

**BỘ GIÁO DỤC VÀ ĐÀO TẠO BỘ NÔNG NGHIỆP VÀ PTNT
TRƯỜNG ĐẠI HỌC THỦY LỢI**



TẠ HỒNG QUÂN

**XÂY DỰNG MÔ HÌNH DỰ ĐOÁN KHÁCH HÀNG TIỀM NĂNG TRỞ
THÀNH KHÁCH HÀNG TRẢ TIỀN (PAID USER)**

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2022

BỘ GIÁO DỤC VÀ ĐÀO TẠO BỘ NÔNG NGHIỆP VÀ PTNT
TRƯỜNG ĐẠI HỌC THỦY LỢI

TẠ HỒNG QUÂN

**XÂY DỰNG MÔ HÌNH DỰ ĐOÁN KHÁCH HÀNG TIỀM NĂNG TRỞ
THÀNH KHÁCH HÀNG TRẢ TIỀN (PAID USER)**

Ngành: Công nghệ thông tin
Mã số: 7480201

NGƯỜI HƯỚNG DẪN ThS. Trương Xuân Nam

HÀ NỘI, NĂM 2022



CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc



NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ và tên sinh viên: Tạ Hồng Quân

Hệ đào tạo: Đại học chính quy

Lớp: 60TH1

Ngành: Công nghệ thông tin

Khoa: Công nghệ thông tin

1- TÊN ĐỀ TÀI: XÂY DỰNG MÔ HÌNH DỰ ĐOÁN KHÁCH HÀNG TIỀM NĂNG
TRỞ THÀNH KHÁCH HÀNG TRẢ TIỀN (PAID USER)

2- CÁC TÀI LIỆU CƠ BẢN:

- [1] iRender Việt Nam: Khát vọng làm chủ công nghệ
- [2] Từ Minh Phương, Giáo trình “*Nhập môn trí tuệ nhân tạo*” 2014 Trường Đại học Công Nghệ Bưu Chính Viễn Thông
- [3] John M. Zelle, Python Programming: An Introduction to Computer Science
- [4] Streamlit Cloud - Streamlit Docs - Streamlit documentation
- [5] Wes McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Ipython
- [6] Microsoft SQL documentation Learn how to use SQL Server and Azure SQL, both on-premises and in the cloud
- [7] Documentation GitHub

3 - NỘI DUNG CÁC PHẦN THUYẾT MINH VÀ TÍNH TOÁN: Tỷ lệ %

Nội dung cần thuyết trình	Tỷ lệ %
Chương 1: Tổng quan về đề tài	5%
Chương 2: Học máy và một số thuật toán phân lớp	30%
Chương 3: Mô hình dự đoán và web app	35%
Chương 4: Thực nghiệm và đánh giá	30%

4 - GIÁO VIÊN HƯỚNG DẪN TỪNG PHẦN

Phần nội dung	Họ tên giáo viên hướng dẫn
Chương 1: Tổng quan về đề tài	ThS. Trương Xuân Nam
Chương 2: Học máy và một số thuật toán phân lớp	ThS. Trương Xuân Nam
Chương 3: Mô hình dự đoán và web app	ThS. Trương Xuân Nam
Chương 4: Thực nghiệm và đánh giá	ThS. Trương Xuân Nam

5 - NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Ngày tháng năm 20

Trưởng Bộ môn
(Ký và ghi rõ Họ tên)

Giáo viên hướng dẫn chính
(Ký và ghi rõ Họ tên)

Nhiệm vụ Đồ án tốt nghiệp đã được Hội đồng thi tốt nghiệp của Khoa thông qua

Ngày. . . . tháng. . . . năm 20

Chủ tịch Hội đồng
(Ký và ghi rõ Họ tên)

Sinh viên đã hoàn thành và nộp bản Đồ án tốt nghiệp cho Hội đồng thi ngày...
tháng... năm 20

Sinh viên làm Đồ án tốt nghiệp
(Ký và ghi rõ Họ tên)



TRƯỜNG ĐẠI HỌC THỦY LỢI
KHOA CÔNG NGHỆ THÔNG TIN

BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP

TÊN ĐỀ TÀI: XÂY DỰNG MÔ HÌNH DỰ ĐOÁN KHÁCH HÀNG TIỀM NĂNG
TRỞ THÀNH KHÁCH HÀNG TRẢ TIỀN (PAID USER)

Sinh viên thực hiện: TẠ HỒNG QUÂN

Lớp: 60TH1

Giáo viên hướng dẫn: Ths. TRƯƠNG XUÂN NAM

TÓM TẮT ĐỀ TÀI

Hiện nay, công ty iRender Việt Nam đang cung dịch vụ cho phép khách hàng thuê và sử dụng các server, GPU Server để render dữ liệu, hình ảnh. Với sự lớn mạnh nhanh chóng, công ty đang phục vụ một lượng khách hàng lớn. Sự tăng trưởng nhanh của lượng người đăng ký mới và sử dụng dịch vụ, bộ phận chăm sóc khách hàng chưa thể quan tâm và hỗ trợ người dùng một cách tốt ưu và hiệu quả, đem lại năng suất cũng như độ chính xác cao. Do vậy, một trong những mục tiêu hiện tại là làm sao vừa chăm sóc được các khách hàng thân thiết, vừa biết trước những khách hàng nào sẽ trở thành khách hàng nạp tiền để sử dụng dịch vụ, giúp gia tăng khả năng trở thành khách hàng thân thiết của những khách mới sử dụng dịch vụ. Một trong những chiến lược hàng đầu là sử dụng các kỹ thuật khai phá dữ liệu vào hoạt động chăm sóc, dự đoán khách hàng.

Với nhu cầu của công ty, em sẽ lựa chọn giải quyết *bài toán dự đoán khách hàng tiềm năng trở thành khách hàng trả tiền (PAID USER)*. Với việc sử dụng học máy kết hợp với khai thác dữ liệu và sử dụng ngôn ngữ python để lập trình. Sản phẩm cuối cùng là đưa mô hình dự đoán khách hàng và tích hợp đưa lên web app

CÁC MỤC TIÊU CHÍNH

Các mục tiêu chính của đề tài:

- Sử dụng SQL Server để lấy thông tin, dữ liệu cần thiết cho bài toán.
- Tìm hiểu và sử dụng các thuật toán trong học máy để triển khai bài toán.
- Dùng thành thạo ngôn ngữ lập trình Python trong học máy, sử dụng thư viện streamlit trong Python để tạo web app.
- Deploy web app lên internet sử dụng streamlit.

KẾT QUẢ DỰ KIẾN

- Nắm bắt được công nghệ, các thuật toán machine learning về phân lớp dữ liệu ứng dụng vào xây dựng mô hình dự đoán.
- Xây dựng thành công mô hình dự đoán khách hàng tiềm năng sử dụng thuật toán tốt nhất với dữ liệu bài toán
- Phát triển được thành công web app giúp người dùng có thể thao tác, tương tác với mô hình dự đoán.
- Deploy web app lên internet.

LỜI CAM ĐOAN

Em xin cam đoan đây là Đồ án tốt nghiệp của bản thân em. Các kết quả trong Đồ án tốt nghiệp này là trung thực, và không sao chép từ bất kỳ một nguồn nào và dưới bất kỳ hình thức nào. Việc tham khảo các nguồn tài liệu (nếu có) đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo đúng quy định.

Sinh viên thực hiện

Chữ ký

Tạ Hồng Quân

LỜI CẢM ƠN

Em xin trân trọng cảm ơn ThS. Trương Xuân Nam đã đưa ra những lời khuyên, chỉ bảo trong quá trình thực hiện Đồ án, giúp em định hình được mình cần phải làm gì. Sự chỉ bảo của thầy giúp em có thể nhanh chóng hoàn thiện báo cáo cũng như sản phẩm cuối cùng.

Em xin cảm ơn anh Dương Văn Phụng – cựu sinh viên K58 khoa Kỹ thuật phần mềm đã giúp đỡ em về ý tưởng, kế hoạch và thực hiện Đồ án. Với sự hướng dẫn của anh em đã có thể hoàn thiện Đồ án tốt nghiệp.

Em xin cảm ơn công ty iRender đã hỗ trợ em giúp em có một môi trường làm việc tốt nhất. Được làm việc trong công ty giúp em có thể hoàn thành được Đồ án dưới sự chỉ bảo và quan tâm nhắc nhở của các thành viên trong công ty.

Em xin chân thành cảm ơn trường Đại học Thủy Lợi nói chung và khoa Công nghệ thông tin nói riêng, đã luôn tạo điều kiện tốt nhất về cả cơ sở vật chất và giảng dạy, hỗ trợ sinh viên một cách tối đa. Từ đó, sinh viên có cơ hội học tập, phấn đấu, trau dồi kiến thức trên trường lớp lẫn thực hành trong thực tiễn nhằm tạo ra những con người có ích cho đất nước, xã hội trong tương lai.

Em xin chân thành cảm ơn!

MỤC LỤC

MỤC LỤC.....	iii
DANH MỤC HÌNH ẢNH	v
DANH MỤC BẢNG BIỂU	vii
DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ....	viii
CHƯƠNG 1 TỔNG QUAN VỀ ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Phát biểu bài toán dự đoán	3
1.3 Mục tiêu đề tài	3
1.4 Ý nghĩa của đề tài	4
CHƯƠNG 2 HỌC MÁY VÀ MỘT SỐ THUẬT TOÁN PHÂN LỚP PHỔ BIẾN	5
2.1 Giới thiệu học máy	5
2.1.1 Khái niệm học máy (Machine learning)	5
2.1.2 Một số phương pháp học máy.....	5
2.1.3 Một số ứng dụng của học máy	6
2.2 Các thuật toán phân lớp nhị phân phổ biến	7
2.2.1 Phương pháp cây quyết định.....	7
2.2.2 Phương pháp kNN (k láng giềng gần nhất)	9
2.2.3 Phương pháp SVM.....	11
2.2.4 Phương pháp hồi quy logistic	14
2.2.5 Phương pháp đánh giá mô hình.	20
2.3 Các công cụ hỗ trợ cho bài toán	20
2.3.1 Ngôn ngữ lập trình Python.....	20

2.3.2	SQL Server.....	28
2.3.3	Git và Github.....	30
2.3.4	Google Colaboratory	33
2.3.5	Streamlit Cloud	35
CHƯƠNG 3 MÔ HÌNH DỰ ĐOÁN VÀ WEB APP		37
3.1	Tập dữ liệu và tiền xử lý dữ liệu	37
3.2	Xây dựng mô hình dự đoán	40
3.3	Xây dựng web app	41
CHƯƠNG 4 THỰC NGHIỆM VÀ ĐÁNH GIÁ.....		45
4.1	Môi trường thực nghiệm.....	45
4.2	Kịch bản thực hiện.....	45
4.3	Kết quả thực nghiệm và đánh giá	46
4.4	Kết quả xây dựng web app	51
4.4.1	Giao diện web app.....	51
4.4.2	Kiểm thử web app	58
KẾT LUẬN		61
TÀI LIỆU THAM KHẢO.....		62

DANH MỤC HÌNH ẢNH

Hình 1.1 iRender nhận giải Sao Khuê 2021 cho 2 lĩnh vực Sản phẩm giải pháp khởi nghiệp số và Điện toán đám mây & Big Data	1
Hình 2.1 Machine learning	5
Hình 2.2 Phát triển trợ lý cá nhân.....	6
Hình 2.3 Ví dụ minh họa trợ lý ảo Google Assistant hỗ trợ trên thiết bị di động	7
Hình 2.4 Mô tả thuật toán cây quyết định	8
Hình 2.5 Kết quả phân loại của k-NN với $k = 3$ và $k = 5$	11
Hình 2.6 Siêu phẳng H chia dữ liệu huấn luyện thành 2 lớp với khoảng cách biên lớn nhất (các điểm gần H nhất nằm trên H1 và H2 là vector hỗ trợ)	13
Hình 2.7 Hồi quy tuyến tính với một đặc trưng đầu vào.....	15
Hình 2.8 Đồ thị hàm logit.....	16
Hình 2.9 Giá trị β đạt giá trị cực đại.....	19
Hình 2.10 Logo python sử dụng từ những năm 1990 đến 2006.....	21
Hình 2.11 Biểu tượng của NumPy	22
Hình 2.12 Pandas làm việc với DataFrame	23
Hình 2.13 Sklearn với Python	24
Hình 2.14 Các loại biểu đồ trong Matplotlib.....	24
Hình 2.15 Trường region ở dạng chữ	25
Hình 2.16 Dữ liệu trường region sau khi encoder	26
Hình 2.17 Streamlit	27
Hình 2.18 Một web app đơn giản sử dụng thư viện Streamlit	28
Hình 2.19 SQL.....	29
Hình 2.20 Logo SQL Server.....	30
Hình 2.21 Chỗ để lấy địa chỉ repository	32
Hình 2.22 Logo của Colap.....	33
Hình 2.23 Deploy web app với Streamlit Cloud	35
Hình 2.24 Chia sẻ web app qua URL	36
Hình 2.25 Cài đặt người có thể truy cập vào ứng dụng.....	36
Hình 3.1 Câu truy vấn trên SQL Server	38
Hình 3.2 Dữ liệu trước khi đưa vào huấn luyện	39

Hình 3.3 Tương quan dữ liệu của hai lớp nhãn dân	39
Hình 3.4 Mô hình chọn lựa thuật toán để xây dựng	40
Hình 3.5 Đăng nhập streamlit.io.....	42
Hình 3.6 Đăng nhập với Github	42
Hình 3.7 Màn hình làm việc của Streamlit Cloud	43
Hình 3.8 Kết nối Streamlit với kho lưu trữ dự án trên Github	43
Hình 3.9 Triển khai web app	44
Hình 3.10 Triển khai thành công web app	44
Hình 4.1 Biểu đồ kết quả thực nghiệm thuật toán cây quyết định	47
Hình 4.2 Biểu đồ kết quả thực nghiệm thuật toán kNN	48
Hình 4.3 Biểu đồ kết quả thực nghiệm thuật toán SVM	49
Hình 4.4 Biểu đồ kết quả thực nghiệm thuật toán hồi quy logistic	50
Hình 4.5 Giao diện tổng thể web app với người triển khai	51
Hình 4.6 Giao diện người được chia sẻ web app	51
Hình 4.7 Phân nhập dữ liệu đầu vào.....	52
Hình 4.8 Danh sách quốc gia đầu vào của bài toán.....	53
Hình 4.9 Giao diện thông tin timezone	53
Hình 4.10 Giao diện phân nhập ngôn ngữ.....	54
Hình 4.11 Giao diện phân nhập gói dịch vụ.....	54
Hình 4.12 Giao diện phân số giờ sử dụng lần đầu	55
Hình 4.13 Giao diện phân nhập dung lượng ổ Z.....	55
Hình 4.14 Giao diện hiện thị thông tin người dùng nhập vào	55
Hình 4.15 Giao diện kết quả dự đoán.....	55
Hình 4.16 Giao diện xác suất dự đoán của hai lớp.....	56
Hình 4.17 Giao diện dữ liệu bài toán	56
Hình 4.18 Giao diện phần log của web app.....	57
Hình 4.19 Kết quả kiểm thử trường hợp 1 và khách hàng 1	58
Hình 4.20 Kết quả kiểm thử trường hợp 1 và khách hàng 2	59
Hình 4.21 Kết quả kiểm thử trường hợp 2 và khách hàng 1	60
Hình 4.22 Kết quả kiểm thử trường hợp 2 và khách hàng 2	60

DANH MỤC BẢNG BIỂU

Bảng 4.1 Kết quả thực nghiệm sử dụng thuật toán cây quyết định.....	46
Bảng 4.2 Kết quả thực nghiệm sử dụng thuật toán kNN.....	47
Bảng 4.3 Kết quả thực nghiệm của thuật toán SVM.....	48
Bảng 4.4 Kết quả thực nghiệm của thuật toán hồi quy logistic.....	49
Bảng 4.5 Dữ liệu trường hợp 1 cho kiểm tra web app	58
Bảng 4.6 Dữ liệu trường hợp 2 cho kiểm tra web app	59

DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ

Từ viết tắt	Ý nghĩa
Colab	Colaboratory
CPU	Central Processing Unit (bộ xử lý trung tâm)
CSDL	Cơ sở dữ liệu
ETL	Extract - Transform – Load (trích xuất – biến đổi – tải)
FIRCA	Viện Nghiên cứu về Khoa học Máy tính và Tự động hóa
FPT	Công ty Cổ phần FPT
GPU	Graphics Processing Unit (bộ xử lý tác vụ đồ họa)
kNN	KnearestNeighbors
ML	Machine Learning
MLE	Maximum Likelihood estimation
PC	Personal Computer (máy tính)
SQL	Structured Query Language (ngôn ngữ truy vấn cấu trúc dữ liệu)
SVM	Support vector machines
URL	Uniform Resource Locator
Viettel	Tập đoàn Công nghiệp – Viễn thông Quân đội (Viettel)

CHƯƠNG 1 TỔNG QUAN VỀ ĐỀ TÀI

1.1 Đặt vấn đề

iRender là công ty công nghệ tiên phong tại Việt Nam cung cấp 2 dịch vụ điện toán đám mây chính là Cloud Rendering cho lĩnh vực đồ họa 3D, kiến trúc và GPU Cloud cho lĩnh vực AI /Deep Learning. Giải pháp được tạo ra nhằm giải quyết các vấn đề về cơ sở hạ tầng công nghệ cho những cá nhân, doanh nghiệp cần tận dụng cấu hình máy chủ mạnh mẽ để thực hiện các tác vụ đòi hỏi tính toán hiệu suất lớn. Hiện tại, iRender đang phục vụ hơn 30 nghìn khách hàng đến từ hơn 100 quốc gia trên thế giới, tăng trưởng bình quân theo quý đạt trên 70%. Với mục tiêu “**Bình dân hóa điện toán đám mây**”, iRender mang tới cho người dùng một dịch vụ tiện lợi, dễ dàng sử dụng, với mức chi phí bình dân, nhưng dịch vụ đạt tiêu chuẩn chất lượng quốc tế. Giải pháp của iRender giúp khách hàng tiết kiệm ít nhất 50% chi phí và 80% thời gian kết xuất 3D hay đào tạo mô hình trí tuệ nhân tạo, học sâu.[1]



Hình 1.1 iRender nhận giải Sao Khuê 2021 cho 2 lĩnh vực Sản phẩm giải pháp khởi nghiệp số và Điện toán đám mây & Big Data

Ngày nay, các công ty kinh doanh đang ngày càng chú tâm đến việc phân tích dữ liệu, đưa ra các báo cáo thống kê, đánh giá về khách hàng để đưa ra những quyết cải thiện khả năng ra quyết định và thúc đẩy tăng trưởng doanh nghiệp. Việc phân tích và thấu

hiểu những dữ liệu sẵn có sẽ giúp doanh nghiệp hiểu được các sai lầm trong quá khứ và tìm ra phương hướng giải quyết, đồng thời khám phá ra những cơ hội mới để phát triển doanh nghiệp. Cụ thể, các khu vực trong phân tích bao gồm phân tích dự đoán, phân tích theo quy tắc, quản lý quyết định doanh nghiệp, phân tích mô tả, phân tích nhận thức, phân tích bán lẻ, phân loại cửa hàng và tối ưu hoá lưu trữ đơn vị hàng hóa tồn kho, tối ưu hoá tiếp thị và các mô hình tiếp thị kết hợp, phân tích web, phân tích cuộc gọi, phân tích giọng nói, nhân lực bán hàng và tối ưu hoá, mô hình định giá bán và khuyến mãi, khoa học dự đoán, phân tích rủi ro tín dụng và phân tích gian lận.

Một trong những bài toán mà doanh nghiệp nào cũng cần đó là phân loại phân khúc khách hàng và quan trọng hơn nữa là dự đoán khách hàng. Thay vì đưa ra các quyết định dựa trên cảm tính hay kinh nghiệm phán đoán, việc dự đoán khách hàng dựa trên cơ sở các số liệu khoa học rút ra từ dữ liệu đã thu thập được giúp doanh nghiệp có thể đưa ra được những quyết định đúng đắn hơn, chính xác và hiệu quả hơn. Việc phân loại phân khúc khách hàng giúp cho các chiến dịch tiếp thị tốt hơn với từng khách hàng, có các ý tưởng rõ ràng về tối tượng khách hàng để biết được họ muốn gì khi sử dụng các dịch vụ của doanh nghiệp. Giúp tinh chỉnh và tối ưu hóa các dịch vụ để đáp ứng nhu cầu của từng loại khách hàng. Vì dành ít thời gian, nguồn lực vào nỗ lực tiếp thị và chăm sóc khách hàng vào các phân khúc khách hàng ít sinh lời và dành thêm thời gian phát triển dịch vụ, chăm sóc vào các phân khúc khách hàng thành công nhất của doanh nghiệp. Kết quả nó làm tăng doanh thu, lợi nhuận cũng như giảm chi phí cho doanh nghiệp.

