HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

NGUYỄN XUÂN SANG

CẢI TIẾN THUẬT TOÁN SVM VỚI SVM SONG SONG, ỨNG DỤNG VÀO PHÂN LỚP VÀ DỰ BÁO SỐ KHÁCH HÀNG SỬ DỤNG DI ĐỘNG

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 NGUYỄN ĐÌNH THUÂN

THÀNH PHỐ HỒ CHÍ MINH – NĂM 2022

LỜI CAM ĐOAN

Tôi xin cam đoan:

1. Những nội dung trong luận văn này là do tôi thực hiện dưới sự hướng dẫn trực tiếp của Thầy **PGS. TS Nguyễn Đình Thuân**.
2. Mọi tham khảo trong luận văn đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian công bố.

Mọi sao chép không hợp lệ, vi phạm quy chế đào tạo tôi xin chịu hoàn toàn trách nhiệm.

Tp. Hồ Chí Minh, ngày 25 tháng 01 năm 2022

Học viên thực hiện luận văn

Nguyễn Xuân Sang

LỜI CÁM ƠN

Em xin dành lời cảm ơn chân thành và sâu sắc nhất đến Thầy **PGS. TS Nguyễn Đình Thuân** người đã truyền cảm hứng về mảng khai phá dữ liệu, khuyến khích và chỉ dẫn tận tình cho em trong từng bước từ khi bắt đầu cho đến khi hoàn thành luận văn của mình.

Em cũng xin dành lời cảm ơn chân thành đến Thầy Cô Học viện Bưu Chính Viễn Thông đã truyền đạt kiến thức vô cùng quý giá và tạo điều kiện thuận lợi cho em trong suốt thời gian học tập và nghiên cứu tại trường.

Cũng xin gửi lời cám ơn đến Viễn Thông Tây Ninh đã tạo điều kiện để em hoàn thành đề tài luận văn này. Đặc biệt em xin gửi lời cám ơn đến anh Nguyễn Văn Đời, Phó giám đốc Trung Tâm Công Nghệ Thông Tin – Viễn Thông Tây Ninh, cám ơn anh đã hỗ trợ và tạo điều kiện để em thực hiện tốt đề tài.

Cuối cùng em xin gửi lời cám ơn đến Cha Mẹ, gia đình, người thân, bạn bè và đồng nghiệp đã quan tâm, ủng hộ trong suốt quá trình học tập cao học.

Tp. Hồ Chí Minh, ngày 25 tháng 01 năm 2022

Học viên thực hiện luận văn

Nguyễn Xuân Sang

MỤC LỤC

DANH MỤC KÝ HIỆU, CHỮ VIẾT TẮT

Acc	Accurary
FP	False Positive
FN	False Negative
LR	Logistic Regression
MA	Moving Average
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
NB	Naive Bayes
RMSE	Root Mean Square Error
RF	Random Forest
SVM	Support Vector Machine
TP	True Positive
TN	True Negative

DANH SÁCH BẢNG

DANH SÁCH HÌNH VẼ

MỞ ĐẦU

Dịch vụ thông tin di động ngày càng phát triển mạnh mẽ, trở thành một phần tất yếu trong cuộc sống của mỗi người dân Việt Nam. Quản lý khách hàng ngày càng nhận được sự quan tâm vì việc giữ chân khách hàng hiện tại mang lại lợi nhuận và quan trọng đối với các công ty viễn thông. Chi phí để tìm khách hàng mới lớn hơn nhiều so với chi phí để giữ chân khách hàng hiện tại trong kinh doanh, đặc biệt là trong thị trường viễn thông bão hòa. Hơn nữa, khách hàng dài hạn ít biến động hơn trong thị trường cạnh tranh, ví dụ: những khách hàng lâu năm ít có xu hướng chuyển sang công ty khác vì được khuyến mãi và góp nhiều lợi nhuận hơn cho công ty hiện tại.

Vì những nhu cầu đặt ra, các công ty viễn thông đang rất chú trọng và đầu tư nhiều hơn vào việc phát triển một mô hình dự báo khách hàng rời mạng. Nhiều phương pháp tiếp cận máy học đã được các nhà nghiên cứu đề xuất để dự báo khách hàng rời mạng, đặc biệt là trong lĩnh vực kinh doanh viễn thông. Các phương pháp tiếp cận máy học như vậy bao gồm các phương pháp phân lớp truyền thống như thuật toán Random Forest (RF), Naive Bayes (NB), Logistic Regression (LR) và Support Vector Machine (SVM) [2],[3],[4].

Tuy nhiên, với từng mô hình dự báo đều có những hạn chế riêng, ví dụ NB cần lượng dữ liệu lớn để đạt độ chính xác cao hay SVM có thời gian thực thi cao và độ phức tạp lớn [5]. Để có thể giải quyết những hạn chế đó, trong những năm gần đây nhiều nhà khoa học cũng bắt đầu nghiên cứu các phương pháp khai phá dữ liệu dựa trên sự kết hợp của hai hay nhiều phương pháp khai phá dữ liệu đã có. Sự kết hợp này bước đầu đã mang lại những kết quả tích cực khi các phương pháp khai phá dữ liệu kết hợp đã phát huy được phần nào những ưu điểm cũng như khắc phục được một số hạn chế của từng phương pháp khai phá dữ liệu đơn lẻ.

Nhằm mục đích tìm hiểu về hướng tiếp cận mới này trong lĩnh vực khai phá dữ liệu, cũng như khả năng ứng dụng vào thực tế, luận văn xin trình bày về phương pháp dự báo dữ liệu khách hàng rời mạng kết hợp giữa mô hình Logistic Regression

(LR) và Support Vector Machine (SVM), cùng ứng dụng mô hình kết hợp này vào dự báo khách hàng rời mạng tại Viễn Thông Tây Ninh.

Đối tượng nghiên cứu của đề tài tập trung vào các mô hình dự báo dữ liệu khách hàng rời mạng, đặc biệt là mô hình LR, thuật giải SVM và phương pháp kết hợp mô hình LR và SVM trong dự báo dữ liệu khách hàng rời mạng. Bên cạnh đó đề tài còn trình bày kết quả áp dụng các mô hình dự báo dữ liệu khách hàng rời mạng vào trong thực tế dựa trên bộ dữ liệu được thu thập tại Viễn Thông Tây Ninh.

Phạm vi nghiên cứu của đề tài giới hạn trong việc tìm hiểu và ứng dụng các mô hình dự báo dữ liệu khách hàng rời mạng, mô hình LR, thuật giải SVM và mô hình kết hợp LR và SVM.

Tuy phạm vi nghiên cứu của đề tài giới hạn trong việc tìm hiểu và ứng dụng các mô hình dự báo dữ liệu khách hàng rời mạng nhưng đề tài cũng đã mang lại một số ý nghĩa về khoa học và thực tiễn. Về khoa học, kết quả thực nghiệm của đề tài cũng cố thêm tính đúng đắn của hướng tiếp cận kết hợp các mô hình dự báo dữ liệu khách hàng rời mạng nói chung và mô hình dự báo khách hàng rời mạng kết hợp LR và SVM nói riêng. Về thực tiễn, kết quả dự báo của mô hình kết hợp LR và SVM giúp ích cho Viễn Thông Tây Ninh dự báo được khách hàng rời mạng để có thể lên kế hoạch tiếp cận và khuyến mãi hợp lý nhằm giữ chân khách hàng.

Luận văn được trình bày thành 4 chương:

Chương 1. Tổng quan: Giới thiệu về khách hàng rời mạng và dự báo khách hàng rời mạng. Trình bày về tình hình nghiên cứu trong và ngoài nước, xác định những vấn đề còn tồn tại trong các mô hình dự khách hàng rời mạng. Xác định mục tiêu, nội dung và phương pháp nghiên cứu của đề tài.

Chương 2: Mô hình kết hợp Logistic Regression và Support Vector Machine: Giới thiệu về mô hình kết hợp Logistic Regression và Support Vector Machine trong dự báo khách hàng rời mạng.

Chương 3: Dự báo tại Viễn Thông Tây Ninh: Giới thiệu về vấn để cần dự báo và ứng dụng mô hình kết hợp Logistic Regression và Support Vector Machine vào dự báo tại Viễn Thông Tây Ninh.

Chương 4: Kết luận và khuyến nghị: Đánh giá về các kết quả đạt được và hướng phát triển tiếp theo của đề tài.

Chương 1. TỔNG QUAN

Trong chương này sẽ trình bày các khái niệm về khách hàng rời mạng, tổng quan về các phương pháp dự báo khách hàng rời mạng. Ngoài ra chương này còn trình bày về những khó khăn, thách thức còn tồn tại trong các mô hình dự báo khách hàng rời mạng.

1.1 Khách hàng rời mạng và dự báo khách hàng rời mạng

1.1.1 Khách hàng rời mạng

Trong ngành viễn thông di động, thuật ngữ khách hàng rời mạng (churn customer), còn được gọi là khách hàng tiêu hao hoặc xáo trộn thuê bao, dùng để chỉ hiện tượng mất khách hàng. Quá trình di chuyển từ nhà cung cấp dịch vụ viễn thông này sang nhà cung cấp khác thường xảy ra do giá hoặc dịch vụ tốt hơn, hoặc do các lợi ích khác nhau mà công ty đối thủ cạnh tranh cung cấp.

Để thu hút thuê bao mới, các mạng di động phải thi nhau khuyến mại liên tục các tháng trong năm. Tuy nhiên, sau khi kết thúc mỗi đợt khuyến mại, số lượng thuê bao sử dụng hết tài khoản ngay lập tức rời mạng, tạm ngưng hoặc chuyển sang mạng khác lại tăng lên đáng kể, số thuê bao rời mạng nhiều hơn số thuê bao hòa mạng mới. Số lượng thuê bao đang hoạt động tăng giảm bất thường, doanh thu không tăng theo tốc độ phát triển của số lượng thuê bao. Đây là kiểu cạnh tranh đang đi ngược lại với xu thế hội nhập của ngành thông tin di động Việt Nam. Ở góc độ quản lý vĩ mô, thực trạng trên cho thấy tiêu cực thị trường và gây lãng phí nguồn lực của ngành.

Tỷ phú Jeff Bezos từng nói: “Chúng tôi coi khách hàng của mình là khách của một bữa tiệc, và chúng tôi là chủ nhà. Công việc của chúng tôi hàng ngày là làm cho mọi khía cạnh trải nghiệm khách hàng trở nên tốt hơn một chút “. Cải thiện tỷ lệ giữ chân khách hàng là một quá trình liên tục và hiểu được tỷ lệ khách hàng rời mạng là bước đầu tiên đúng hướng.

1.1.2 Dự báo khách hàng rời mạng

Trong một thị trường gần như bão hòa, các công ty đang sử dụng chiến lược tiếp thị để giữ khách hàng hiện tại. Để đạt được điều này, cần một phương pháp có thể xác định những khách hàng có nhiều khả năng bỏ đi nhất để có thể triển khai các chiến dịch giữ chân một cách chủ động. Để tối đa hóa hiệu quả và giảm chi phí cao liên quan đến các chiến dịch giữ chân này, dự đoán khách hàng rời mạng phải cực kỳ chính xác, để đảm bảo rằng các khuyến mãi chỉ đạt được những khách hàng có nhiều khả năng đổi nhà cung cấp dịch vụ của nhất.

Trong dự báo khách hàng rời mạng, những giá trị trong quá khứ được thu thập và phân tích để tìm ra các mô hình phù hợp. Giá trị tương lai của khách hàng rời mạng được dự báo từ các mô hình đó. Do đó, dữ liệu trong quá khứ ảnh hưởng rất lớn đến quá trình xây dựng mô hình và cải thiện kết quả dự báo của mô hình.

1.2 Tình hình dự báo khách hàng rời mạng

Chính vì có nhiều ý nghĩa quan trọng nên từ lâu đã có nhiều nhà khoa học tìm hiểu, nghiên cứu và mô hình hóa khách hàng rời mạng để ứng dụng trong phân tích, dự báo. Trong những năm gần đây nhiều mô hình, phương pháp được đề xuất để cải thiện kết quả, tăng độ chính xác cho dự báo dữ liệu khách hàng rời mạng nhưng nhìn chung các mô hình, phương pháp dự báo dữ liệu khách hàng rời mạng tập trung vào các hướng nghiên cứu chính là:

- Các mô hình dự báo dựa trên mô hình xác suất, thống kê như mô hình hồi quy logistic (Logistic Regression) [9].
- Các mô hình máy học (Machine Learning) như mô hình Random Forest [6], thuật giải SVM (Support Vector Machine)[7].
- Một hướng nghiên cứu khác có nền tảng dựa trên lý thuyết logic mờ, là phương pháp dự khách hàng rời mạng Neuro – Fuzzy [8].
- Hướng nghiên cứu kết hợp các mô hình dự báo khách hàng rời mạng. Tiêu biểu là mô hình kết hợp dự báo dữ liệu tuyến tính và phi tuyến Bayesian Model Averaging (BMA) và Frequentist Model Averaging [10].

1.3 Những vấn đề còn tồn tại

Mỗi một mô hình, phương pháp dự báo khách hàng rời mạng đều chỉ có thể phù hợp với một số dạng dữ liệu đặc thù, mà chưa có một mô hình nào có thể dự báo tốt được cho tất cả các dạng dữ liệu, ví dụ như những mô hình dựa trên xác xuất thống kê như mô hình hồi quy Logistic Regression chỉ phù hợp để dự báo cho các dữ liệu dạng tuyến tính (linear), còn các mô hình máy học như SVM lại chỉ phù hợp để dự báo cho các dạng dữ liệu phi tuyến tính [11]. Mặt khác, dữ liệu trong thực tế đa số đều tính tuyến tính và phi tuyến tính, nên việc chỉ sử dụng một mô hình, phương pháp để dự báo dữ liệu khách hàng rời mạng thường chưa mang lại kết quả như mong đợi. Do đó việc tìm hiểu và áp dụng kết hợp các mô hình, phương pháp dự báo dữ liệu khách hàng rời mạng vào trong thực tế là cần thiết để tăng độ chính xác của kết quả dự báo.

Thứ hai, với tình hình thị trường viễn thông hiện nay, dữ liệu về khách hàng viễn thông rất lớn. Vấn đề đặt ra cần xây dựng một mô hình tối ưu về thời gian để có thể đáp ứng ngay lập tức nhu cầu tốc độ dự báo của viễn thông hiện nay.

