count_distinct(x).alias("{0}".format(x)) for x in
df.columns])
    dict_of_columns = count_distinct_df.toPandas().to_dict(orient='list')
    distinct_columns=[k for k,v in dict_of_columns.items() if v == [1]]
    return distinct_columns
```

Loại bỏ các trường dữ liệu có số lượng bản ghi NULL nhỏ hơn 500000

```
%pyspark
no_drop =500000
a = drop_null_columns(base2,no_drop)

base = base2.drop(*a)
base.columns
```

Loại bỏ các trường dữ liệu chỉ gồm duy nhất một giá trị

```
b = drop_only_1_value_cols(base)
base = base.drop(*b)
```

Ép kiểu dữ liệu với từng trường dữ liệu chỉ định

```
%pyspark
base = base.withColumn("sin_month", base["sin_month"].cast(FloatType()))
base = base.withColumn("cos_month", base["cos_month"].cast(FloatType()))
base = base.withColumn("total_charge_max_price_pkg_current_month",
base["total_charge_max_price_pkg_current_month"].cast(FloatType()))
base = base.withColumn("total_charge_max_price_pkg_month_n_1",
base["total_charge_max_price_pkg_month_n_1"].cast(FloatType()))
base = base.withColumn("total_charge_max_price_pkg_month_n_2",
base["total_charge_max_price_pkg_month_n_2"].cast(FloatType()))
base = base.withColumn("total_charge_max_price_pkg_month_n_3",
base["total_charge_max_price_pkg_month_n_3"].cast(FloatType()))
base = base.withColumn("total_charge_max_price_pkg_current_qtr",
base["total_charge_max_price_pkg_current_qtr"].cast(FloatType()))
base = base.withColumn("total_charge_max_price_pkg_qtr_1",
base["total_charge_max_price_pkg_qtr_1"].cast(FloatType()))
base = base.withColumn("flag_change_in_province_current_qtr",
base["flag_change_in_province_current_qtr"].cast(IntegerType()))
base = base.withColumn("flag_change_in_province_current_month",
base["flag_change_in_province_current_month"].cast(IntegerType()))
```

Thay thế các giá trị NULL bằng -1

```
%pyspark
base =base.fillna(-1, subset =
['p_age','p_is_dcom','p_is_199_197','p_is_vip','p_is_attach'
,'v_out_used_cell_top1_latitude','v_out_used_cell_top2_latitude','flag_data_non
_cycle_month_n_1'
,'flag_data_non_cycle_month_n_2','flag_data_non_cycle_month_n_3','flag_data_non
_cycle_current_qtr'])
```

Thay thế các giá trị NULL bằng 0

```
%pyspark
base =base.fillna(0, subset =
['number_distinct_viettel_products_current_month','number_distinct_viettel_prod
ucts_month_n_1'
,'number_distinct_viettel_products_month_n_2','number_distinct_viettel_products
_current_qtr'
,'delta_number_distinct_viettel_products_current_month','delta_number_distinct_
viettel_products_month_n_1'
,'delta_number_distinct_viettel_products_month_n_2','delta_number_distinct_viet
tel_products_current_qtr','delta_perc_number_distinct_viettel_products_current_
month'])
```

Thay thế các giá trị NULL bằng 1

```
%pyspark
#fill null vào multisim
base =base.fillna(1, subset = ['group_number_of_isdn'])
```

Tạo nhãn khi tiêu dùng data tháng hiện tại = 0, tháng kế tiếp lớn hơn 0, có sử dụng gói

```
%pyspark
main3 = sqlContext.sql(" select *, case when
((data_total_cost_adjust_refund_next_month >
data_total_cost_adjust_refund_current_month) and
(type_data_package_current_month in ('goi_n_ngay')) and
(type_data_package_next_month in ('goi_combo_thang',
'goi_thang_co_mua_them','goi_thang_ko_mua_them')) and
(nvl(normalised_price_of_max_normalised_price_data_product_next_month,0) >=
nvl(normalised_price_of_max_normalised_price_data_product_current_month,0)) and
nvl(normalised_price_of_max_normalised_price_data_product_current_month,0) > 0
) then 1 else 0 end as upgrade_next_month_normalized_price from main2 where
nvl(data_total_cost_adjust_refund_current_month,0) > 0 and
type_data_package_current_month in ('goi_n_ngay') ")
```

Chia tập dữ liệu huấn luyện, test và infer

```
%pyspark
test_df      =      df.filter("""year      =      {}      and      month      in
({})""".format(year_n,','.join(map(str, TEST_MONTH))))
```

```
train_df = df.filter("""year in ({}) and month in ({})""".format(year_n,','.join(map(str, TRAIN_VALID_MONTH))))
infer_df = df.filter("""year = {} and month in ({})""".format(year_n,','.join(map(str, INFER_MONTH))))
```

Chia tập dữ liệu train và valid bằng cách lấy ngẫu nhiên từ cùng một tập dữ liệu với tỉ lệ xác định

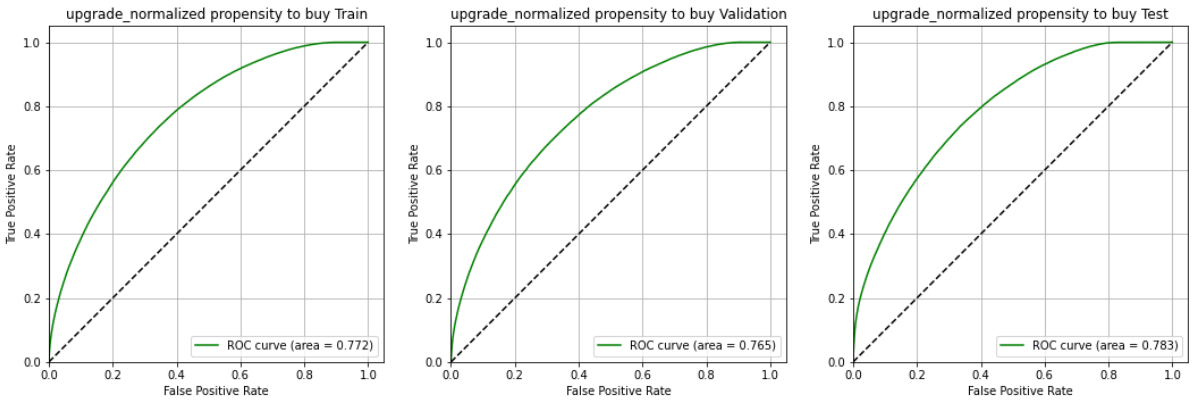
```
%pyspark
#chia train set_ test set
def train_val_split(df, train_fraction, val_fraction, class_col):
    '''
    df: dataframe
    train_fraction: fraction used for training [0-1)
    val_fraction: fraction used for validation [0-1)
    class_col: col name for class
    '''
    if train_fraction + val_fraction > 1:
        raise ValueError('Train val fraction must have sum less than or equal to 1')
    train = df.sampleBy(class_col, fractions ={0: train_fraction, 1: train_fraction}, seed=2019)
    left_over = df.subtract(train)
    valid_fraction_from_left = val_fraction / (1 - train_fraction)
    valid = left_over.sampleBy(class_col, fractions ={0: valid_fraction_from_left, 1: valid_fraction_from_left}, seed=2019)
    return train, valid
TRAIN_FRACTION = 0.5
VAL_FRACTION = 0.2
# SAMPLING_RATIO = 9
train, valid = train_val_split(train_df, TRAIN_FRACTION, VAL_FRACTION, CLASS_COL)
```

```
df_writer =
pyspark.sql.DataFrameWriter(infer_pred_export.select(['p_sub_id',CLASS_COL,'pro
bability','partition']))
df_writer.partitionBy('partition').saveAsTable('f_upgrade_all_goi_n_ngay_monthl
y_auto', format='parquet', mode='overwrite',
path='/work_zone/upsell_core/data_all/tungnx15/f_upgrade_all_goi_n_ngay_monthly
_auto')
```

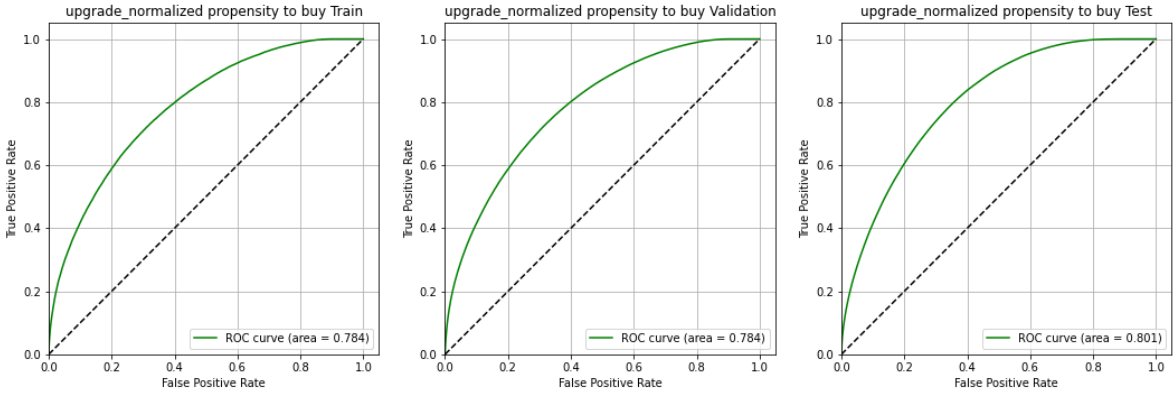
3.5 Kết quả thực nghiệm

Trong khuôn khổ bài luận văn em đã thử nghiệm xây dựng 3 mô hình dự đoán nhu cầu dùng tăng dịch vụ data của nhà mạng Viettel. Với cùng một bộ dữ liệu huấn luyện mô hình bao gồm các dữ liệu liên quan tới lịch sử sử dụng data, gọi thoại, nhắn tin, nạp tiền của các thuê bao sử dụng dịch vụ viễn thông của nhà mạng Viettel. Cả ba tập dữ liệu đầu vào 3 mô hình này là giống nhau và cùng được tiền xử lý dữ liệu như nhau để đảm bảo công bằng trong việc so sánh hiệu năng dự đoán của các mô hình.