Là một công ty công nghệ tiên phong, iRender rất quan tâm đến việc phân tích, khai phá dữ liệu và đặc biệt là tập dữ liệu về khách hàng. Sự tăng trưởng nhanh chóng của lượng người đăng ký mới và có khả năng nạp tiền để sử dụng dịch vụ. iRender muốn chăm sóc tập khách hàng mới một cách thông minh nhất, tiết kiệm tài nguyên, nhân lực, cải thiện dịch vụ,... Do vậy để đáp ứng nhu cầu trên, trong đề tài này em sẽ đưa ra một phương pháp dự đoán loại khách hàng bằng việc giải quyết bài toán ***“Xây dựng mô hình dự đoán khách hàng tiềm năng trở thành khách hàng trả tiền (Paid User)”*** sử dụng thuật toán học máy và ngôn ngữ python để hỗ trợ đưa ra mô hình dự đoán.

1.2 Phát biểu bài toán dự đoán

Tại iRender, doanh thu mang lại chủ yếu cho công ty đến từ việc cho thuê server, CPU và GPU để phục vụ nhu cầu render dữ liệu, hình ảnh, AI/Deep Learning. Việc tập chung chăm sóc những khách hàng tiềm năng có khả năng nạp tiền vào hệ thống là một mục tiêu chiến lược mà công ty hướng đến nhằm tăng doanh thu. Tuy nhiên với sự tăng trưởng vượt bậc, khách hàng đăng ký mới ngày càng nhiều dẫn đến việc chăm sóc khách hàng gặp nhiều khó khăn hơn, việc tiếp cận khách hàng đúng mục tiêu, thời điểm mà không làm ảnh hưởng tới trải nghiệm của khách hàng cần đòi hỏi một nghiên cứu chuyên sâu từ kỹ thuật khai phá dữ liệu. Việc có thể giúp các nhân viên chăm sóc khách hàng có thể tập chung chăm sóc những khách hàng tiềm năng có thể nạp tiền vào hệ thống để sử dụng dịch vụ. Với các phân tích trên em lựa chọn bài toán dự đoán khách hàng tiềm năng trở thành khách hàng trả tiền làm đề tài nghiên cứu. Bài toán được phát biểu cụ thể như sau:

- Đầu vào:

Thông tin của khách hàng bao gồm thông tin về quốc gia, ngôn ngữ, múi giờ, gói sử dụng, số giờ sử dụng, tổng dung lượng ổ Z sử dụng.

Trạng thái của khách hàng đang là khách hàng miễn phí hay khách hàng trả tiền.

- Đầu ra:

Mô hình dự đoán khách hàng có là khách hàng tiềm năng trở thành khách hàng trả tiền hay không.

Với dữ liệu bài toán, trạng thái khách hàng là khách hàng miễn phí hay khách hàng trả tiền. Chúng ta có thể thấy hướng tiếp cận giải quyết bài toán chính là phân lớp dữ liệu, cụ thể hơn là bài toán phân lớp nhị phân.

1.3 Mục tiêu đề tài

Mục tiêu của đề tài là nghiên cứu các thuật toán học máy ứng dụng cho việc xây dựng mô hình dự đoán khách hàng. Với các mục tiêu chính:

- Xây dựng mô hình dự đoán khách hàng tiềm năng trở thành khách hàng trả tiền.
Phân tích đánh giá mô hình dự đoán
- Dùng web app sử dụng mô hình đã xây dựng để tương tác, đưa ra những dự đoán với dữ liệu mà người sử dụng đưa vào.
- Sử dụng Github và Streamlit Cloud để triển khai web app.

1.4 Ý nghĩa của đề tài

Đề tài có ý nghĩa quan trọng, mang tính chất quyết định hướng đi của doanh nghiệp. Góp phần giúp giúp doanh nghiệp có cái nhìn tổng quan về khách hàng, nắm rõ được thị trường, tiếp cận với các khách hàng tiềm năng, có định hướng để phát triển sản phẩm, chất lượng dịch vụ, xây dựng mối quan hệ bền vững với khách hàng. Ngoài ra, việc dự đoán khách hàng cũng giúp doanh nghiệp giảm chi phí, tiết kiệm nhân lực, xây dựng được các chiến dịch quảng bá sản phẩm tới đúng khách hàng mà mình mong muốn.

CHƯƠNG 2 HỌC MÁY VÀ MỘT SỐ THUẬT TOÁN PHÂN LỚP PHỔ BIẾN

2.1 Giới thiệu học máy

2.1.1 Khái niệm học máy (Machine learning)

Học máy là một lĩnh vực của trí tuệ nhân tạo và khoa học máy tính. Học máy liên quan đến việc nghiên cứu cung cấp cho máy tính khả năng học hỏi và xây dựng các kỹ thuật cho phép các hệ thống “học” tự động từ dữ liệu để giải quyết những vấn đề cụ thể dựa trên kinh nghiệm được đưa vào đào tạo.



Hình 2.1 Machine learning

Machine learning là một phần quan trọng của lĩnh vực khoa học dữ liệu đang rất phát triển ngày nay. Học máy ngày càng mang tính phổ biến trên toàn thế giới. Sự tăng trưởng vượt bậc của dữ liệu lớn (Big data) và các thuật toán Học máy đã cải thiện độ chính xác của những mô hình và dự đoán tương lai.

2.1.2 Một số phương pháp học máy

Dựa trên tính chất của các tập dữ liệu, các thuật toán machine learning có thể phân loại thành hai nhóm chính là: **Học có giám sát** (*Supervised learning*) và **Học không giám sát** (*Unsupervised learning*)

- Học có giám sát (*Supervised learning*)

Học có giám sát là thuật toán dự đoán đầu ra của một hoặc nhiều dữ liệu mới dựa trên các cặp (đầu vào, đầu ra) đã biết từ trước. Ứng dụng của học có giám sát chính là giúp xác định tín hiệu tốt nhất để dự báo xu hướng, lợi nhuận trong tương lai trong lĩnh vực cổ phiếu, chứng khoán. Một số thuật toán được sử dụng trong học máy có giám sát bao gồm mạng nơ-ron, Navie Bayes, hồi quy tuyến tính, hồi quy logistic, Random Forest, thuật toán SVM,...

Học không giám sát là thuật toán cho máy tính học trên dữ liệu mà không được gán nhãn, thuật toán sẽ tìm ra sự tương quan dữ liệu, mô hình hóa dữ liệu hay chính là làm cho máy tính hiểu về dữ liệu, từ đó chúng có thể phân loại các dữ liệu về sau thành các nhóm, lớp (clustering) giống nhau. Học không giám sát cũng được sử dụng để giảm số lượng tính năng trong một mô hình thông qua quá trình giảm kích thước. Một số thuật toán được sử dụng trong học không giám sát bao gồm neural network, phân cụm K-mean,...

2.1.3 Một số ứng dụng của học máy

- Nhận diện khuôn mặt/ giọng nói: Đây có thể xem là ứng dụng phổ biến nhất của Machine learning cụ thể như: điều tra, xác định tội phạm, hỗ trợ pháp y, mở khóa điện thoại,...



Hình 2.2 Phát triển trợ lý cá nhân

- Tự động nhận diện giọng nói, phát triển trợ lý cá nhân: trợ lý các nhân ảo hỗ trợ tìm kiếm thông tin thông qua văn bản, giọng nói hoặc hình ảnh ví dụ như Google Assistant, Siri ...



Hình 2.3 Ví dụ minh họa trợ lý ảo Google Assistant hỗ trợ trên thiết bị di động

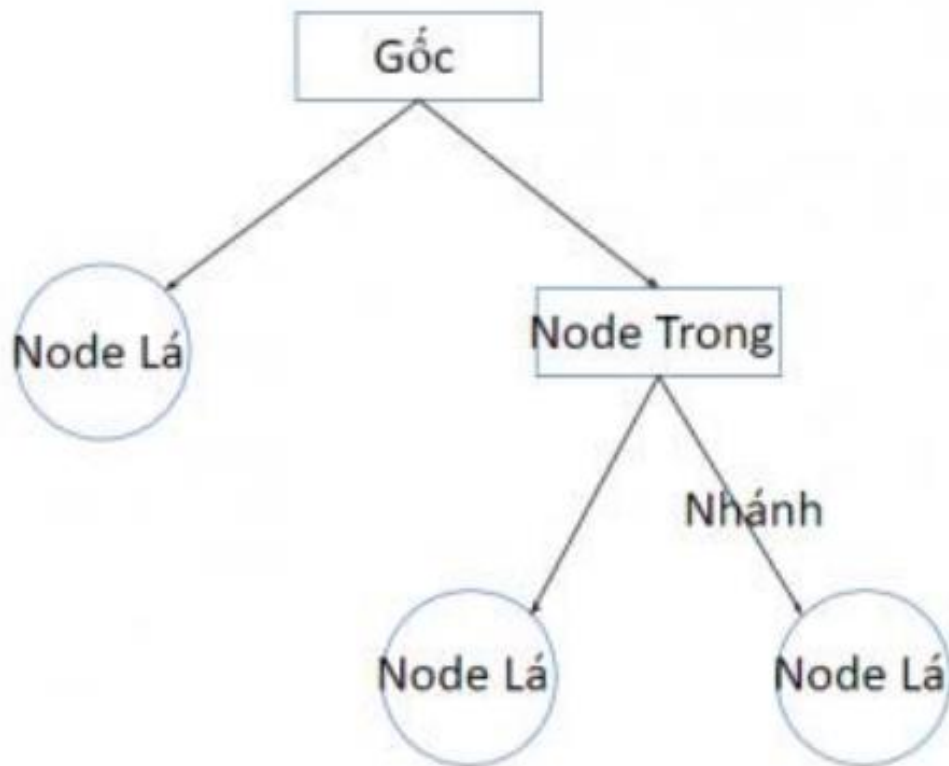
- Hệ khuyến nghị khách hàng: sử dụng dữ liệu hành vi tiêu dùng trong quá khứ, học máy giúp đưa ra các dự đoán xu hướng có thể xảy ra từ đó phát triển các chiến lược bán hàng hiệu quả hơn.
- Chăm sóc khách hàng: Chatbots hỗ trợ con người trong hành trình chăm sóc và nâng cao trải nghiệm khách hàng. Chatbots giúp trả lời những câu hỏi thường gặp của khách hàng

2.2 Các thuật toán phân lớp nhị phân phổ biến

2.2.1 Phương pháp cây quyết định

Mô hình cây quyết định là một mô hình được sử dụng khá phổ biến và hiệu quả trong cả hai lớp bài toán phân loại và dự báo của học có giám sát. Khác với những thuật toán khác trong học có giám sát, mô hình cây quyết định không tồn tại phương trình dự báo. Thay vào đó, chúng ta xây dựng một cây quyết định dự báo tốt trên tập huấn luyện và sử dụng cây quyết định này dự báo trên tập kiểm tra. Cây quyết định với 3 loại nút. Nút gốc là nút ban đầu đại diện cho toàn bộ tập dữ liệu và có thể chia thành các nút khác. Nút

trong là nút đại diện cho các nhãn dán của tập dữ liệu và các nhánh đại diện cho các quy tắc quyết định. Cuối cùng, các nút lá đại diện cho giá trị phân loại mà ta thu được.



Hình 2.4 Mô tả thuật toán cây quyết định

Thuật toán học cho cây quyết định [2]

Input: Tập dữ liệu huấn luyện

Output: Cây quyết định

Khởi đầu: Nút hiện thời là nút gốc chứa toàn bộ tập dữ liệu huấn luyện

Tại nút hiện thời n , lựa chọn thuộc tính:

- Chưa được sử dụng ở nút tổ tiên (tức là nút nằm trên đường đi từ gốc tới nút hiện thời)
- Cho phép phân chia tập dữ liệu hiện thời thành các tập con một cách tốt nhất

Với mỗi giá trị thuộc tính được chọn:

- Thêm một nút con bên dưới

- Chia các mẫu ở nút hiện thời về các nút con theo giá trị thuộc tính được chọn

Lặp (đệ quy) với mỗi nút con cho tới khi:

- Tất cả các thuộc tính đã được sử dụng ở các nút phía trên, hoặc
- Tất cả các mẫu tại nút hiện thời có cùng nhãn phân loại
- Nhãn của nút được lấy theo đa số nhãn của mẫu tại nút hiện thời

Một điểm quan trọng trong thuật toán xây dựng cây quyết định là lựa chọn thuộc tính tốt nhất tại mỗi nút. Trong trường hợp lý tưởng, thuộc tính lựa chọn là thuộc tính cho phép chia tập dữ liệu thành các tập con có cùng một nhãn, và do vậy chỉ cần một phép kiểm tra thuộc tính khi phân loại. Trong trường hợp nói chung, thuộc tính lựa chọn cần cho phép tạo ra những tập con có độ đồng nhất cao nhất. Trong khi xây dựng cây quyết định (hay bộ phân loại nói chung), thuật toán học máy thường cố gắng để cây phù hợp với dữ liệu, tức là phân loại đúng các mẫu huấn luyện, ở mức tối đa. Tuy nhiên, mục đích học cây quyết định không phải để phân loại dữ liệu mẫu, mà để phân loại dữ liệu nói chung, tức là dữ liệu mà thuật toán chưa biết trong thời gian học. Có thể xảy ra tình huống cây quyết định có độ chính xác tốt trên dữ liệu huấn luyện nhưng lại cho độ chính xác không tốt trên dữ liệu nói chung (overfitting).

2.2.2 Phương pháp kNN (k láng giềng gần nhất)

K - láng giềng gần nhất (k - nearest neighbors, viết tắt là k-NN) là phương pháp tiêu biểu nhất của học dựa trên mẫu (Instance-based learning) trong học có giám sát. Nguyên tắc của phương pháp này là đặc điểm của mẫu được quyết định dựa trên đặc điểm của k mẫu giống mẫu đang xét nhất. Ví dụ, muốn xác định nhãn phân loại, ta tìm k mẫu gần nhất và xem những mẫu này mang nhãn gì. Phương pháp k-NN thường làm việc với dữ liệu trong đó các thuộc tính được cho dưới dạng vec tơ các số thực. Như vậy, mỗi mẫu tương ứng với một điểm trong không gian O clit. Giả sử mẫu x có giá trị thuộc tính là $\langle a_1(x), a_2(x), \dots, a_n(x) \rangle$. Để xác định các mẫu giống x , cần có độ đo khoảng cách giữa các mẫu. Do mẫu tương ứng với điểm trong không gian, khoảng cách O clit thường được dùng cho mục đích này. Khoảng cách O clit giữa hai mẫu x_i và x_j được tính như sau[2]:

$$d(x_i, y_j) = \sqrt{\sum_{l=1}^n (a_l(x_i) - a_l(x_j))^2} \quad (2.1)$$

Với khoảng cách $d(x_i, x_j)$ vừa được định nghĩa, phương pháp k-NN cho hai trường hợp: phân loại và hồi quy (regression). Trong đề tài này, chúng ta chỉ xem xét trường hợp phân loại.

Mỗi mẫu x có thể nhận phân loại $f(x)$ với $f(x)$ nhận một giá trị trong tập hữu hạn các phân loại C . Thuật toán k-NN cho phân loại

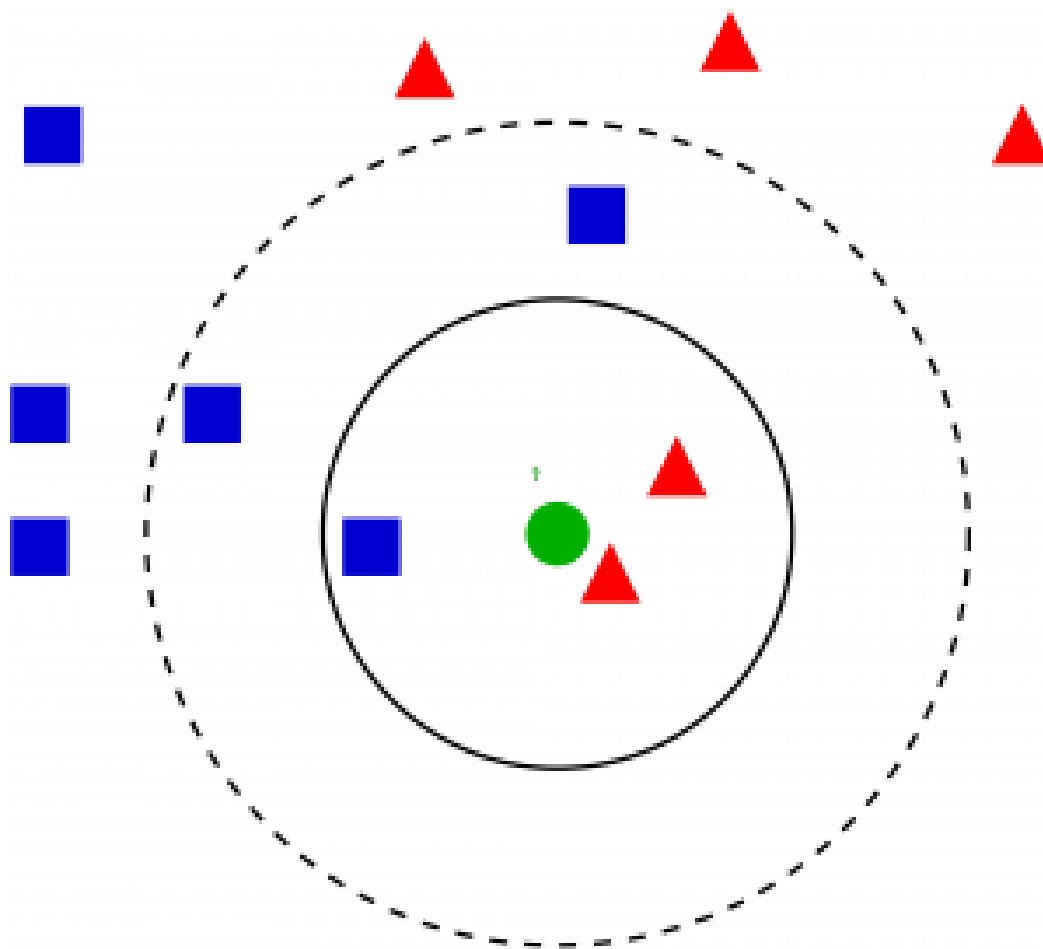
Giai đoạn học (huấn luyện)

- Lưu các mẫu huấn luyện có dạng $\langle x, f(x) \rangle$ vào cơ sở dữ liệu

Giai đoạn phân loại

- Đầu vào: tham số k
- Với mẫu x cần phân loại:
 - Tính khoảng cách $d(x, x_i)$ từ x tới tất cả mẫu x_i trong cơ sở dữ liệu
 - Tìm k mẫu có $d(x, x_i)$ nhỏ nhất, giả sử k mẫu đó là x_1, x_2, \dots, x_k .
 - Xác định nhãn phân loại $f'(x)$ là nhãn chiếm đa số trong tập $\{x_1, x_2, \dots, x_k\}$ [2]

Kết quả phân loại của thuật toán k-NN được minh họa trên dưới đây cho trường hợp phân loại hai lớp. Các hình vuông biểu diễn các mẫu huấn luyện thuộc một lớp, hình tam giác biểu diễn các mẫu huấn luyện thuộc lớp còn lại. Ví dụ cần phân loại được biểu diễn bởi hình tròn. Khoảng cách giữa các mẫu là khoảng cách Oclit trên mặt phẳng. Với $k = 3$, mẫu được phân loại thành tam giác do có 2 trong số 3 láng giềng gần nhất là tam giác, trong khi với $k = 5$, mẫu đang xét được phân loại thành hình vuông do có 3 trong số 5 láng giềng gần nhất là hình vuông. Với $k = 1$, mẫu sẽ được phân loại thành tam giác.