1.4 Mục tiêu, nội dung, phương pháp nghiên cứu

Mục tiêu của đề tài nhằm tìm hiểu và áp dụng kết hợp mô hình Logistic Regression và SVM song song trong dự báo dữ liệu khách hàng rời mạng. Ứng dụng mô hình này vào dự báo số khách hàng sử dụng dịch vụ viễn thông của Viễn Thông Tây Ninh. Lý do đề tài lựa chọn mô hình Logistic Regression và phương pháp SVM song sing để kết hợp dự báo vì:

- Mô hình LR và phương pháp SVM trong ước lượng hồi quy đều là những mô hình, phương pháp dự báo khách hàng rời mạng cho kết quả dự báo tương đối tốt. Tùy thuộc vào đặc tính của dữ liệu khách hàng rời mạng mà mô hình LR và phương pháp SVM thường được lựa chọn để thực hiện dự báo. Mô hình LR được chọn để dự báo cho thành phần tuyến tính của dữ liệu khách hàng rời mạng, còn phương pháp SVM thường được chọn để dự báo cho thành phần phi tuyến tính của dữ liệu khách hàng rời mạng. Do đó mà mô hình kết hợp LR và SVM trong dự báo dữ liệu khách hàng rời mạng hy vọng sẽ phát huy được các ưu điểm

của mô hình LR cũng như phương pháp SVM để cho kết quả dự báo chính xác hơn là sử dụng một mô hình, phương pháp dự báo đơn lẻ.

- Thực tế đã có những nghiên cứu và ứng dụng cho thấy hiệu quả của phương pháp kết hợp LR và SVM trong dự báo như Ứng dụng mô hình kết hợp LR và SVM trong dự báo tín dụng [12]. Mô hình kết hợp LR và SVM trong dự báo các chứng bệnh tim mạch trong y tế [13]. Tất cả các nghiên cứu và ứng dụng trên đều cho thấy kết quả dự báo của mô hình kết hợp LR và SVM hiệu quả hơn so với các mô hình, phương pháp dự báo đơn lẻ.
- Tuy nhiên với hạn chế về độ phức tạp và thời gian của SVM, mô hình sẽ rất tốn tài nguyên khi sử dụng SVM truyền thống. Chính vì vậy việc cài đặt sẽ sử dụng SVM song song thay thế cho SVM truyền thống. SVM song song sử dụng các GPUs nhằm tăng tốc độ tính toán nhưng vẫn đạt được độ chính xác tương đương với SVM truyền thống [14].
- Mô hình LR và phương pháp SVM đều là những mô hình, phương pháp dự báo dữ liệu khách hàng rời mạng hiệu quả và đã được nghiên cứu từ lâu. Do đó mà các thư viện hỗ trợ cài đặt các mô hình, phương pháp này trong các ngôn ngữ lập trình nói chung và ngôn ngữ R nói riêng là tương đối đầy đủ. Chính vì vậy mà việc cài đặt và thử nghiệm mô hình kết hợp LR và phương pháp SVM là tương đối thuận lợi và nhanh chóng. Bên cạnh đó các tài liệu nghiên cứu về mô hình LR và phương pháp SVM cũng rất đa dạng và phong phú.

Nội dung nghiên cứu của đề tài bao gồm:

- Tìm hiểu các mô hình dự báo dữ liệu khách hàng rời mạng, tập trung tìm hiểu về mô hình LR, mô hình SVM và mô hình kết hợp LR với SVM.
- Tiền xử lý dữ liệu để biến đổi dữ liệu về dạng phù hợp với các mô hình dự báo.
- Tiến hành cài đặt và thử nghiệm các mô hình dự báo dựa trên tập dữ liệu được thu thập từ dữ liệu của Viễn Thông Tây Ninh.

- So sánh, đánh giá kết quả dự báo của các mô hình với nhau và với dữ liệu thực tế.

Phương pháp nghiên cứu của đề tài:

- Tìm hiểu các mô hình, phương pháp trong dự báo khách hàng rời mạng.
- Tìm hiểu mô hình LR.
- Tìm hiểu về SVM và SVM song song.
- Tìm hiểu phương pháp kết hợp mô hình LR và SVM để tăng độ chính xác kết quả dự báo.
- Cài đặt thử nghiệm các mô hình, phương pháp dự báo dữ liệu khách hàng rời mạng.

Chương 2. MÔ HÌNH KẾT HỢP LOGISTIC REGRESSION VÀ SUPPORT VECTOR MACHINE

Cả mô hình Logistic Regression và thuật giải Support Vector Machine (SVM) đều là những mô hình, phương pháp nỗi bật trong lĩnh vực dự báo. Mỗi mô hình đều mang những đặc điểm riêng biệt phù hợp với từng loại hình dữ liệu khác nhau. Trong chương này sẽ trình bày chi tiết về hai mô hình dự báo dữ liệu là LR và SVM, giới thiệu về SVM song song, cũng như mô hình kết hợp LR và SVM.

2.1 Mô hình Logistic Regression

Mô hình LR là một mô hình được sử dụng nhiều trong số các mô hình dự báo dữ liệu khách hàng rời mạng. Trong mục này sẽ trình bày về mô hình LR và giới thiệu mô hình LR.

2.1.1 Giới thiệu

Trong thống kê, mô hình logistic (hay mô hình logit) được sử dụng để lập mô hình xác suất của một lớp hoặc sự kiện nhất định đang tồn tại như đạt / không đạt, thắng / thua, sống / chết hoặc khỏe mạnh / bệnh. Điều này có thể được mở rộng để mô hình hóa một số lớp sự kiện như xác định xem một hình ảnh có chứa mèo, chó, sư tử, v.v. Mỗi đối tượng được phát hiện trong hình ảnh sẽ được gán một xác suất từ 0 đến 1, với tổng là 1.

Logistic Regression là một mô hình thống kê ở dạng cơ bản sử dụng một hàm logistic để mô hình hóa một biến phụ thuộc nhị phân. Trong hồi quy logistic, ước lượng các tham số của mô hình logistic. Về mặt toán học, mô hình logistic nhị phân có một biến phụ thuộc với hai giá trị có thể có, chẳng hạn như đạt / không đạt được biểu thị bằng một biến chỉ báo, trong đó hai giá trị được gắn nhãn "0" và "1". Xác suất tương ứng của giá trị được gắn nhãn "1" có thể thay đổi giữa 0 (chắc chắn là giá trị "0") và 1 (chắc chắn là giá trị "1"), do đó việc ghi nhãn; hàm chuyển đổi tỷ lệ thành xác suất là hàm logistic. Đặc điểm xác định của mô hình logistic là việc tăng một trong các biến độc lập nhân lên

tỷ lệ của kết quả đã cho với tỷ lệ không đổi, với mỗi biến độc lập có tham số riêng; đối với một biến phụ thuộc nhị phân, điều này tổng quát tỷ lệ chênh lệch.

2.1.2 Mô hình Logistic

Xét một mô hình logistic với các tham số cho trước, sau đó xem cách các hệ số có thể được ước tính từ dữ liệu. Hãy xem xét một mô hình có hai yếu tố dự báo: x_1 và x_2 và một biến nhị phân Bernoulli Y với tham số p = P(Y = 1). Ta giả định mối quan hệ tuyến tính giữa các biến dự báo và tỷ lệ logit là Y = 1.

Mối quan hệ tuyến tính này có thể được viết ở dạng toán học như sau. Trong đó ℓ là tỷ lệ logit, b là cơ số logarit và β_i là các tham số của mô hình. Ta có:

$$\ell = log_b \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

Ta có thể khôi phục tỷ lệ logit bằng cách lũy thừa cả hai vế trên:

$$\frac{p}{1-p} = b^{\beta_0+\beta_1 x_1+\beta_2 x_2}$$

Chuyển vế p để ta có xác suất Y = 1:

$$p = \frac{b^{\beta_0+\beta_1 x_1+\beta_2 x_2}}{b^{\beta_0+\beta_1 x_1+\beta_2 x_2}+1} = \frac{1}{1+b^{-(\beta_0+\beta_1 x_1+\beta_2 x_2)}} = S_b(\beta_0 + \beta_1 x_1 + \beta_2 x_2)$$

trong đó đẳng thức thứ hai theo sau bằng cách chia tử số và mẫu số của phân số cho $b^{\beta_0+\beta_1 x_1+\beta_2 x_2}$ và trong đó $S_b$là hàm Sigmoid với cơ số b.

Công thức trên cho thấy rằng một khi β_i cố định, chúng ta có thể dễ dàng tính toán tỷ lệ logit Y = 1 cho một quan sát nhất định hoặc xác suất Y = 1 cho một quan sát nhất định. Trường hợp sử dụng chính của mô hình logistic là đưa ra một quan sát x và ước tính xác suất p mà Y = 1. Trong hầu hết các ứng dụng, cơ số b của logarit thường được coi là e. Tuy nhiên, trong một số trường hợp, kết quả có thể dễ dàng hơn bằng sử dụng cơ số 2 hoặc cơ số 10.

2.1.3 Hàm Sigmoid

Hàm sigmoid là một hàm toán học có đường cong hình chữ "S" hoặc đường cong sigmoid đặc trưng.

Một ví dụ phổ biến về hàm sigmoid là hàm logistic được hiển thị trong hình đầu tiên và được xác định bởi công thức:

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} = 1 - S(-x)$$

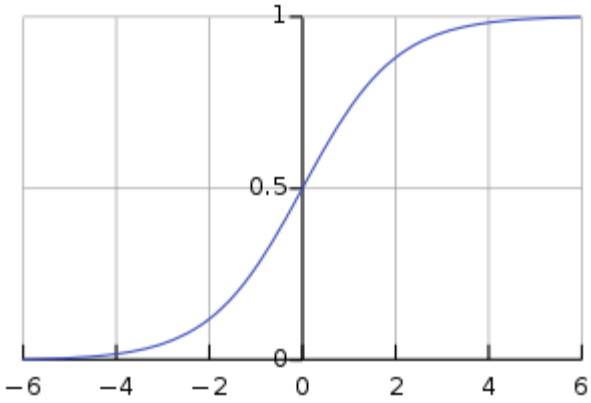
Hàm sigmoid là một hàm có giới hạn, có thể phân biệt, thực được xác định cho tất cả các giá trị đầu vào thực và có đạo hàm không âm tại mỗi điểm và chính xác một điểm uốn. Một "hàm" sigmoid và một "đường cong" sigmoid đề cập đến cùng một đối tượng. Một hàm sigmoid là đơn điệu, và có đạo hàm cấp một là hình chuông. Ngược lại, tích phân của bất kỳ hàm liên tục, không âm, hình chuông nào (với một cực đại cục bộ và không có cực tiểu cục bộ, trừ khi suy biến) sẽ là dấu hiệu. Do đó, các hàm phân phối tích lũy cho nhiều phân phối xác suất chung là sigmoidal. Một ví dụ như vậy là hàm lỗi, có liên quan đến hàm phân phối tích lũy của phân phối chuẩn; một hàm khác là hàm arctan, có liên quan đến hàm phân phối tích lũy của phân phối Cauchy.

2.1.4 Hàm mất mát và phương pháp tối ưu

Hàm logistic là một hàm sigmoid, nhận bất kỳ đầu vào thực tế nào và xuất ra giá trị từ 0 đến 1. Đối với logit, điều này được hiểu là lấy tỷ lệ logit đầu vào và có xác suất đầu ra. Hàm logit tiêu chuẩn: $\sigma: \mathbb{R} \rightarrow (0,1)$ được định nghĩa như sau:

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

Đồ thị của hàm logistic trên khoảng t (−6,6) được thể hiện trong Hình 2.1.



Hình 2.1: Đồ thị hàm logistic trong khoảng t(-6,6)

Giả sử t là một hàm tuyến tính của một biến giải thích duy nhất x (trường hợp t là một tổ hợp tuyến tính của nhiều biến giải thích được xử lý tương tự). Sau đó, ta có thể biểu diễn t như sau:

$$t = \beta_0 + \beta_1 x$$

Hàm logit tiêu chuẩn: σ: $\mathbb{R} \rightarrow (0,1)$ được viết lại như sau:

$$p(x) = \sigma(t) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

Trong mô hình logistic, p(x) được hiểu là xác suất của biến phụ thuộc Y bằng một trường hợp thành công chứ không phải là một trường hợp thất. Rõ ràng là các biến phản hồi Y không được phân phối giống nhau.

Với mô hình logistic, ta có thể giả sử rằng xác suất để một điểm dữ liệu x rơi vào lớp 1 là $f(w^T x)$ và rơi vào lớp 0 là $1 - f(w^T x)$. Với mô hình được giả sử như vậy, với các điểm dữ liệu training (đã biết đầu ra y), ta có thể viết như sau:

$$P(y_i = 1|x_i; w) = f(w^T x_i) \quad (1) \; P(y_i = 0|x_i; w) = 1 - f(w^T x_i)$$

trong đó $P(yi = 1|xi; W)$ được hiểu là xác suất xảy ra sự kiện đầu ra $y_i = 1$ khi biết tham số mô hình w và dữ liệu đầu vào x_i. Mục đích của chúng ta là tìm các hệ số w sao cho là $f(w^T x_i)$ càng gần với 1 càng tốt với các điểm dữ liệu thuộc lớp 1 và càng gần với 0 càng tốt với những điểm thuộc lớp 0.

Ký hiệu $z_i = f(w^T x_i)$ và viết gộp lại hai biểu thức bên trên ta có:

$$P(\text{yi} = 1|\text{xi}; \text{W}) = z_i^{y_i}(1 - z_i)^{1-y_i}$$

Ta cần mô hình gần với dữ liệu đã cho nhất, tức là xác suất này đạt giá trị cao nhất. Xét toàn bộ training set với $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{d \times N}$ và $y = [y_1, y_2, \dots, y_n]$ ta cần tìm w để biểu thức sau đây đạt giá trị lớn nhất:

$$P(y|X; w)$$

2.2 Support Vector Machine

Support Vector Machine (SVM) là một thuật giải quan trọng và được biết đến nhiều trong lĩnh vực máy học. SVM được ứng dụng rộng rãi trong rất nhiều lĩnh vực khác của khoa học máy tính như trong các bài toán về nhận diện hay trong các bài toán về phân lớp, gom cụm,…Trong mục này sẽ giới thiệu về thuật giải SVM và ứng dụng.

2.2.1 Giới thiệu

Trong không gian 2 chiều, ta biết rằng khoảng cách từ một điểm có toạ độ (x_0, y_0) tới đường thẳng có phương trình $w_1 x + w_2 y + b = 0$ được xác định bởi:

$$\frac{|w_1 x_0 + w_2 y_0 + b|}{\sqrt{w_1^2 + w_2^2}}$$

Trong không gian 3 chiều, khoảng cách từ một điểm có toạ độ (x_0, y_0, z_0) tới một mặt phẳng có phương trình $w_1 x + w_2 y + w_3 z + b = 0$ được xác định bởi:

$$\frac{|w_1 x_0 + w_2 y_0 + w_3 z_0 + b|}{\sqrt{w_1^2 + w_2^2 + w_3^2}}$$

Hơn nữa, nếu bỏ trị tuyệt đối ở tử số, có thể xác định được điểm đó nằm về phía nào của đường thẳng đang xét. Những điểm làm cho biểu thức trong trị tuyệt đối mang dấu dương nằm về cùng 1, những điểm làm cho biểu thức trong

dấu giá trị tuyệt đối mang dấu âm nằm về phía còn lạiNhững điểm nằm trên đường thẳng sẽ làm cho tử số có giá trị bằng 0, tức khoảng cách bằng 0.

