**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA HỆ THỐNG THÔNG TIN**



**DỰ ĐOÁN KHẢ NĂNG KHÁCH HÀNG ĐĂNG KÝ GỬI TIỀN CÓ KỲ HẠN TỪ CÁC CHIẾN DỊCH TIẾP THỊ CỦA MỘT TỔ CHỨC NGÂN HÀNG VIỆT NAM**

ĐỒ ÁN CUỐI KÌ KHAI THÁC DỮ LIỆU

Lớp: IS252.O21

GVHD: Th.S Mai Xuân Hùng

GV Phạm Nguyễn Thanh Bình

**Nhóm sinh viên thực hiện:**

Nguyễn Hoàng Việt MSSV: 21522791

Nguyễn Hoàng Phúc MSSV: 21520400

Lê Quang Nhân MSSV: 21522402

Nguyễn Văn Phát MSSV: 21522448

**TP. HỒ CHÍ MINH, NĂM 2024**

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA HỆ THỐNG THÔNG TIN**

**DỰ ĐOÁN KHẢ NĂNG KHÁCH HÀNG ĐĂNG KÝ GỬI TIỀN CÓ KỲ HẠN TỪ CÁC CHIẾN DỊCH TIẾP THỊ CỦA MỘT TỔ CHỨC NGÂN HÀNG VIỆT NAM**

ĐỒ ÁN CUỐI KÌ KHAI THÁC DỮ LIỆU

Lớp: IS252.O21

GVHD: Th.S Mai Xuân Hùng

GV Phạm Nguyễn Thanh Bình

**Nhóm sinh viên thực hiện:**

Nguyễn Hoàng Việt MSSV: 21522791

Nguyễn Hoàng Phúc MSSV: 21520400

Lê Quang Nhân MSSV: 21522402

Nguyễn Văn Phát MSSV: 21522448

**TP. HỒ CHÍ MINH, NĂM 2024**

**LỜI CẢM ƠN**

Chúng em xin chân thành cảm ơn trường Đại học Công nghệ Thông tin và Khoa Hệ Thống Thông Tin đã tạo điều kiện cho chúng em hoàn thành tốt đồ án môn học Khai thác dữ liệu. Trong quá trình học tập môn này đã giúp cho nhóm chúng em có được rất nhiều kinh nghiệm quý báu. Đặc biệt, nhóm chúng em xin gửi lời biết ơn sâu sắc đến thầy Mai Xuân Hùng cũng như giảng viên hướng dẫn thực hành Phạm Nguyễn Thanh Bình môn Khai thác dữ liệu (IS252.021) đã dành thời gian quý báo và trực tiếp hướng dẫn tận tình, đóng góp ý kiến giúp nhóm hoàn thành tốt báo cáo môn học.

Thông qua quá trình thực hiện đồ án, chúng em đã phần nào củng cố, tích lũy được những kiến thức về môn Khai thác dữ liệu. Bên cạnh đó là biết sử dụng thêm một số công cụ để phân tích, rất có ích cho chúng em sau này. Bên cạnh đó, chúng em còn rút ra được nhiều kinh nghiệm trong công việc làm nhóm. Hi vọng mọi thứ sẽ được áp dụng và phát triển trong tương lai.

Trong thời gian thực hiện đề tài, vì thời gian và kiến thức có hạn nên không tránh khỏi nhiều sai sót. Vì vậy, nhóm rất mong nhận được những góp ý bổ sung từ thầy để đề tài ngày một hoàn thiện hơn.

Chúng em xin chân thành cảm ơn.

Nhóm sinh viên thực hiện

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*Thành phố Hồ Chí Minh, ngày……...tháng……năm 2024*

**Người nhận xét**

*(Ký tên và ghi rõ họ tên****)***

# MỤC LỤC

[MỤC LỤC 1](#_Toc167743132)

[DANH MỤC BẢNG 4](#_Toc167743133)

[DANH MỤC HÌNH ẢNH 5](#_Toc167743134)

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 7](#_Toc167743135)

[1.1 Lý do chọn đề tài 7](#_Toc167743136)

[1.2 Mô tả bài toán 7](#_Toc167743137)

[1.3 Mô tả dữ liệu 8](#_Toc167743138)

[1.3.1 Mô tả thành phần dữ liệu 8](#_Toc167743139)

[1.3.2 Phân tích dữ liệu 10](#_Toc167743140)

[1.3.2.1 Giá trị lớn nhất, giá trị nhỏ nhất: 10](#_Toc167743141)

[1.3.2.2 Giá trị phổ biến nhất, hiếm nhất 11](#_Toc167743142)