Hình 2.5 Kết quả phân loại của k -NN với $k = 3$ và $k = 5$

2.2.3 Phương pháp SVM

Support vector machines (SVM) là kỹ thuật học có giám sát tương đối mới, được đề xuất lần đầu vào năm 1995 bởi Vladimir N. Vapnik và được áp dụng nhiều từ khoảng cuối những năm chín mươi thế kỷ trước. SVM được đề xuất ban đầu cho bài toán phân loại nhị phân, và có thể mở rộng cho phân loại đa lớp tương tự như hồi quy logistic.

SVM dựa trên hai nguyên tắc chính:

- Thứ nhất, SVM tìm cách phân chia mẫu có nhãn khác nhau bằng một siêu phẳng sao cho khoảng cách từ siêu phẳng tới những mẫu có nhãn khác nhau là lớn nhất. Nguyên tắc này được gọi là nguyên tắc lề cực đại (max margin). Trong quá trình huấn luyện, thuật toán SVM xác định siêu phẳng có lề cực đại bằng cách giải bài toán tối ưu cho một hàm mục tiêu bậc 2.

- Thứ hai, để giải quyết trường hợp các mẫu không thể phân chia bằng một siêu phẳng, phương pháp SVM sẽ ánh xạ không gian ban đầu của các mẫu sang một không gian khác thường là có số chiều cao hơn, sau đó tìm siêu phẳng với lề cực đại trong không gian này. Để tăng tính hiệu quả khi ánh xạ, một kỹ thuật được sử dụng là kỹ thuật dùng hàm nhân (kernel function) thay cho tích có hướng của các vec tơ. [2]

Với nguyên tắc thứ nhất, ta xét bài toán phân lớp văn bản thành các lớp mẫu dương và mẫu âm:

$$D = \{(x_i, y_i) \mid i = 1, 2, \dots, N, x_i \in \mathbb{R}^n, y = \pm 1\}$$

Trong đó mẫu là các vector đối tượng được phân lớp thành các mẫu dương và âm:

- Các mẫu dương là các mẫu x_i được gán nhãn $y_i = 1$
- Các mẫu âm là các mẫu x_i được gán nhãn $y_i = -1$

Một siêu phẳng trong không gian có thể được biểu diễn như sau: $w \cdot x + b = 0$ trong

đó w là vector trọng số, $w = (w_1, w_2, \dots, w_n)$ với n là số đặc trưng, b là độ lệch.

Bộ phân lớp SVM:

$$f(x) = \text{sign}(w \cdot x + b)$$

Trong đó:

$$\text{sign}(x) = 1 \text{ nếu } x \geq 0$$

$$\text{sign}(x) = -1 \text{ nếu } x < 0$$

Nếu $f(x) = 1$ thì x thuộc về lớp dương, ngược lại nó thuộc về lớp âm.

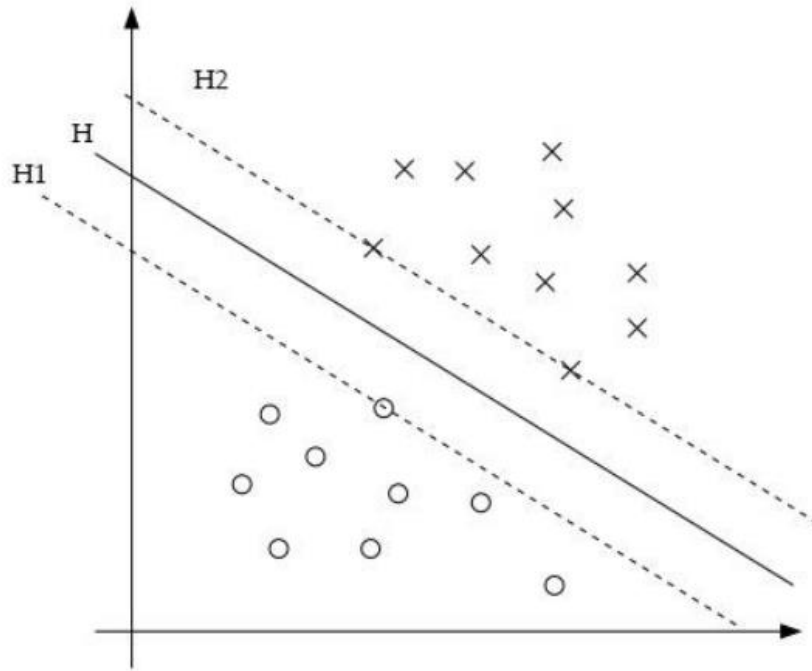
Khoảng cách từ mỗi điểm trong tập mẫu đến siêu phẳng bằng:

$$M_i = \frac{Y_i(w \cdot X_i + b)}{\|w\|} \quad (2.2)$$

Suy ra lề của siêu phẳng là:

$$M = \min_i M_i = \min_i \frac{Y_i(w \cdot X_i + b)}{\|w\|} \quad (2.3)$$

Các vector nằm trên hai siêu phẳng H_1 và H_2 song song với siêu phẳng H và cách một khoảng M gọi là vector hỗ trợ (support vector).



Hình 2.6 Siêu phẳng H chia dữ liệu huấn luyện thành 2 lớp với khoảng cách biên lớn nhất (các điểm gần H nhất nằm trên H_1 và H_2 là vector hỗ trợ)

Bài toán tìm siêu phẳng có lề lớn nhất có thể phát biểu như một bài toán tối ưu hóa $\max_{w,b,M} M$ với các ràng buộc $y_i (w \cdot x_i + b) \geq M \|w\|, \forall i = 1, \dots, N$

Với mỗi bài toán phù hợp với một dạng hàm nhân cụ thể. Sau đây là một số dạng hàm nhân thường được sử dụng với SVM:

(2.4)

Hàm đa thức: $K(x, x') = (x^T x' + 1)^d$

Trong trường hợp $d = 1$, hàm nhân gọi là hàm tuyến tính (không ánh xạ gì cả) và SVM trở thành SVM tuyến tính. Trong trường hợp nói chung, d là bậc của hàm nhân đa thức.

Hàm Gauss: $K(x, x') = \exp(-\gamma \|x - x'\|^2)$ (2.5)

trong đó $\|x\|$ là độ dài của vec tơ x : $\|x\| = \sqrt{x^T x}$ (2.6); γ là tham số thể hiện độ rộng của nhân.

(2.7)

Hàm nhân sigmoid: $K(x, x') = \tanh(\kappa x^T x + \theta)$

2.2.4 Phương pháp hồi quy logistic

Hồi quy logistic là phần mở rộng của hồi quy tuyến tính “thông thường”. Tên gọi là hồi quy nhưng thuật toán hồi quy logistic là thuật toán phân loại và được dùng cho bài toán phân loại nhị phân. Thuật ngữ logistic xuất phát từ hàm logit được dùng để biểu diễn mô hình phân loại và không liên quan tới từ logistic có nghĩa là “hậu cần”.

Để đi sâu và làm rõ được mô hình hồi quy logistic, ta cần tìm hiểu về mô hình tuyến tính là gì và tại sao hồi quy logistic lại là trường hợp mở rộng của hồi quy tuyến tính.

- Mô hình hồi quy tuyến tính

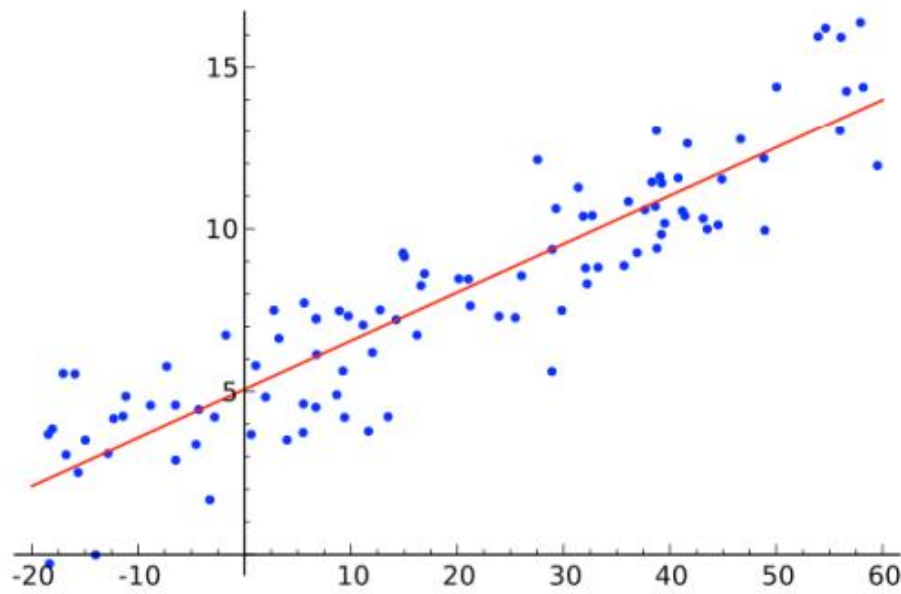
Để xây dựng mô hình học có giám sát, ta cần chọn dạng mô hình, hay dạng hàm đích, chẳng hạn như mô hình dạng cây như trong cây quyết định. Trong hồi quy tuyến tính, mô hình được sử dụng là mô hình có dạng sau:

$$h(x) = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n \quad (2.8)$$

Trong công thức này, giá trị biến đầu ra $h(x)$ được tính bằng tổng của các thuộc tính, mỗi thuộc tính được nhân với một tham số β_i . Ta có thể bổ sung tham số giả $x_0 = 1$. Khi đó, công thức của mô hình hồi quy logistic có thể viết lại thành như sau:

$$h(x) = \sum_{i=0}^n \beta_i x_i \quad (2.9)$$

Với n là số đặc trưng đầu vào. Khi $n = 1$, công thức trên trở thành phương trình đường thẳng, $n = 2$ là phương trình mặt phẳng, $n > 2$ công thức trên xác định một siêu phẳng (hyperplane)



Hình 2.7 Hồi quy tuyến tính với một đặc trưng đầu vào

- Mô hình hồi quy logistic

Trong bài toán nhị phân, hàm đích hay đầu ra của bài toán có thể nhận một trong hai nhãn dán phân loại. Hai nhãn dán này thường được ký hiệu là 0 và 1 (“không” và “có”)

Để phân loại nhị phân, ta có thể sử dụng thuật toán hồi quy tuyến tính với độ phức tạp thấp. Tuy nhiên, việc sử dụng thuật toán hồi quy tuyến tính cho phân loại nhị phân cho kết quả không được tốt trong nhiều trường hợp. Ngoài ra, thuật toán hồi quy tuyến tính có thể cho ra kết quả không mong muốn là đầu ra có thể lớn hơn 1 và nhỏ hơn 0.

Để giới hạn đầu ra cho mô hình luôn nằm trong khoảng $[0,1]$, thay vì sử dụng mô hình tuyến tính, thuật toán hồi quy logistic sử dụng mô hình sau:

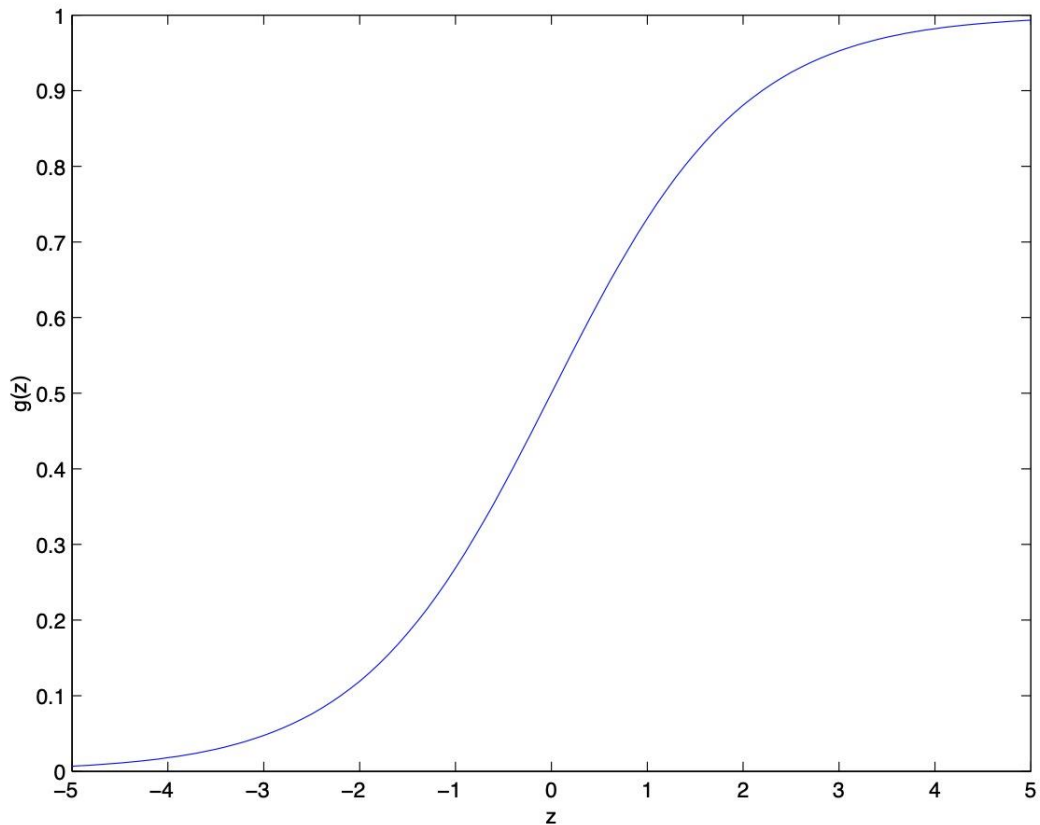
$$h(x) = g(\beta^T x) = \frac{1}{1 + e^{-\beta^T x}} \quad (2.10)$$

Trong đó:

$$\beta^T x = \sum_{i=0}^n \beta_i x_i \text{ với } x_0 = 1 \quad (2.11)$$

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

được gọi là hàm *logit* hay hàm *sigmoid*. Dạng đồ thị của hàm logit $g(z)$ như sau:



Hình 2.8 Đồ thị hàm logit

Hàm logit có hai tính chất quan trọng.

- Thứ nhất, như đồ thị trong hình trên cho thấy, giá trị của hàm logit $g(z)$ tiến tới 1 khi z tiến tới $+\infty$ và $g(z)$ tiến tới 0 khi z tiến tới $-\infty$. Bằng cách sử dụng hàm logit, giá trị hàm đích $h(x)$ luôn nằm trong khoảng $[0,1]$.
- Thứ hai, ngoài việc làm cho giá trị luôn nằm trong khoảng $[0,1]$, hàm này còn là hàm khả vi với công thức tính đạo hàm g' tương đối đơn giản:

$$g'(z) = g(z)(1-g(z)) \quad (2.12)$$

Đạo hàm này sẽ được sử dụng để tính gradient trong quá trình huấn luyện và ước lượng tham số của mô hình.

Do giá trị $h(z)$ luôn nằm trong khoảng $[0,1]$, thuật toán hồi quy logistic sử dụng giá trị này như xác suất đầu ra $y = 1$ khi biết giá trị đầu vào x và tham số β :

$$P(y = 1 | x, \beta) = h(x) = g(\beta^T x) \quad (2.13)$$

Do giá trị đầu ra chỉ có thể nhận hai giá trị là 0 hoặc 1 và tổng xác suất bằng 1 nên ta có xác suất đầu ra $y = 0$ là:

$$P(y = 0 | x, \beta) = 1 - h(x) = 1 - g(\beta^T x) \quad (2.14)$$

Với hai công thức ở trên, giá trị đầu ra dự đoán được xác định bằng cách so sánh $h(x)$ với một ngưỡng nằm ở giữa khoảng $[0,1]$, ví dụ như 0.5. nếu $h(x) > 0.5$, tương đương với xác suất của đầu ra $y = 1$ lớn hơn 50% thì nhãn đầu ra sẽ nhận giá trị bằng 1. Nếu ngược lại nhãn dán đầu ra sẽ nhận giá trị là 0. Trong một số trường hợp ta có thể tăng hoặc giảm ngưỡng này.

- **Huấn luyện mô hình hồi quy logistic**

Mục tiêu của quá trình huấn luyện cần xác định các tham số β cho phép tối ưu một hàm mục tiêu nào đó. Trong hồi quy tuyến tính ta có thể chọn hàm mục tiêu là hàm tổng bình phương lỗi, sau đó tìm được các tham số làm cho hàm này có giá trị cực tiểu bằng phương pháp phân tích hoặc phương pháp gradient giảm dần. Hàm logistic phức tạp hơn và không thể tìm được lời giải bằng phương pháp tương tự như trong hồi quy tuyến tính. Với hồi quy logistic, việc ước lượng các tham số khi huấn luyện mô hình thực hiện bằng phương pháp thủ tục ước lượng độ phù hợp tối đa - Maximum Likelihood estimation (MLE)

Độ phù hợp (likelihood) $L(\beta)$ của tham số β được định nghĩa là xác suất điều kiện của các giá trị đầu ra \mathbf{y} khi biết giá trị đầu vào \mathbf{X} và tham số β .

$$L(\beta) = P(\mathbf{y} | \mathbf{X}, \beta) \quad (2.15)$$

Khi xây dựng mô hình, ta cần xác định tham số β sao cho giá trị đầu ra của dự đoán gần với giá trị đầu ra thực tế trên tập dữ liệu huấn luyện, hay nói cách khác là xác suất giá trị nhãn dán đầu ra khi biết đầu vào \mathbf{X} và tham số β càng lớn càng tốt. Nguyên tắc xác định tham số cho độ phù hợp lớn nhất được gọi là nguyên lý độ phù hợp cực đại

Để có thể xác định được tham số β , ta triển khai công thức tính $L(\beta)$.

Viết gọn lại công thức:

$$\begin{aligned} P(y = 1 | x, \beta) &= h(x) \\ P(y = 0 | x, \beta) &= 1 - h(x) \end{aligned} \quad (2.16)$$

$$\text{thành: } P(y | x, \beta) = (h(x))^y (1 - h(x))^{1-y} \quad (2.17)$$

Do có thể coi các mẫu huấn luyện là độc lập xác suất với nhau, ta có:

$$\begin{aligned} L(\beta) &= P(\mathbf{y} | \mathbf{X}, \beta) \\ &= \prod_{i=1}^m P(y_i | x_i, \beta) \quad (\text{bằng tích xác suất cho từng mẫu huấn luyện}) \\ &= \prod_{i=1}^m (h(x_i))^{y_i} (1 - h(x_i))^{1-y_i} \end{aligned} \quad (2.18)$$

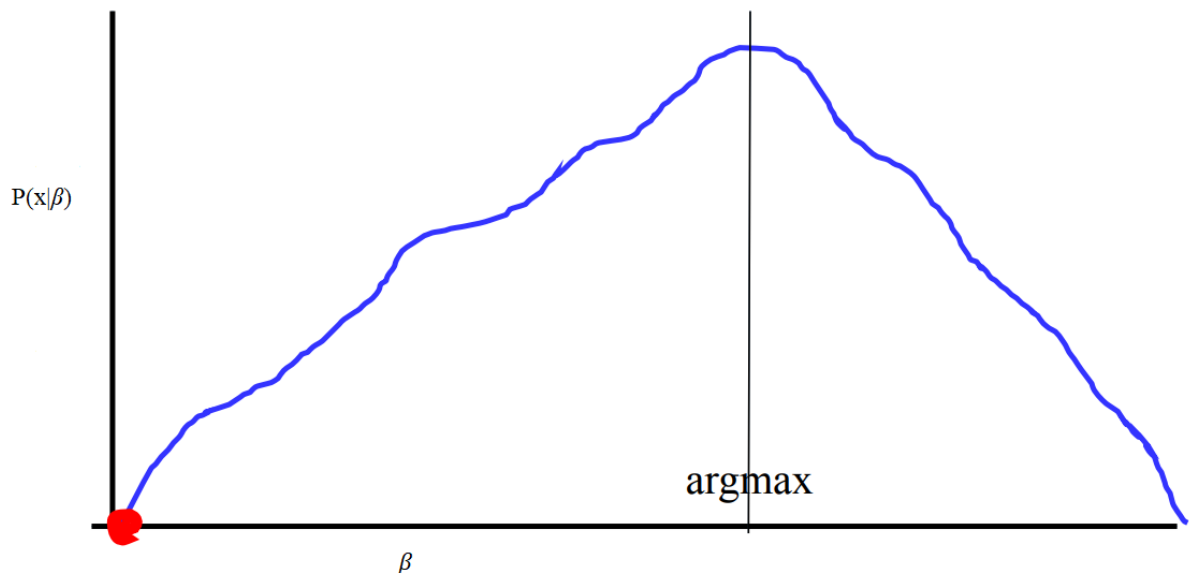
Để tiện tính toán, thay vì cực đại hóa hàm $L(\beta)$ ta cực đại hóa hàm $\log L(\beta)$ vì hàm logarit là hàm đơn điệu:

$$\begin{aligned} l(\beta) &= L(\beta) \\ &= \sum_{i=1}^m [y_i \log h(x_i) + (1 - y_i) \log(1 - h(x_i))] \end{aligned} \quad (2.19)$$

Để cực đại hóa hàm $l(\beta)$, ta có thể sử dụng hàm gradient tăng dần (gradient ascent) với công thức cập nhật tham số như sau:

$$\beta_j \leftarrow \beta_j + \alpha \frac{\partial}{\partial \beta_j} l(\beta)$$

trong đó, α là tham số có tên là tốc độ học. Giá trị α cho phép thay đổi tốc độ cập nhật tham số sau mỗi bước học của mô hình. Giá trị càng lớn thì tham số thay đổi càng nhanh, tức là càng nhanh đạt giá trị cực trị. Khác với mô hình hồi quy tuyến tính, thay vì trừ đi giá trị gradient nhân với tốc độ học, logistic lại cộng đại lượng đó vào giá trị tham số trước đó do ta cần cực đại chứ không phải cực tiểu. Do vậy, phương pháp này gọi là gradient tăng dần.