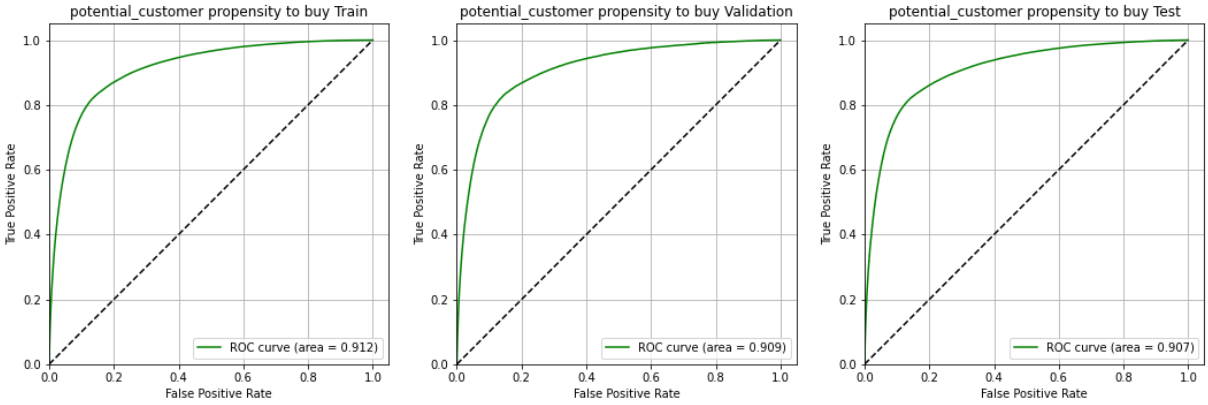
Dưới đây lần lượt là biểu đồ ROC của thuật toán phân loại Naïve Bayes, hồi quy Logistic, rừng ngẫu nhiên.



Hình 3.2: ROC thuật toán Naïve Bayes

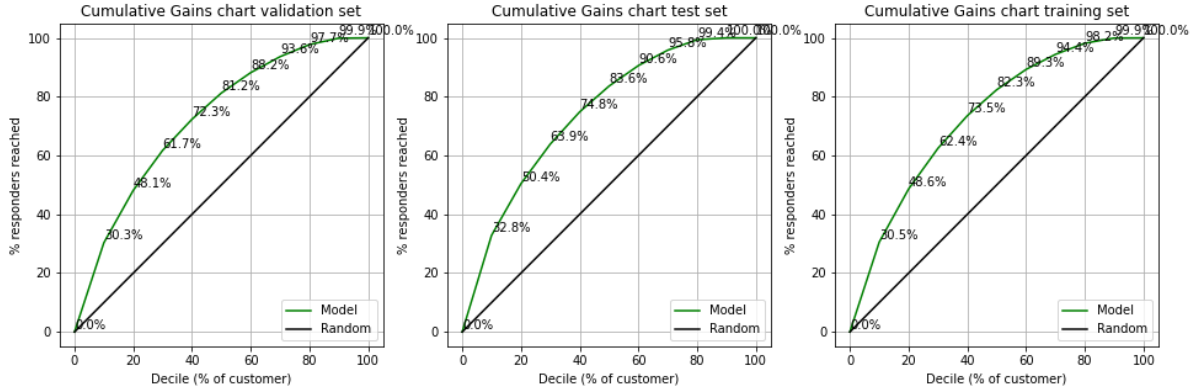


Hình 3.3: ROC thuật toán hồi quy Logistic

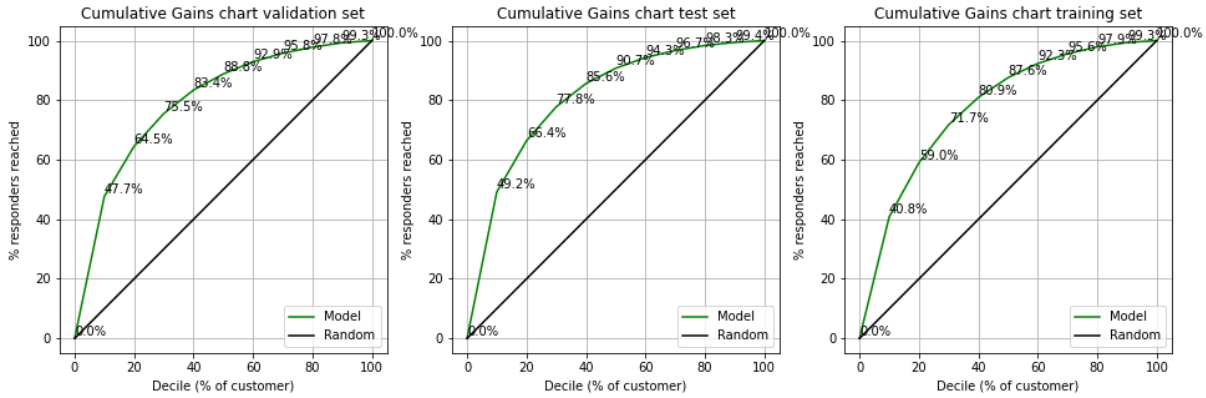


Hình 3.4: ROC thuật toán rừng ngẫu nhiên

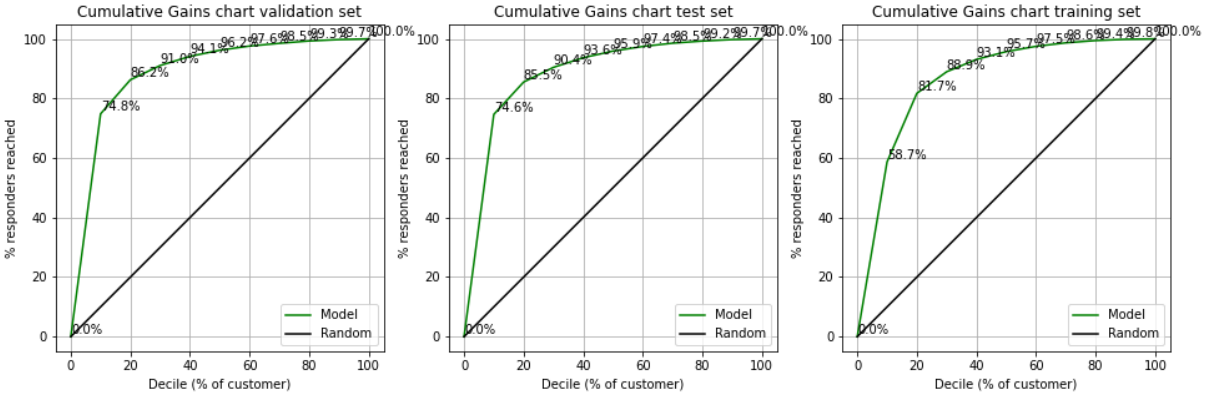
Biểu đồ cumulative thể hiện tỉ lệ tổng số dự đoán đúng khách hàng có nhu cầu dùng tăng dịch vụ trong tổng số khách hàng có nhu cầu dùng tăng dịch vụ trên từng tập dữ liệu. Theo thứ tự lần lượt là kết quả trên tập xác thực, tập huấn luyện và tập kiểm tra.