[1.3.2.3 Giá trị trung bình, trung vị 14](#_Toc167743143)

[1.4 Công cụ sử dụng 16](#_Toc167743144)

[CHƯƠNG 2: TRỰC QUAN HÓA DỮ LIỆU 18](#_Toc167743145)

[2.1 Thống kê tỉ lệ khách hàng đăng ký gửi tiền có kỳ hạn theo từng nhóm tuổi 18](#_Toc167743146)

[2.2 Thống kê tỉ lệ khách hàng đăng ký gửi tiền có kỳ hạn theo mức độ học vấn 19](#_Toc167743147)

[2.3 Thống kê tỉ lệ khách hàng đăng ký gửi tiền có kỳ hạn theo từng loại hình nghề nghiệp 21](#_Toc167743148)

[2.4 Thống kê tỉ lệ khách hàng đăng ký gửi tiền có kỳ hạn theo tình trạng hôn nhân 22](#_Toc167743149)

[2.5 So sánh tỉ lệ đăng ký gửi tiền có kì hạn giữa các nhóm khách hàng có khoản vay nhà và có khoản vay cá nhân 23](#_Toc167743150)

[2.6 Phân tích tần suất đăng ký gửi tiền có kỳ hạn theo từng tháng trong năm 25](#_Toc167743151)

[2.7 Thống kê thời gian và số lượng liên hệ trung bình giữa khách hàng đăng ký và không đăng ký gửi tiền có kỳ hạn 25](#_Toc167743152)

[2.8 Phân tích sự ảnh hưởng của số ngày kể từ lần liên lạc cuối cùng và số lần liên hệ trước đó đến việc đăng ký gửi tiền có kỳ hạn 27](#_Toc167743153)

[CHƯƠNG 3: TIỀN XỬ LÝ DỮ LIỆU 29](#_Toc167743154)

[3.1 Làm sạch dữ liệu 29](#_Toc167743155)

[3.1.1 Loại bỏ thuộc tính không cần thiết 29](#_Toc167743156)

[3.1.2 Nhận diện dữ liệu bị nhiễu 29](#_Toc167743157)

[3.1.3 Xử lý dữ liệu bị nhiễu 31](#_Toc167743158)

[3.1.4 Nhận diện dữ liệu bị thiếu 34](#_Toc167743159)

[3.1.5 Ép kiểu thuộc tính dữ liệu 35](#_Toc167743160)

[3.1.6 Xử lý ý nghĩa dữ liệu 36](#_Toc167743161)

[3.1.7 Nhận diện dữ liệu trùng lặp 39](#_Toc167743162)

[3.2 Tích hợp dữ liệu 39](#_Toc167743163)

[3.3 Biến đổi và thu giảm chiều dữ liệu 41](#_Toc167743164)

[3.3.1 Thu giảm chiều dữ liệu 41](#_Toc167743165)

[3.3.2 Chuẩn hóa dữ liệu 43](#_Toc167743166)

[3.4 Rời rạc hóa dữ liệu 44](#_Toc167743167)

[3.5 Tách dữ liệu huấn luyện và kiểm thử 46](#_Toc167743168)

[CHƯƠNG 4: ÁP DỤNG CÁC GIẢI THUẬT KHAI THÁC DỮ LIỆU 47](#_Toc167743169)

[4.1 Giải thuật Naïve Bayes 47](#_Toc167743170)

[4.1.1 Khái niệm 47](#_Toc167743171)

[4.1.2 Lý do lựa chọn 47](#_Toc167743172)

[4.1.3 Áp dụng 48](#_Toc167743173)

[4.1.3.1. Tính toán thủ công 48](#_Toc167743174)

[4.1.3.2. Sử dụng thư viện 53](#_Toc167743175)

[4.2 Giải thuật Logistic Regression 54](#_Toc167743176)

[4.2.1 Khái niệm 54](#_Toc167743177)

[4.2.2 Lý do lựa chọn 56](#_Toc167743178)

[4.2.3 Áp dụng 56](#_Toc167743179)

[4.2.3.1 Tính toán thủ công 56](#_Toc167743180)

[4.2.3.2 Sử dụng thư viện 61](#_Toc167743181)

[4.3 Giải thuật Decision Tree 63](#_Toc167743182)

[4.3.1 Khái niệm 63](#_Toc167743183)

[4.3.2 Lý do lựa chọn 63](#_Toc167743184)

[4.3.3 Áp dụng 63](#_Toc167743185)

[4.4 Giải thuật Random Forest 67](#_Toc167743186)

[4.4.1 Khái niệm 67](#_Toc167743187)