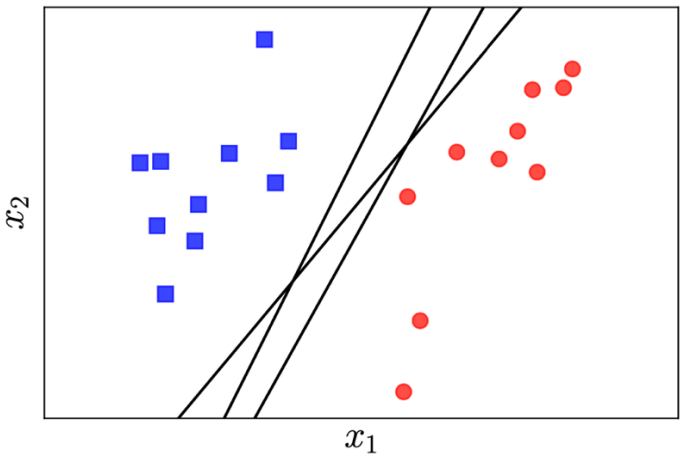
Việc này có thể được tổng quát lên không gian nhiều chiều: Khoảng cách từ một điểm (vector) có toạ độ x_0 tới siêu mặt phẳng (hyperplane) có phương trình $w^T x + b = 0$ được xác định bởi:

$$\frac{w^T x_0 + b}{||w||_2}$$

Với $||w||_2 = \sqrt{\sum_{i=1}^{d} w_i^2}$ với d là số chiều của không gian.

Giả sử rằng có hai lớp khác nhau được mô tả bởi các điểm trong không gian nhiều chiều, hai lớp này phân tách tuyến tính, tức tồn tại một siêu phẳng phân chia chính xác hai lớp đó. Hãy tìm một siêu mặt phẳng phân chia hai lớp đó, tức tất cả các điểm thuộc một lớp nằm về cùng một phía của siêu mặt phẳng đó và ngược phía với toàn bộ các điểm thuộc lớp còn lại. Thuật toán Perceptron Learning Algorithm (PLA) [15] có thể làm được việc này nhưng nó có thể cho chúng ta vô số nghiệm như Hình 2.2.

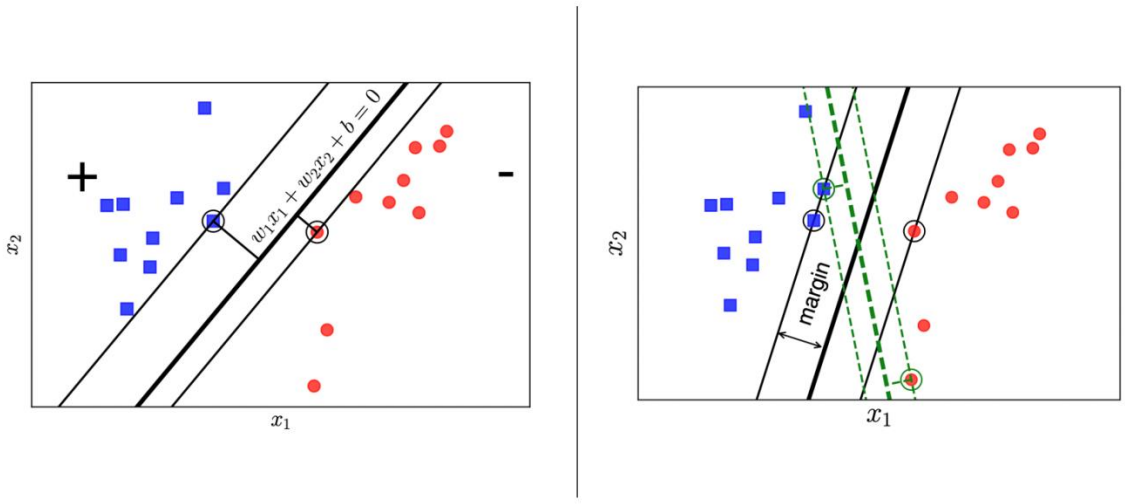
Vấn đề đặt ra là: trong vô số các mặt phân chia, đâu là mặt phân chia tốt nhất theo một tiêu chuẩn nào đó? Trong 3 đường thẳng minh họa trong Hình 2.8 phía trên, có hai đường thẳng khá lệch về phía lớp hình tròn đỏ. Điều này có thể khiến cho lớp màu đỏ không thõa mãn bị lấn nhiều quá. Liệu có cách nào để tìm được đường phân chia mà cả hai lớp đều cảm thõa mãn nhất hay không?



Hình 2.2: Các mặt phân cách hai lớp

2.2.2 Độ rộng của margin

Nếu ta định nghĩa độ thõa mãn của một lớp tỉ lệ thuận với khoảng cách gần nhất từ một điểm của lớp đó tới đường/mặt phân chia, thì ở Hình 2.2 trái, lớp tròn đỏ sẽ không thõa mãn vì đường phân chia gần nó hơn lớp vuông xanh rất nhiều. Chúng ta cần một đường phân chia sao cho khoảng cách từ điểm gần nhất của mỗi lớp (các điểm được khoanh tròn) tới đường phân chia là như nhau. Khoảng cách như nhau này được gọi là margin.



Hình 2.3: Margin của hai lớp

Xét tiếp Hình 2.2 bên phải khi khoảng cách từ đường phân chia tới các điểm gần nhất của mỗi lớp là như nhau. Xét hai cách phân chia bởi đường nét

liền màu đen và đường nét đứt màu lục, đường nào sẽ làm cho cả hai lớp thõa mãn. Rõ ràng đó phải là đường nét liền màu đen vì nó tạo ra một margin rộng hơn.

Việc margin rộng hơn sẽ mang lại hiệu quả phân lớp tốt hơn vì sự phân chia giữa hai lớp là rạch ròi hơn. Bài toán tối ưu trong SVM chính là bài toán đi tìm đường phân chia sao cho margin là lớn nhất.

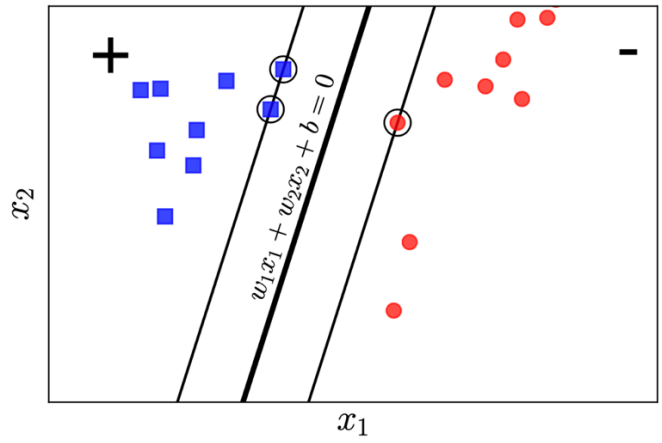
3.2.3 Tìm kiếm siêu phẳng tối ưu

Giả sử rằng các cặp dữ liệu của training set là:

$(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ với vector $x_i \in \mathbb{R}^d$ thể hiện đầu vào của một điểm dữ liệu và y_i là nhãn của điểm dữ liệu đó, d là số chiều của dữ liệu và N là số điểm dữ liệu. Giả sử rằng nhãn của mỗi điểm dữ liệu được xác định bởi $y_i = 1$ (lớp 1) hoặc $y_i = -1$ (lớp 2) giống như trong PLA.

Để dễ hình dung, chúng ta cùng xét trường hợp trong không gian hai chiều dưới đây. Không gian hai chiều để dễ hình dung, các phép toán hoàn toàn có thể được tổng quát lên không gian nhiều chiều.

Giả sử rằng các điểm vuông xanh thuộc lớp 1, các điểm tròn đỏ thuộc lớp -1 và mặt $w^T x + b = w_1 x_1 + w_2 x_2 + b = 0$ là mặt phân chia giữa hai lópes (Hình 2.3). Hơn nữa, lớp 1 nằm về phía dương, lớp -1 nằm về phía âm của mặt phân chia. Nếu ngược lại, ta chỉ cần đổi dấu của w và b. Chú ý rằng chúng ta cần đi tìm các hệ số w và b.



Hình 2.4: Phân tích bài toán tối ưu SVM

Ta có một điểm quan trọng sau đây: với cặp dữ liệu (x_n, y_n) bất kỳ, khoảng cách từ điểm đó tới mặt phân chia là:

$$\frac{y_n(w^T x_n + b)}{||w||_2}$$

Giả sử ở trên, y_n luôn cùng dấu với *phía* của x_n. Từ đó suy ra y_n cùng dấu với $(w^T x_n + b)$, và tử số luôn là 1 số không âm. Với mặt phần chia như trên, *margin* được tính là khoảng cách gần nhất từ 1 điểm tới mặt đó (bất kể điểm nào trong hai lớpes):

$$margin = \min_n \frac{y_n(w^T x_n + b)}{||w||_2}$$

Bài toán tối ưu trong SVM chính là bài toán tìm w và b sao cho margin này đạt giá trị lớn nhất:

$$\begin{aligned}(w, b) &= arg\max_{w,b}\left\{\min_n \frac{y_n(w^T x_n + b)}{||w||_2}\right\} \\ &= arg\max_{w,b}\left\{\frac{1}{||w||_2}\min_n y_n(w^T x_n + b)\right\}\end{aligned} \tag{2-8}$$

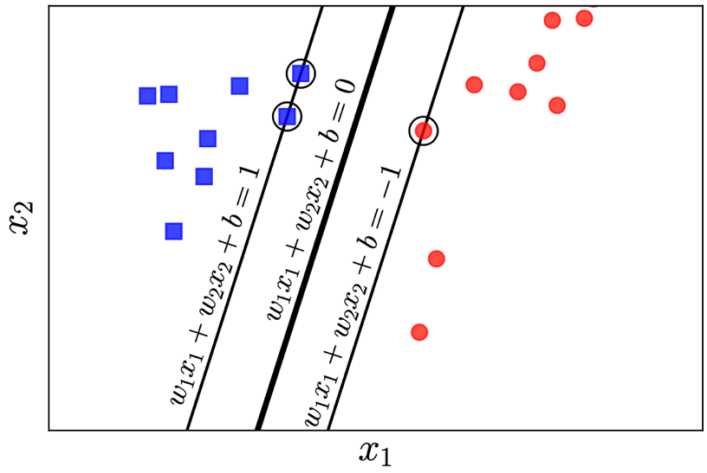
Việc trực tiếp giải bài toán này sẽ rất phức tạp, nhưng ta thấy có cách để đưa về bài toán đơn giản hơn.

Nhận xét quan trọng nhất:

Nếu ta thay vector hệ số w bởi kw và b bởi kb trong đó k là một hằng số dương thì mặt phân chia không thay đổi, tức khoảng cách từ từng điểm đến mặt phân chia không đổi, tức $margin$ không đổi. Dựa trên tính chất này, ta có thể giả sử:

$$y_n(w^T x_n + b) = 1$$

Với những điểm nằm gần mặt phân chia nhất như Hình 2.4 dưới đây:



Hình 2.5: Các điểm gần mặt phân cách nhất của hai lớp

Như vậy, với mọi n, ta có:

$$y_n(w^T x_n + b) \geq 1$$

Vậy bài toán tối ưu (2-8) có thể đưa về bài toán tối ưu có ràng buộc sau đây:

$$(w, b) = arg \max_{w,b} \frac{1}{||w||_2} \quad (2\text{-}9)$$

Với điều kiện: $y_n(w^T x_n + b) \geq 1, \forall n = 1,2, \dots, N$

Bằng 1 biến đổi đơn giản, ta có thể đưa bài toán này về bài toán dưới đây:

$$(w, b) = arg \min_{w,b} \frac{1}{2} ||w||_2^2 \quad (2\text{-}10)$$

Với điều kiện: $1 - y_n(w^T x_n + b) \leq 0, \forall n = 1,2, \dots, N$

Ở đây, ta lấy nghịch đảo hàm mục tiêu, bình phương nó để được một hàm khả vi, và nhân với 1/2 để biểu thức đạo hàm đẹp hơn.

Quan sát quan trọng**:** Trong bài toán (2-10), hàm mục tiêu là một hàm lồi (convex function) [16]. Các hàm bất đẳng thức ràng buộc là các hàm tuyến tính theo w và b, nên chúng cũng là các hàm lồi. Vậy bài toán tối ưu (2-10) có

hàm mục tiêu là lồi, và các hàm ràng buộc cũng là lồi, nên nó là một bài toán lồi. Tuy nhiên, việc giải bài toán này trở nên phức tạp khi số chiều d của không gian dữ liệu và số điểm dữ liệu N tăng lên cao.

Người ta thường giải bài toán đối ngẫu Lagrange của bài toán này. Đầu tiên, bài toán đối ngẫu có những tính chất thú vị khiến nó được giải hiệu quả. Thứ hai, trong quá trình xây dựng bài toán đối ngẫu, ta thấy rằng SVM có thể được áp dụng cho những bài toán mà dữ liệu phi tuyến, tức các đường phân chia không phải là một mặt phẳng mà có thể là các mặt có hình phức tạp.

Xác định lớp cho một điểm dữ liệu mới: Sau khi tìm được mặt phân cách $w^T x_n + b = 0$, lớp của bất kỳ một điểm nào sẽ được xác định đơn giản bằng cách:

$$class(x) = sgn(w^T x_n + b)$$

Trong đó hàm sgn là hàm xác định dấu, nhận giá trị 1 nếu đối số là không âm và -1 nếu ngược lại.

Trong thực tế, dữ liệu thường không phân chia tuyến tính như âm thanh bị nhiễu, … Nên ta cũng mong muốn rằng SVM có thể làm việc với dữ liệu gần tuyến tính như Logistic Regression.

2.2.4 Phương pháp Lagrange multipliers

Để tìm nghiệm theo công thức chúng ta sẽ dùng bài toán đối ngẫu Lagrange, công thức Lagrange được biểu diễn như sau:

$$\lambda = arg\ \max_\lambda g(\lambda), với: \lambda \geqslant 0\ và\ \sum_1^N \lambda_n y_n = 0 \text{ (2-21)}$$

Trong đó $g(\lambda) = -\frac{1}{2}\lambda^T K\lambda + 1^T\lambda, với\ K = V^T V$, K là ma trận nữa xác định dương, V là ma trận kết hợp của hai tập dữ liệu đầu vào.