Hình 3.5: Cumulative gain thuật toán Naïve Bayes

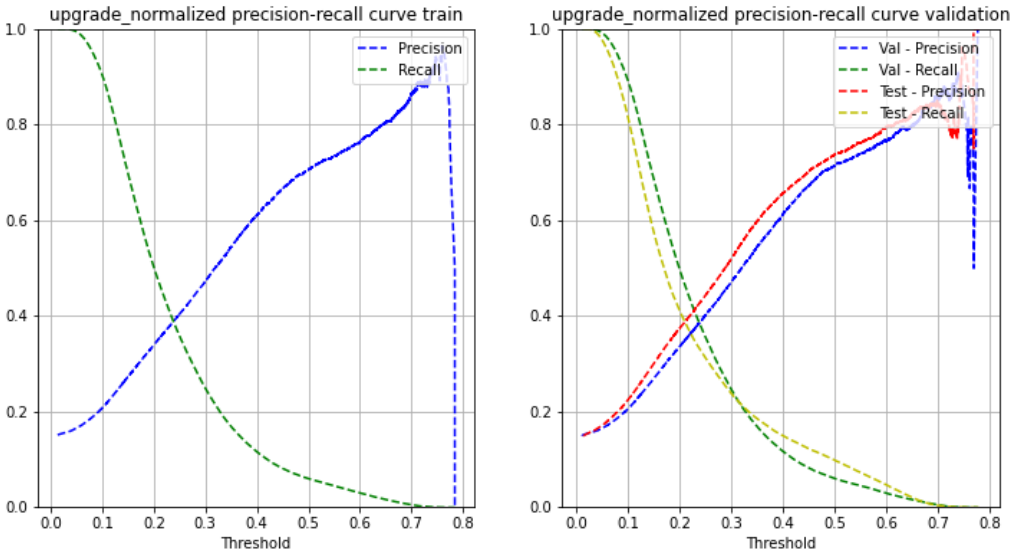


Hình 3.6: Cumulative gain hồi quy Logistic

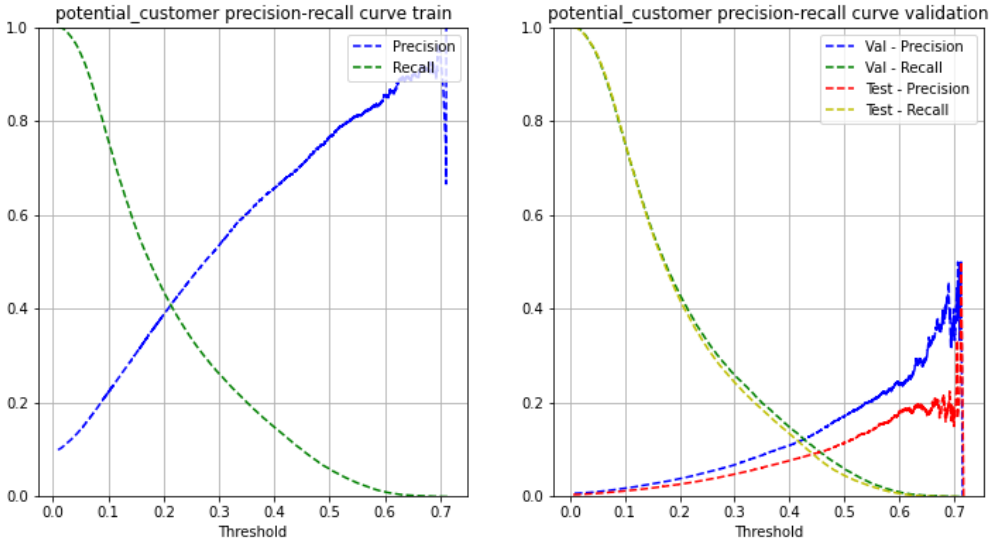


Hình 3.7: Cumulative gain thuật toán rừng ngẫu nhiên

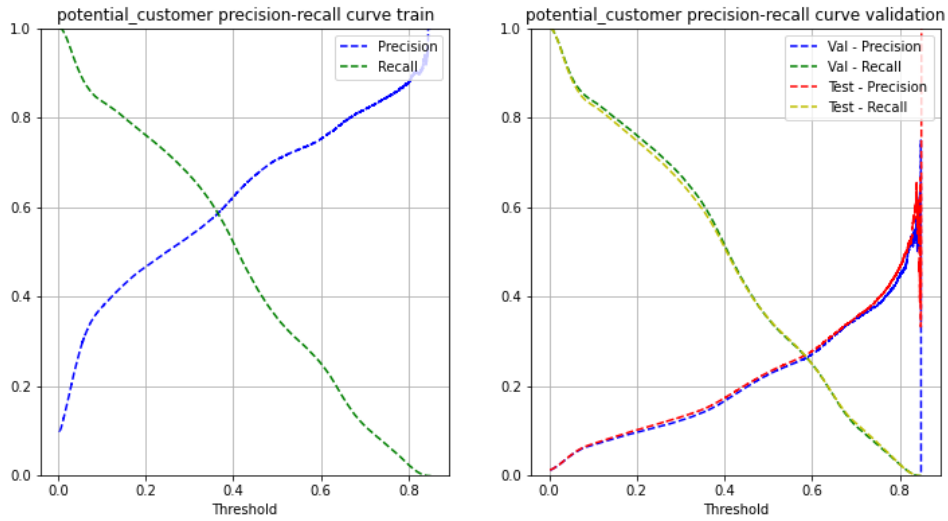
Biểu đồ biểu diễn độ đo precision và recall thể hiện tỉ lệ dự đoán đúng và độ phủ của mô hình trên hai tập dữ liệu huấn luyện và xác thực. Từ mỗi điểm trên hình ta sẽ xác định được ứng với từng phần trăm của tập dữ liệu dự đoán thì sẽ có độ chính xác và độ phủ là bao nhiêu. Nếu phần trăm của tập dữ liệu càng lớn thì độ chính xác càng giảm và độ phủ càng tăng.



Hình 3.8: Precision-Recall thuật toán Naïve Bayes

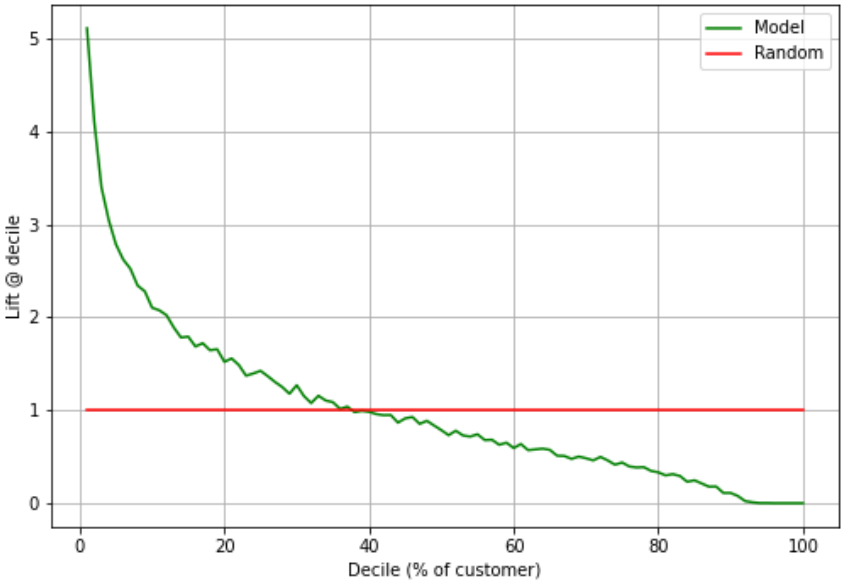


Hình 3.9: Precision-Recall thuật toán hồi quy Logistic

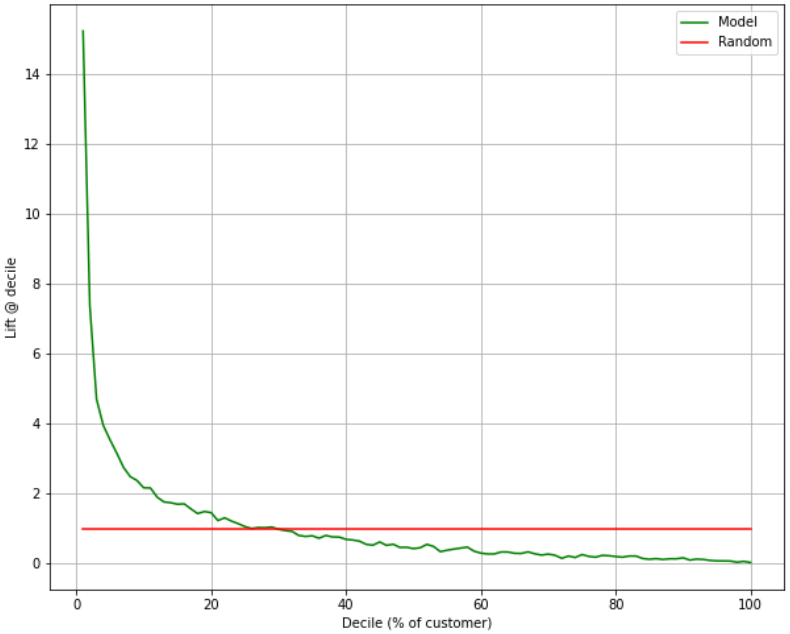


Hình 3.10: Precision-Recall thuật toán rừng ngẫu nhiên

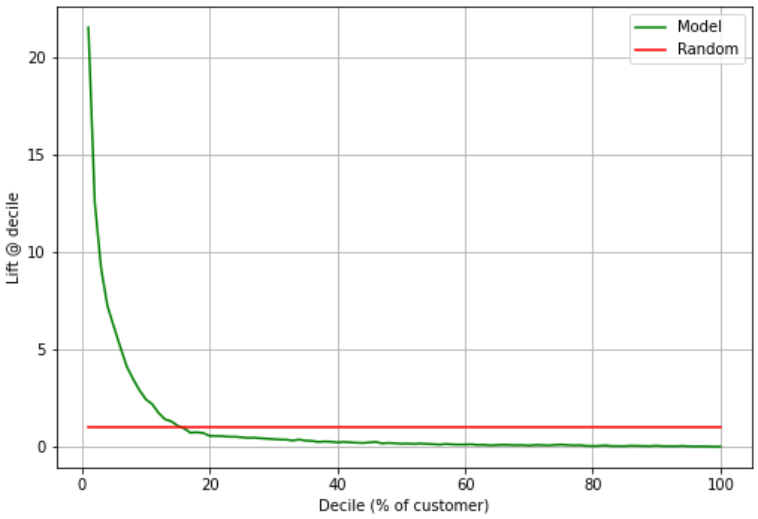
Biểu đồ Lift thể hiện hiệu quả dự đoán chính xác so với tỉ lệ dự đoán tự nhiên trên từng phần trăm của tập dữ liệu dự đoán. Đường màu xanh được vẽ từ kết quả dự đoán của mô hình còn đường màu đỏ là tỉ lệ dự đoán tự nhiên. Có thể thấy đường màu xanh có tỉ lệ dự đoán đúng cao hơn đường màu đỏ ở những phần trăm đầu của tập dữ liệu dự đoán và giảm dần ở những phần trăm sau do mô hình đã nhận diện được các khách hàng có nhu cầu dùng tăng dịch vụ data và gán xác suất sự kiện giảm dần cho những khách hàng ít nhu cầu dùng tăng và không có nhu cầu. Mô hình nào cho ra tỉ lệ chênh lệch so với dự đoán tự nhiên càng lớn ở những dải điểm đầu chứng tỏ khả năng phân loại nhãn càng tốt. Việc này sẽ giúp đơn vị kinh doanh triển khai các chiến dịch với quy mô nhỏ hơn mà vẫn đảm bảo mang lại hiệu quả lớn hơn.



Hình 3.11: Biểu đồ Lift thuật toán Naïve Bayes



Hình 3.12: Biểu đồ Lift thuật toán hồi quy Logistic



Hình 3.13: Biểu đồ Lift thuật toán Rừng ngẫu nhiên

3.6 Xây dựng hệ thống

3.6.1 Giới thiệu hệ thống

Mục đích: Xây dựng hệ thống quản lý luồng công việc của ứng dụng học máy, bao gồm toàn bộ các quá trình huấn luyện, thử nghiệm, dự đoán mà một nhà khoa học dữ liệu thường phải thực hiện.