[4.4.2 Lý do lựa chọn 67](#_Toc167743188)

[4.4.3 Áp dụng 68](#_Toc167743189)

[4.5 Giải thuật K-Nearest Neighbor 72](#_Toc167743190)

[4.5.1 Khái niệm 72](#_Toc167743191)

[4.5.2 Lý do lựa chọn 73](#_Toc167743192)

[4.5.3 Áp dụng 73](#_Toc167743193)

[4.5.3.1 Tính toán thủ công 74](#_Toc167743194)

[4.5.3.2 Sử dụng thư viện 76](#_Toc167743195)

[4.6 Giải thuật Perceptron Learning Algorithm 79](#_Toc167743196)

[4.6.1 Khái niệm 79](#_Toc167743197)

[4.6.2 Lý do lựa chọn 80](#_Toc167743198)

[4.6.3 Áp dụng 80](#_Toc167743199)

[4.6.3.1 Tính toán thủ công 81](#_Toc167743200)

[4.6.3.2 Sử dụng thư viện 84](#_Toc167743201)

[4.7 Giải thuật Neural Network 86](#_Toc167743202)

[4.7.1 Khái niệm 86](#_Toc167743203)

[4.7.2 Lý do lựa chọn 88](#_Toc167743204)

[4.7.3 Áp dụng 89](#_Toc167743205)

[4.7.3.1 Tính toán thủ công 89](#_Toc167743206)

[4.7.3.2 Sử dụng thư viện 95](#_Toc167743207)

[CHƯƠNG 5: ĐÁNH GIÁ CÁC THUẬT KHAI THÁC DỮ LIỆU 98](#_Toc167743208)

[5.1 Ma trận nhầm lẫn 98](#_Toc167743209)

[5.2 Đánh giá tỷ lệ TPR (True positive rate) trên tỷ lệ FPR (False positive rate) 98](#_Toc167743210)

[5.3 Đánh giá về thời gian chạy thuật toán 98](#_Toc167743211)

[5.4 Đánh giá về độ chính xác tổng quát của thuật toán (Accuracy) 99](#_Toc167743212)

[5.4.1 So sánh độ chính xác của tập huấn luyện 100](#_Toc167743213)

[5.4.2 So sánh độ chính xác của tập thử nghiệm 101](#_Toc167743214)

[5.5 Đánh giá về độ chính xác của thuật toán (Precision) 102](#_Toc167743215)

[5.5.1 So sánh độ chính xác của thuật toán trên dữ liệu 0 102](#_Toc167743216)

[5.5.2 So sánh độ chính xác của thuật toán trên dữ liệu 1 103](#_Toc167743217)

[5.6 Đánh giá về độ nhạy của thuật toán (Recall) 104](#_Toc167743218)

[5.6.1 So sánh độ nhạy của thuật toán trên dữ liệu 0 104](#_Toc167743219)

[5.6.2 So sánh độ nhạy của thuật toán trên dữ liệu 1 105](#_Toc167743220)

[5.7 Nhận xét các thuật toán 105](#_Toc167743221)

[5.7.1 Naïve Bayes 106](#_Toc167743222)

[5.7 2 Logistic Regression 106](#_Toc167743223)

[5.7.3 Decision Tree 106](#_Toc167743224)

[5.7.4 Random Forest 106](#_Toc167743225)

[5.7.5 K-nearest Neighbor 107](#_Toc167743226)

[5.7.6 Perceptron Learning Algorithm 107](#_Toc167743227)

[5.7.7 Neural Network 108](#_Toc167743228)

[CHƯƠNG 6: TỔNG KẾT 109](#_Toc167743229)

[6.1 Kết quả đạt được 109](#_Toc167743230)

[6.2 Hướng phát triển 110](#_Toc167743231)

[BẢNG PHÂN CHIA CÔNG VIỆC 111](#_Toc167743232)

[BẢNG ĐÁNH GIÁ CHÉO CÁC THÀNH VIÊN TRONG NHÓM 113](#_Toc167743233)

[DANH MỤC TÀI LIỆU THAM KHẢO 116](#_Toc167743234)

# DANH MỤC BẢNG

# DANH MỤC HÌNH ẢNH

[Hình 1. 1 Dataset 8](#_Toc167743235)

[Hình 1. 2 Phân tích dữ liệu Min Max 10](#_Toc167743236)

[Hình 1. 3 Phân tích dữ liệu Most Least 12](#_Toc167743237)

[Hình 1. 5 Phân tích dữ liệu Most Least – Kết quả 12](#_Toc167743238)

[Hình 1. 6 Phân tích dữ liệu – Mean Median 15](#_Toc167743239)