Hình 2.9 Giá trị β đạt giá trị cực đại

Để thực hiện thuật toán, ta cần tính đạo hàm riêng của l theo β_i . Để đơn giản hóa, tạm giả thiết dữ liệu huấn luyện chỉ gồm một mẫu duy nhất (x, y) . Đạo hàm riêng theo β_j tính được như sau:

$$\frac{\partial}{\partial \beta_i} l(\beta) = (y - h(x))x_j \quad (2.20)$$

Trong trường hợp dữ liệu huấn luyện nhiều mẫu, công thức trên cho phép xây dựng quy tắc tăng gradient ngẫu nhiên (stochastic gradient ascent):

$$\beta_j \leftarrow \beta_j + \alpha(y_i - h(x_i))x_{ij}$$

Và được thực hiện:

```

For( $i = 1$  to  $m$ ){
  For( $j = 1$  to  $n$ ){
     $\beta_j \leftarrow \beta_j + \alpha(y_i - h(x_i))x_{ij}$ 
  }
}
```

Trong đó m là số mẫu dữ liệu, n là số đặc trưng đầu vào.

Công thức này rất giống với công thức giảm gradient ngẫu nhiên sử dụng trong hồi quy tuyến tính. Điểm khác duy nhất là hàm $h(x)$ không còn là hàm tuyến tính của x .

Phương pháp phân loại hồi quy logistic là phương pháp phân loại theo nguyên tắc xác suất có độ chính xác tương đối cao. Tuy nhiên thời gian huấn luyện của logistic lâu hơn các phương pháp phân loại khác và đòi hỏi thuộc tính dữ liệu đầu vào phù hợp.

2.2.5 Phương pháp đánh giá mô hình.

Việc đánh giá các thuật toán phân lớp nhị phân thường sử dụng các số liệu như: precision (độ chính xác), recall (độ hồi tưởng), độ đo F-score (F1) để tính toán hiệu năng của mô hình. Với bài toán này, em cũng sử dụng các độ đo này để đánh giá độ tốt của mô hình.

- Precision (độ chính xác): Số liệu do mô hình dự đoán đúng / Tổng số dữ liệu do mô hình dự đoán ra.
- Recall (độ hồi tưởng): Số liệu do mô hình dự đoán đúng / Tổng số dữ liệu thực tế.
- F1-score: Độ hài hòa giữa độ chính xác và độ hồi tưởng. Được tính toán như sau:

$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.21)$$

2.3 Các công cụ hỗ trợ cho bài toán

2.3.1 Ngôn ngữ lập trình Python

2.3.1.1 Python

Python là một ngôn ngữ lập trình phổ biến. Nó được tạo ra bởi Guido van Rossum, và phát hành năm 1991 [3]. Python được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và học máy (ML). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.

Những lợi ích của Python bao gồm:

- Các nhà phát triển có thể dễ dàng đọc và hiểu một chương trình Python vì ngôn ngữ này có cú pháp cơ bản giống tiếng Anh.

- Python giúp cải thiện năng suất làm việc của các nhà phát triển vì so với những ngôn ngữ khác, họ có thể sử dụng ít dòng mã hơn để viết một chương trình Python.
- Python có một thư viện tiêu chuẩn lớn, chứa nhiều dòng mã có thể tái sử dụng cho hầu hết mọi tác vụ. Nhờ đó, các nhà phát triển sẽ không cần phải viết mã từ đầu.
- Các nhà phát triển có thể dễ dàng sử dụng Python với các ngôn ngữ lập trình phổ biến khác như Java, C và C++.
- Cộng đồng Python tích cực hoạt động bao gồm hàng triệu nhà phát triển nhiệt tình hỗ trợ trên toàn thế giới. Nếu gặp phải vấn đề, ta có thể nhận được sự hỗ trợ nhanh chóng từ cộng đồng.
- Trên Internet có rất nhiều tài nguyên hữu ích nếu muốn học Python. Ví dụ: ta có thể dễ dàng tìm thấy video, chỉ dẫn, tài liệu và hướng dẫn dành cho nhà phát triển.
- Python có thể được sử dụng trên nhiều hệ điều hành máy tính khác nhau, chẳng hạn như Windows, macOS, Linux và Unix.



Hình 2.10 Logo python sử dụng từ những năm 1990 đến 2006

2.3.1.2 Các thư viện của python

a. Thư viện Numpy

Numpy là một gói Python. Nó là viết tắt của “Numerical Python”. Nó là một thư viện bao gồm các đối tượng mảng đa chiều và một tập hợp các thủ tục để xử lý mảng.

Numeric, tiền thân của NumPy, được phát triển bởi Jim Hugunin. Một gói khác Numarray cũng được phát triển, có một số chức năng bổ sung. Năm 2005, Travis Oliphant đã tạo gói NumPy bằng cách kết hợp các tính năng của Numarray vào gói Numeric. Có nhiều người đóng góp cho dự án mã nguồn mở này.[5]

Sử dụng NumPy, nhà phát triển có thể thực hiện các thao tác sau:

- Các phép toán và logic trên mảng.
- Biến đổi Fourier và các thói quen để thao tác hình dạng.
- Các phép toán liên quan đến đại số tuyến tính. NumPy có các hàm dựng sẵn để tạo đại số tuyến tính và số ngẫu nhiên.



Hình 2.11 Biểu tượng của NumPy

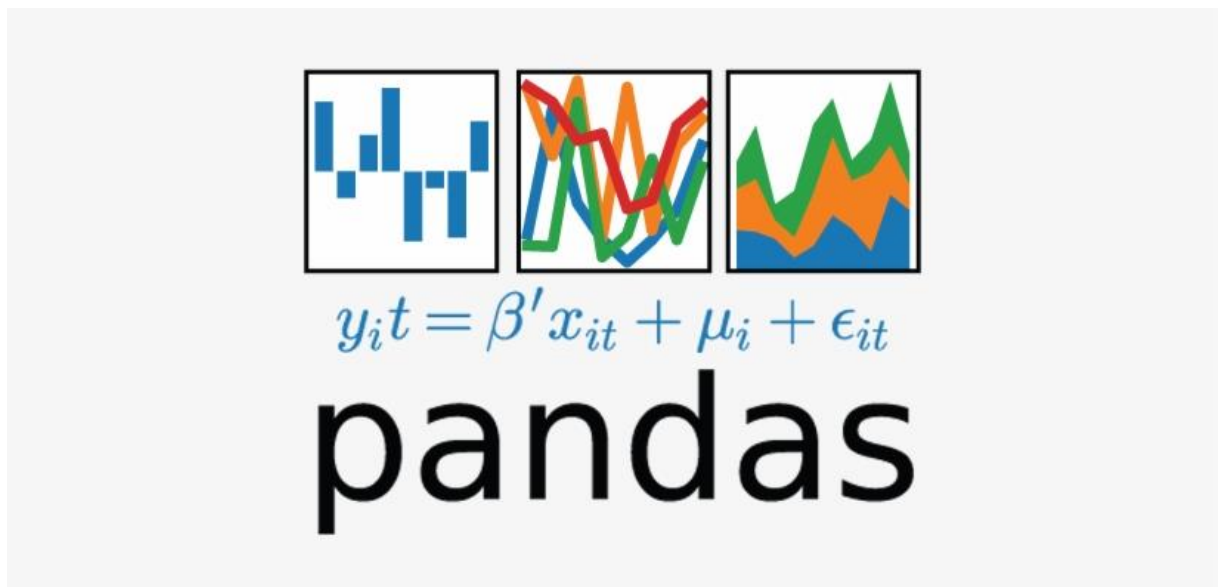
b. Thư viện Pandas

Pandas là một gói Python mã nguồn mở được sử dụng rộng rãi nhất cho các nhiệm vụ học máy/khoa học dữ liệu/phân tích dữ liệu. Nó được xây dựng trên một gói khác là Numpy, cung cấp hỗ trợ cho các mảng đa chiều. Là một trong những gói sắp xếp dữ liệu phổ biến nhất, Pandas hoạt động tốt với nhiều mô-đun khoa học dữ liệu khác bên trong hệ sinh thái Python và thường được bao gồm trong mọi bản phân phối Python.

Pandas giúp ta dễ dàng thực hiện nhiều tác vụ lặp đi lặp lại, tốn thời gian liên quan đến làm việc với dữ liệu, bao gồm:

- Dọn dẹp dữ liệu
- Điền dữ liệu
- Chuẩn hóa dữ liệu
- Hợp nhất và tham gia
- Trực quan hóa dữ liệu
- Phân tích thống kê
- Kiểm tra dữ liệu
- Tải và lưu dữ liệu

Trên thực tế, với Pandas, ta có thể làm mọi thứ khiến các nhà khoa học dữ liệu hàng đầu thế giới bình chọn Pandas là công cụ thao tác và phân tích dữ liệu tốt nhất hiện có.[5]



Hình 2.12 Pandas làm việc với DataFrame

c. Scikit – Learn (Sklearn)

Scikit-learning (Sklearn) là thư viện mạnh mẽ và hữu ích nhất dành cho máy học trong Python. Nó cung cấp một loạt các công cụ hiệu quả để học máy và lập mô hình thống kê bao gồm phân loại, hồi quy, phân cụm và giảm kích thước thông qua giao diện nhất quán trong Python. Thư viện này, phần lớn được viết bằng Python, được xây dựng dựa trên NumPy, SciPy và Matplotlib.

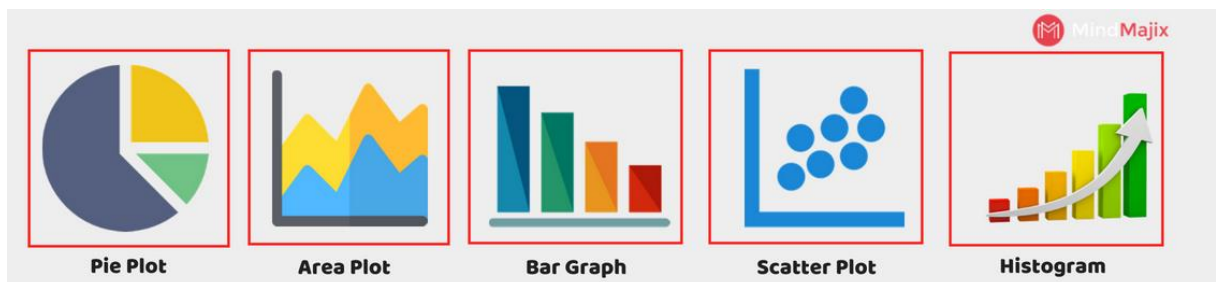
Ban đầu nó được gọi là scikits.learn và ban đầu được phát triển bởi David Cournapeau như một dự án viết mã mùa hè của Google vào năm 2007. Sau đó, vào năm 2010, Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort và Vincent Michel, từ FIRCA (Viện Nghiên cứu Pháp về Khoa học Máy tính và Tự động hóa), đã đưa dự án này lên một cấp độ khác và phát hành lần đầu ra công chúng (v0.1 beta) vào ngày 1 tháng 2 năm 2010.



Hình 2.13 Sklearn với Python

d. Thư viện Matplotlib

Matplotlib là một thư viện toàn diện để tạo các hình ảnh tĩnh, hoạt ảnh và tương tác trong Python. Matplotlib làm cho những điều khó khăn có thể trở nên dễ dàng.



Hình 2.14 Các loại biểu đồ trong Matplotlib

e. Thư viện Category Encoders

Category Encoders là một tập hợp các kỹ thuật của sklearn biến đổi các biến phân loại thành dạng số với các cách khác nhau. Mặc dù các bộ mã hóa ký tự, “one-hot” và “băm” có các giá trị tương đương trong phiên bản sklearn hiện có, các bộ chuyển đổi trong thư viện này đều có chung một số thuộc tính hữu ích như:

- Hỗ trợ đầu cho Pandas dataframe làm đầu vào (và tùy chọn làm đầu ra)

- Có thể định cấu hình rõ ràng cột nào trong dữ liệu mã hóa theo tên hoặc theo chỉ mục hoặc suy ra các cột không phải là bất kể loại đầu vào.
- Có thể bỏ bất kỳ cột nào có phương sai rất thấp dựa trên huấn luyện dữ liệu tùy chọn.
- Tính di động: đào tạo một biến đổi dữ liệu, chọn nó, tái sử dụng nó sau này và lấy ra thứ tự tương tự.
- Tương thích hoàn toàn với các đường ống sklearn, nhập vào một mảng dữ liệu giống bất kỳ một biến đổi nào khác.

Ví dụ: trường region có kiểu dữ liệu là object, vì là kiểu dữ liệu chữ nên mô hình không thể hiểu được.

0	AU
1	SA
2	ZA
3	US
4	BE
5	VN
6	NL
7	KR
8	CH
9	ID
10	VN
11	IN
12	VN
13	RU
14	JP

Hình 2.15 Trường region ở dạng chữ

Sau khi đưa qua Category Encoders, dữ liệu được biến đổi thành:

	region
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	10
10	6
11	11
12	6
13	12
14	13

Hình 2.16 Dữ liệu trường region sau khi encoder

Như ta có thể thấy, từ “VN” được chuyển thành dạng số là số 6. Sau khi “fit_transform” trường region đã được biến đổi và ghi lại, với những lần lấy giá trị tiếp theo của trường region, ta chỉ cần “transform”.

f. Thư viện Streamlit

Streamlit là một khung mã nguồn mở và miễn phí để nhanh chóng xây dựng và chia sẻ các ứng dụng web khoa học dữ liệu và máy học tuyệt đẹp. Nó là một thư viện dựa trên Python được thiết kế dành riêng cho các kỹ sư máy học. Các nhà khoa học dữ liệu hoặc kỹ sư máy học không phải là nhà phát triển web và họ không quan tâm đến việc dành hàng tuần để học cách sử dụng các khung này để xây dựng ứng dụng web. Thay vào đó, họ muốn một công cụ dễ học và dễ sử dụng hơn, miễn là nó có thể hiển thị dữ liệu và thu thập các tham số cần thiết để lập mô hình. Streamlit cho phép chúng ta tạo một ứng dụng có giao diện bắt mắt chỉ với một vài dòng mã.



Streamlit

Hình 2.17 Streamlit

Streamlit là cách dễ dàng nhất, đặc biệt đối với những người không có kiến thức về front-end để đưa mã của họ vào một ứng dụng web:

- Không yêu cầu kinh nghiệm hoặc kiến thức về front-end (html, js, css).
- Không cần phải dành nhiều ngày hoặc nhiều tháng để tạo một ứng dụng web, ta có thể tạo một ứng dụng khoa học dữ liệu hoặc máy học thực sự đẹp mắt chỉ trong vài giờ hoặc thậm chí vài phút.
- Nó tương thích với phần lớn các thư viện Python (ví dụ: pandas, matplotlib, seaborn, plotly, Keras, PyTorch, SymPy(latex)).
- Cần ít mã hơn để tạo các ứng dụng web.
- Bộ nhớ đệm dữ liệu đơn giản hóa và tăng tốc các đường ống tính toán.[5]



Hình 2.18 Một web app đơn giản sử dụng thư viện Streamlit

2.3.2 SQL Server

2.3.2.1 Ngôn ngữ SQL

SQL (**Structured Query Language**) là Ngôn ngữ truy vấn có cấu trúc, là ngôn ngữ máy tính để lưu trữ, thao tác và truy xuất dữ liệu được lưu trữ trong cơ sở dữ liệu quan hệ.

SQL là ngôn ngữ tiêu chuẩn cho Hệ thống cơ sở dữ liệu quan hệ. Tất cả các Hệ thống quản lý cơ sở dữ liệu quan hệ (RDMS) như MySQL, MS Access, Oracle, Sybase, Informix, Postgres và SQL Server đều sử dụng SQL làm ngôn ngữ cơ sở dữ liệu tiêu chuẩn của chúng.

Ngoài ra, họ đang sử dụng các phương ngữ khác nhau, chẳng hạn như:

- Máy chủ MS SQL sử dụng T-SQL,
- Oracle sử dụng PL/SQL,
- Phiên bản MS Access của SQL được gọi là JET SQL (định dạng gốc), v.v.

SQL là một trong những ngôn ngữ truy vấn được sử dụng rộng rãi nhất trên cơ sở dữ liệu. Một số ứng dụng của SQL:

- Cho phép người dùng truy cập dữ liệu trong các hệ quản trị cơ sở dữ liệu quan hệ.
- Cho phép người dùng mô tả dữ liệu.
- Cho phép người dùng xác định dữ liệu trong cơ sở dữ liệu và thao tác với dữ liệu đó.
- Cho phép nhúng vào các ngôn ngữ khác bằng các mô-đun SQL, thư viện và trình biên dịch trước.
- Cho phép người dùng tạo và thả cơ sở dữ liệu và bảng.
- Cho phép người dùng tạo dạng xem, thủ tục lưu trữ, hàm trong cơ sở dữ liệu.
- Cho phép người dùng đặt quyền trên bảng, quy trình và chế độ xem.[6]



Hình 2.19 SQL

2.3.2.2 SQL Server

SQL Server là phần mềm (Hệ quản trị cơ sở dữ liệu quan hệ) được phát triển bởi Microsoft. Nó còn được gọi là MS SQL Server. Nó được triển khai từ đặc điểm kỹ thuật của RDBMS. Hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS) là một sản phẩm phần mềm của Microsoft chủ yếu được sử dụng để lưu trữ và truy xuất dữ liệu cho cùng một ứng dụng hoặc các ứng dụng khác. Chúng ta có thể chạy các ứng dụng này trên cùng một máy tính hoặc một máy tính khác.

Sau đây là cách sử dụng chính của MS SQL Server:

- Mục đích chính của nó là xây dựng và duy trì cơ sở dữ liệu.

- Nó được sử dụng để phân tích dữ liệu bằng Dịch vụ phân tích máy chủ SQL (SSAS).
- Nó được sử dụng để tạo báo cáo bằng Dịch vụ Báo cáo Máy chủ SQL (SSRS).
- Nó được sử dụng để thực hiện các hoạt động ETL bằng Dịch vụ tích hợp máy chủ SQL (SSIS).[6]



Hình 2.20 Logo SQL Server

2.3.3 Git và Github

2.3.3.1 Git

Git là một hệ thống kiểm soát phiên bản cho phép ta theo dõi các thay đổi mà ta thực hiện đối với các tệp của mình theo thời gian. Với Git, ta có thể hoàn nguyên các trạng thái khác nhau của tệp (giống như cỗ máy du hành thời gian).Hoặc cũng có thể tạo một bản sao của tệp của mình, thực hiện các thay đổi đối với bản sao đó, rồi hợp nhất những thay đổi này với bản gốc.