Một số mục tiêu cụ thể:

- Cung cấp giao diện quản lý các tiến trình huấn luyện, thử nghiệm, dự đoán
- Cung cấp giao diện quản lý các mô hình học máy
- Cơ chế sử dụng AutoML để tự động chọn ra mô hình và bộ tham số tối ưu
- Cơ chế khai báo thủ công tham số và huấn luyện nhiều mô hình đồng thời

Các chức năng chính:

➢ Tạo mới mô hình:

- Tạo mới các mô hình với 2 chế độ: người dùng tự khai báo tham số, hoặc sử dụng AutoML tự động chọn lựa tham số

- Khi tạo mới mô hình xong có thể chọn đồng thời các tùy chọn để huấn luyện, thử nghiệm hoặc dự đoán

- Chọn tập dữ liệu tương ứng với các tùy chọn trên

- Chọn địa chỉ lưu trữ trên HDFS để lưu trữ báo cáo và đầu ra

- Sử dụng mô hình:

- Chọn mô hình trong danh sách mô hình đã được huấn luyện
- Khi chọn mô hình xong có thể chọn đồng thời các tùy chọn để huấn luyện, thử nghiệm, dự đoán
- Chọn tập huấn luyện tương ứng với các tùy chọn trên
- Chọn địa chỉ trên HDFS để lưu trữ báo cáo và đầu ra

- Quản lý các mô hình:

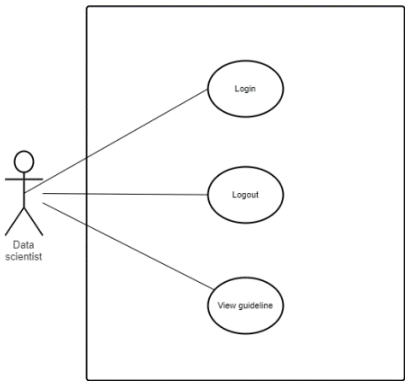
- Hệ thống lưu lại và quản lý các mô hình đã được huấn luyện
- Hệ thống hiển thị danh sách các mô hình của người dùng
- Hệ thống hiển thị chi tiết của từng mô hình (tham số, điểm huấn luyện, điểm xác nhận, ...)
- Hệ thống lưu lại logs, báo cáo cho các lần chạy

- Quản lý các dự án:

- Các mô hình sẽ thuộc 1 dự án nào đó với mục tiêu gom nhóm và quản lý dễ dàng

3.6.2 Biểu đồ ca sử dụng hành vi người dùng

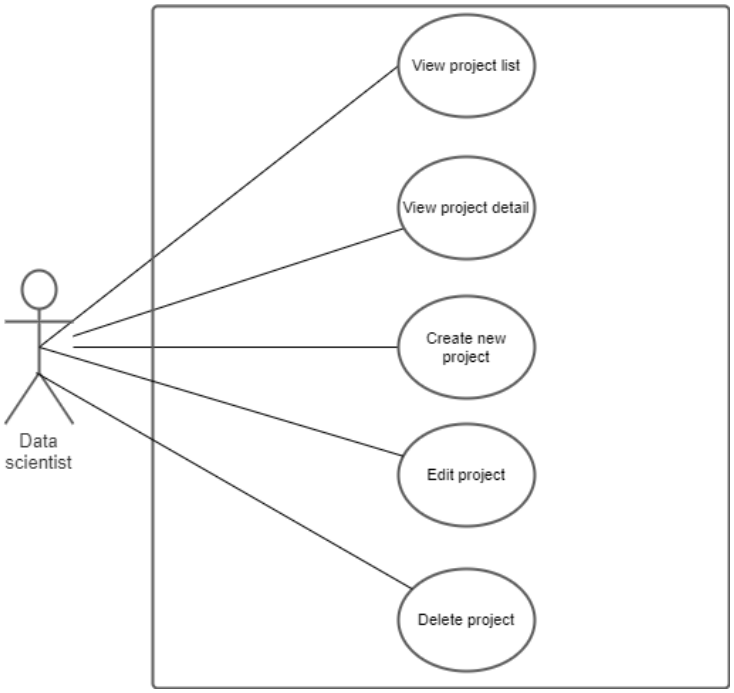
Giúp giảm thiểu quá trình thao tác với mã nguồn, tránh sai sót trong quá trình triển khai chiến dịch. Mô hình hành vi người dùng được biểu diễn dưới ba chức năng chính là đăng nhập, đăng xuất và sử dụng giao diện phần mềm để xây dựng các mô hình dự đoán. Hệ thống được sinh ra không chỉ để dành cho các kỹ sư phân tích dữ liệu sử dụng để xác định mô hình ban đầu mà còn là cơ sở để các đơn vị phân tích nghiệp vụ kinh doanh đưa ra quyết định kinh doanh dựa trên các kết quả dự đoán từ mô hình.