[Hình 1. 8 Đánh giá – Thời gian chạy 99](#_Toc167743240)

[Hình 1. 9 Đánh giá – Accuracy trên tập huấn luyện 100](#_Toc167743241)

[Hình 1. 10 Đánh giá – Accuracy trên tập thử nghiệm 101](#_Toc167743242)

[Hình 2. 1 Code Trực quan hóa dữ liệu 2.5 24](#_Toc165341578)

[Hình 2. 2 Kết quả Trực quan hóa dữ liệu 2.5 24](#_Toc165341579)

[Hình 2. 3 Code Trực quan hóa dữ liệu 2.6 25](#_Toc165341580)

[Hình 3. 1 Tiền xử lý: Loại bỏ thuộc tính ID 29](#_Toc165341541)

[Hình 3. 2 Tiền xử lý: Code vẽ biểu đồ BoxPlot xác định Outlier 29](#_Toc165341542)

[Hình 3. 3 Tiền xử lý: Kết quả vẽ biểu đồ BoxPlot xác định Outlier 30](#_Toc165341543)

[Hình 3. 4 Tiền xử lý: Thống kê Outlier bằng phương pháp Box plot 30](#_Toc165341544)

[Hình 3. 5 Tiền xử lý: Thống kê Outlier bằng phương pháp Z-score 31](#_Toc165341545)

[Hình 3. 6 Tiền xử lý: Class ZscoreOutlierClipper 32](#_Toc165341546)

[Hình 3. 7 Tiền xử lý: Code Áp dụng kỹ thuật Zscore để xử lý giá trị nhiễu 32](#_Toc165341547)

[Hình 3. 8 Tiền xử lý: Kết quả áp dụng kỹ thuật Zscore để xử lý giá trị nhiễu – Cột age 33](#_Toc165341548)

[Hình 3. 9 Tiền xử lý: Kết quả áp dụng kỹ thuật Zscore để xử lý giá trị nhiễu – Cột balance 33](#_Toc165341549)

[Hình 3. 10 Tiền xử lý: Kết quả áp dụng kỹ thuật Zscore để xử lý giá trị nhiễu – Cột day 33](#_Toc165341550)

[Hình 3. 11 Tiền xử lý: Kết quả áp dụng kỹ thuật Zscore để xử lý giá trị nhiễu – Cột duration 33](#_Toc165341551)

[Hình 3. 12 Tiền xử lý: Kết quả áp dụng kỹ thuật Zscore để xử lý giá trị nhiễu – Cột campaign 34](#_Toc165341552)

[Hình 3. 13 Tiền xử lý: Kết quả áp dụng kỹ thuật Zscore để xử lý giá trị nhiễu – Cột pdays 34](#_Toc165341553)

[Hình 3. 14 Tiền xử lý: Kết quả áp dụng kỹ thuật Zscore để xử lý giá trị nhiễu – Cột previous 34](#_Toc165341554)

[Hình 3. 15 Tiền xử lý: Nhận diện giá trị bị thiếu 35](#_Toc165341555)

[Hình 3. 16 Tiền xử lý: Xem xét kiểu dữ liệu của các thuộc tính 35](#_Toc165341556)

[Hình 3. 17 Tiền xử lý: Ép kiểu thuộc tính dữ liệu 35](#_Toc165341557)

[Hình 3. 18 Tiền xử lý: Đếm giá trị cột job 36](#_Toc165341558)

[Hình 3. 19 Tiền xử lý: Xử lý ý nghĩa dữ liệu cột job 36](#_Toc165341559)

[Hình 3. 20 Tiền xử lý: Phân tích cột education 37](#_Toc165341560)

[Hình 3. 21 Tiền xử lý: Xử lý ý nghĩa dữ liệu cột education 38](#_Toc165341561)

[Hình 3. 22 Tiền xử lý: Đếm giá trị cột month 38](#_Toc165341562)

[Hình 3. 23 Tiền xử lý: Xử lý ý nghĩa dữ liệu – Cột month 39](#_Toc165341563)

[Hình 3. 24 Tiền xử lý: Nhận diện dữ liệu trùng lặp 39](#_Toc165341564)

[Hình 3. 25 Tiền xử lý: Code Thể hiện mức tương quan của term\_deposit và các thuộc tính còn lại 40](#_Toc165341565)

[Hình 3. 26 Tiền xử lý: Kết quả Thể hiện mức tương quan của term\_deposit và các thuộc tính còn lại 41](#_Toc165341566)

[Hình 3. 27 Tiền xử lý: Biểu đồ thể hiện mức độ tương quan giữa các thuộc tính 42](#_Toc165341567)