Ví dụ: Có thể đang làm việc trên trang đích của một trang web và phát hiện ra rằng thanh điều hướng đang gặp vấn đề. Nhưng đồng thời, có thể không muốn bắt đầu thay đổi các thành phần của nó vì nó có thể trở nên tồi tệ hơn.

Với Git, ta có thể tạo một bản sao giống hệt của tệp đó và thao tác với thành điều hướng. Sau đó, khi đã hài lòng với những thay đổi của mình, ta hoàn toàn có thể hợp nhất bản sao vào tệp gốc.

Không bị giới hạn chỉ sử dụng Git cho các tệp mã nguồn – cũng có thể sử dụng nó để theo dõi các tệp văn bản hoặc thậm chí cả hình ảnh. Điều này có nghĩa là Git không chỉ dành cho các nhà phát triển – bất kỳ ai cũng có thể thấy nó hữu ích.[7]

2.3.3.2 Github

GitHub là một dịch vụ lưu trữ trực tuyến cho các kho lưu trữ Git. Hãy tưởng tượng rằng ta đang làm một dự án ở nhà và trong khi đi vắng, có thể là ở chỗ của một người bạn hay đi du lịch, đột nhiên nhớ ra giải pháp cho một lỗi mã khiến ta bôn chôn trong nhiều ngày.

Chúng ta không thể thực hiện những thay đổi này vì PC không ở bên người. Nhưng nếu dự án được lưu trữ trên GitHub, ta có thể truy cập và tải xuống dự án đó bằng một lệnh trên bất kỳ máy tính nào mà ta có quyền truy cập. Sau đó, ta có thể thực hiện các thay đổi của mình và đẩy phiên bản mới nhất trở lại GitHub.

Tóm lại, GitHub cho phép lưu trữ repo trên nền tảng của họ. Một tính năng tuyệt vời khác đi kèm với GitHub là khả năng cộng tác với các nhà phát triển khác từ bất kỳ vị trí nào.

- **Cách đẩy kho lưu trữ lên Github bằng Terminal:**

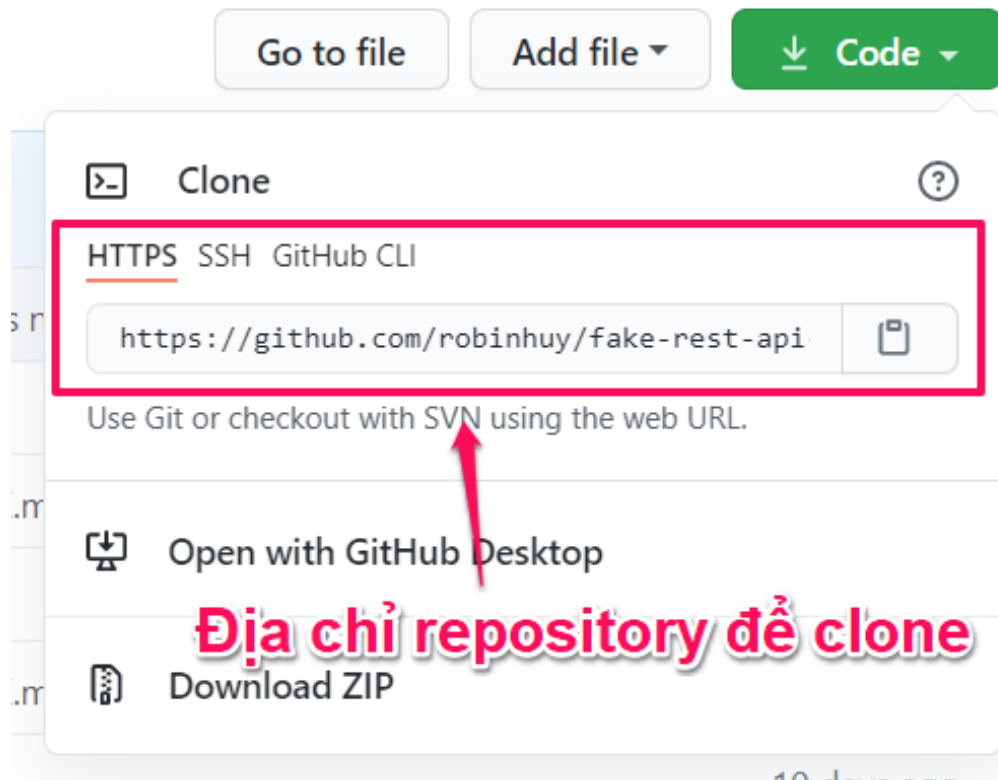
Terminal là phần mềm thao tác với máy tính qua các mã lệnh. Trên các hệ điều hành Mac, Linux thì đều có sẵn. Còn trên Windows thì ta phải cài thêm Git.

Sau khi cài xong thì chúng ta có thể thao tác với Git qua câu lệnh. Và đi kèm với Git sẽ có 1 phần mềm là Git Bash tương tự như Terminal trên Linux. Có thể bấm chuột phải vào màn hình và chọn **Git Bash Here** để bật Git Bash lên ở ngay tại thư mục hiện hành.

Một số lệnh hay dùng:

Clone 1 repository về máy:

```
git clone [địa chỉ repository]
```



Hình 2.21 Chỗ để lấy địa chỉ repository

Các lệnh sau cần vào trong thư mục chứa source code (local git repository) thì mới gõ được:

cd [thư mục chứa source code]

- Add files để chuẩn bị Commit:

git add --all (add toàn bộ các file trong project)

hoặc

git add . (add các file ở thư mục hiện tại)

- Commit:

git commit -m "Chú thích cho lần commit này"

- Push code:

git push origin main

Chú ý *main* là tên branch (nhánh) mặc định khi tạo repository, với các repository cũ thì tên nhánh mặc định là *master*.^[7]

2.3.4 Google Colaboratory

Colaboratory (viết tắt là "Colab") là một công cụ máy học và phân tích dữ liệu cho phép ta kết hợp mã Python thực thi và văn bản đa dạng thức cùng với biểu đồ, hình ảnh, HTML, LaTeX, v.v. vào một tài liệu duy nhất được lưu trữ trong Google Drive. Nó kết nối với thời gian chạy Google Cloud Platform mạnh mẽ và cho phép ta dễ dàng chia sẻ công việc của mình cũng như cộng tác với những người khác.

Colab là một dịch vụ máy tính xách tay Jupyter được lưu trữ trên máy tính, không yêu cầu thiết lập để sử dụng, đồng thời cung cấp quyền truy cập miễn phí vào các tài nguyên máy tính bao gồm cả GPU.



Hình 2.22 Logo của Colap

Một số tính năng có thể kể đến như sau:

- Tạo mục lục dựa trên các Heading viết bằng ngôn ngữ markdown giúp ta dễ dàng cấu trúc Notebook làm việc của mình. Ta cũng có thể Thu gọn (Collapse) hay Mở rộng (Expand) các phần nội dung khi soạn thảo cực kỳ tiện lợi.

- Thêm hình ảnh, biểu mẫu dễ dàng với markdown giúp trình bày báo cáo hoặc làm dashboard cực tiện lợi. Thậm chí ta có thể ẩn các dòng code để trông Notebook gọn gàng hơn với tính năng biểu mẫu.
- Kết nối dễ dàng với Google Drive, Google Sheets để bắt tay vào phân tích dữ liệu “trên mây” hoàn toàn.
- Chạy Python trên Cloud hay Local Runtime (Python trên máy tính cá nhân) đều cho trải nghiệm tốt. Ta vẫn tận dụng được tính năng tuyệt vời của Google Colab khi chạy với Python trên Local Runtime trong khi không bị Google tự động xóa dữ liệu khi kết thúc phiên làm việc như khi chạy trên Cloud.
- Tự động lưu lịch sử chỉnh sửa thành các phiên bản giúp dễ dàng khôi phục lại phiên bản gần nhất nếu cần khi gặp lỗi. Tính năng này tương tự như trên Google Sheets hay Google Docs, ta thậm chí không cần đến Github để lưu trữ các phiên bản chỉnh sửa này.
- Cho phép tìm kiếm và chèn các đoạn mã được soạn thảo sẵn trong các Template vào Notebook. Tính năng này rất hay bởi ta không cần phải mở thêm nhiều file lưu trữ để tìm lại các đoạn code mẫu mình đã biết khi cần. Workflow lập trình Python trở nên đơn giản và hiệu quả hơn rất nhiều.
- Tạo dashboard viết bằng Python và chia sẻ với team dễ dàng nếu cần tương tự như Google DataStudio nhưng linh hoạt và mạnh mẽ hơn rất nhiều.

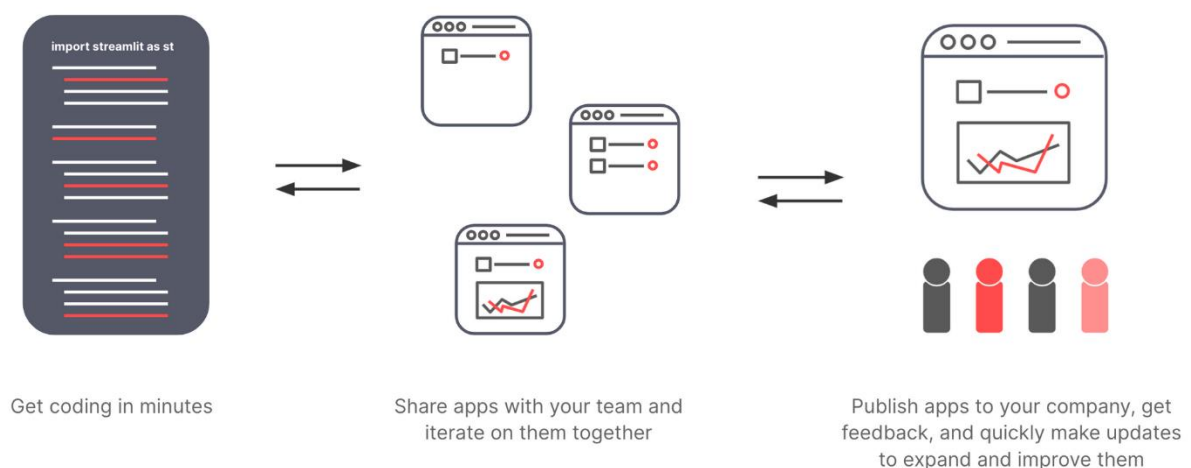
Tuy nhiên Google Colab có 1 nhược điểm là dữ liệu (bộ nhớ tạm) của phiên làm việc sẽ bị xóa sau khi ta không active trong 1 thời gian nhất định để Colab đảm bảo có thể cung cấp tài nguyên miễn phí cho nhiều người.

Do đó mỗi khi mở Google Colab, nếu cần sử dụng các thư viện của bên thứ 3 thì cần install và import lại từ đầu để có thể sử dụng. Phiên bản Colab Pro giúp khắc phục điều này nhưng hiện tại không áp dụng cho thị trường Việt Nam.

2.3.5 Streamlit Cloud

Streamlit Cloud là không gian làm việc để triển khai, quản lý và cộng tác trên các ứng dụng Streamlit. Ta kết nối trực tiếp tài khoản Streamlit Cloud của mình với kho lưu trữ GitHub (công khai hoặc riêng tư) và sau đó Streamlit Cloud khởi chạy ứng dụng trực tiếp từ mã ta đã lưu trữ trên GitHub. Hầu hết các ứng dụng sẽ khởi chạy chỉ sau vài phút và bất cứ khi nào cập nhật mã trên GitHub, ứng dụng sẽ tự động cập nhật. Điều này tạo ra một chu kỳ lặp lại nhanh chóng cho các ứng dụng đã triển khai, để các nhà phát triển và người xem ứng dụng có thể nhanh chóng tạo nguyên mẫu, khám phá và cập nhật ứng dụng.

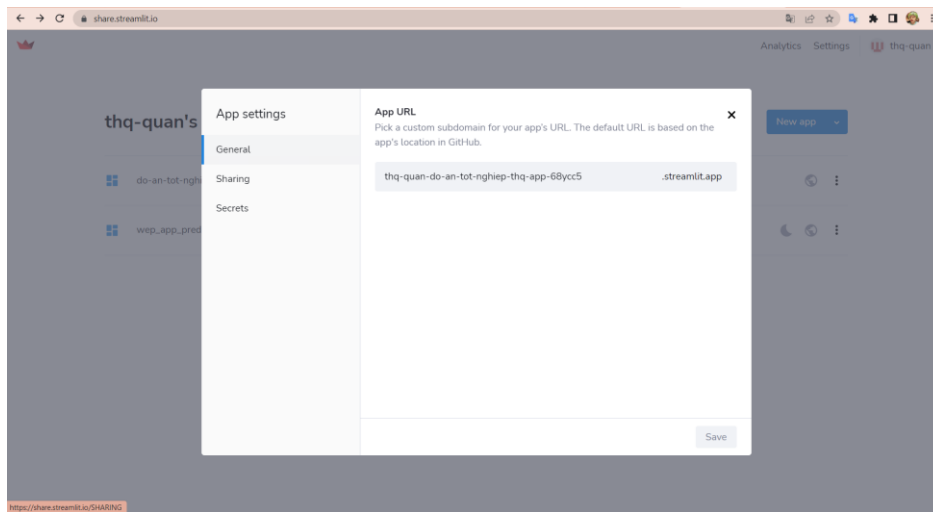
Phía dưới, Streamlit Cloud xử lý tất cả quá trình chứa, xác thực, chia tỷ lệ, bảo mật và mọi thứ khác để tất cả những gì cần làm là tạo ứng dụng, duy trì ứng dụng. Các vùng chứa nhận các bản vá bảo mật mới nhất, được theo dõi tích cực về tình trạng của các vùng chứa. Streamlit cũng đang xây dựng khả năng quan sát và giám sát các ứng dụng.[4]



Hình 2.23 Deploy web app với Streamlit Cloud

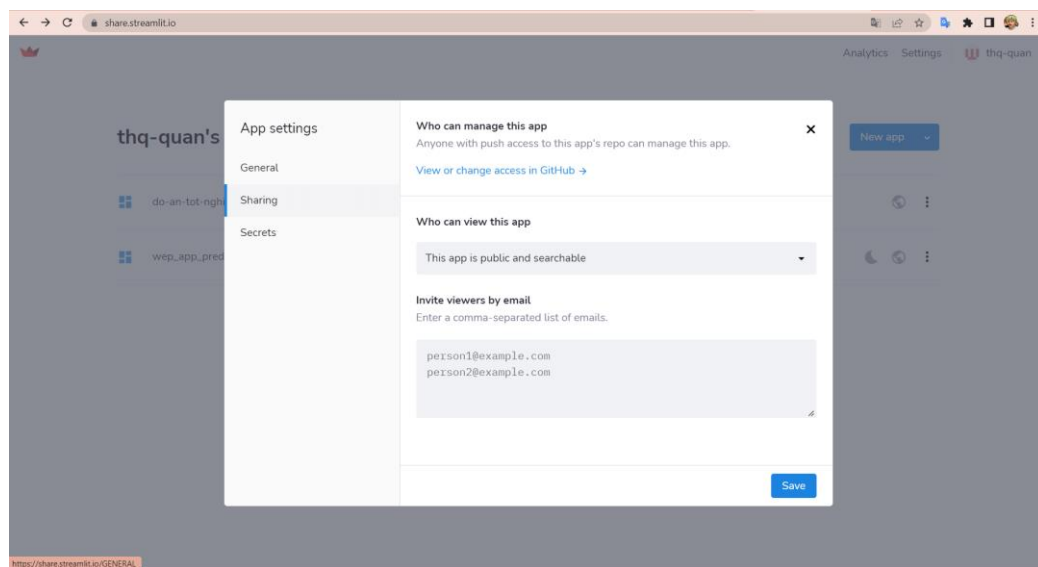
Sau khi đã hoàn tất việc tạo ứng dụng, ta có thể chia sẻ và cộng tác trên ứng dụng đó. Ứng dụng hiện tại đang hoạt động tại URL cố định đó, vì vậy ta có thể chia sẻ ứng dụng với bất kỳ ai. Ứng dụng sẽ kế thừa các quyền từ repo Github, nghĩa là nếu repo ở chế độ riêng tư thì ứng dụng sẽ ở chế độ riêng tư và nếu repo ở chế độ công khai thì ứng dụng sẽ ở chế độ công khai.[4]

Chia sẻ qua URL:



Hình 2.24 Chia sẻ web app qua URL

Cài đặt xem ai có thể có thể truy cập qua Github, mời người khác qua Email:



Hình 2.25 Cài đặt người có thể truy cập vào ứng dụng

CHƯƠNG 3 MÔ HÌNH DỰ ĐOÁN VÀ WEB APP

3.1 Tập dữ liệu và tiền sử lý dữ liệu

iRender bắt đầu triển khai dịch vụ kinh doanh và thu thập, lưu trữ thông tin của khách hàng từ tháng 6 năm 2019, hiện nay, CSDL của iRender đã lưu trữ thông tin của hơn 30 nghìn khách hàng. Thông tin này được lưu trữ trên SQL Server. Với bài toán khách hàng tiềm năng, xét các yếu tố ảnh hưởng nhiều đến việc có nạp tiền của khách hàng hay không như:

- Quốc gia: Các khách hàng tiềm năng nạp tiền chủ yếu đến từ Việt Nam và các nước phát triển từ châu Âu, châu Mỹ. Việt Nam là thị trường chính của iRender, nhắm đến nhu cầu sử dụng các CPU, GPU tốt của khách hàng là các studio, designer và nhu cầu thuê server để lưu trữ dữ liệu của các công ty lớn như FPT, Viettel,...
- Ngôn ngữ cài đặt trong ứng dụng của iRender: Việc cài đặt ngôn ngữ tương chừng không liên quan nhưng với một số ngôn ngữ được cài đặt có lượng khách hàng nạp tiền cao hơn với những ngôn ngữ khác. Ví dụ, khách hàng là người Việt Nam cài đặt ngôn ngữ là tiếng Việt có tỉ lệ nạp tiền ít hơn những khách hàng người Việt cài đặt sử dụng ngôn ngữ là tiếng Anh.
- Timezone: Việc sử dụng tiêu chí timezone khác với tiêu chí quốc gia. Người dùng có thể ở mang quốc tịch quốc gia khác và đến làm việc tại một quốc gia khác, dẫn tới việc tiêu chí quốc gia bị kém độ ảnh hưởng. Ví dụ, khách hàng là người Việt nhưng làm việc tại Nhật Bản, tuy quốc gia là Việt Nam nhưng múi giờ là của Nhật Bản.
- Gói sử dụng: iRender chia dịch vụ của mình thành từng gói cho khách hàng lựa chọn. Với từng gói dịch vụ, nhu cầu của khách hàng và khả năng chi trả cho dịch vụ đó là khác nhau.
- Số giờ sử dụng lần đầu: Khi khách hàng đăng ký tài khoản tại iRender, mỗi tài khoản sẽ có 5\$ miễn phí để trải nghiệm sản phẩm và dịch vụ. Với những khách hàng tiềm năng, có thể họ sẽ sử dụng hết luôn 5\$ và nạp luôn để tiếp tục sử dụng

dịch vụ. Nhưng với khách hàng chỉ muốn xem xét dịch vụ thì họ chỉ sử dụng chưa hết 5\$ và số giờ sử dụng dưới 1 giờ đồng hồ.

- Dung lượng ổ Z sử dụng: Sau khi đăng ký và bắt đầu thuê dịch vụ, khách hàng có thể tải lên những dữ liệu của mình lên ổ Z, có thể là hình ảnh, video, dữ liệu, triển khai các mô hình học máy, hoặc các phần mềm ứng dụng phục vụ nhu cầu sử dụng. Việc sử dụng nhiều hay ít dung lượng có thể cho biết mức độ công việc và nhu cầu của khách hàng. Những khách hàng có dung lượng cao thường đa phần là khách hàng tiềm năng và nạp tiền vào để sử dụng.