Để giải bài toán tối ưu này ta sử dụng phương pháp Lagrange multipliers, hàm Lagrange được biểu diễn như sau [16]:

$$\mathcal{L}(x, y, \lambda) = f(x, y) - \lambda . g(x, y) \\ = 2 - x^2 - 2y^2 - \lambda(x + y - 1) \tag{3.32}$$

Phương trình đạo hàm riêng của hàm Lagrange

$$\nabla\mathcal{L}(x, y, \lambda) = \nabla f(x, y) - \lambda . \nabla g(x, y) = 0 \tag{3.33}$$

Các đạo hàm riêng của hàm Lagrange

$$\frac{\partial}{\partial x}\mathcal{L}(x, y, \lambda) = -2x - \lambda = 0 \tag{3.34.1}$$

$$\frac{\partial}{\partial y}\mathcal{L}(x, y, \lambda) = -4y - \lambda = 0 \tag{3.34.2}$$

$$\frac{\partial}{\partial \lambda}\mathcal{L}(x, y, \lambda) = x + y - 1 = 0 \tag{3.34.3}$$

Giải hệ phương trình (3.34.1), (3.34.2) và (3.34.3) ta tìm được $x = \frac{2}{3}; y = \frac{1}{3}; \lambda = -\frac{4}{3}$ và giá trị của $f\left(\frac{2}{3}; \frac{1}{3}\right) = \frac{4}{3}$. Đến đây ta chỉ biết được $f\left(\frac{2}{3}; \frac{1}{3}\right) = \frac{4}{3}$ là một cực trị của hàm *f*. Để kiểm tra xem $\frac{4}{3}$ có phải là giá trị lớn nhất của hàm *f* hay không, ta sẽ tính giá trị của hàm *f* tại một điểm bất kỳ thỏa hàm *g*, sau đó so sánh kết quả với cực trị mà ta tìm được. $g(0; 1) = 0; f(0; 1) = 0$. Vậy $f\left(\frac{2}{3}; \frac{1}{3}\right) = \frac{4}{3}$ là giá trị lớn nhất của hàm *f*.

Tuy nhiên dữ liệu trong thực tế thường bị tác động của nhiều yếu tố, dẫn đến dữ liệu thường xuyên bị nhiễu và không được phân lớp một cách tuyến tính. Vì vậy mà thuật giải SVM có thêm hai cải tiến là Soft margin và Kernel để thích nghi với các đặc tính của dữ liệu.

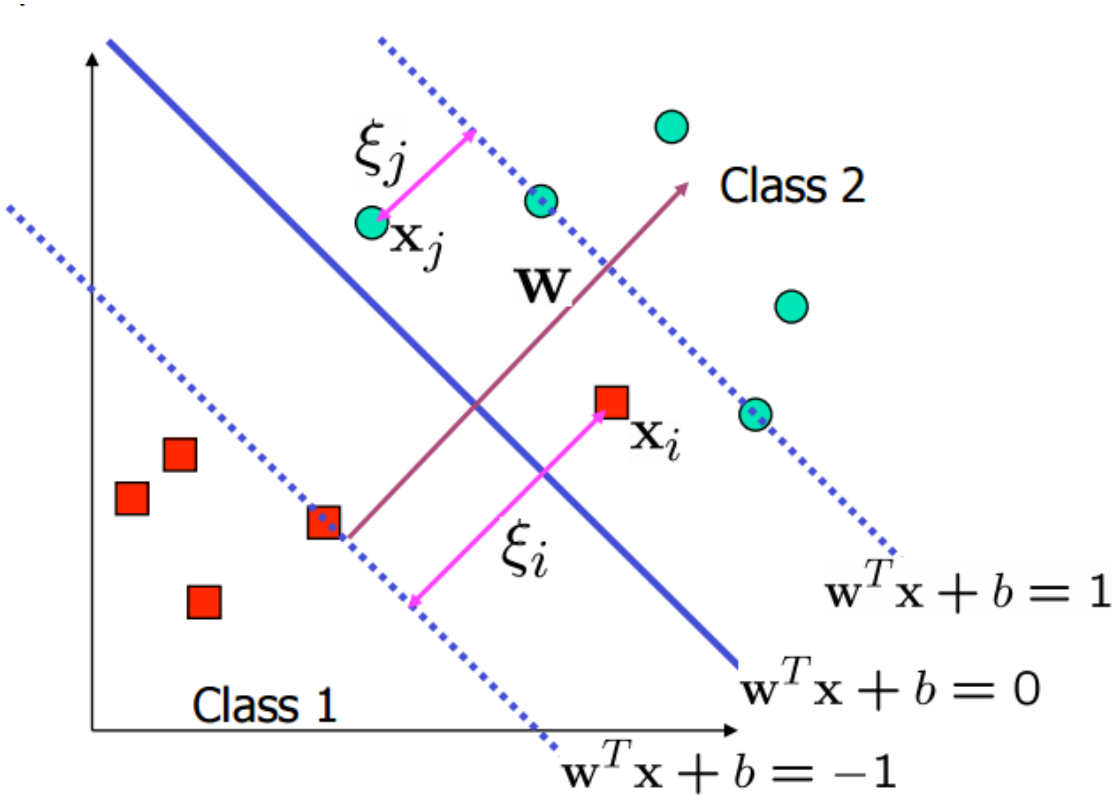
2.2.5 Soft Margin và Kernel

Soft Margin: Hình 2.6 là một ví dụ về trường hợp phân lớp dữ liệu trong đó có 2 điểm dữ liệu nhiễu là x_i và x_j. Trong trường hợp này nếu xem hai điểm dữ liệu nhiễu này là các điểm dữ liệu bình thường và áp dụng thuật giải SVM sẽ dẫn đến kết quả là không tìm được một siêu phẳng tối ưu nào để phân lớp dữ

liệu. Vì vậy mà thuật giải SVM được cải tiến cho trường hợp phân lớp dữ liệu bị nhiễu như sau:

$$\begin{aligned} &Minimize\ \frac{\|\vec{w}\|^2}{2} + C\sum_{i=1}^{n}\xi_i \\ &subject\ to\ y_i(\vec{w}.\vec{x_i} + b) \geq 1 - \xi_i\ and\ \xi_i \geq 0 \\ &(\forall i\ 1 \leq i \leq n) \end{aligned} \tag{3.42}$$

Trong đó C là một hằng số được dùng để tinh chỉnh vấn đề overfitting. Bài toán tối ưu này được giải theo cách tương tự như bài toán tối ưu (3.29). Margin



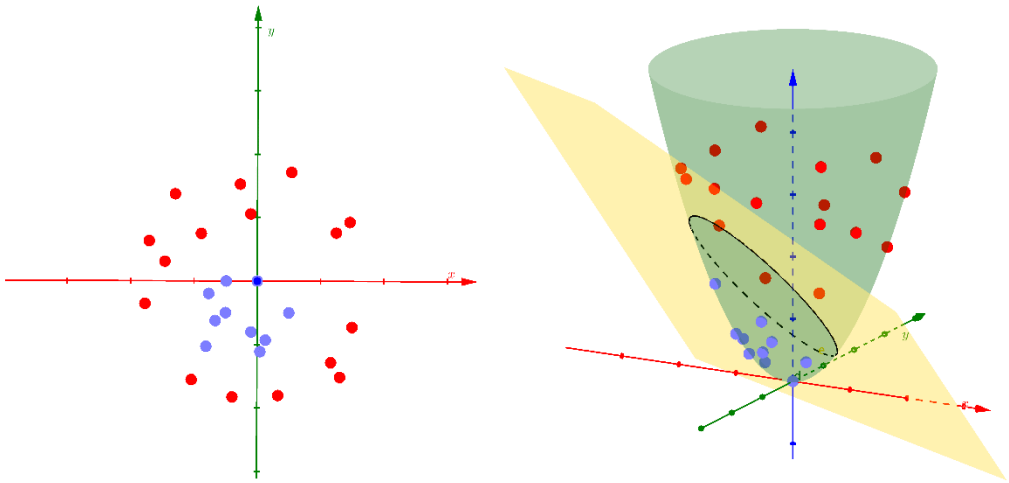
Hình 2.6: Ví dụ về Soft Margin

trong trường hợp này được gọi là Soft Margin.

Kernel: Trong thực tế có rất nhiều dữ liệu không tuyến tính, dữ liệu có thể không được biểu diễn trong không gian vector. Trong khi đó, hàm phân lớp tuyến tính thì đơn giản và thuận lợi hơn nhiều. Điều này đã đặt ra yêu cầu phân lớp mở rộng cho phi tuyến.

Ý tưởng cơ bản của Kernel SVM [9] và các phương pháp kernel nói chung là tìm một phép biến đổi nào đó làm cho dữ liệu ban đầu phi tuyến tính được biến sang không gian mới. Ở không gian mới, dữ liệu trở nên tuyến tính.

Xét ví dụ dưới đây với việc biến dữ liệu phi tuyến tính trong không gian hai chiều thành tuyến tính trong không gian ba chiều bằng cách giới thiệu thêm một chiều mới.



Hình 2.7: Ví dụ về Kernel trong SVM

Không gian đầu vào ban đầu có thể được ánh xạ tới một số không gian vector nào đó, gọi là không gian đặc trưng có nhiều chiều hơn mà tập huấn luyện có thể tách rời được, bằng cách sử dụng hàm f như sau:

$$f(x) =< w, \theta(x) > +b \tag{2-12}$$

Trong đó $f(x)$ là tuyến tính trong không gian đặc trưng. Φ là một hàm phi tuyến được định nghĩa bởi ánh xạ $\Phi: x \to \varphi(x)$. Với mỗi ánh xạ này, xem xét tất cả các tích của các cặp $< \theta(x), \theta(x') >$. Kết quả là một bộ phân lớp có dạng hàm phân tách bậc hai có dạng:

$$f(x) = \sum_{1}^{n} y_i a_i < \theta(x_i), \theta(x) > +b \tag{2-13}$$

Với $w = \sum_1^n y_i a_i x_i$, nghĩa là các vector trọng số của một mặt phẳng phân tách với biên độ lớn có thể được biểu diễn như một tổ hợp tuyến tính của các điểm huấn luyện.

Hàm kernel được định nghĩa như là một chức năng tương ứng với một đối tượng điểm của hai vector đặc trưng trong một số không gian đặc trưng mở rộng. Như vậy hàm kernel $k(x, x')$ có dạng là:

$$k(x, x') =< \theta(x), \theta(x') > \quad (2\text{-}14)$$

Kernel cho các dữ liệu thực

Dữ liệu thực là dữ liệu mà các mẫu là các vector có số chiều xác định.

Có 4 loại Kernel phổ biến:

- Linear kernel:

$$K(x_i, x_j) = x_i^T x_j \quad (2\text{-}15)$$

- Polynomial kernel:

$$K(x_i, x_j) = (1 + x_i^T x_j)^p \quad (2\text{-}16)$$

- Gaussian (Radial-Basis Function (RBF)) kernel

$$K(x_i, x_j) = \exp(-\frac{||x_i - x_j||^2}{2\sigma^2}) \quad (2\text{-}17)$$

- Sigmoid:

$$k(x, x') = \tanh(\beta_0 x_i^T x_j + \beta_1) \quad (2\text{-}18)$$

Trong đó, đa thức kernel và Gaussian kernel được sử dụng phổ biến nhất.

a. Đa thức kernel

Bậc *d* của đa thức kernel được định nghĩa là:

$$k_{d,k}^{polynomial}(x, x') = ((< x, x' > +k)^d \quad (2\text{-}19)$$

k là thường được chọn là 0 (đồng nhất) hoặc 1 (không đồng nhất). Không gian đặc trưng cho các hàm kernel không đồng nhất bao gồm tất cả các đơn thức bậc nhỏ hơn d. Nhưng, thời gian tính toán của nó là tuyến tính với số chiều của không gian đầu vào. Kernel với d =1 và k = 0, biểu hiện bằng klinear, là kernel tuyến tính dẫn đến một hàm phân tách tuyến tính.

Bậc của kernel đa thức kiểm soát sự linh hoạt của bộ phân lớp (Hình 2.14). Đa thức bậc thấp nhất là kernel tuyến tính. Hàm kernel này không đủ tốt nếu không gian đặc trưng là phi tuyến. Đối với các dữ liệu trong Hình 2.14 ở đa thức bậc 2 đã đủ linh hoạt để phân biệt giữa hai lớp với một biên cong tốt hơn.

b. Gaussian kernel:

Gaussian kernel được định nghĩa là:

$$k_{\delta}^{Gaussian}(x, x') = exp(-\frac{1}{\delta}||x - x'||)^2 \quad (2\text{-}20)$$

Trong đó $\delta > 0$ là một tham số điều khiển độ rộng của Gaussian. Nó đóng một vai trò tương tự như bậc của kernel đa thức trong việc kiểm soát sự linh hoạt của bộ phân lớp (Hình 2.15). Gaussian kernel cơ bản là bằng 0 nếu khoảng cách bình phương $|x - x'|^2$ là lớn hơn nhiều so với δ.

2.2.6 SVM song song và bộ công cụ ThunderSVM

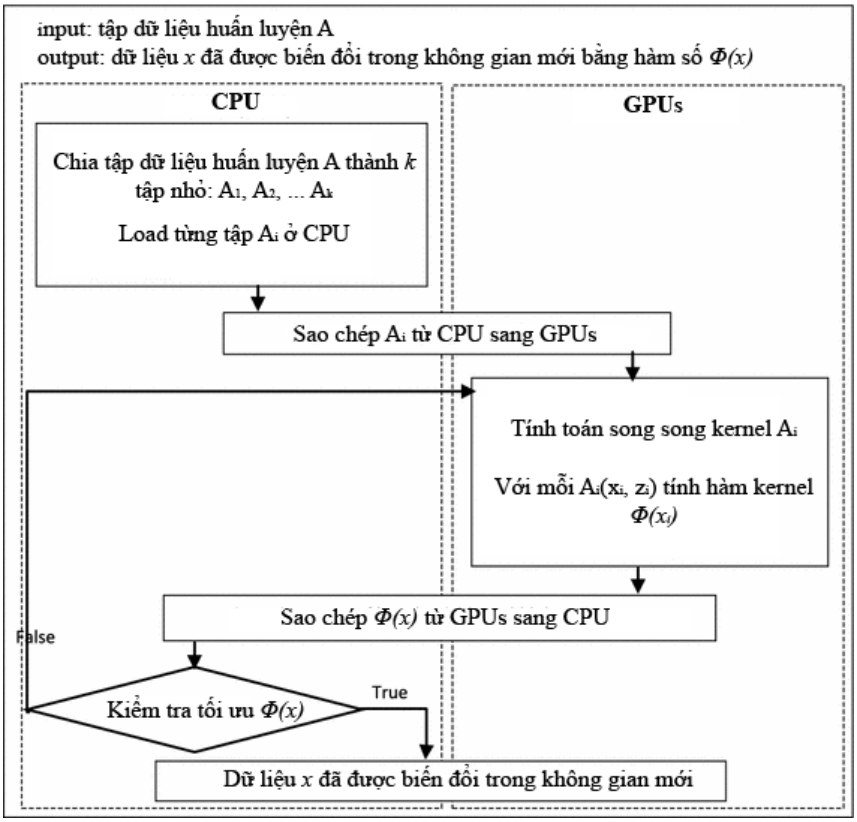
Mặc dù SVM dù được sử dụng rộng rãi trong nhiều ứng dụng, nhưng việc dự báo và huấn luyện SVM cho những bài toán với số chiều không gian dữ liệu lớn là rất phức tạp và chi phí vô cùng đắt đỏ. Bộ xử lý đồ họa GPUs được sử dụng để gia tốc cho nhiều giải pháp xử lý của các ứng dụng trong thế giới thực nhờ vào việc giải toán đa nhân và băng thông bộ nhớ lớn của GPUs.