[Hình 3. 28 Tiền xử lý: Loại bỏ thuộc tính có độ tương quan cao 42](#_Toc165341568)

[Hình 3. 29 Tiền xử lý: Thống kê thuộc tính 43](#_Toc165341569)

[Hình 3. 30 Tiền xử lý: Code Chuẩn hóa dữ liệu 44](#_Toc165341570)

[Hình 3. 31 Tiền xử lý: Kết quả chuẩn hóa dữ liệu 44](#_Toc165341571)

[Hình 3. 32 Tiền xử lý: Rời rạc hóa cột month 45](#_Toc165341572)

[Hình 3. 33 Tiền xử lý: Rời rạc hóa cột age 45](#_Toc165341573)

[Hình 3. 34 Tiền xử lý: Kết quả rời rạc hóa cột age 45](#_Toc165341574)

[Hình 3. 35 Tiền xử lý: Dữ liệu mới sau xử lý 46](#_Toc165341575)

[Hình 3. 36 Tiền xử lý: Tách dữ liệu thành X, y 46](#_Toc165341576)

[Hình 3. 37 Tiền xử lý: Tách train & test 46](#_Toc165341577)

[Hình 4. 1 Thủ công - KNN không trọng số 74](#_Toc167639997)

[Hình 4. 2 Thủ công - KNN có tham số 75](#_Toc167639998)

[Hình 4. 3 Thủ công – KNN dự đoán 76](#_Toc167639999)

[Hình 4. 4 Thư viện – KNN không trọng số 76](#_Toc167640000)

[Hình 4. 5 Thư viện – KNN có trọng số 77](#_Toc167640001)

[Hình 4. 6 Thư viện – KNN dự đoán 77](#_Toc167640002)

[Hình 4. 7 Thư viện – Bước 4 78](#_Toc167640003)

[Hình 4. 8 Thư viện – KNN Ma trận nhầm lẫn 79](#_Toc167640004)

[Hình 4. 9 PLA – Thủ công: Bước 1 82](#_Toc167640005)

[Hình 4. 10 PLA – Thủ công: Bước 2 82](#_Toc167640006)

[Hình 4. 11 PLA – Thủ công: Bước 3 83](#_Toc167640007)

[Hình 4. 12 PLA – Thủ công: Bước 4 83](#_Toc167640008)

[Hình 4. 13 PLA – Thủ công: Bước 4 – Kết quả 84](#_Toc167640009)

[Hình 4. 14 PLA – Thủ công: Bước 5 84](#_Toc167640010)

[Hình 4. 15 PLA – Thư viện: Bước 1 85](#_Toc167640011)

[Hình 4. 16 PLA - Thư viện: Bước 2 85](#_Toc167640012)

[Hình 4. 17 PLA - Thư viện: Bước 3 85](#_Toc167640013)

[Hình 4. 18 PLA - Thư viện: Bước 4 86](#_Toc167640014)

[Hình 5. 1 Đánh giá – Thời gian chạy 99](#_Toc167640015)

[Hình 5. 2 Đánh giá – Accuracy trên tập huấn luyện 100](#_Toc167640016)

[Hình 5. 3 Đánh giá – Accuracy trên tập thử nghiệm 101](#_Toc167640017)

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

## Lý do chọn đề tài

* Trong thời đại kinh tế khó khăn hiện nay, việc gửi tiền có kỳ hạn trong ngân hàng có tầm ảnh hưởng rất lớn.
* Đối với khách hàng, giúp tăng giá trị tiền gửi. Đây là hình thức lưu trữ tiền an toàn và ổn định, được bảo vệ bởi ngân hàng. Khách hàng có thể quản lý tài chính hiệu quả hơn, tích lũy tiền cho các mục tiêu dài hạn và lựa chọn đa dạng về kỳ hạn để phù hợp với nhu cầu cá nhân. Ngoài ra, gửi tiền có kỳ hạn tạo nguồn dự trù tài chính cho tương lai và hỗ trợ kiểm soát chi tiêu.
* Đối với ngân hàng, các khoản tiền gửi có kỳ hạn là nguồn vốn quan trọng để duy trì hoạt động kinh doanh, đầu tư và cung cấp dịch vụ tài chính. Việc tăng cường lượng tiền gửi có kỳ hạn từ khách hàng giúp ngân hàng có thể quản lý tài sản hiệu quả hơn, cung cấp các sản phẩm và dịch vụ tốt hơn cho khách hàng.
* Trong bối cảnh cạnh tranh gay gắt trong ngành ngân hàng tại Việt Nam, các tổ chức tài chính cần có chiến lược tiếp thị hiệu quả để thu hút và giữ chân khách hàng. Việc dự đoán khả năng khách hàng đăng ký gửi tiền có kỳ hạn sẽ giúp ngân hàng tối ưu hóa chiến lược tiếp thị, tập trung vào những khách hàng có khả năng cao nhất để tăng doanh số.
* Nhận thấy được điều đó, nhóm đã quyết định chọn đề tài phân tích dữ liệu về các chiến dịch tiếp thị trực tiếp của một tổ chức ngân hàng Việt Nam.