Hình 3.14: Biểu đồ ca sử dụng hành vi người dùng

3.6.3 Biểu đồ ca sử dụng giám sát dự án

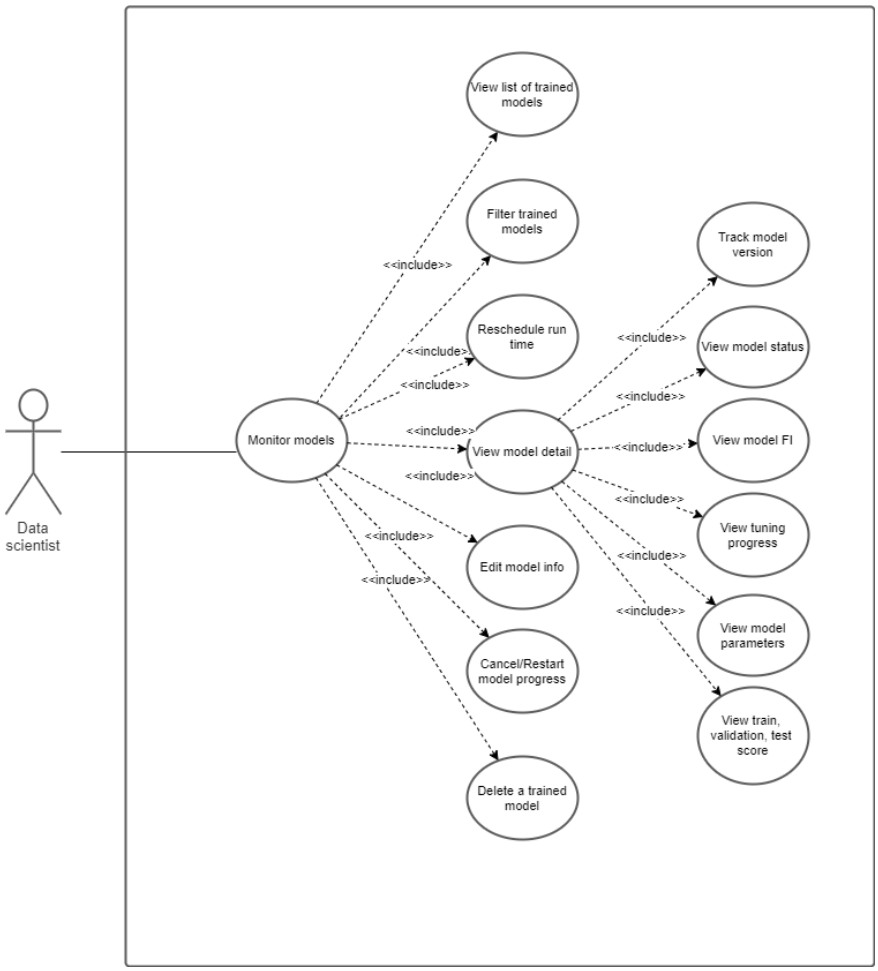
Ở ca sử dụng chức năng giám sát dự án mỗi người dùng có thể xem toàn bộ danh sách dự án hiện tại, xem chi tiết thông tin của từng dự án, khởi tạo dự án mới, chỉnh sửa dự án và xóa những dự án không còn sử dụng nữa. Tuy nhiên đối với từng vị trí, chức năng nhiệm vụ của mỗi cá nhân ở trong dự án sẽ được cấp quyền khác nhau. Ví dụ như chỉ vị trí quản lý dự án mới được phân quyền đầy đủ còn với các thành viên dự án thì sẽ được cấp ít quyền tác động tới dự án hơn. Tránh những trường hợp sự cố đáng tiết do lỗi cá nhân.



Hình 3.15: Biểu đồ ca sử dụng chức năng giám sát dự án

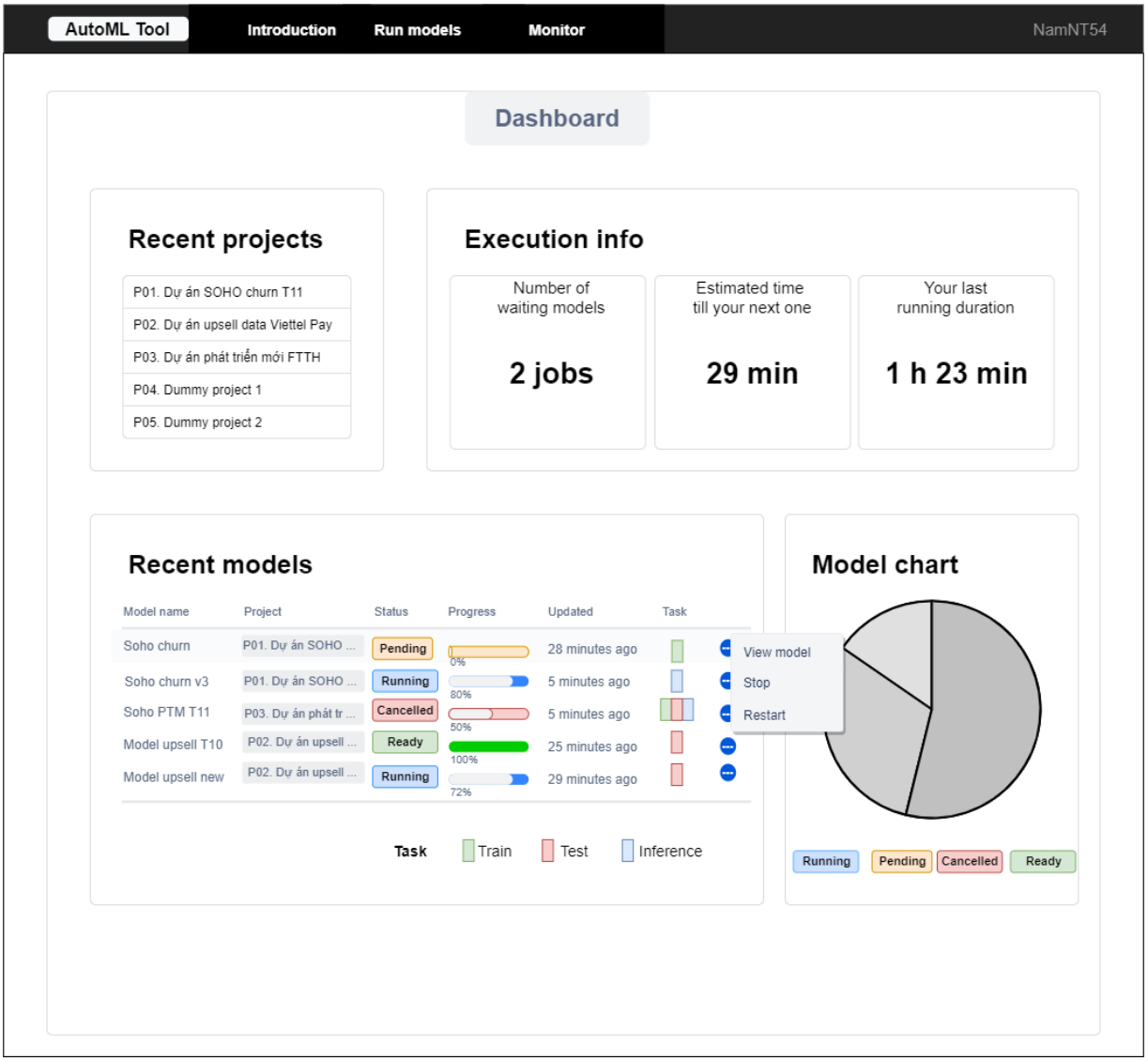
3.6.4 Biểu đồ ca sử dụng giám sát mô hình

Ca sử dụng chức năng giám mô hình bao gồm các mô-đun hiển thị phiên bản mô hình, hiển thị thông tin trạng thái mô hình, xem thông tin liên quan tới mô hình gốc, theo dõi tiến trình hiệu chỉnh mô hình, xem tham số mô hình, xem các thông tin của tập dữ liệu huấn luyện, kiểm tra và thông số hiệu năng dự đoán của mô hình dự đoán. Từ các mô- đun kể trên người sử dụng có thể dễ dàng xem lại lịch sử huấn luyện mô hình, kết quả huấn luyện mô hình và thay đổi tham số, tỉ lệ train test để tìm ra mô hình đáp ứng tốt nhất các yêu cầu của bài toán.