Sử dụng lợi thế của GPUs, giới thiệu một bộ công cụ gọi là ThuderSVM [18] dùng để khai thác GPUs và CPUs đa nhân. Nhiệm vụ của bộ công cụ này là để giúp người dùng có thể dễ dàng ứng dụng SVMs một cách hiệu quả để

giải quyết các bài toán. Từ đó chỉ ra rằng có thể huấn luyện SVM nhanh hơn bằng cách sử dụng xấp xỉ kernel SVM và tìm một phép biến đổi sao cho dữ liệu ban đầu là không tuyến tính được biến sang không gian mới. Ở không gian mới này, dữ liệu trở nên tuyến tính. ThunderSVM hướng tới việc tìm kiếm một giải pháp chính xác. ThunderSVM hỗ trợ tất cả các chức năng mà công cụ LibSVM cung cấp như SVC, SVR và đơn lớp SVMs cùng với các tham số đầu vào như nhau, từ đó giúp cho người dùng đã quen với LibSVM hiện tại có thể dễ dàng chuyển đổi sang ThunderSVM.

Ngoài ra, ThunderSVM còn hỗ trợ nhiều API trên các nên tảng ngôn ngữ khác nhau như C/C++, Python, R, MATLAB, Java, … giúp cho lập trình viên có thể linh động trong việc áp dụng vào bài toán của mình. ThunderSVM có thể chạy trên các hệ điều hành như Linux, Windows hay Macintosh có hoặc không có GPUs. Các kết quả thực nghiệm cho thấy ThunderSVM nhìn chung chạy nhanh hơn gấp nhiều lần so với LibSVM trong tất cả các tác vụ.

2.2.7 Minh họa thuật toán SVM song song



Hình 2.8: Minh họa luồng tác vụ của SVM song song

Hình 2.8 mô tả luồng tác vụ của thuật toán SVM song song. Đầu vào input là tập dữ liệu cần huấn luyện A, đầu ra output của thuật toán chính là dữ liệu đã được biến đổi trong không gian mới bằng hàm số $\phi(x)$. Thuật toán tiến hành theo các bước sau:

- Bước 1: (Thực hiện ở GPU) Chia nhỏ tập dữ liệu A thành k tập nhỏ.
- Bước 2: Sao chép từng tập nhỏ sang GPUs
- Bước 3: Lặp từng A_i
    - (Thực hiện ở GPUs) Tính toán song song kernel A_i
    - Với mỗi A_i tính toán hàm kernel $\phi(x)$
    - Sao chép $\phi(x)$ từ GPUs sang CPU
    - Kiểm tra tối ưu $\phi(x)$. Nếu đã tối ưu kết thúc vòng lặp và qua bước 4, nếu chưa tối ưu quay lại tính toán ở bước 3 và sử dụng lại dữ liệu lưu trên bộ nhớ đệm của GPUs
- Bước 4: Trả về dữ liệu x đã được biến đổi ở không gian mới

2.3 Mô hình kết hợp Logistic Regression và Support Vector Machine

2.3.1 Giới thiệu

Như đã trình bày trong chương 1, dữ liệu khách hàng di động trong thực tế thường bị ảnh hưởng hoặc tác động bởi nhiều yếu tố khác nhau.

Thứ nhất là yếu tố phát sinh từ trong doanh nghiệp và công ty, yếu tố này yếu tố chính thường xuyên chi phối sự thay đổi của khách hàng. Ví dụ như giá cổ phiếu của một công ty, tình hình kinh doanh của công ty, các chính sách khuyến mãi, khiếu nại hoặc bảo vệ khách hàng... Các yếu tố này quyết định đặc tính tuyến tính của dữ liệu.

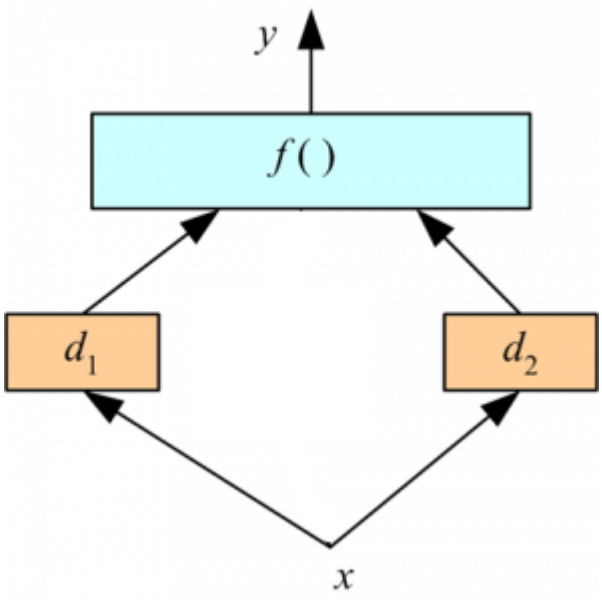
Thứ hai là các yếu tố ngoài doanh nghiệp, các yếu tố này thường ngẫu nhiên và bất ngờ đối với doanh nghiệp viễn thông. Ví dụ các thay đổi về địa điểm cư trú của khách hàng, nhu cầu sử dụng trong gia đình,... . Các yếu tố tnày quy định đặc tính phi tuyến tính của dữ liệu.

Chính vì dữ liệu luôn có những đặc tính tuyến tính và phi tuyến mà kết quả dự báo của các mô hình riêng biệt đôi khi không được như mong đợi. Lý do là bởi các mô hình dự báo riêng biệt thường chỉ phù hợp để dự báo cho một số thành phần của dữ liệu. Ví dụ mô hình LR chỉ phù hợp để dự báo cho thành phần tuyến tính, trong khi thành phần phi tuyến tính mô hình LR thường bỏ qua, không dự báo được. Ngược lại, các mô hình máy học như SVM hay mạng neural lại thích hợp để dự báo cho thành phần phi tuyến tính của khách hàng rời mạng hơn là thành phần tuyến tính. Vì vậy mà việc cần thiết là tìm cách kết hợp các mô hình dự báo riêng biệt này lại với nhau sao cho có thể phát huy các ưu điểm cũng như khắc phục được các nhược điểm của từng mô hình.

Mô hình kết hợp LR và SVM là một trong những mô hình tiếp cận theo hướng trên. Mô hình này sử dụng mô hình LR để dự báo cho thành phần tuyến tính của dữ liệu, đồng thời sử dụng phương pháp SVM để dự báo cho thành phần phi tuyến tính của dữ liệu. Sau đó kết quả dự báo của hai mô hình này sẽ được kết hợp lại với nhau để cho kết quả dự báo sau cùng.

2.3.2 Nội dung

Để dự báo dữ liệu khách hàng rời mạng sử dụng mô hình kết hợp LR và SVM. Ở đây, sử dụng phương pháp xếp chồng, chia dữ liệu thành hai phần tuyến tính và phi tuyến để huấn luyện. Trong việc xếp chồng nhiều lớp mô hình học máy được đặt chồng lên nhau trong đó mỗi mô hình chuyển các dự đoán của chúng đến mô hình ở lớp phía trên và mô hình lớp trên cùng sẽ đưa ra quyết định dựa trên kết quả đầu ra của các mô hình ở lớp bên dưới.



Hình 2.9: Ví dụ minh họa kết hợp LR và SVM

Dữ liệu x_t bao gồm hai thành phần tuyến tính L_t và phi tuyến N_t. Do đó dữ liệu có thể mô hình hóa thành: $x_t = L_t + N_t$.

Đầu tiên LR dùng để dự báo cho thành phần tuyến tính L_t của dữ liệu. Gọi d_t là kết quả dự báo của LR.

Thành phần còn lại e_t của dữ liệu sau khi trừ đi kết quả dự báo của LR được xác định:

$$e_t = x_t - d_1$$

Tiếp theo dự báo e_t chính là thành phần phi tuyến của dữ liệu bằng SVM. Gọi d_2 là kết quả dự báo của SVM

Sau cùng, kết quả dự báo của mô hình kết hợp chính là tổng hợp kết quả dự báo của LR và SVM:

$$y_t = d_1 + d_2$$

2.3.3 Một số kết quả tham khảo và đánh giá

Mô hình kết hợp Logistic Regression và Support Vector Machine đã được nghiên cứu và ứng dụng trong một số lĩnh vực như dự báo tín dụng, dự báo sức khỏe, ung thư,…Sau đây là kết quả dự báo của mô hình kết hợp này trong một số nghiên cứu đã được công bố.

Bảng 2.1: Dự báo chẩn đoán ung thư vú

Loại mô hình	Độ chính xác
SVM R	0.77
SVM L	0.80
Hybrid	0.89

Chú thích: Bảng so sánh được trích dẫn từ [19]. Hybrid: SVM + LR, SVM L: SVM tuyến tính, SVM R: SVM phi tuyến

Bảng 2.2: Dự báo rượu vang

Loại mô hình	Độ chính xác
SVM	94.29
SVM Linear	94.29
SVM Poly	94.29
Logistic Regression	54.29
Hybrid SVM - LR	97.14

Chú thích: Bảng so sánh được trích dẫn từ [20]

Dựa trên các kết quả nghiên cứu trên có thể thấy mô hình kết hợp LR và Support Vector Machine cho kết quả dự báo tốt hơn so với các mô hình riêng biệt. Do đó có thể sử dụng mô hình kết hợp này để dự báo khách hàng rời mạng. Từ những ưu điểm và kết quả dự báo của mô hình kết hợp LR và Support Vector Machine chương tiếp theo của báo cáo sẽ trình bày một ứng dụng cụ thể của mô hình này vào dự báo tại Viễn Thông Tây Ninh.

Chương 3. DỰ BÁO KHÁCH HÀNG RỜI MẠNG TẠI VIỄN THÔNG TÂY NINH

Chương này sẽ trình bày về một ứng dụng cụ thể của mô hình kết hợp Logistic Regression và Support Vector Machine trong dự báo dữ liệu khách hàng rời mạng nhằm đánh giá kết quả dự báo của mô hình kết hợp này tại Viễn Thông Tây Ninh.

3.1 Giới thiệu về công ty và bài toán dự báo

Đã gần khoảng 30 năm, kể từ khi Vinaphone - mạng di động đầu tiên của Việt Nam chính thức đi vào hoạt động. Tại thời điểm đó, di động còn là khái niệm xa lạ với đa số người tiêu dùng, số lượng thuê bao của mạng di động không nhiều vì vùng phủ sóng hạn chế và giá cước cũng như thiết bị đầu cuối còn cao. Điện thoại di động rất hiếm, giá mỗi máy khá cao khoảng 1.000 USD. Ngoài việc khan hiếm máy, tiền thuê bao và cước cuộc gọi di động cũng rất đắt, phí hòa mạng 200 USD/thuê bao, thuê bao tháng khoảng 30 USD, cước cuộc gọi cho nội hạt TP Hồ Chí Minh hoặc Hà Nội là 0,3 USD/ phút. Riêng với các cuộc gọi liên tỉnh, mức cước phí là 0,3 USD/ phút + cước liên tỉnh.

Sự bùng nổ của thị trường thông tin di động Việt Nam chỉ thực sự diễn ra trong 10 năm trở lại đây, khi Viettel chính thức bước chân vào thị trường di động năm 2004. Theo thống kê, giá cước di động Việt Nam trong 30 năm gần đây đã giảm hơn 3 lần. Cuộc cạnh tranh trở nên nóng hơn trên thị trường di động đã đưa Việt Nam từ nước có giá cước thuộc hàng cao trên thế giới đã trở thành nước có mức cước thuộc hàng rẻ nhất thế giới.

Bảng 3.1: Tình hình phát triển di động tại Việt Nam đến năm 2017 [1]

TT	Chỉ tiêu	Đơn vị tính	2015	2016	2017
1	Số thuê bao di động phát sinh lưu lượng	Thuê bao	126.499.499	128.996.179	120.016.181
2	Số thuê bao di động phát sinh lưu lượng/100 dân	%	137,90	139,20	128,08
3	Số thuê bao di động chỉ phát dinh lưu lượng thoại, tin nhắn	Thuê bao	94.552.934	92.807.762	72.265.524
4	Số thuê bao di động phát sinh lưu lượng dữ liệu	Thuê bao	31.946.565	36.188.417	47.750.657

Nhìn vào Bảng 3.1 ta có thể thấy chỉ tiêu về thoại và SMS có xu hướng giảm từ khoản 94 triệu thuê bao ở năm 2015 xuống còn khoản 72 triệu thuê bao ở năm 2017, tuy nhiên thuê báo phát sinh lưu lượng (data) có xu hướng tăng từ khoản 31 triệu thuê bao năm 2015 lên khoản 47 triệu thuê bao năm 2017 do người dùng ngày càng sử dụng điện thoại thông minh nhiều và luôn có một ứng dụng OTT (Zalo, Viber, Facebook, …) trên điện thoại để phục vụ việc gọi điện thoại hoặc nhắn tin góp phần làm doanh thu từ mạng viễn thông cũng tăng nhẹ. Tính đến hết năm 2017 doanh thu dịch vụ di động trên cả nước là 4.539,34 triệu USD [1] và có xu hướng giảm qua từng năm, nhìn chung dù dịch vụ data có tăng nhưng doanh thu các nhà mạng vẫn sụt giảm vì doanh thu data vẫn chưa bù lại được doanh thu thoại và SMS. Đây là vấn đề tất yếu khi công nghệ ngày càng phát triển và vấn đề đặt ra là các nhà mạng phải có định hướng cho tương lai về dịch vụ của mình khi mà các dịch vụ truyền thống là thoại và SMS không còn nữa.

Khách hàng rời mạng thường được phân thành 2 nhóm: chủ động rời mạng và bị động rời mạng. Chủ động rời mạng là trường hợp những khách hàng chọn để rời mạng, việc rời mạng là do chính lựa chọn của khách hàng. Ví dụ, khách hàng chuyển sang mạng đối thủ hoặc chuyển đổi sang hợp đồng thuê bao trả sau. Bị động rời mạng là trường hợp khách hàng bị nhà cung cấp ngừng cung cấp dịch vụ, thường là vì lý do gian lận hoặc nợ cước. Rời mạng vì lý do gian lận dường như rất hiếm xảy ra. Rời mạng do nợ cước thì chỉ xảy ra với thuê bao trả sau. Như đã đề cập ở trên, trong nghiên cứu này, chỉ tập trung vào vấn đề rời mạng của thuê bao trả trước. Vì vậy, rời mạng bị động xảy ra khi khách hàng không nạp lại tiền trong một khoảng thời gian đủ dài theo quy định.

Một trong những vấn đề quan trọng nhất của thuê bao trả trước là thiếu một định nghĩa đủ rõ ràng. Khi xem xét rời mạng đối với thuê bao trả sau, ngày thuê bao bị khóa 2 chiều (ngày thuê bao ngừng kết nối với mạng) chính là ngày rời mạng, đây là ngày thuê bao thực sự ngừng sử dụng dịch vụ của nhà cung cấp. Tuy nhiên, trường hợp thuê bao trả trước, ngày khóa 2 chiều cũng không thực sự là ngày rời mạng. Điều này có thể được nhìn một cách rõ ràng hơn thông qua các giai đoạn khác nhau của thuê bao trả trước.