Với những tiêu chí đã chọn lọc được từ CSDL. Phần dữ liệu này được lấy từ hai server là IRENDER_DATA và IRENDER_RENTAL. Để lấy dữ liệu ra, ta sử dụng câu truy vấn trong SQL Server như sau:

```
01. ;WITH cte AS
02. (
03.     SELECT USER_ID, [VM_PACKAGE_ID],[START_TIME],[END_TIME],[MACHINE_ID],
04.           ROW_NUMBER() OVER (PARTITION BY USER_ID ORDER BY USER_ID,[TRANS_AT] ASC) AS rn
05.     FROM IRENDER_RENTAL.dbo.[RENTAL_BILLING]
06. )
07. SELECT
08.     a.region,
09.     a.timezone,
10.     a.language,
11.     VM_PACKAGE_ID,
12.     DATEDIFF(hh, cte.START_TIME, cte.END_TIME) AS hours_use,
13.     b.VHDX_LENGTH/1000000000 as sum_length,
14.     a.is_paid
15. from [IRENDER_DATA].[dbo].[users] as a, cte, [IRENDER_RENTAL].[dbo].[VM_MACHINE] as b
16. where rn = 1 and cte.USER_ID = a.id and cte.MACHINE_ID = b.ID
17. and a.region is not null
18. and a.timezone is not null
19. and a.language is not null
20. and cte.VM_PACKAGE_ID is not null
```

Hình 3.1 Câu truy vấn trên SQL Server

Sau khi thực hiện câu truy vấn, ta thu được tập dữ liệu với 5045 bản ghi. Dữ liệu được thu thập từ tháng 6 năm 2019 đến tháng 8 năm 2022.

Tiền xử lý dữ liệu là một bước không thể thiếu trong Machine Learning vì như ta đã biết, dữ liệu là một phần rất quan trọng, ảnh hưởng trực tiếp tới việc Training Model. Do vậy, tiền xử lý dữ liệu trước khi đưa nó vào model là rất quan trọng, giúp loại bỏ hoặc bù đắp những dữ liệu còn thiếu. Việc tiền xử lý dữ liệu được thực hiện ngay trong câu truy vấn. Việc loại bỏ các trường bị “null” trong khi truy vấn giúp lấy ra các bản ghi có đầy đủ dữ liệu ở tất cả các trường trong tập dữ liệu.

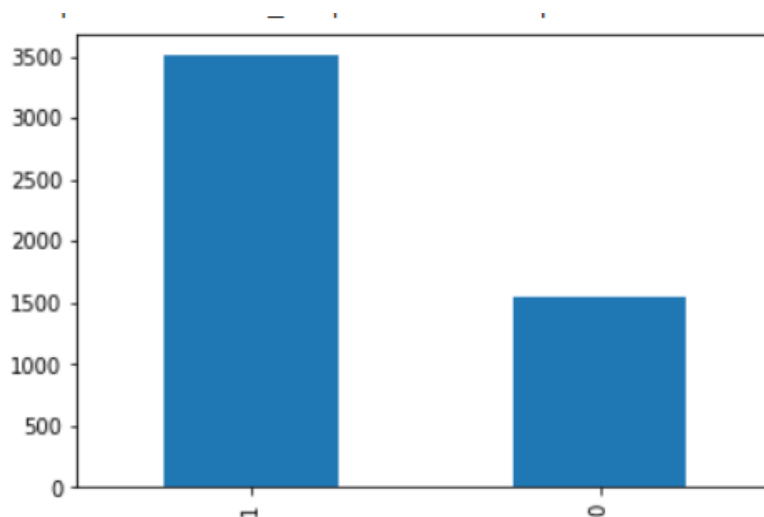
```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5045 entries, 0 to 5044
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  -
0   region          5045 non-null   object
1   timezone        5045 non-null   int64
2   language        5045 non-null   object
3   package         5045 non-null   object
4   hours_use       5045 non-null   int64
5   sum_length      5045 non-null   int64
6   is_paid         5045 non-null   int64
dtypes: int64(4), object(3)
memory usage: 276.0+ KB

```

Hình 3.2 Dữ liệu trước khi đưa vào huấn luyện

Kiểm tra dữ liệu tại hai lớp, số bản ghi tại lớp có nhãn là 1 (khách hàng trả tiền) có 3504 bản ghi và số bản ghi tại lớp có nhãn là 0 (khách hàng miễn phí) có 1541 bản ghi. Như vậy, số bản ghi của lớp có nhãn là 1 gấp xấp xỉ 2.27 lần số bản ghi của lớp có nhãn 0. Suy ra, tập dữ liệu bị mất cân bằng. Điều này gây ảnh hưởng lớn tới việc huấn luyện và kiểm thử mô hình.



Hình 3.3 Tương quan dữ liệu của hai lớp nhãn dân

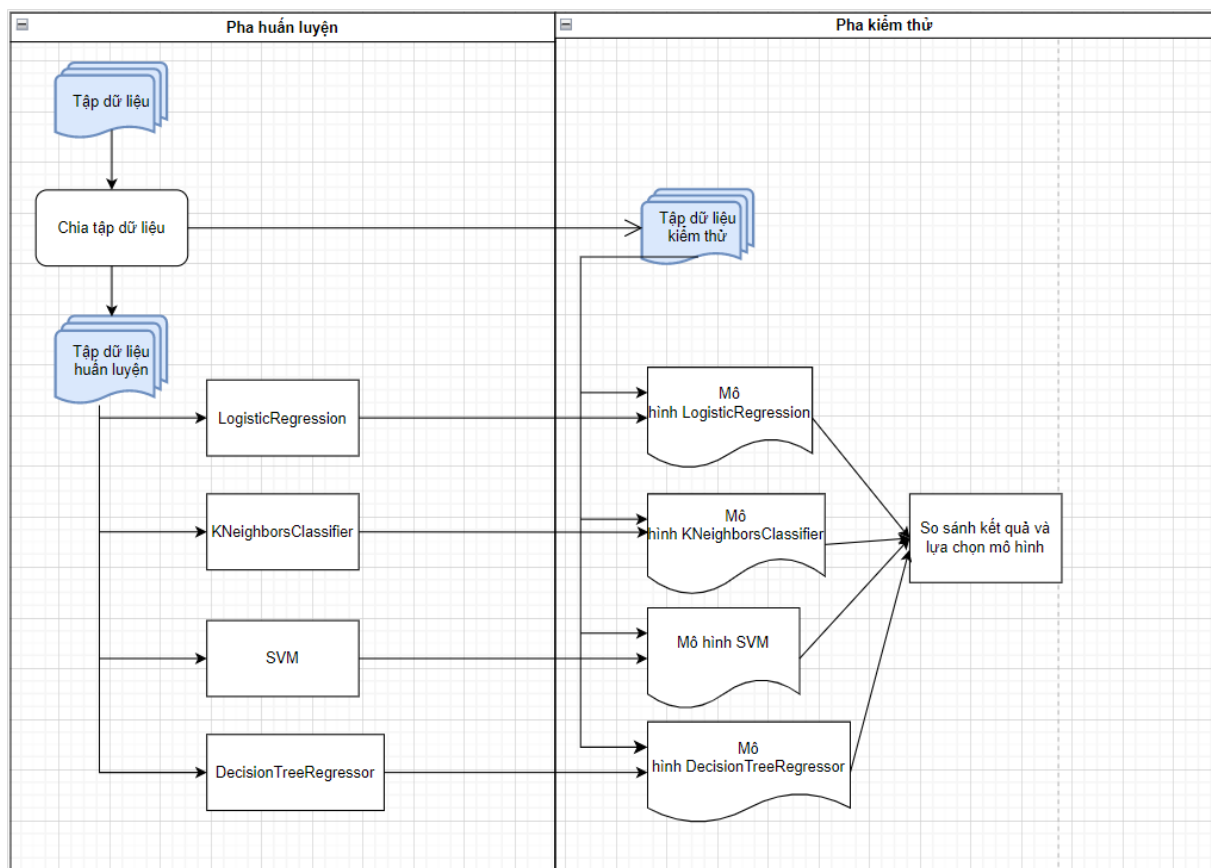
Với việc dữ liệu bị mất cân bằng, ta xử lý thông qua “class_weight” của thư viện sklearn.

Sau khi đã lấy được tệp dữ liệu từ SQL Server, ta bắt đầu vào việc xây dựng mô hình dự đoán với các thuật toán phân lớp.

3.2 Xây dựng mô hình dự đoán

Việc xây dựng mô hình dự đoán được chia làm hai pha:

- Pha huấn luyện mô hình: Thực hiện việc huấn luyện mô hình dự đoán nhóm khách hàng tiềm năng bằng các thuật toán phân lớp nhị phân là hồi quy tuyến logistic (LogisticRegression), k-NN (KNeighborsClassifier), SVM, cây quyết định (DecisionTreeRegressor).
- Pha kiểm thử: Thực hiện việc kiểm chứng độ hiệu quả mô hình, dữ liệu kiểm thử được đưa qua từng mô hình. Dựa trên kết quả thu được, chọn ra thuật toán tốt nhất để xây dựng web app dự đoán.



Hình 3.4 Mô hình chọn lựa thuật toán để xây dựng

3.3 Xây dựng web app

Sau khi xây dựng mô hình, ta tiến hành xây dựng web app để tương tác với mô hình. Web app (web application) hay ứng dụng web là một loại chương trình máy tính thường chạy với sự hỗ trợ của trình duyệt web và công nghệ web để thực hiện các tác vụ khác nhau trên internet. Web app được sử dụng với nhiều mục đích khác nhau và có thể truy cập tại bất cứ đâu. Với phạm vi bài toán, ta sử dụng web app để tương tác với mô hình học máy đã tạo từ trước.

Kết hợp thư viện Streamlit và ngôn ngữ Python, ta xây dựng web app với các chức năng chính sau:

- Cho phép người dùng nhập thông tin khách hàng muốn dự đoán.
- Hiển thị lại thông tin mà khách hàng đang nhập.
- Hiển thị kết quả mà mô hình tính toán ra.
- Hiển thị tỷ lệ phần trăm giữa hai lớp dữ liệu.

Sau khi tạo thành công một web app sử dụng mô hình học máy vừa tạo, ta tiến hành deploy web app trên internet bằng Streamlit Cloud.

Để triển khai, ta thực hiện như sau:

1. Đăng ký Streamlit Cloud: Streamlit's Community Cloud cho phép triển khai, quản lý, chia sẻ ứng dụng web với thế giới một cách hoàn toàn miễn phí.
2. Đăng nhập vào share.streamlit.io: Có thể đăng nhập vào streamlit.io với Google, Github, Email. Trong dự án này, ta nên sử dụng Github để đăng nhập và sử dụng kho lưu trữ này để phát triển web.

Sign in

Continue with Google

Continue with GitHub

Continue with SSO

OR


Your email...


Continue with email

New to Streamlit? [Sign up, it's free!](#)

Hình 3.5 Đăng nhập streamlit.io

Chọn “Continue with GitHub”:





Sign in to **GitHub**
to continue to **Streamlit**

Username or email address

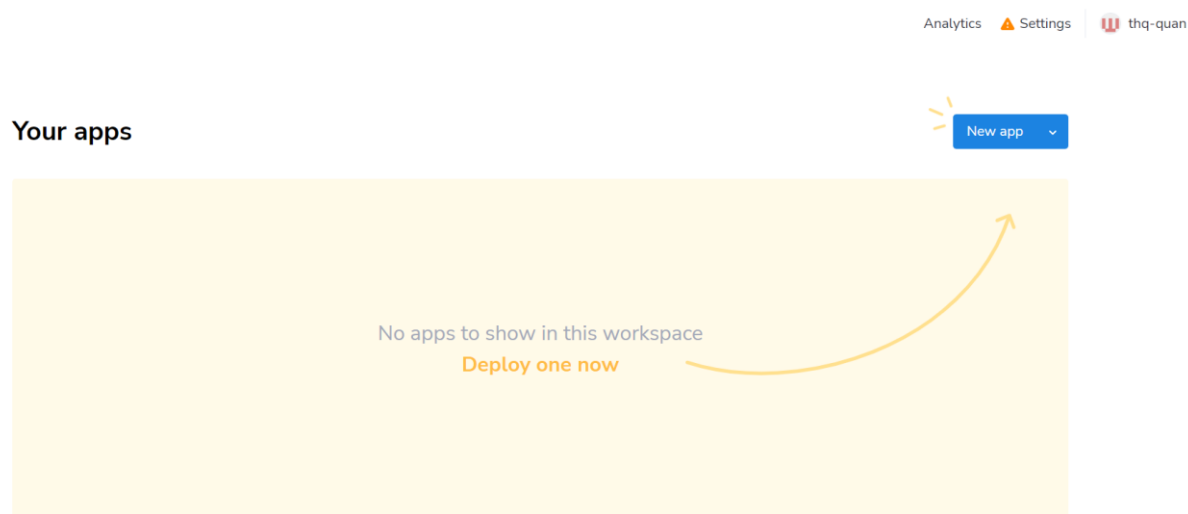
Password [Forgot password?](#)

Sign in

New to GitHub? [Create an account.](#)

Hình 3.6 Đăng nhập với Github

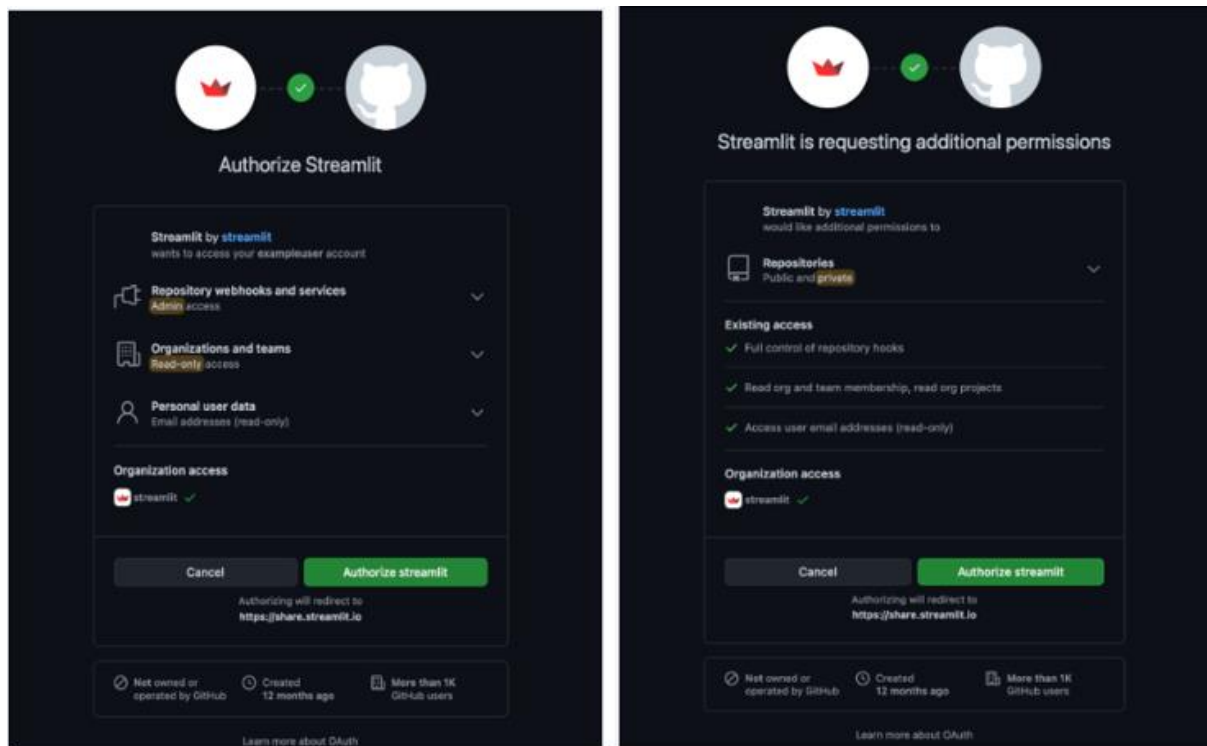
Sau khi đăng nhập, ta được đi đến màn hình làm việc của Streamlit Cloud.



Hình 3.7 Màn hình làm việc của Streamlit Cloud

3. Kết nối với tài khoản Github.

Ta cần cấp quyền cho Streamlit để kết nối với tài khoản GitHub của mình. Điều này cho phép không gian làm việc Streamlit Cloud khởi chạy ứng dụng trực tiếp từ các tệp ứng dụng lưu trữ trong kho lưu trữ của mình.



Hình 3.8 Kết nối Streamlit với kho lưu trữ dự án trên Github

4. Triển khai web app.

Nhấn vào “New app”, chọn kho lưu trữ chứa dự án và file chính để chạy web app. Sau đó nhấn “Deploy!”.

[← Back](#)

Deploy an app

Repository [Paste GitHub URL](#)

thq-quan/Do-an-tot-nghiep-THQ

Branch

main

Main file path

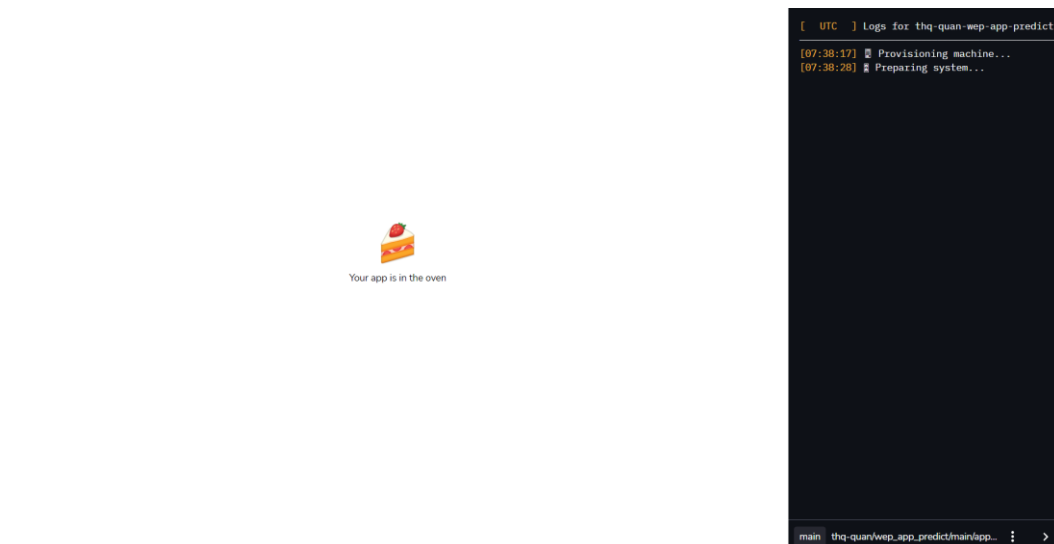
app.py

[Advanced settings...](#)

[Deploy!](#)

Hình 3.9 Triển khai web app

Khi thấy màn hình này có nghĩa là đã khởi chạy web app thành công



Hình 3.10 Triển khai thành công web app

CHƯƠNG 4 THỰC NGHIỆM VÀ ĐÁNH GIÁ

4.1 Môi trường thực nghiệm

Để xây dựng được mô hình đã đề xuất, em đã sử dụng 2 môi trường. Google Colaboratory để xây dựng mô hình dự đoán và máy chủ cục bộ để xây dựng web app.

- Google Colaboratory có tài nguyên:
 - Kết nối với Python 3
 - RAM: 12.68G
 - Ổ đĩa: 107.72GB
- Máy chủ cục bộ: Với bài toán này, em sử dụng laptop cá nhân với cấu hình sau:
 - Hệ điều hành Windows 11 - 64bit
 - Ổ cứng SSD 512GB
 - RAM 8GB
 - CPU Intel(R) Core(TM) i5-9300H @ 2.40GHz

4.2 Kịch bản thực hiện

Đề án thực hiện 4 kịch bản thực nghiệm: thực nghiệm đánh giá kết quả phân loại dựa trên cây quyết định (DecisionTreeRegressor), thực nghiệm đánh giá kết quả phân loại dựa trên kNN (KNearestNeighbors), thực nghiệm đánh giá dựa trên thuật toán SVM và thực nghiệm đánh giá trên thuật toán hồi quy logistic (LogisticRegression). Các thực nghiệm được đánh giá trên cùng tập dữ liệu huấn luyện, kiểm thử và phương pháp đánh giá là độ chính xác (precision), độ hồi tưởng (recall) và độ đo F-score (F1).

Với các tham số thực nghiệm được lựa chọn thông qua các thực nghiệm thay đổi tham số, kết quả được đưa ra dưới đây là các tham số đạt kết quả tốt nhất.

Sau khi lựa chọn được thuật toán học máy có kết quả tốt nhất cho tập dữ liệu, tiến hành xây dựng web app trên mô hình đã lựa chọn. Tiến hành triển khai web app với Streamlit.