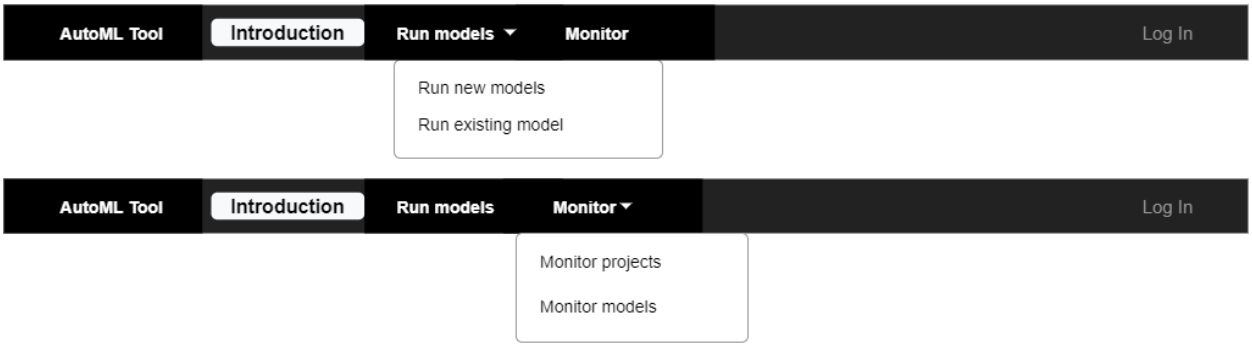


Hình 3.16: Biểu đồ ca sử dụng giám sát mô hình

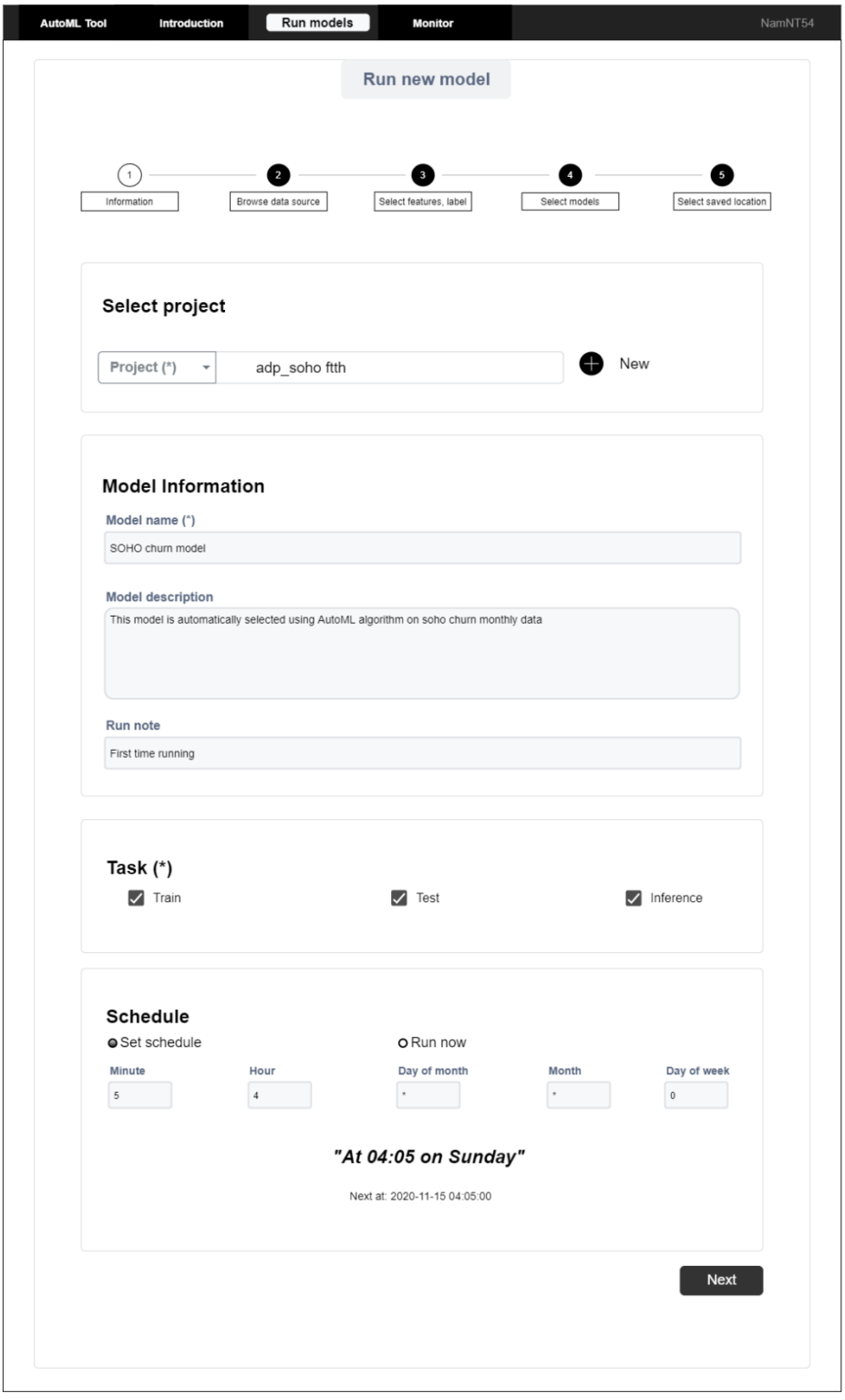
3.6.5 Giao diện Home



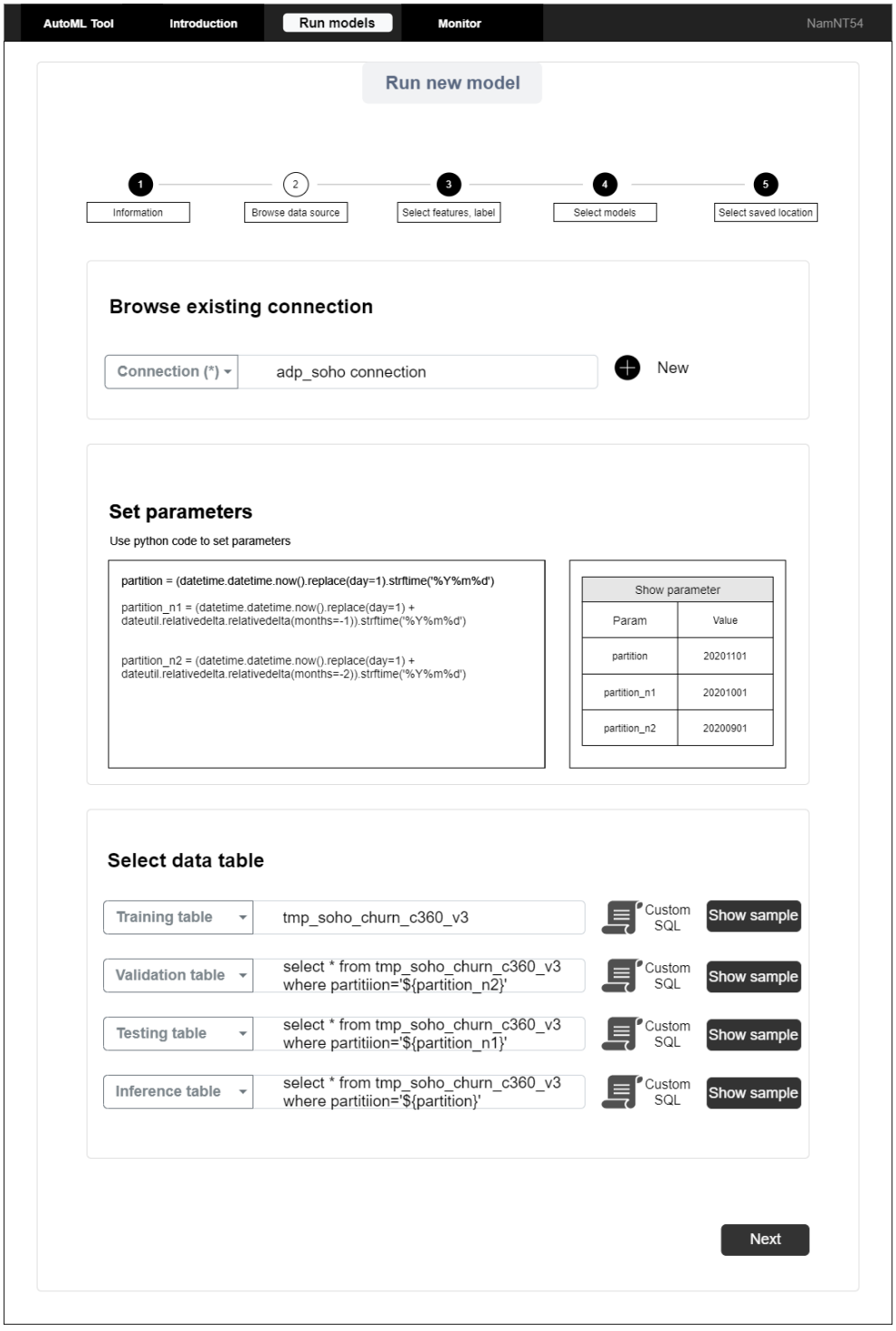
3.6.6 Giao diện thanh điều hướng



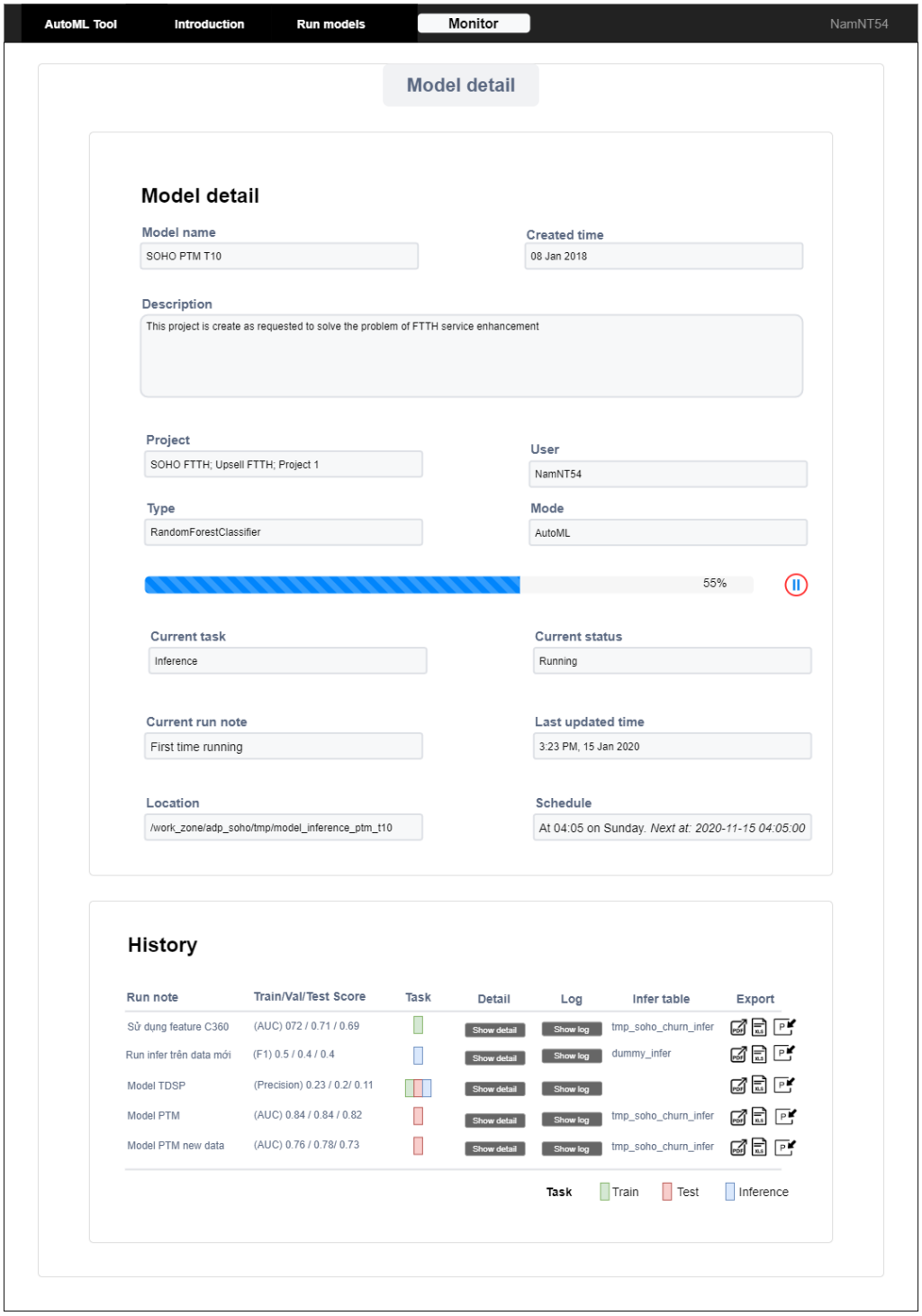
3.6.7 Giao diện thông tin chung



3.6.8 Giao diện nguồn dữ liệu



3.6.9 Giao diện thông tin mô hình



3.7 Kết quả trong triển khai thực tế

3.7.1 Các chỉ số tính hiệu quả triển khai

- **ARPU UPLIFT**

ARPU uplift = ARPU after/ARPU before - 1

**ARPU before: ARPU trước triển khai là tổng tiêu dùng gốc trung bình/khách hàng trong vòng 30 ngày trước ngày triển khai*

**ARPU after: ARPU sau triển khai được tính bằng giá gói khách hàng đăng ký(riêng gói ST30K thì giá gói quy đổi tháng sẽ là 90k)*

- **REVENUE UPLIFT**

REVENUE uplift = (ARPU after - ARPU before) * Number of Take-up Customer

**ARPU before: ARPU trước triển khai là tổng tiêu dùng gốc trung bình/khách hàng trong vòng 30 ngày trước ngày triển khai*

**ARPU after: ARPU sau triển khai được tính bằng giá gói khách hàng đăng ký(riêng gói ST30K thì giá gói quy đổi tháng sẽ là 90k)*

**Number of Take-up Customer: Số lượng khách hàng mua gói*

- **TAKE-UP RATE (TUR)**

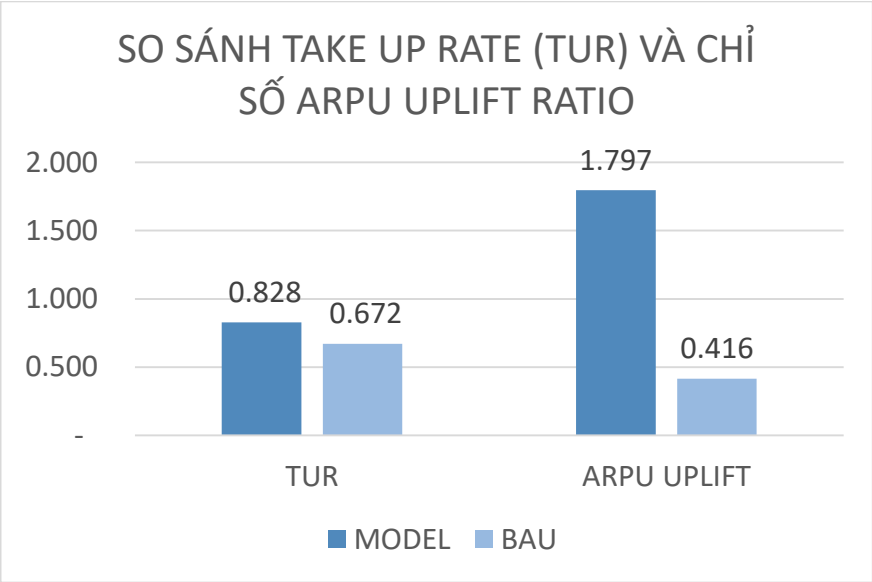
TAKE-UP rate = Number of Take-up Customer/Number of Receivers*100

**Number of Take-up Customer: Số lượng khách hàng mua gói*

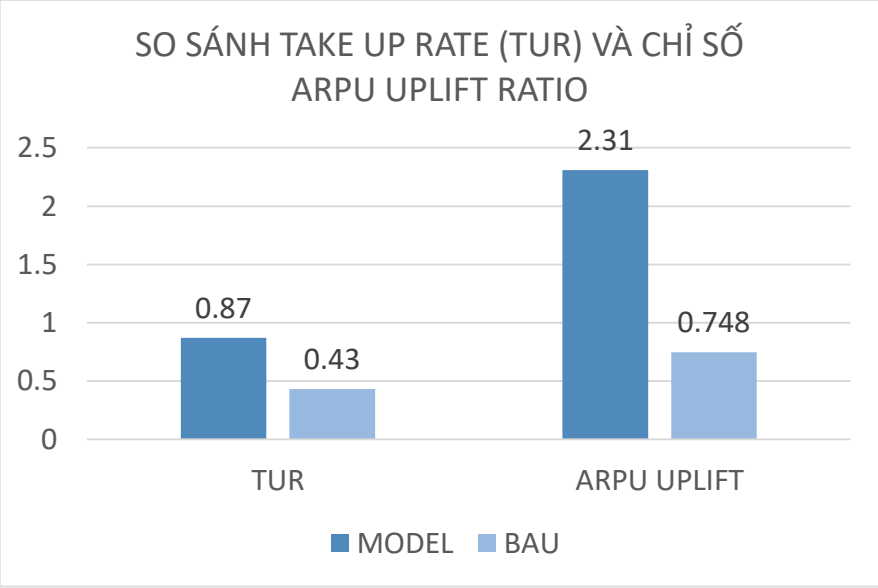
**Number of Receivers: Số lượng khách hàng nhận được tin truyền thông gói*

3.7.2 Kết quả triển khai thực tế

Sản phẩm MIMAX90



Sản phẩm ST70



Giá	Sản phẩm	Tổng số lượng thuê bao nhận được truyền thông	Số lượng thuê bao nhận được truyền thông tập model	Số lượng thuê bao nhận được truyền thông tập BAU	Tổng số lượng thuê bao đăng ký gói - tập model	Tổng số lượng thuê bao đăng ký gói - tập BAU	Tỷ lệ nâng gói tập model	Tỷ lệ nâng gói tập BAU	Tỷ lệ nâng gói Mode / Tỷ lệ nâng gói BAU	Doanh thu tập model	Doanh thu tập BAU
90,000	MIMAX90	552,707	426,016	126,691	2,128	244	0.50%	0.19%	3	191,520,000	21,960,000
70,000	MIMAX70	309,333	222,783	86,550	612	115	0.27%	0.13%	2	42,840,000	8,050,000
25,000	MIMAX25	100,217	100,217	-	317	-	0.32%	-	-	7,925,000	-
90,000	ST90	59,516	58,923	593	235	-	0.40%	-	-	21,150,000	-
70,000	ST70	390,969	269,915	121,054	1,023	149	0.38%	0.12%	3	71,610,000	10,430,000
90,000	F90	94,226	42,167	52,059	240	13	0.57%	0.02%	28.5	21,600,000	1,170,000
TOTAL		1,506,968	1,120,021	386,947	4,555	521	0.41%	0.13%	3.02	356,645,000	41,610,000

KẾT CHƯƠNG

- Mục đích thử nghiệm đánh giá
- Phát biểu ý nghĩa các chỉ số độ đo
- Thử nghiệm
- Xây dựng mô hình dự đoán khả năng gia tăng nhu cầu sử dụng dịch vụ với thuật toán Rừng ngẫu nhiên