Vì lý do thời điểm tác động được đến thuê bao quan trọng nên việc xác định thời điểm nào được coi là rời mạng sẽ rất quan trọng trong việc dự đoán rời mạng và thực hiện các tác động để duy trì, ngăn chặn thuê bao rời mạng. Trong phạm vi đề tài này, khái niệm "rời mạng" được xác định là trường hợp khách hàng không phát sinh cước (không phát sinh bất cứ giao dịch nào hoặc không có biến động về tài khoản trong vòng một tháng). Lý do sử dụng khái niệm rời mạng này như sau:

Theo kinh nghiệm thực tế, thuê bao trả trước chuyển sang giai đoạn khóa 1 chiều thì hầu như rất khó liên lạc, thậm chí đã vứt bỏ thẻ sim ra khỏi điện thoại. Do vậy, việc tác động đến thuê bao ở giai đoạn này hầu như không có hiệu quả.

Mốc “không phát sinh cước” cho phép dự đoán thuê bao rời mạng khi thuê bao vẫn còn đang ở giai đoạn đang sử dụng, đảm bảo còn đủ thời gian để thực hiện tác động trước khi thuê bao chuyển sang giai đoạn khóa 1 chiều.

Mục đích của nghiên cứu: Phát hiện các thuê bao trả trước lâu năm có khả năng rời mạng bằng cách phân lớp kho thuê bao này với nhẫn gán trước là “rời mạng” và “không rời mạng” để có thể tác động và duy trì thuê bao.

Mục tiêu của nghiên cứu: Dự đoán các thuê bao trả trước dài hạn có khả năng rời mạng khi vẫn đang ở giai đoạn đang sử dụng của vòng đời thuê bao, tức là không phát sinh cước trong thời gian 30 ngày. Sau khi có mô hình phân tích tốt và chính xác cho dữ liệu, do kho dữ liệu rất lớn nên sẽ tìm giải pháp đẩy nhanh quá trình huấn luyện nhằm tối ưu thời gian chạy mô hình.

3.2 Chuẩn bị và tiền xử lý dữ liệu

Giai đoạn chuẩn bị và xử lý dữ liệu ban đầu luôn là một giai đoạn quan trọng trong quy trình khai phá dữ liệu. Dữ liệu là một trong hai thành phần của phân lớp dữ liệu. Truy cập dữ liệu thực hiện việc trích xuất và thu thập dữ liệu cần thiết cho việc phân lớp mô hình thuê bao gửi đi. Thông tin khách hàng cần thiết để dự đoán thuê bao rời mạng bao gồm: quản lý dữ liệu khách hàng thuê bao, chi tiết dữ liệu sử dụng dịch vụ của thuê bao, thanh toán và khuyến mại của thuê bao, dữ liệu thuê bao rời mạng. Từ các dữ liệu khác nhau, một cơ sở dữ liệu đưa ra dự đoán về việc rời khỏi mạng được xây dựng với dữ liệu thu thập mục tiêu.

Dữ liệu thu thập được sau khi lọc và loại bỏ các thông tin không chính xác, không cần thiết thì gồm các thông tin:

Dữ liệu quản lý khách hàng: loại thuê bao, bưu cục thu, thời gian hoạt động.

Dịch vụ sử dụng dữ liệu di động: số lượng dịch vụ sử dụng, gọi nội mạng, ngoại mạng, quốc tế, nội mạng, ngoại mạng, quốc tế, SMS.

Dữ liệu thanh toán: tiền phát sinh gọi nội mạng, tiền phát sinh gọi ngoại mạng, tiền phát sinh gọi quốc tế, tiền phát sinh SMS, tiền phát sinh Data, tổng số tiền phát sinh, số tiền được khuyến mãi.

Results Messages

	PHONE	GRAND_PACKAGE_ID	AREA_SK_PSC_N	TOTAL_TKC	DTKC_THOAI_NOI_N	TKKM_THOAI_NOI_N	DTTKC_SMS_NOI_N	DTTKM_SMS_NOI_N
1	912898810	XTRA	ThanhPho	36913.6	23945.1	0	750	0
2	942836827	XTRA	Huyen	99	0	0	99	0
3	944612352	VINA690	ThanhPho	80717.9	47000.5	0	198	0
4	913052762	XTRA	Huyen	24836	24836	0.0800000000017462	0	0
5	824984039	CARD	ThanhPho	90493.3	15743.3	0	750	0
6	943819615	VINA365	Huyen	73450	100	996.75	350	1050
7	919012501	CARD	Huyen	161944.7	158564.7	0	3380	0
8	945917875	XTRA	Huyen	111801.8	41326.8	12000.14	0	0
9	949049478	XTRA	ThanhPho	110472.6	86419.7	0.630000000004657	5250	0
10	942983801	CARD	Huyen	140694.4	64429.8	0	2270	0
11	815219315	XTRA	Huyen	10337.7	10337.7	0	0	0
12	944838752	XTRA	Huyen	74778.5	52380.6	0.209999999999127	250	0
13	911486624	XTRA	ThanhPho	52809.4	37550.9	0	99	0
14	919749202	CARD	ThanhPho	213555.4	158765.4	9999.95999999999	290	0
15	849927992	XTRA	Huyen	7341	7332.2	0.0299999999997453	0	0
16	813613806	MYZONE	Huyen	9703.9	703.9	0.0299999999999727	0	0
17	917323330	CARD	Huyen	100275.7	0	0	0	0
18	852000407	MYZONE	Huyen	105730.8	84390.8	0.529999999998836	1340	0
19	813205679	SINHVIEN	Huyen	39998.2	1848.2	0	0	0
20	947191057	CARD	ThanhPho	111499.5	15005.4	0	0	0

Hình 3.1: Dữ liệu thực tế SQL tại VNPT Tây Ninh

Dữ liệu thu thập là dữ liệu di động trả trước vinaphone tại đơn vị tổng hợp từ nhiều nguồn như: thông tin thuê bao, lịch sử nạp thẻ, lịch sử gọi, SMS, … trong 6 tháng từ tháng 07/2019 đến 12/2019. Chi tiết các trường dữ liệu được mô tả trong Bảng 3.2.

Bảng 3.2: Mô tả các trường dữ liệu

STT	**Mô tả**	**Kiểu dữ liệu**
1	Nơi phát sinh cước	Chuỗi
2	Số điện thoại	Chuỗi
3	Gói dịch vụ	Chuỗi
4	Thời gian bắt đầu hoà mạng	Ngày giờ
5	Khu vực phát sinh cước	Chuỗi
6	Tài khoản chính	Số thực
7	Doanh thu gọi nội mạng tài khoản chính	Số thực
8	Doanh thu gọi nội mạng tài khoản khuyến mãi	Số thực
9	Doanh thu sms nội mạng tài khoản chính	Số thực

10	Doanh thu sms nội mạng tài khoản khuyến mãi	Số thực
11	Doanh thu data tài khoản chính	Số thực
12	Doanh thu data tài khoản khuyến mãi	Số thực
13	Doanh thu GTGT tài khoản chính	Số thực
14	Doanh thu GTGT tài khoản khuyến mãi	Số thực
15	Doanh thu tài khoản chính khác	Số thực
16	Doanh thu tài khoản khuyến mãi khác	Số thực
17	Tài khoản còn lại	Số thực
18	Số lượng cuộc thoại chiều đi	Số nguyên
19	Số lượng lưu lượng thoại chiều đi	Số nguyên
20	Số lượng lưu lượng thoại chiều đến	Số nguyên
21	Số lượng SMS chiều đi	Số nguyên
22	Số lượng SMS chiều đến	Số nguyên
23	Số tiền nạp thẻ	Số thực
24	Số lượng thẻ nạp	Số nguyên
25	Tháng	Số nguyên
26	Trạng thái rời mạng	**1: Rời mạng** **-1: Không rời mạng**

Từ bảng dữ liệu 3.2 tiến hành làm sạch dữ liệu bằng cách loại bỏ các dòng dữ liệu có trường trống hoặc null, các trường dữ liệu xuất hiện nhiều giá trị 0 có ảnh hưởng đến quá trình chạy mô hình. Loại bỏ một số trường mang tính bảo mật người dùng: họ tên, địa chỉ, số điện thoại... Tiến hành chuyển đổi kiểu dữ liệu từ dạng chữ (chuỗi) sang dạng số bằng cách mã hóa các kí tự bằng số.

Bảng 3.3: Bảng dữ liệu và mã hoá dữ liệu

1	GRAND_PACKAGE_ID	Gói dịch vụ	VINA690: 0, XTRA: 1, MYZONE: 2, CARD: 3, SINHVIEN: 4, VINA360: 5, EZCOM: 6, DAILY: 7, HOCSINH: 8, VINA088: 9
2	AREA_SK_PSC_N	Khu vực phát sinh cước	Thành phố: 0, Huyện: 1
3	TOTAL_TKC	Tài khoản chính	Dưới 10k: 0, Trên 10k: 1
4	DTKC_THOAI_NOI_N	Doanh thu gọi nội mạng tài khoản chính	Có: 1, Không: 0
5	TKKM_THOAI_NOI_N	Doanh thu gọi nội mạng tài khoản khuyến mãi	Có: 1, Không: 0
6	DTTKC_SMS_NOI_N	Doanh thu sms nội mạng tài khoản chính	Có: 1, Không: 0
7	DTTKM_SMS_NOI_N	Doanh thu sms nội mạng tài khoản khuyến mãi	Có: 1, Không: 0
8	DTTKC_DATA_N	Doanh thu data tài khoản chính	Có: 1, Không: 0
9	DTTKM_DATA_N	Doanh thu data tài khoản khuyến mãi	Có: 1, Không: 0
10	DTTKC_GTGT	Doanh thu GTGT tài khoản chính	Có: 1, Không: 0
11	REMAIN_CREDIT	Tài khoản còn lại	Có: 1, Không: 0
12	NUM_OG_CALLS	Số lượng cuộc thoại chiều đi	Có: 1, Không: 0
13	SUM_DURATION_OG	Số lượng lưu lượng thoại chiều đi	Có: 1, Không: 0
14	SUM_DURATION_IC	Số lượng lưu lượng thoại chiều đến	Có: 1, Không: 0

15	SCR_AMOUNT_N	Số tiền nạp thẻ	Có: 1, Không: 0
16	RM	Trạng thái rời mạng	Rời mạng: 1, Không rời mạng: -1

Thu được kết quả ở dạng mã hóa theo chuẩn đầu vào huấn luyện và dự báo của mô hình:

Results Messages

	GRAND_PACKAGE_ID	AREA_SK_PSC_N	TOTAL_TKC	DTKC_THOAI_NOI_N	TKKM_THOAI_NOI_N	DTTKC_SMS_NOI_N	DTTKM_SMS_NOI_N	DTTKC_DATA_N
1	2	1	1	1	1	1	0	0
2	1	1	1	1	0	0	0	0
3	0	0	1	1	0	0	0	1
4	2	1	1	1	1	1	0	1
5	5	1	0	1	1	1	0	0
6	4	1	1	1	0	0	0	0
7	1	1	1	1	1	1	0	1
8	2	0	1	0	0	0	0	0
9	4	1	1	1	0	1	0	1
10	0	1	1	1	0	1	0	0
11	3	1	1	1	0	1	0	1
12	3	1	1	1	1	0	0	1
13	2	1	1	1	1	1	0	1
14	4	1	0	1	1	0	0	0
15	1	1	0	1	0	1	0	1
16	1	1	0	0	0	0	0	1
17	3	1	1	1	1	1	1	0
18	2	1	1	1	1	1	0	0
19	3	1	1	1	0	0	0	1
20	4	1	1	1	0	1	0	1

Hình 3.2: Dữ liệu đầu vào đã mã hóa

3.3 Dự báo

Mô hình kết hợp Logistic và Support Vector Machine thực hiện hai bước dự báo. Bước đầu tiên sử dụng mô hình LR để dự báo thành phần tuyến tính. Bước thứ hai sử dụng phương pháp SVM để dự báo thành phần phi tuyến.

3.3.1 Dự báo thành phần tuyến tính bằng mô hình LR

Như đã trình bày trong chương 2, mô hình LR có thể dự báo cho dữ liệu tuyến tính. Đầu tiên xây dựng mô hình LR để dự báo cho phần tuyến tính của dữ liệu:

Mô hình	**Độ chính xác**
Logistic Regression	0.830

Chú thích: Mã nguồn và thông số sử dụng từ phần mềm R

3.3.2 Dự báo thành phần phi tuyến bằng SVM

Sau khi dự báo thành phần tuyến tính bằng mô hình LR, tiếp tục huấn luyện dự báo thành phần phi tuyến bằng SVM.

Mô hình	Độ chính xác	Thời gian (s)
SVM	0.829	200.4576
SVM song song	0.828	0.36385

Chú thích: Các giá trị được tính bằng phần mềm thống kê R

3.3.3 Kết hợp các kết quả dự báo

Kết quả dự báo của mô hình kết hợp Logistic Regression và Support Vector Machine là tổng hợp kết quả dự báo của thành phần tuyến tính bằng mô hình LR và kết quả dự báo của thành phần phi tuyến bằng SVM. Như đã mô tả về mô hình kết hợp. Mô hình lai giữa LR và SVM sau khi huấn luyện và dự báo các mô hình riêng lẻ, mô hình lớp trên sẽ sử dụng hai kết quả dự báo của từng mô hình riêng lẻ để kết hợp lại tạo thành một mô hình mới.

3.4 Kết quả dự báo và đánh giá

3.4.1 Độ chính xác của thuật toán

Cách đơn giản và hay được sử dụng nhất là accuracy (độ chính xác). Cách đánh giá này đơn giản tính tỉ lệ giữa số điểm được dự đoán đúng và tổng số điểm trong tập dữ liệu kiểm thử.

Giả sử ta có bài toán phân lớp với đầu ra là 2 lớp Đúng/Sai, kết quả phân lớp trên tập mẫu so với thực tế có 4 khả năng thể hiện … Bảng này được gọi là ma trận sai số (confusion matrix).

Bảng 3.4: Bảng ma trận sai số

		Lớp dự đoán (predicted class)	
		Đúng	**Sai**
Lớp thực tế (actual class)	**Đúng**	True Positive (TP)	False Negative (FN)
	Sai	False Positive (FP)	True Negative (TN)

True Positive thể hiện khả năng dự đoán phân lớp phân lớp đúng của một mẫu dự đoán thuộc phân lớp Đúng, False Positive thể hiện khả năng dự đoán phân lớp sai của một mẫu được dự đoán thuộc phân lớp Đúng.

False Negative thể hiện khả năng dự đoán phân lớp phân lớp đúng của một mẫu dự đoán thuộc phân lớp Sai, True Negative thể hiện khả năng dự đoán phân lớp sai của một mẫu được dự đoán thuộc phân lớp Sai.