4.3 Kết quả thực nghiệm và đánh giá

- ❖ Thực nghiệm đánh giá kết quả phân loại sử dụng thuật toán cây quyết định (DecisionTreeRegressor)

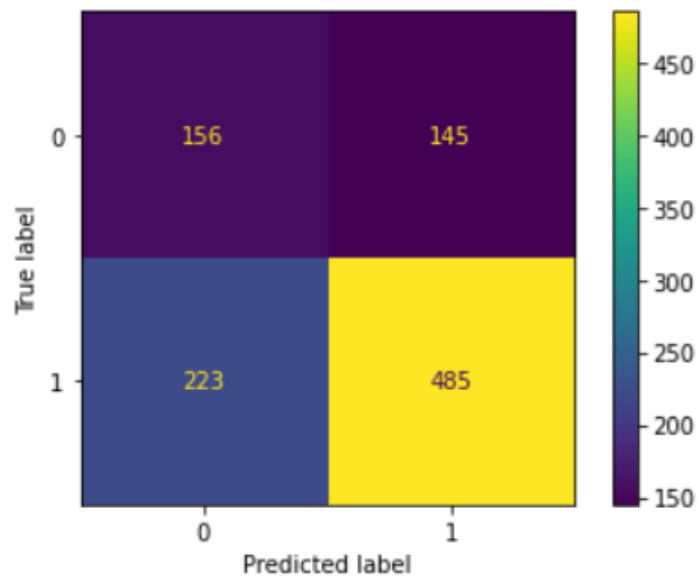
Với thực nghiệm trên thuật toán cây quyết định, tham số liên quan đến thuật toán:

- Tiêu chí: bình phương lỗi (squared_error)
- Bộ chia: ngẫu nhiên (random)
- Số lượng mẫu tối thiểu để tách một nút trong: 3

Bảng 4.1 Kết quả thực nghiệm sử dụng thuật toán cây quyết định

	Tập mẫu	Số dự đoán đúng	Tổng số dự đoán	Độ chính xác	Độ hồi tưởng	Độ đo F-score
0	301	156	379	0.41	0.52	0.46
1	708	485	630	0.77	0.69	0.72
accuracy	1009	641				0.64

Kết quả cho thấy, độ chính xác của lớp 1 cao hơn hẳn so với lớp 0 do sự ảnh hưởng của mất cân bằng dữ liệu. Sự chính xác (accuracy) đạt 0.64 tương ứng với 64% tỷ lệ dự đoán đúng. Tuy nhiên, việc thuật toán cho ra độ chính xác khá chênh lệch giữa hai lớp khiến ta không thể chọn thuật toán này làm mô hình.



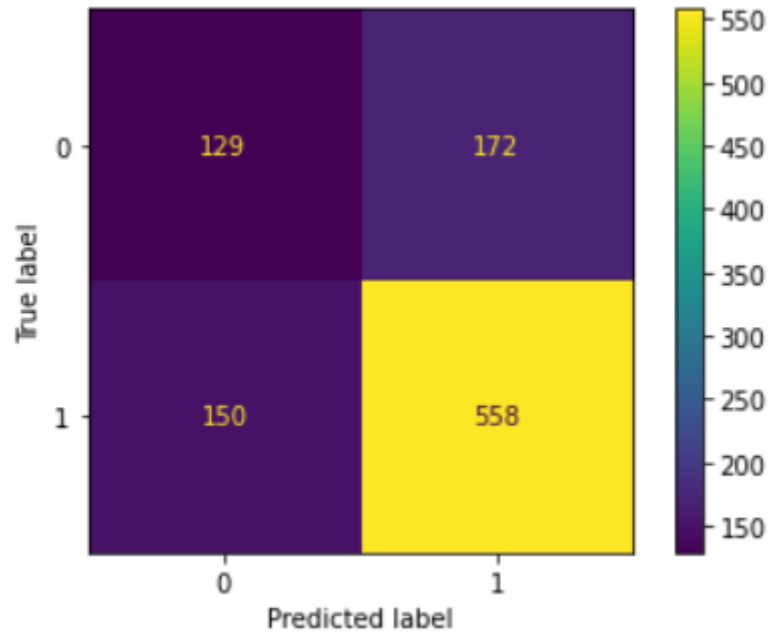
Hình 4.1 Biểu đồ kết quả thực nghiệm thuật toán cây quyết định

- ❖ Thực nghiệm đánh giá kết quả phân loại sử dụng thuật toán kNN (KNearestNeighbors)

Với thực nghiệm trên thuật toán kNN, thay đổi tham số thực nghiệm với $k = 3$ ($n_neighbors = 3$) cho ra kết quả tốt nhất.

Bảng 4.2 Kết quả thực nghiệm sử dụng thuật toán kNN

	Tập mẫu	Số dự đoán đúng	Tổng số dự đoán	Độ chính xác	Độ hồi tưởng	Độ đo F-score
0	301	129	279	0.46	0.43	0.44
1	708	558	730	0.76	0.79	0.78
accuracy	1009	641				0.68



Hình 4.2 Biểu đồ kết quả thực nghiệm thuật toán kNN

Kết quả cho thấy, độ chính xác của lớp 0 tăng lên nhưng độ hồi tưởng lại giảm. Mô hình bị nghiêng về lớp 1 vì khoảng cách giữa 2 lớp là lớn. Tuy độ chính xác đã tăng lên 0.4 so với cây quyết định nhưng vì nghiêng về lớp 1 nhiều khiến mô hình không còn khách quan. Vì vậy ta không sử dụng thuật toán kNN để xây dựng mô hình dự đoán.

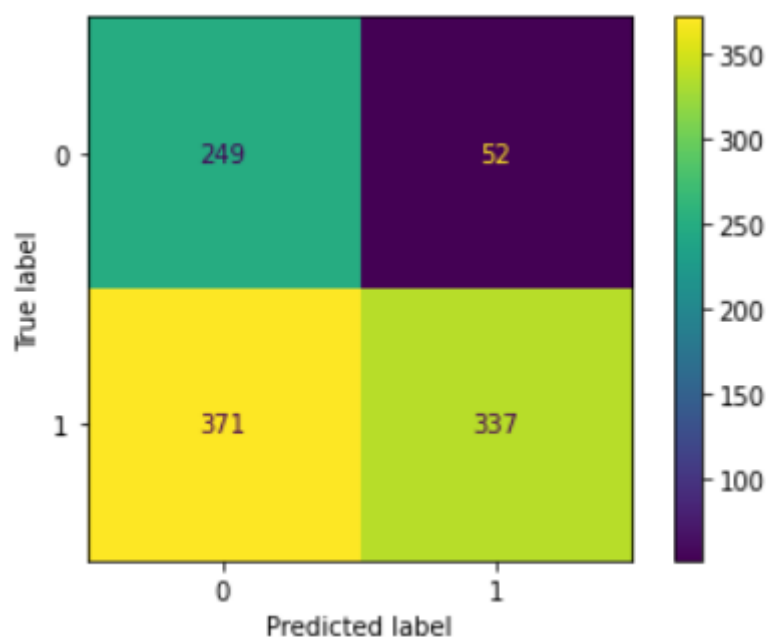
❖ Thực nghiệm đánh giá kết quả phân loại sử dụng thuật toán SVM

Với thuật toán SVM, tham số thực nghiệm trên mô hình đạt kết quả tốt nhất là:

- Nhân (kernel): linear
- Cân đối (class_weight): balanced

Bảng 4.3 Kết quả thực nghiệm của thuật toán SVM

	Tập mẫu	Số dự đoán đúng	Tổng số dự đoán	Độ chính xác	Độ hồi tưởng	Độ đo F-score
0	301	249	620	0.4	0.83	0.54
1	708	337	389	0.87	0.48	0.61
accuracy	1009	641				0.58



Hình 4.3 Biểu đồ kết quả thực nghiệm thuật toán SVM

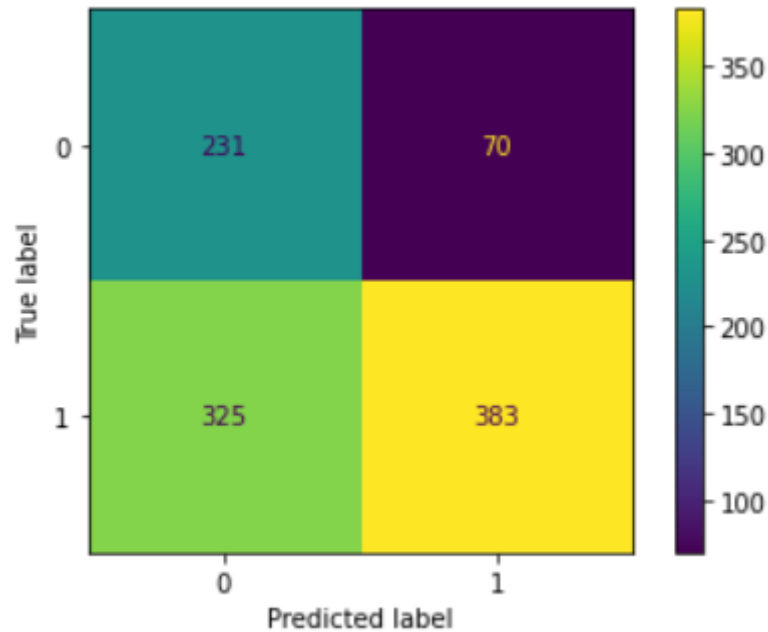
Kết quả cho thấy, độ chính xác của lớp 0 giảm còn 0.4 (giảm 0.06 so với kNN, 0.01 so với cây quyết định). Độ hồi tưởng lại tăng mạnh lên 0.83 cho thấy dự đoán của lớp 0 đã tăng lên đáng kể nhưng lớp 1 lại giảm mạnh xuống 0.48. Sự chính xác cũng giảm còn 0.58 cho nên thuật toán SVM cũng không phù hợp với dữ liệu bài toán.

❖ Thực nghiệm đánh giá kết quả phân loại sử dụng thuật toán hồi quy logistic (LogisticRegression)

Với thuật toán hồi quy logistic, tham số thực nghiệm để mô hình đạt tốt nhất có sự tham gia của độ cân đối giữa hai lớp (class_weight) có giá trị là “balanced”

Bảng 4.4 Kết quả thực nghiệm của thuật toán hồi quy logistic

	Tập mẫu	Số dự đoán đúng	Tổng số dự đoán	Độ chính xác	Độ hồi tưởng	Độ đo F-score
0	301	231	556	0.42	0.77	0.54
1	708	383	453	0.85	0.54	0.66
accuracy	1009	614				0.61



Hình 4.4 Biểu đồ kết quả thực nghiệm thuật toán hồi quy logistic

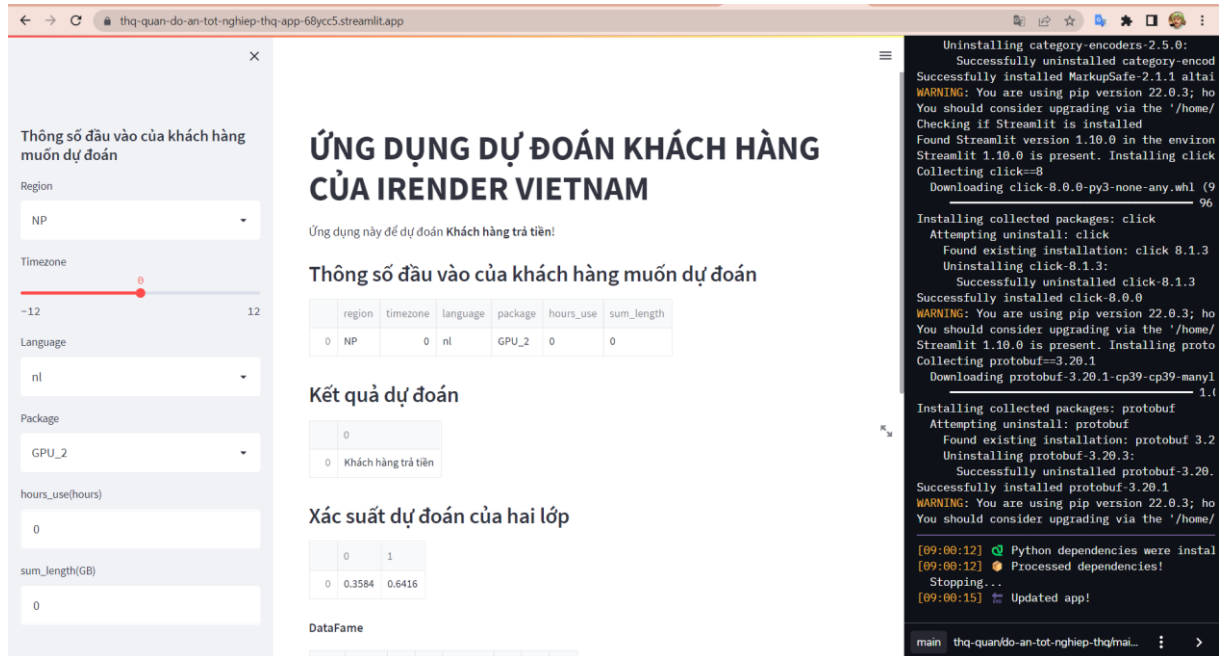
Kết quả cho thấy, việc sử dụng tham số “balanced” trong mô hình giúp cải thiện việc dự đoán lớp 0. Độ hồi tưởng, độ chính xác, độ đo F1 đều tăng lên với lớp 0 và đạt giá trị tốt nhất trong bốn mô hình. Sự chênh lệch giữa dự đoán của hai lớp cũng giảm đi đáng kể, việc nhận biết lớp 0 tốt giúp cho mô hình có độ khách quan. Tuy rằng sự chính xác thấp hơn kNN (0.68) và cây quyết định (0.64) nhưng sự nhận biết của hai lớp tăng lên giúp mô hình hồi quy logistic là mô hình tốt nhất trong bốn thuật toán phân lớp nhị phân.

Qua việc đánh giá kết quả thu được cho thấy, thuật toán hồi quy logistic đạt các tiêu chí đánh giá cao và tốt nhất. Theo đó, em sẽ sử dụng mô hình hồi quy logistic để xây dựng mô hình và triển khai xây dựng web app trên streamlit của Python.

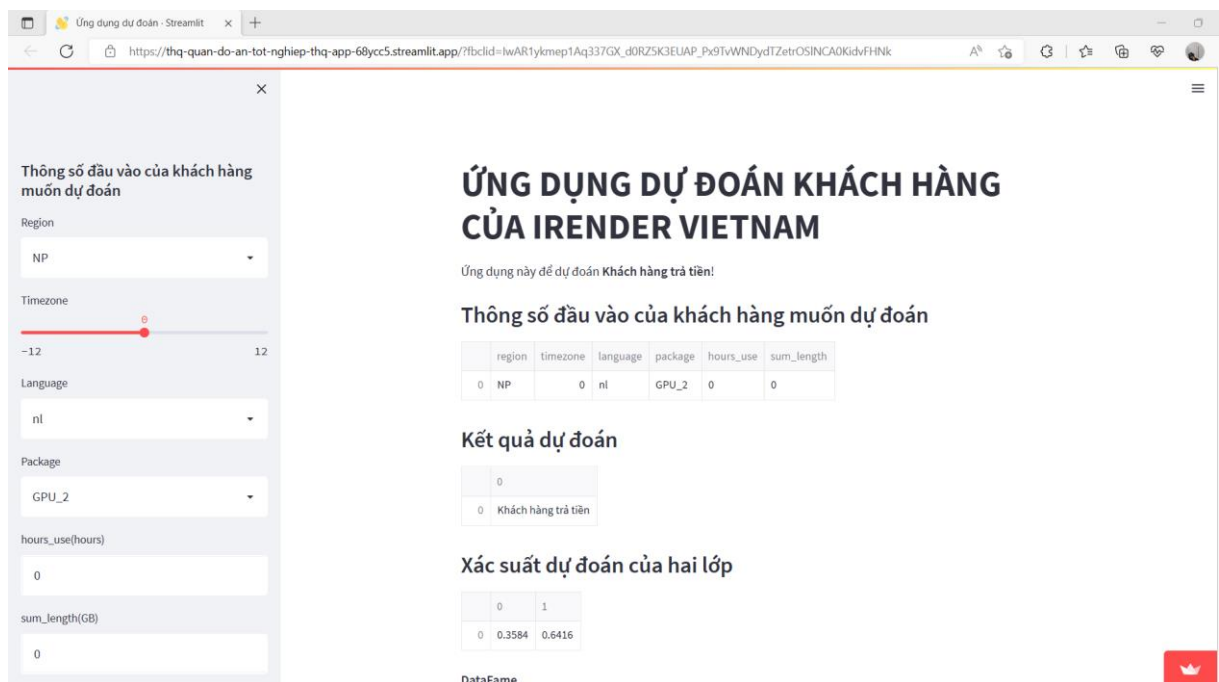
4.4 Kết quả xây dựng web app

4.4.1 Giao diện web app

Giao diện tổng thể của web app:

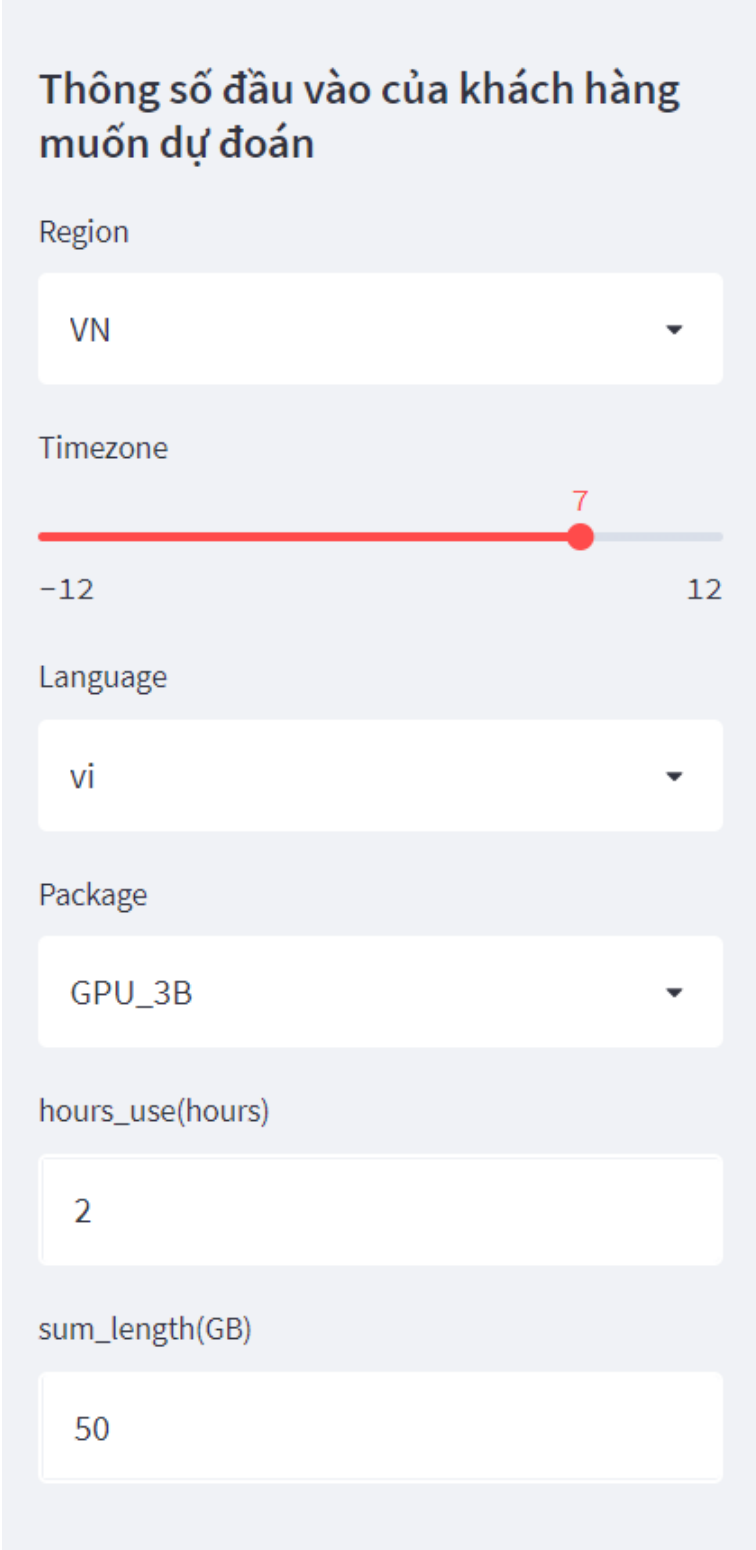


Hình 4.5 Giao diện tổng thể web app với người triển khai



Hình 4.6 Giao diện người được chia sẻ web app

Với chức năng cho phép người dùng nhập thông tin khách hàng muốn dự đoán, giao diện chức năng nhập dữ liệu như sau:



Thông số đầu vào của khách hàng muốn dự đoán

Region

VN

Timezone

7

-12 12

Language

vi

Package

GPU_3B

hours_use(hours)

2

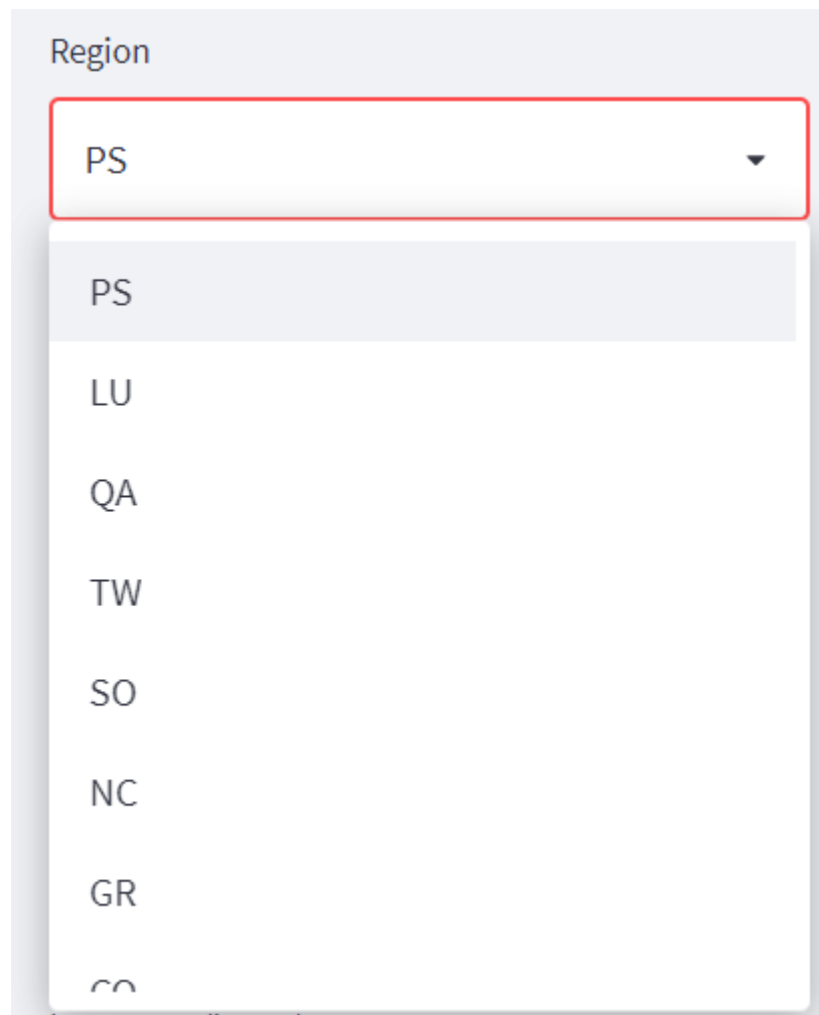
sum_length(GB)

50

Hình 4.7 Phần nhập dữ liệu đầu vào

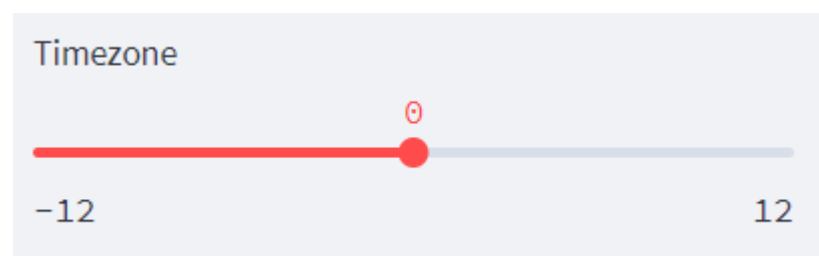
Trong đó:

- Region: thông tin quốc gia, được tạo từ danh sách các quốc gia có trong tập dữ liệu của bài toán



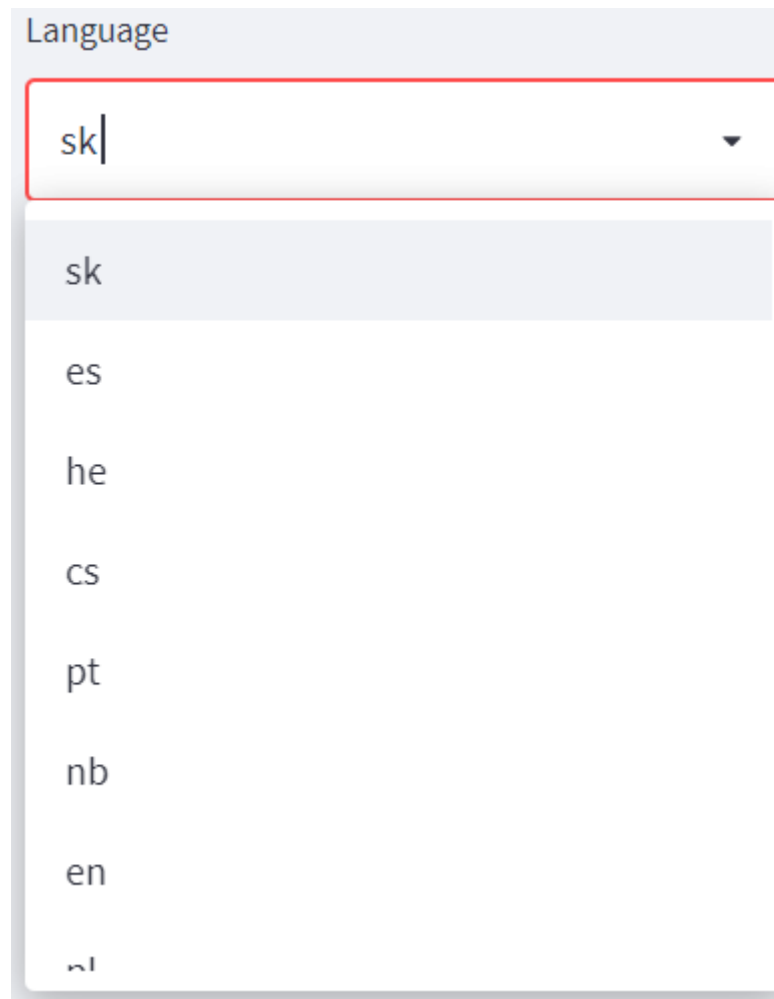
Hình 4.8 Danh sách quốc gia đầu vào của bài toán

- Timezone: múi giờ của khách hàng, được mặc định là từ múi giờ -12 đến múi giờ 12.



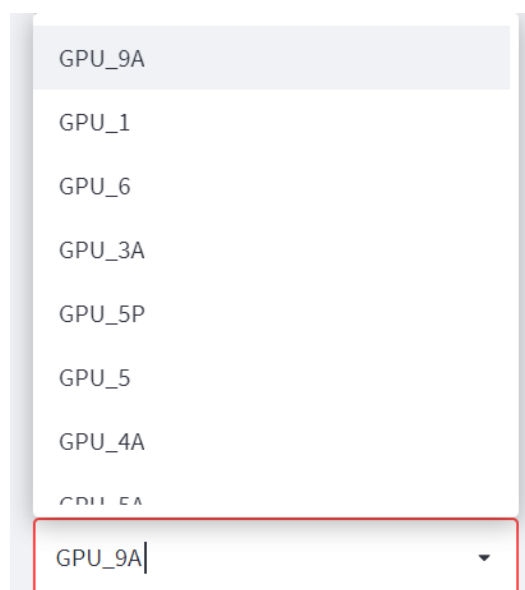
Hình 4.9 Giao diện thông tin timezone

- Language: ngôn ngữ cài đặt của khách hàng, được tạo ra từ danh sách ngôn ngữ trong tập dữ liệu đầu vào.



Hình 4.10 Giao diện phần nhập ngôn ngữ

- Package: thông tin gói dịch vụ sử dụng, được tạo từ danh sách gói sử dụng trong tập dữ liệu bài toán



Hình 4.11 Giao diện phần nhập gói dịch vụ

- Hours_user (hours): Số giờ sử dụng lần đầu.

hours_use(hours)

Hình 4.12 Giao diện phần số giờ sử dụng lần đầu

- Sum_length(GB): dung lượng ổ Z sử dụng.

sum_length(GB)

Hình 4.13 Giao diện phần nhập dung lượng ổ Z

Giao diện hiển thị kết quả thông tin dữ liệu đầu vào:

Thông số đầu vào của khách hàng muốn dự đoán

	region	timezone	language	package	hours_use	sum_length
0	PS	0	sk	GPU_9A	0	0

Hình 4.14 Giao diện hiển thị thông tin người dùng nhập vào

Giao diện hiển thị kết quả dự đoán:

Kết quả dự đoán

	0
0	Khách hàng miễn phí

Hình 4.15 Giao diện kết quả dự đoán

Giao diện hiển thị xác suất giữa hai lớp:

Xác suất dự đoán của hai lớp

	0	1
0	0.6607	0.3393

Hình 4.16 Giao diện xác suất dự đoán của hai lớp

Giao diện hiển thị tập dữ liệu bài toán:

DataFame

	region	time	lang	package	hour	sum	is_pa
0	AU	10	en	GPU_4A	1	25	1
1	SA	3	en	GPU_3A	2	54	1
2	ZA	2	en	GPU_3A	0	48	1
3	US	8	en	GPU_4A	0	31	1
4	BE	1	en	GPU_3A	1	58	0
5	VN	7	vi	GPU_1	1	38	0
6	NL	3	ru	GPU_3A	0	36	1
7	KR	9	ko	GPU_3A	1	8	1
8	CH	1	fr	CPU_1	1	11	0
9	ID	7	en	GPU_3A	1	28	0
			.				

Hình 4.17 Giao diện dữ liệu bài toán

Giao diện của phần log:

```
Successfully built validators
Installing collected packages: pytz, commonma
Successfully installed MarkupSafe-2.1.1 altai
WARNING: You are using pip version 22.0.3; ho
You should consider upgrading via the '/home/
Checking if Streamlit is installed
Found Streamlit version 1.10.0 in the environ
Streamlit 1.10.0 is present. Installing click
Collecting click==8
  Downloading click-8.0.0-py3-none-any.whl (9
  _____ 96
Installing collected packages: click
  Attempting uninstall: click
    Found existing installation: click 8.1.3
    Uninstalling click-8.1.3:
      Successfully uninstalled click-8.1.3
Successfully installed click-8.0.0
WARNING: You are using pip version 22.0.3; ho
You should consider upgrading via the '/home/
Streamlit 1.10.0 is present. Installing proto
Collecting protobuf==3.20.1
  Downloading protobuf-3.20.1-cp39-cp39-manyl
  _____ 1.0
Installing collected packages: protobuf
  Attempting uninstall: protobuf
    Found existing installation: protobuf 3.2
    Uninstalling protobuf-3.20.3:
      Successfully uninstalled protobuf-3.20.
Successfully installed protobuf-3.20.1
WARNING: You are using pip version 22.0.3; ho
You should consider upgrading via the '/home/

[09:13:03] 🔄 Python dependencies were instal
Check if streamlit is installed
Streamlit is already installed
[09:13:04] 📦 Processed dependencies!
```

main thq-quan/do-an-tot-nghiep-thq/mai... ⋮ >

Hình 4.18 Giao diện phần log của web app

4.4.2 Kiểm thử web app

Với web app đã xây dựng, ta đưa dữ liệu vào để kiểm tra mô hình dự đoán.

- Trường hợp 1: Đưa hai bản ghi bất kỳ trong tập dữ liệu của bài toán. Ta sử dụng hai bản ghi sau:

Bảng 4.5 Dữ liệu trường hợp 1 cho kiểm tra web app

STT	region	timezone	language	package	hours_use	sum_length	is_paid
1	US	-6	En	GPU_4A	0	28	0
2	VN	7	En	CPU_1	5	36	1

Ta nhận được kết quả kiểm thử như sau:

- Với khách hàng thứ 1:

Thông số đầu vào của khách hàng muốn dự đoán

Region
US

Timezone
-6

Language
en

Package
GPU_4A

hours_use(hours)
0

sum_length(GB)
28

ỨNG DỤNG DỰ ĐOÁN KHÁCH HÀNG CỦA IRENDER VIETNAM

Ứng dụng này để dự đoán Khách hàng trả tiền!

Thông số đầu vào của khách hàng muốn dự đoán

	region	timezone	language	package	hours_use	sum_length
0	US	-6	en	GPU_4A	0	28

Kết quả dự đoán

0
Khách hàng miễn phí

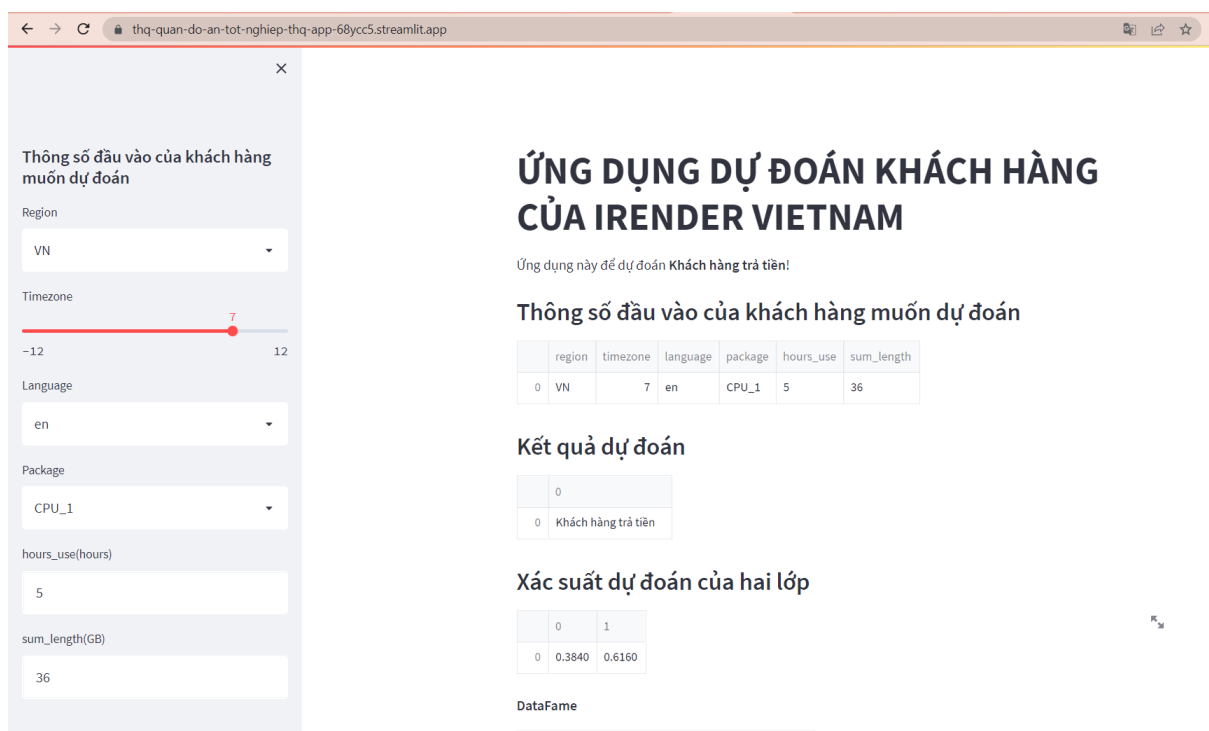
Xác suất dự đoán của hai lớp

	0	1
0	0.6012	0.3988

DataFame

Hình 4.19 Kết quả kiểm thử trường hợp 1 và khách hàng 1

– Với khách hàng thứ 2:



Hình 4.20 Kết quả kiểm thử trường hợp 1 và khách hàng 2

Nhận xét: Với trường hợp 1, web app sử dụng mô hình học máy dự đoán chính xác kết quả so với tập dữ liệu bài toán.

- Trường hợp 2: Đưa 2 bản ghi có số liệu bất kỳ.

Bảng 4.6 Dữ liệu trường hợp 2 cho kiểm tra web app

STT	region	timezone	language	package	hours_use	sum_length
1	IN	5	En	GPU_3A	0	55
2	FR	1	En	GPU_7A	8	20

Ta nhận được kết quả kiểm thử như sau:

– Với khách hàng thứ 1:

Thông số đầu vào của khách hàng muốn dự đoán

Region

IN

Timezone

5

Language

en

Package

GPU_3A

hours_use(hours)

0

sum_length(GB)

55

ỨNG DỤNG DỰ ĐOÁN KHÁCH HÀNG CỦA IRENDER VIETNAM

Ứng dụng này để dự đoán Khách hàng trả tiền!

Thông số đầu vào của khách hàng muốn dự đoán

	region	timezone	language	package	hours_use	sum_length
0	IN	5	en	GPU_3A	0	55

Kết quả dự đoán

0
0 Khách hàng miễn phí

Xác suất dự đoán của hai lớp

	0	1
0	0.5943	0.4057

DataFame

Hình 4.21 Kết quả kiểm thử trường hợp 2 và khách hàng 1

– Với khách hàng thứ 2:

Thông số đầu vào của khách hàng muốn dự đoán

Region

FR

Timezone

1

Language

en

Package

GPU_7A

hours_use(hours)

8

sum_length(GB)

20

ỨNG DỤNG DỰ ĐOÁN KHÁCH HÀNG CỦA IRENDER VIETNAM

Ứng dụng này để dự đoán Khách hàng trả tiền!

Thông số đầu vào của khách hàng muốn dự đoán

	region	timezone	language	package	hours_use	sum_length
0	FR	1	en	GPU_7A	8	20

Kết quả dự đoán

0
0 Khách hàng trả tiền

Xác suất dự đoán của hai lớp

	0	1
0	0.1933	0.8067

DataFame

Hình 4.22 Kết quả kiểm thử trường hợp 2 và khách hàng 2

KẾT LUẬN

❖ Nội dung đã đạt được

- ✓ Nghiên cứu tìm hiểu bài toán dự đoán khách hàng tiềm năng trở thành khách hàng trả tiền cho công ty iRender và hướng tiếp cận giải quyết bài toán.
- ✓ Phân tích, tìm hiểu các đặc trưng, điểm nổi bật của phần dữ liệu về khách hàng nhằm đưa ra các tiêu chí phù hợp nhất với bài toán.
- ✓ Đưa ra được các thuật toán phổ biến trong phân lớp dữ liệu, sử dụng các thuật toán cây quyết định, kNN, SVM, hồi quy logistic cho bài toán phân lớp khách hàng.
- ✓ Xây dựng được web app trên nền tảng Python và Streamlit. Sử dụng mô hình trong web app. Triển khai được web app với Streamlit Cloud.

❖ Hướng tiếp cận trong tương lai

Do dữ liệu chưa có nhiều tiêu chí đánh giá, mang tính chủ quan dẫn tới việc dự đoán khách hàng còn gặp nhiều hạn chế. Vì vậy, hướng tiếp theo để nghiên cứu bài toán dự đoán khách hàng là mở rộng tập dữ liệu, khai thác thêm các tiêu chí đánh giá khách hàng.

TÀI LIỆU THAM KHẢO

TIẾNG VIỆT

- [1] iRender Việt Nam: Khát vọng làm chủ công nghệ
- [2] Từ Minh Phương, Giáo trình “*Nhập môn trí tuệ nhân tạo*” 2014 Trường Đại học Công Nghệ Bưu Chính Viễn Thông

TIẾNG ANH

- [3] John M. Zelle, Python Programming: An Introduction to Computer Science
- [4] Streamlit Cloud - Streamlit Docs - Streamlit documentation
- [5] Wes McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Ipython
- [6] Microsoft SQL documentation Learn how to use SQL Server and Azure SQL, both on-premises and in the cloud
- [7] Documentation GitHub