- Xây dựng mô hình dự đoán khả năng gia tăng nhu cầu sử dụng dịch vụ với thuật toán Naive Bayes
- Xây dựng mô hình dự đoán khả năng gia tăng nhu cầu sử dụng dịch vụ với thuật toán Hồi quy Logistic
- Xây dựng ứng dụng
- Kết quả triển khai trong thực tế

KẾT LUẬN CHUNG

Các kết quả thu được trong luận văn

Sau khi thử nghiệm lần lượt 3 thuật toán phân loại Naïve Bayes, hồi quy Logistic, rừng ngẫu nhiên đối trên cùng một tập dữ liệu thì thuật toán rừng ngẫu nhiên cho kết quả dự đoán chính xác nhất sau đó tới thuật toán hồi quy Logistic và thuật toán Naïve Bayes cho kết quả dự đoán kém chính xác nhất.

Hiện nay các bài toán mô hình upgrade tương đối tốt và đã đạt tới ngưỡng gần như không thể improve bằng các kĩ thuật thông thường mà chỉ có thể improve dựa trên việc xây dựng các features có giá trị phân loại tốt hơn; thêm vào đó cách xây dựng mô hình upgrade tương đối đơn giản nên trước mắt chưa có phương pháp để improve mô hình này. Thêm vào đó, việc mất cân đối giữa số lượng các sản phẩm trong mô hình cũng ảnh hưởng lớn đến performance chung của mô hình khi đưa ra dự đoán cho các sản phẩm thiểu số.

Định hướng nghiên cứu tiếp theo

Dựa trên những vấn đề đó định hướng hiện tại đang thử nghiệm các phương pháp sau:

- Từ bộ p1 thu được tương ứng với từng sản phẩm, giả định đầu tiên là với score p1 cao hơn tương ứng với thuê bao có khả năng mua sản phẩm ấy cao hơn. Chọn ra sản phẩm có p1 cao nhất để tiến hành back test. Kết hợp kết quả back test với phân tích cluster.
- Từ kết quả thu được của trial 1st cũng như từ performance của các model riêng biệt. Đánh giá rằng các mô hình thu được hoạt động tốt trong khả năng dự đoán xu hướng sử dụng gói của thuê bao, tuy nhiên để đưa đến một xếp hạng score cuối cùng rằng sản phẩm nào sẽ được mua thì cần một phương pháp rõ ràng và chính xác hơn. Vì vậy, thử áp dụng các mô hình phân lớp cho bộ score p1 thu được từ các model trên với mục tiêu là khi đó có thể xây dựng được một "MODEL MASTER" có khả năng tổng hợp kết quả từ các mô hình nhỏ.

Từ kết quả quan sát tại các thử nghiệm trước cũng như quá trình xây dựng mô hình riêng biệt, nhận thấy: Tuy các model đều bị ảnh hưởng bởi hiện tượng imblance và đã sử dụng các phương pháp Downsampling để cải thiện performance. Nhưng các yếu tố có thể ảnh hưởng đến ranking scores của các mô hình không chỉ là tỉ lệ nhãn giữa 0 và 1 (0:1) mà còn một yếu tố chưa được đề cập tới đó chính là số lượng tuyệt đối của các nhãn trong mô hình. Bởi số lượng các thuê bao có mua gói ngày/n-ngày so với số lượng các thuê bao mua gói tháng có số lượng chênh lệch đáng kể.