Ta có độ đo đánh giá hiệu quả của kết quả phân lớp như sau:

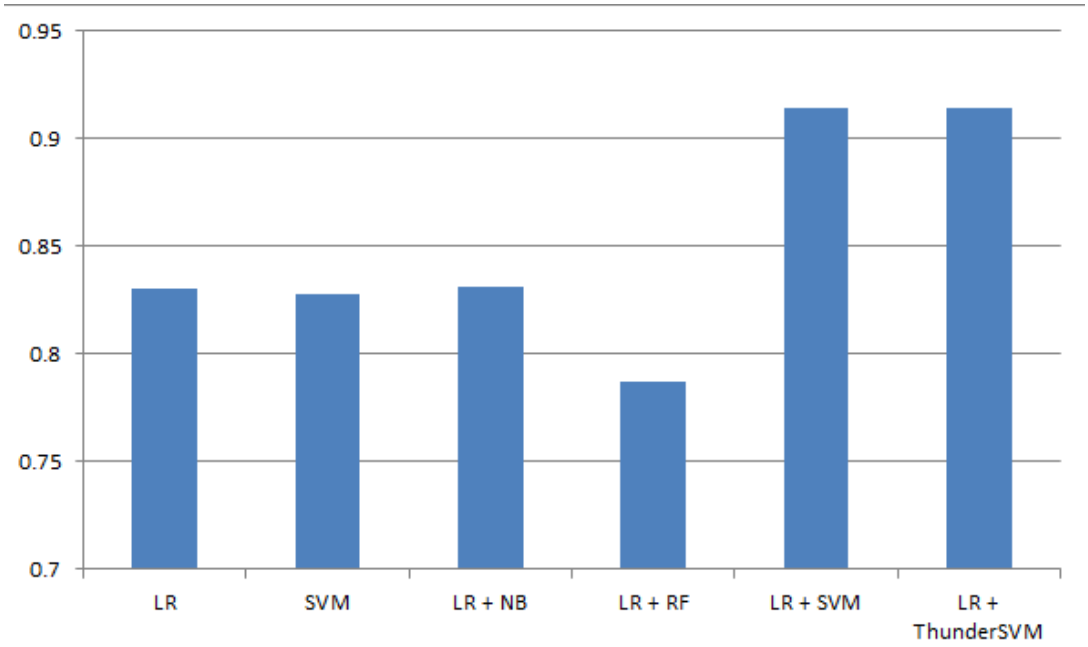
Bảng 3.5: Cách tính độ chính xác

Tên độ đo	**Công thức**	**Diễn giải**
Độ chính xác	$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$	Tỷ lệ các mẫu được phân lớp đúng trên toàn bộ tập mẫu

3.4.2 Kết quả dự báo và đánh giá

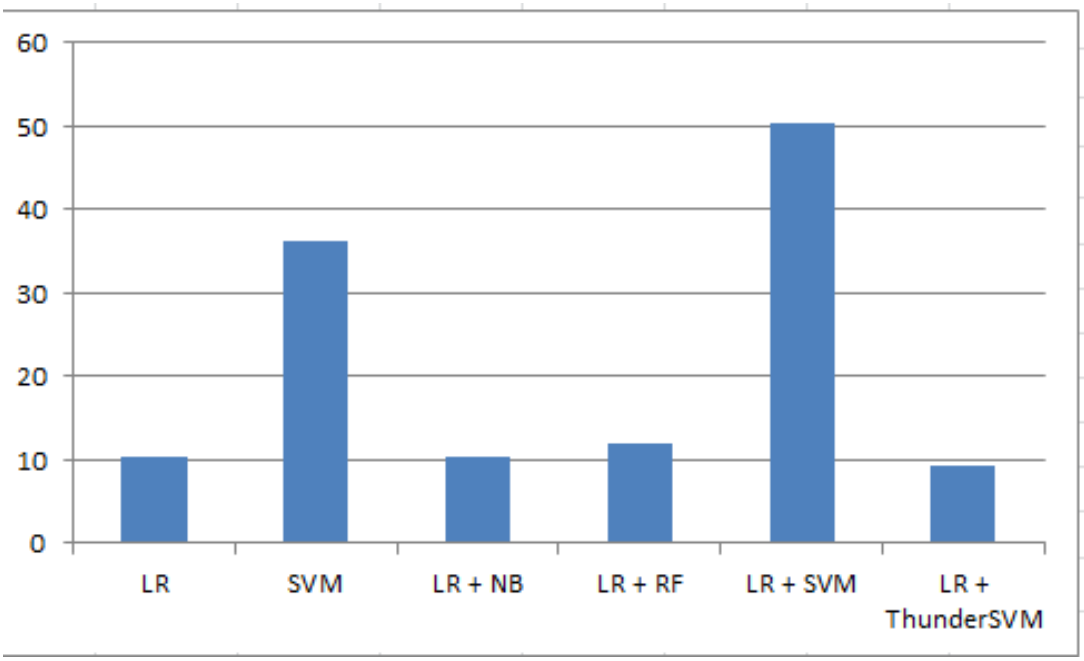
Bảng 3.6: Kết quả dự báo của các mô hình

Mô hình	Độ chính xác	Thời gian(s)
Logistic Regression	0.830	10.2433
SVM	0.828	36.0965
Logistic Regression + Naïve Bayes	0.831	10.3104
Logistic Regression + Random Forest	0.787	11.8493
Logistic Regression + SVM	0.914	50.4576
Logistic Regression + ThunderSVM	0.914	9.36385



Hình 3.3: Biểu đồ so sánh độ chính xác của các thuật toán phân lớp

Bảng 3.6 là kết quả dự báo của các mô hình dựa trên các độ đo được trình bày trong mục 3.3. Từ kết quả dự báo này có thể thấy mô hình kết hợp LR và Support Vector Machine cho kết quả dự báo tốt nhất trên cùng một tập dữ liệu so với các mô hình khác là mô hình LR, mô hình SVM, mô hình kết hợp LR và NB, mô hình kết hợp LR và RF.



Hình 3.4: Biểu đồ so sánh thời gian huấn luyện của các thuật toán phân lớp (đơn vị giây)

Điều đó chứng tỏ mô hình kết hợp LR và SVM phù hợp để dự báo cho dữ liệu tại Viễn Thông Tây Ninh. Do đó có thể sử dụng mô hình kết hợp LR và Support Vector Machine vào dự báo số khách hàng rời mạng theo từng tháng, quý hoặc năm với tốc độ tối ưu nhất.

Chương 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết luận

Kết quả dự báo số khách hàng rời mạng củng cố thêm tính đúng đắn của hướng tiếp cận kết hợp các mô hình dự báo dữ liệu khách hàng nói chung và mô hình dự báo dữ liệu khách hàng rời mạng kết hợp Logistic Regression và Support Vector Machine nói riêng.

Mô hình kết hợp Logistic Regression và Support Vector Machine thể hiện kết quả dự báo vượt trội hơn so với các mô hình khác như mô hình LR đơn lẻ hay mô hình SVM đơn lẻ trong dự báo dữ liệu khách hàng rời mạng. Phương pháp Support Vector Machine giúp tăng độ chính xác và công cụ ThunderSVM sử dụng GPUs vừa giúp cho kết quả dự báo cũng như thời gian huấn luyện của mô hình.

Lý do chính giúp kết quả dự báo của mô hình kết hợp Logistic và Support Vector Machine vượt trội hơn so với các mô hình khác là do dữ liệu khách hàng rời mạng trong thực tế thường bao gồm hai phần tuyến tính và phi tuyến tính. Nếu một mô hình dự báo chỉ có thể dự báo tốt cho một trong hai phần đó thì kết quả dự báo thường không chính xác với thực tế.

Mặc dù kết quả dự báo của mô hình kết hợp Logistic Regression và Support Vector Machine là vượt trội hơn so với các mô hình khác nhờ sử dụng vào GPUs tăng tốc độ huấn luyện nhưng cũng sinh ra chi phí để xây dựng mô hình kết hợp cũng lớn hơn so với các mô hình đơn lẻ khác. Việc lắp ráp hai mô hình có thể rất khó và tốn thời gian để thực thi ở các doanh nghiệp.

Từ việc nghiên cứu những yêu cầu cấp thiết đặt ra trong công tác duy trì và phát triển thuê bao của mạng di động, luận văn đã đạt được một số kết quả chính sau đây:

- Xây dựng mô hình dự báo áp dụng kỹ thuật khai phá dữ liệu để phát hiện nhanh chính xác các thuê bao di động có khả năng rời mạng từ đó áp dụng các giải pháp để duy trì thuê bao.

- Triển khai mô hình đề xuất, áp dụng trên dữ liệu thực tế, so sánh với các giải pháp đã sử dụng được áp dụng. Các kết quả đạt được đã cho thấy được tiềm năng áp dụng phương pháp đề xuất vào thực tiễn

Trong thời gian tới chúng tôi sẽ nghiên cứu tích hợp các kỹ thuật này vào các chương trình hỗ trợ kinh doanh của Vinaphone Tây Ninh đồng thời cải tiến thời gian dự báo cũng như kết quả dự báo. Trong thời gian tới tôi sẽ tiếp tục cập nhật mô hình với dữ liệu của Vinaphone Tây Ninh để kết quả dự đoán được cải thiện hơn.

4.2 Hướng phát triển

Trong hầu hết các nghiên cứu hay ứng dụng về mô hình kết hợp Logistic Regression cho dữ liệu tuyến tính và các phương pháp máy học như Support Vector Machine cho dữ liệu phi tuyến, sự kết hợp này chỉ dừng lại ở việc tổng hợp các kết quả dự báo của các mô hình đơn lẻ lại với nhau để cho ra kết quả dự báo cuối cùng, nên giữa hai mô hình này không có liên kết gì với nhau.

Do đó để kết quả dự báo hiệu quả hơn cần có sự kết hợp chặt chẽ giữa các mô hình sao cho các mô hình này có thể hỗ trợ cho nhau trong việc dự báo. Chính vì vậy mà vấn đề làm thế nào để kết hợp chặt chẽ các phương pháp dự báo trong các mô hình kết hợp cũng là một hướng phát triển của để tài. Ngoài ra, cũng có thể phát triển mô hình được kết hợp từ 3 hoặc hơn các mô hình đơn lẻ để tăng độ chính xác của mô hình kết hợp.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] Bộ Thông tin và Truyền thông. *Sách Trắng về Công nghệ thông tin và Truyền thông (CNTT-TT) Việt Nam 2018.* Hà Nội: Nhà xuất bản Thông tin và Truyền thông, 2018, tr.31-33.

[2] B. Huang, M. T. Kechadi, and B. Buckley, "*Customer churn prediction in telecommunications*," Expert Syst. Appl., vol. 39, pp. 1414–1425, Jan. 2012.

[3] M. Owczarczuk, "*Churn models for prepaid customers in the cellular telecommunication industry using large data marts*," Expert Systems with Applications, vol. 37,no. 6, pp. 4710 – 4712, 2010.

[4] G. Li and X. Deng, "*Customer churn prediction of china telecom based on cluster analysis and decision tree algorithm*," in Emerging Research in Artificial Intelligence and Computational Intelligence (J. Lei, F. Wang, H. Deng, and D. Miao, eds.), Communications in Computer and Information Science, pp. 319–327, Springer Berlin Heidelberg, 2012.

[5]SAGAR S. NikAM. "*A Comparative Study of Lớpifcation Techniques in Data Mining Algorithms*".Department of Computer Science, K.K.Wagh College of Agriculture, Nashik, India. April 10, 2015

[6] Lingling Yang, Dongyang Li, Yao Lu, "*Prediction Modeling and Analysis for Telecom Customer Churn in Two Months*" School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China Guangdong Province Key Laboratory of Computational Science, 135 Xingang Xi Road, Guangzhou, China, January. 2019

[7] XIA Guo-en, JIN Wei-dong "*Model of Customer Churn Prediction on Support Vector Machine*", Volume 28, Issue 1, January 2008

[8] Hossein Abbasimehr, Mostafa Setak, M. J. Tarokh. "*A Neuro-Fuzzy Lǿpifier for Customer Churn Prediction*", K. N. Toosi University of Tech Tehran, Iran, Volume 19– No.8, April 2011

[9] Hemlata Jain, Ajay Khunteta, Sumit Srivastava. "*Churn Prediction in Telecommunication using Logistic Regression and Logit Boost*". Computer Science, School of Basic and Applied Sciences, Poornima University Jaipur-303905, India, January 2020.

[10] Mamdouh A. M. Abdelsalam & Doaa Akl Ahmed. "*Combining Forecasts from Linear and Nonlinear Models Using Sophisticated Approaches*". Astley Clarke Building, University of Leicester, Leicester. October 25, 2015.

[11] ALEX J. SMOLA and BERNHARD SCHOLKOPF. "*A tutorial on support vector regression*". RSISE, Australian National University, Canberra 0200, Australia. November 2003.

[12] Tony Van Gestel, Bart Baesens, Peter Van Dijcke, Johan A. K. Suykens, Joao Garcia, Thomas Alderweireld. "*Linear and non-linear credit scoring by combining logistic regression and support vector machines*". Group Risk Management, Dexia Group, Square Meeus 1, B-1000 Brussels, Belgium. Volume 1/Number 4, Fall 2005.

[13] Mythili T., Dev Mukherji, Nikita Padalia, and Abhiram Naidu. "*A Heart Disease Prediction Model using SVM-Decision Trees-Logistic Regression (SDL)"*. School of Computing Sciences and Engineering, VIT University Vellore – 632014, Tamil Nadu, India. Volume 68– No.16, April 2013.

[14] Zeyi Wen, Jiashuai Shi, Qinbin Li, Bingsheng He, Jian Chen. "*ThunderSVM: a fast SVM library on GPUs and CPUs*". The Journal of Machine Learning ResearchVolume 19Issue 1January 2018 pp 797–801.

[15] S. I. Gallant. "*Perceptron-Based Learning Algorithms*," in IEEE transaction on neural. February 1990.

[16] S. Boyd and L. Vandenberghe. "*Convex Optimization*". Cambridge University Press, 2014.

[17] Univesity of Florida. The Foundation for The Gator Nation (Nov. 2016), http://www.cise.ufl.edu/lớp/cis4930sp11dtm/notes/intro_svm_new.pdf

[18] Z. Wen et al. "*ThunderSVM*." Internet: https://github.com/Xtra-Computing/thundersvm

[19] Yuan-chin Ivan Chang. "*Boosting SVM Lớpifiers with Logistic Regression*". Institute of Statistical Science, Academia Sinica, Taipei, Taiwan. June 2003.

[20] Eviyana Atmanegara, Taly Purwa. "*Hybrid Support Vector Machine and Logistic Regression for Multilớp Lớpification: A Case Study on Wine Dataset*". Indonesian Journal of Data ScienceVol.1, No.1, April 2021, pp. 1~7.

PHỤ LỤC

MÃ NGUỒN

Báo cáo sử dụng phần mềm thống kê RStudio phiên bản 1.4.1717-3 để tính toán các giá trị và cài đặt chương trình. Phần này trình bày các đoạn mã nguồn được thực hiện trong báo cáo.