## Mô tả bài toán

* Bài toán của đề tài là dự đoán khả năng khách hàng đăng ký gửi tiền có kỳ hạn từ các chiến dịch tiếp thị của một tổ chức ngân hàng Việt Nam. Cụ thể, mục tiêu là áp dụng các thuật toán phân lớp sử dụng dữ liệu khách hàng, thông tin từ các chiến dịch tiếp thị và các yếu tố liên quan khác để dự đoán khả năng khách hàng sẽ đăng ký gửi tiền có kỳ hạn. Nhằm tối ưu hóa chiến lược tiếp thị, tập trung vào những khách hàng có khả năng cao nhất để tăng doanh số.

Các bước thực hiện trong báo cáo có thể bao gồm:

* **Thu thập dữ liệu**: sử dụng bộ dữ liệu về chiến dịch tiếp thị trực tiếp của một tổ chức ngân hàng Việt Nam từ Kaggle.
* **Trực quan và xử lý tiền dữ liệu**:Phân tích dữ liệu để tìm hiểu các yếu tố ảnh hưởng đến quyết định gửi tiền có kỳ hạn của khách hàng. Dọn dẹp và tiền xử lý dữ liệu để loại bỏ thông tin không cần thiết hoặc không chính xác.
* **Áp dụng các giải thuật toán khai thác dữ liệu**: Sử dụng các giải thuật khai thác dữ liệu để dự đoán dựa trên bộ dữ liệu tìm được.
* **Đánh giá mô hình**: Kiểm tra và đánh giá hiệu suất của các giải thuật khai thác dữ liệu bằng cách sử dụng các chỉ số đo lường như độ chính xác, độ nhạy, độ đặc hiệu, v.v.

Báo cáo sẽ trình bày chi tiết về quy trình trên, kết quả phân tích, và các khuyến nghị dựa trên mô hình dự đoán để giúp ngân hàng tối ưu hóa chiến lược tiếp thị và thu hút khách hàng gửi tiền có kỳ hạn.

## Mô tả dữ liệu

### Mô tả thành phần dữ liệu

- Tên dataset: Bank Customer Data in VietNam

- Nguồn dữ liệu: <https://www.kaggle.com/datasets/tomculihiddleston/bank-customer-data-in-vietnam/data>

- Mô tả dữ liệu: Dữ liệu có liên quan đến các chiến dịch tiếp thị trực tiếp (cuộc gọi điện thoại) của một tổ chức ngân hàng Việt Nam. Mục tiêu là dự đoán xem khách hàng có đăng ký gửi tiền có kỳ hạn hay không.

A screenshot of a computer

Description automatically generated

Hình 1. 1 Dataset

Dữ liệu gồm 42600 dòng và 16 cột dữ liệu, trong đó:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ý nghĩa |
| 1 | ID | int | Số thứ tự của khách hàng đã tư vấn của chiến dịch |
| 2 | age | int | Độ tuổi của khách hàng |
| 3 | job | string | Mô tả công việc của khách hàng |
| 4 | marital | int | Mô tả tình trạng hôn nhân của khách hàng. 0(ly hôn hoặc không xác định), 1(độc thân),2(đã kết hôn cũng như có gia đình) |
| 5 | education | string | Mô tả trình độ học vấn. “primary” (học hết tiểu học), “secondary” (học hết cấp hai),”teriary” (từ cấp ba trở lên: cấp ba, đại học, cao học), “unknown”(không xác định) |
| 6 | default | binary | Mô tả khách hàng có sài thẻ tín dụng mặc định hay không. 0(không), 1(có) |
| 7 | housing | binary | Mô tả khách hàng có khoản vay nhà hay không.0(không), 1(có) |
| 8 | loan | binary | Mô tả khách hàng có khoản vay cá nhân hay không.0(không), 1(có) |
| 9 | day | int | Ngày liên hệ cuối cùng với khách hàng ở chiến dịch lần này |
| 10 | month | string | Tháng của ngày liên hệ cuối cùng với khách hàng ở chiến dịch lần này |
| 11 | duration | int | Tổng số thời gian liên lạc với khách hàng ở chiến dịch lần này(tính bằng giây) |
| 12 | campaign | int | Số lần liên hệ với khách hàng ở chiến dịch này |
| 13 | pdays | int | Số ngày trôi qua kể từ lần liên hệ cuối cùng với khách hàng này ở chiến dịch lần trước(-1 tức là khách hàng này chưa được liên hệ trước đó) |
| 14 | previous | int | Số lần liên hệ được thực hiện trước với khách hàng này ở chiến dịch này |
| 15 | term\_deposit | binary | Khách hàng có đăng kí gửi tiền kì hạn không?0(không), 1(có) |