Vì vậy, để calibrate thành công kết quả các mô hình cần một hàm có khả năng cover được tất cả các yếu tố trên. Từ đó, tiến hành nghiên cứu tìm hiểu các phương pháp calibrate khác thường được áp dụng. Kết quả là một số phương pháp phổ biến như Platt's Scaling và Isotonic Regression đã được đề cập đến trong những tài liệu calibration khác.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] Abdelrahim Kasem Ahmad, Assef Jafar and Kadan Aljoumaa, "Customer churn prediction in telecom using machine learning in big data platform", *Journal of Big data*, 2019, pg.1-24. Available at: https://journalofbigdata.springeropen.com/track/pdf/10.1186/s40537-019-0191-6

[2] Gerard Biau, "Analysis of a Random Forests Model", *Journal of Machine Learning Research* 13 (2012) pg. 1063-1095. Available at: http://www.jmlr.org/papers/volume13/biau12a/biau12a.pdf

[3] Gil Press, *6 Predictions About Data In 2020 And The Coming Decade*, Forbes, Jan 6, 2020. Available at: https://www.forbes.com/sites/gilpress/2020/01/06/6-predictions-about-data-in-2020-and-the-coming-decade/#5dbe212d4fc3

[4] R. Masoud et al., "Using data mining in telecommunication industry: Customer's churn prediction model", *Journal of Theoretical and applied information Technology*, Vol.1, No.2, 2016.pp.322-328. Available at: http://www.jatit.org/volumes/Vol91No2/12Vol91No2.pdf

[5] Saad Ahmed Qureshi, Ammar Saleem Rehman, Ali Mustafa Qamar, Aatif Kamal, Ahsan Rehman, *Telecommunication subscribers' churn prediction model using machine learning*, September 2013, pg. 1-6. Available at: https://www.researchgate.net/publication/257201765_Telecommunication_Subscribers'_Churn_Prediction_Model_Using_Machine_Learning

[6] Osisanwo F.Y, Akinsola J.E.T, Awodele O, Hinmikaiye J. O, Olakanmi O, Akinjobi J, "Supervised Machine Learning Algorithms: Classification and Comparison", *International Journal of Computer Trends and Technology (IJCTT)*, Volume 48 Number 3 June 2017, pg. 128-138. Available at: https://www.researchgate.net/publication/318338750_Supervised_Machine_Learning_Algorithms_Classification_and_Comparison

[7] Lian Yan, R.H. Wolniewicz, R. Dodier, *Predicting customer behavior in telecommunications*, April 2004 Intelligent Systems, IEEE 19(2), pg.50 - 58
Available at:
https://www.researchgate.net/publication/3454180_Predicting_Customer_Behavior_in_Telecommunications

BẢN CAM ĐOAN

Tôi cam đoan đã thực hiện việc kiểm tra mức độ tương đồng nội dung luận văn/luận án qua phần mềm DoIT một cách trung thực và đạt kết quả mức độ tương đồng 10% toàn bộ nội dung luận văn/luận án. Bản luận văn/ luận án kiểm tra qua phần mềm là bản cứng luận văn/ luận án 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 tháng năm 2020

HỌC VIÊN CAO HỌC/NCS

(Ký và ghi rõ họ tên)