Cài đặt mô hình dự báo bằng Logistic Regression

```
# Load library
library(caret)
# Read file
churn <- read.csv('C:/R/data_csv.csv', header =
TRUE,colClasses="factor")

str(churn)

intrain<-
createDataPartition(churn$RM,p=0.7,list=FALSE)

set.seed(2017)

training<- churn[intrain,]

testing<- churn[-intrain,]

# Forecast

start_time <- Sys.time()

LogModel <- glm(RM ~
.,family=binomial(link="logit"),data=training)

end_time <- Sys.time()

end_time - start_time

# Predict

testing$RM <- as.character(testing$RM)
```

```
testing$RM[testing$RM=="0"] <- "0"

testing$RM[testing$RM=="1"] <- "1"

fitted.results <-
predict(LogModel,newdata=testing,type='response')

fitted.results <- ifelse(fitted.results > 0.5,1,0)

misClasificError <- mean(fitted.results !=
testing$RM)

# Measured

print(paste('Logistic Regression Accuracy',1-
misClasificError))

print("Confusion Matrix for Logistic Regression");

table(testing$RM, fitted.results > 0.5)
```

Cài đặt mô hình dự báo bằng SVM

```
# Load library

library(caret)

# Read file

data = read.csv("C:/R/data_csv.csv", header =
TRUE,colClasses="factor")

summary(data)

ind = sample(2, nrow(data), replace=TRUE,
prob=c(0.8, 0.2))

trainData = data[ind==1,] #Training Data

testData = data[ind==2,] #Testing Data

# Train

start_time <- Sys.time()
```

```
svm.model_svm <- train(RM ~., data = trainData,
method = "svmPoly", trControl = fitControl,
preProcess = c("center","scale"))

end_time <- Sys.time()

end_time - start_time
# Predict
pred <- predict(svm.model,trainData)
# Measured
caret::confusionMatrix(pred, trainData$RM)
```

Cài đặt mô hình kết hợp Logistic Regression và Naïve Bayes

```
#Loading the required libraries

library('caret')


#Seeting the random seed

set.seed(1)


#Loading the dataset

data_processed<-read.csv('C:/R/data_csv.csv',
header = TRUE,colClasses="factor")


#Splitting training set into two parts based on
outcome: 75% and 25%

index <- createDataPartition(data_processed$RM,
p=0.75, list=FALSE)

trainSet <- data_processed[ index,]
```

```
testSet <- data_processed[-index,]


#Defining the training control

fitControl <- trainControl(

  method = "cv",

  number = 10,

  savePredictions = 'final', # To save out of fold predictions for best parameter combinations

  classProbs = T # To save the class probabilities of the out of fold predictions

)

#Defining the predictors and outcome

predictors<-c("GRAND_PACKAGE_ID",  "AREA_SK_PSC_N", "TOTAL_TKC", "DTKC_THOAI_NOI_N","TKKM_THOAI_NOI_N", "DTTKC_SMS_NOI_N","DTTKM_SMS_NOI_N","DTTKC_DATA_N", "DTTKM_DATA_N","DTTKC_GTGT","DTTKM_GTGT","DTTKC_KHAC_N","REMAIN_CREDIT","NUM_OG_CALLS","SUM_DURATION_OG","SUM_DURATION_IC","SCR_AMOUNT_N")

outcomeName<-'RM'

#Training the naive bayes model

model_nb<-
train(trainSet[,predictors],trainSet[,outcomeName], method='nb',trControl=fitControl,tuneLength=3)


#Training the logistic regression model
```

```
model_lr<-
train(trainSet[,predictors],trainSet[,outcomeName],
method='glm',trControl=fitControl,tuneLength=3)


#Predicting the out of fold prediction
probabilities for training data

trainSet$OOF_pred_nb<-
model_nb$pred$Y[order(model_nb$pred$rowIndex)]

trainSet$OOF_pred_lr<-
model_lr$pred$Y[order(model_lr$pred$rowIndex)]


#Predicting probabilities for the test data

testSet$OOF_pred_nb<-
predict(model_nb,testSet[predictors],type='prob')$Y

testSet$OOF_pred_lr<-
predict(model_lr,testSet[predictors],type='prob')$Y


#Predictors for top layer models

predictors_top<-c('OOF_pred_nb','OOF_pred_lr')


#Logistic regression as top layer model

model_glm<-
train(trainSet[,predictors_top],trainSet[,outcomeNa
me],method='glm',trControl=fitControl,tuneLength=3)
```

```
#predict using logistic regression top layer model

testSet$glm_stacked<-
predict(model_glm,testSet[,predictors_top])


#confusion matrix LR

confusionMatrix(testSet$RM,testSet$glm_stacked)


#Naive bayes as top layer model

model_nbb<-
train(trainSet[,predictors_top],trainSet[,outcomeNa
me],method='nb',trControl=fitControl,tuneLength=3)


#predict using naive bayes top layer model

testSet$nb_stacked<-
predict(model_nbb,testSet[,predictors_top])


#confusion matrix LR

confusionMatrix(testSet$RM,testSet$nb_stacked)
```

Mô hình dự báo kết hợp giữa Logistic Regression và Random Forest

```
#Loading the required libraries

library('caret')


#Seeting the random seed
```

```
set.seed(1)


#Loading the dataset

data_processed<-read.csv('C:/R/data_csv.csv',
header = TRUE,colClasses="factor")


#Does the data contain missing values

sum(is.na(data))


#Splitting training set into two parts based on
outcome: 75% and 25%

index <- createDataPartition(data_processed$RM,
p=0.75, list=FALSE)

trainSet <- data_processed[ index,]

testSet <- data_processed[-index,]


#Defining the training control

fitControl <- trainControl(

  method = "cv",

  number = 10,

  savePredictions = 'final', # To save out of fold
predictions for best parameter combinations
```

```
  classProbs = T # To save the class probabilities
of the out of fold predictions

)

#Defining the predictors and outcome

predictors<-c("GRAND_PACKAGE_ID", "AREA_SK_PSC_N",
"TOTAL_TKC", "DTKC_THOAI_NOI_N","TKKM_THOAI_NOI_N",
"DTTKC_SMS_NOI_N","DTTKM_SMS_NOI_N","DTTKC_DATA_N",
"DTTKM_DATA_N","DTTKC_GTGT","DTTKM_GTGT","DTTKC_KHA
C_N","REMAIN_CREDIT","NUM_OG_CALLS","SUM_DURATION_O
G","SUM_DURATION_IC","SCR_AMOUNT_N")

outcomeName<-'RM'


#Training the random forest model

model_rf<-
train(trainSet[,predictors],trainSet[,outcomeName],
method='rf',trControl=fitControl,tuneLength=3)


#Training the logistic regression model

model_lr<-
train(trainSet[,predictors],trainSet[,outcomeName],
method='glm',trControl=fitControl,tuneLength=3)


#Predicting the out of fold prediction
probabilities for training data

trainSet$OOF_pred_rf<-
model_rf$pred$Y[order(model_rf$pred$rowIndex)]
```

```
trainSet$OOF_pred_lr<-
model_lr$pred$Y[order(model_lr$pred$rowIndex)]


#Predicting probabilities for the test data

testSet$OOF_pred_rf<-
predict(model_rf,testSet[predictors],type='prob')$Y

testSet$OOF_pred_lr<-
predict(model_lr,testSet[predictors],type='prob')$Y


#Predictors for top layer models

predictors_top<-c('OOF_pred_rf','OOF_pred_lr')


#Logistic regression as top layer model

model_glm<-
train(trainSet[,predictors_top],trainSet[,outcomeNa
me],method='glm',trControl=fitControl,tuneLength=3)


#predict using logistic regression top layer model

testSet$glm_stacked<-
predict(model_glm,testSet[,predictors_top])


#confusion matrix LR

confusionMatrix(testSet$RM,testSet$glm_stacked)
```

```
#Random forest as top layer model

model_rff<-
train(trainSet[,predictors_top],trainSet[,outcomeNa
me],method='rf',trControl=fitControl,tuneLength=3)


#predict using random forest top layer model

testSet$rf_stacked<-
predict(model_rff,testSet[,predictors_top])


#confusion matrix random forest

confusionMatrix(testSet$RM,testSet$rf_stacked)
```

Mô hình kết hợp giữa Logistic Regression và SVM

```
#Loading the required libraries

library('caret')

#Seeting the random seed

set.seed(1)


#Loading the dataset

data_processed<-read.csv('C:/R/data_csv.csv',
header = TRUE,colClasses="factor")


#Splitting training set into two parts based on
outcome: 75% and 25%
```

```
index <- createDataPartition(data_processed$RM,
p=0.8, list=FALSE)

trainSet <- data_processed[ index,]

testSet <- data_processed[-index,]


#Defining the training control

fitControl <- trainControl(

  method = "cv",

  number = 10,

  savePredictions = 'final', # To save out of fold
predictions for best parameter combinations

  classProbs = TRUE # To save the class
probabilities of the out of fold predictions

)

#Defining the predictors and outcome

predictors<-c("GRAND_PACKAGE_ID", "AREA_SK_PSC_N",
"TOTAL_TKC", "DTKC_THOAI_NOI_N","TKKM_THOAI_NOI_N",
"DTTKC_SMS_NOI_N","DTTKM_SMS_NOI_N","DTTKC_DATA_N",
"DTTKM_DATA_N","DTTKC_GTGT","DTTKM_GTGT","DTTKC_KHA
C_N","REMAIN_CREDIT","NUM_OG_CALLS","SUM_DURATION_O
G","SUM_DURATION_IC","SCR_AMOUNT_N")

outcomeName<-'RM'


#Training the svm model
```

```
model_svm <- train(RM ~., data = trainSet, method =
"svmPoly", trControl = fitControl,  preProcess =
c("center","scale"))


#Training the logistic regression model

model_lr<-
train(trainSet[,predictors],trainSet[,outcomeName],
method='glm',trControl=fitControl,tuneLength=3)


#Predicting the out of fold prediction
probabilities for training data

trainSet$OOF_pred_svm<-
model_svm$pred$Y[order(model_svm$pred$rowIndex)]

trainSet$OOF_pred_lr<-
model_lr$pred$Y[order(model_lr$pred$rowIndex)]


#Predicting probabilities for the test data

testSet$OOF_pred_svm<-
predict(model_svm,testSet[predictors],type='prob')$
Y

testSet$OOF_pred_lr<-
predict(model_lr,testSet[predictors],type='prob')$Y


#Predictors for top layer models

predictors_top<-c('OOF_pred_svm','OOF_pred_lr')
```

```
#Logistic regression as top layer model

model_glm<-
train(trainSet[,predictors_top],trainSet[,outcomeNa
me],method='glm',trControl=fitControl,tuneLength=3)


#predict using logistic regression top layer model

testSet$glm_stacked<-
predict(model_glm,testSet[,predictors_top])


#confusion matrix LR

confusionMatrix(testSet$RM,testSet$glm_stacked)


#SVM as top layer model

model_svmm<-
train(trainSet[,predictors_top],trainSet[,outcomeNa
me], method = "svmPoly", trControl = fitControl,
preProcess = c("center","scale"))


#predict using SVM top layer model

testSet$svm_stacked<-
predict(model_svmm,testSet[,predictors_top])


#confusion matrix SVM
```

```
confusionMatrix(testSet$RM,testSet$svm_stacked)
```

Mô hình kết hợp giữa Logistic Regression và SVM Song song

```
## Hybrid Model

# Load library

library(hydroGOF)

library(ggplot2)

library(e1071)

library(plyr)

library(corrplot)

library(ggplot2)

library(gridExtra)

library(ggthemes)

library(caret)

library(MASS)

library(randomForest)

library(party)


# Read file

train <- read.csv('C:/R/Train.csv', header =
TRUE,colClasses="factor")

train_svm <- read.csv('C:/R/TestLR.csv', header =
TRUE,colClasses="factor")
```

```
test <- read.csv('C:/R/Test.csv', header =
TRUE,colClasses="factor")


# Forecasting use LR

LogModel <- glm(RM ~
.,family=binomial(link="logit"),data=train)

pred.LogModel <- predict(LogModel,n.ahead=15)


# Calculated Residuals

residuals <- train_svm$RM - LogModel$pred;

# ThunderSVM function

check_location <- function(){

  if(Sys.info()['sysname'] == 'Windows'){


if(!file.exists("C:/Users/Sang/thundersvm/build/bin
/Debug/thundersvm.dll")){

      stop("Please build the library first (or
check you called this while your workspace is set
to the thundersvm/R/ directory)!")

    }


dyn.load("C:/Users/Sang/thundersvm/build/bin/Debug/
thundersvm.dll")

  } else if(Sys.info()['sysname'] == 'Linux'){
```

```
if(!file.exists("../build/lib/libthundersvm.so")){
      stop("Please build the library first (or
check you called this while your workspace is set
to the thundersvm/R/ directory)!")
    }
    dyn.load("../build/lib/libthundersvm.so")
  } else if(Sys.info()['sysname'] == 'Darwin'){

if(!file.exists("../build/lib/libthundersvm.dylib")
){
      stop("Please build the library first (or
check you called this while your workspace is set
to the thundersvm/R/ directory)!")
    }
    dyn.load("../build/lib/libthundersvm.dylib")
  } else{
    stop("OS not supported!")
  }
}
check_location() # Run this when the file is
sourced

svm_train_R <-
```

```
function(
svm_type = 0, kernel = 2,degree = 3,gamma = 'auto',
coef0 = 0.0, nu = 0.5, cost = 1.0, epsilon = 0.1,
tol = 0.001, probability = FALSE, class_weight =
'None', cv = '-1',
verbose = FALSE, max_iter = -1, n_cores = -1,
dataset = 'None', model_file = 'None'
)
{
    check_location()
    if(!file.exists(dataset)){stop("The file
containing the training dataset provided as an
argument in 'dataset' does not exist")}
    res <- .C("train_R", as.character(dataset),
as.integer(kernel), as.integer(svm_type),
    as.integer(degree), as.character(gamma),
as.double(coef0), as.double(nu),
    as.double(cost), as.double(epsilon),
as.double(tol), as.integer(probability),
    as.character(class_weight),
as.integer(length(class_weight)), as.integer(cv),
    as.integer(verbose), as.integer(max_iter),
as.integer(n_cores), as.character(model_file))
}
```

```
svm_predict_R <-

function(

test_dataset = 'None', model_file = 'None',
out_file = 'None'

)

{

    check_location()

    if(!file.exists(test_dataset)){stop("The file
containing the training dataset provided as an
argument in 'test_dataset' does not exist")}

    if(!file.exists(model_file)){stop("The file
containing the model provided as an argument in
'model_file' does not exist")}

    res <- .C("predict_R",
as.character(test_dataset),
as.character(model_file), as.character(out_file))

}

# Forecasting use ThunderSVM

index <- 1:15

svm <- svm_train_R(residuals ~ index,residuals ,
cost = 100, gamma = 0.5)

pred.svm <- svm_train_R(svm)


# Continue forecasting LR
```

```
pred.LogModel <- predict(LogModel,n.ahead=30)

pred.LogModel <- pred.LogModel$pred[16:30]


# Combine two results

pred <- pred.LogModel + pred.svm;


# Print results

caret::confusionMatrix(pred, train$RM)
```