### Phân tích dữ liệu

* + - 1. Giá trị lớn nhất, giá trị nhỏ nhất:

A screenshot of a computer code

Description automatically generated

Hình 1. 2 Phân tích dữ liệu Min Max

Từ kết quả trên, ta có thể rút ra được kết luận:

* age (tuổi)
* Giá trị lớn nhất (max\_age): 95
* Giá trị nhỏ nhất (min\_age): 18
* Kết luận: Độ tuổi của các khách hàng trong tập dữ liệu dao động từ 18 đến 95 tuổi, cho thấy tập dữ liệu bao gồm cả người trẻ tuổi và người cao tuổi.
* balance (số dư tài khoản)
* Giá trị lớn nhất (max\_balance): 102127
* Giá trị nhỏ nhất (min\_balance): -8019
* Kết luận: Số dư tài khoản dao động từ -8019 đến 102127. Điều này cho thấy có khách hàng bị nợ (số dư âm) và có những khách hàng có số dư rất cao.
* duration (thời lượng cuộc gọi)
* Giá trị lớn nhất (max\_duration): 4918
* Giá trị nhỏ nhất (min\_duration): 0
* Kết luận: Thời lượng cuộc gọi dao động từ 0 đến 4918 giây (tương đương hơn 81 phút). Thời lượng bằng 0 có thể cho thấy các cuộc gọi không thành công hoặc bị từ chối ngay lập tức.
* campaign (số lần liên hệ trong chiến dịch)
* Giá trị lớn nhất (max\_campaign): 63
* Giá trị nhỏ nhất (min\_campaign): 1
* Kết luận: Số lần liên hệ trong chiến dịch dao động từ 1 đến 63 lần. Điều này cho thấy có khách hàng đã được liên hệ rất nhiều lần trong một chiến dịch.
* pdays (số ngày kể từ khi khách hàng được liên hệ lần cuối trong chiến dịch trước)
* Giá trị lớn nhất (max\_pdays): 536
* Giá trị nhỏ nhất (min\_pdays): -1
* Kết luận: Giá trị -1 có thể chỉ ra rằng khách hàng chưa từng được liên hệ trước đó. Số ngày kể từ lần liên hệ trước dao động từ 0 đến 536 ngày đối với những khách hàng đã được liên hệ trước đó.
* previous (số lần liên hệ với khách hàng trước chiến dịch hiện tại)
* Giá trị lớn nhất (max\_previous): 275
* Giá trị nhỏ nhất (min\_previous): 0
* Kết luận: Số lần liên hệ trước chiến dịch hiện tại dao động từ 0 đến 275 lần, với nhiều khách hàng chưa từng được liên hệ trước đó.
  + - 1. Giá trị phổ biến nhất, hiếm nhất

A screenshot of a computer code

Description automatically generated

Hình 1. 3 Phân tích dữ liệu Most Least

Kết quả thu được:

A screenshot of a computer

Description automatically generated

Hình 1. 4 Phân tích dữ liệu Most Least – Kết quả

Từ kết quả trên, ta có thể rút ra được kết luận:

* age (tuổi)
* Giá trị phổ biến nhất (most\_common\_age): 32 (1999 lần xuất hiện)
* Giá trị hiếm nhất (least\_common\_age): 94 (1 lần xuất hiện)
* Nhận xét: Độ tuổi 32 là độ tuổi phổ biến nhất trong tập dữ liệu, trong khi độ tuổi 94 chỉ xuất hiện duy nhất một lần.
* job (nghề nghiệp)
* Giá trị phổ biến nhất (most\_common\_job): blue-collar (9536 lần xuất hiện)
* Giá trị hiếm nhất (least\_common\_job): unknown (264 lần xuất hiện)
* Nhận xét: Nghề nghiệp blue-collar là phổ biến nhất, còn nghề nghiệp unknown là hiếm nhất trong dữ liệu, điều này có thể chỉ ra rằng phần lớn khách hàng làm các công việc liên quan đến lao động chân tay.
* marital (tình trạng hôn nhân)
* Giá trị phổ biến nhất (most\_common\_marital): 2 (25868 lần xuất hiện)
* Giá trị hiếm nhất (least\_common\_marital): 0 (4965 lần xuất hiện)
* Nhận xét: Tình trạng hôn nhân 2 là phổ biến nhất, trong khi tình trạng hôn nhân 0 ít phổ biến hơn.
* education (trình độ học vấn)
* Giá trị phổ biến nhất (most\_common\_education): secondary (22066 lần xuất hiện)
* Giá trị hiếm nhất (least\_common\_education): unknown (1690 lần xuất hiện)
* Nhận xét: Trình độ học vấn secondary là phổ biến nhất trong tập dữ liệu, trong khi unknown là ít phổ biến nhất.
* default (Có thẻ tín dụng)
* Giá trị phổ biến nhất (most\_common\_default): 0 (41828 lần xuất hiện)
* Giá trị hiếm nhất (least\_common\_default): 1 (811 lần xuất hiện)
* Nhận xét: Phần lớn khách hàng không có nợ xấu (default = 0), trong khi số lượng khách hàng có nợ xấu (default = 1) là rất ít.
* housing (vay mua nhà)
* Giá trị phổ biến nhất (most\_common\_housing): 1 (24590 lần xuất hiện)
* Giá trị hiếm nhất (least\_common\_housing): 0 (18049 lần xuất hiện)
* Nhận xét: Số lượng khách hàng có vay mua nhà (housing = 1) nhiều hơn so với những khách hàng không vay mua nhà (housing = 0).
* loan (vay cá nhân)
* Giá trị phổ biến nhất (most\_common\_loan): 0 (35554 lần xuất hiện)
* Giá trị hiếm nhất (least\_common\_loan): 1 (7085 lần xuất hiện)
* Nhận xét: Phần lớn khách hàng không có vay cá nhân (loan = 0), trong khi số lượng khách hàng có vay cá nhân (loan = 1) ít hơn.
* day (ngày liên hệ)
* Giá trị phổ biến nhất (most\_common\_day): 20 (2703 lần xuất hiện)
* Giá trị hiếm nhất (least\_common\_day): 1 (235 lần xuất hiện)
* Nhận xét: Ngày 20 là ngày phổ biến nhất để liên hệ khách hàng, trong khi ngày 1 là ít phổ biến nhất.
* month (tháng liên hệ)
* Giá trị phổ biến nhất (most\_common\_month): may (13532 lần xuất hiện)
* Giá trị hiếm nhất (least\_common\_month): dec (214 lần xuất hiện)
* Nhận xét: Tháng 5 là tháng phổ biến nhất để liên hệ khách hàng, trong khi tháng 12 là ít phổ biến nhất.
* term\_deposit (quyết định gửi tiết kiệm)
* Giá trị phổ biến nhất (most\_common\_term\_deposit): 0 (38678 lần xuất hiện)
* Giá trị hiếm nhất (least\_common\_term\_deposit): 1 (3961 lần xuất hiện)
* Nhận xét: Phần lớn khách hàng không chọn gửi tiết kiệm (term\_deposit = 0), trong khi số lượng khách hàng chọn gửi tiết kiệm (term\_deposit = 1) là ít hơn nhiều.
  + - 1. Giá trị trung bình, trung vị

A screenshot of a computer code

Description automatically generated

Hình 1. 5 Phân tích dữ liệu – Mean Median

Từ kết quả trên, ta có thể rút ra được kết luận:

* Age (Tuổi)
* Trung bình (avg\_age): 40.79
* Trung vị (median\_age): 39.0
* Nhận xét: Độ tuổi trung bình của khách hàng là khoảng 40.79, trong khi tuổi trung vị là 39. Điều này cho thấy độ tuổi của khách hàng tập trung xung quanh mức 40, với một sự phân bố khá đồng đều.
* Balance (Số dư)
* Trung bình (avg\_balance): 1331.86
* Trung vị (median\_balance): 417.0
* Nhận xét: Số dư trung bình của khách hàng là khoảng 1331.86, trong khi số dư trung vị là 417.0. Sự khác biệt lớn giữa trung bình và trung vị cho thấy dữ liệu có thể bị ảnh hưởng bởi một số giá trị cực đại (outliers) lớn.
* Duration (Thời gian cuộc gọi)
* Trung bình (avg\_duration): 255.96
* Trung vị (median\_duration): 175.0
* Nhận xét: Thời gian cuộc gọi trung bình là khoảng 255.96 giây, trong khi trung vị là 175 giây. Điều này cho thấy phần lớn cuộc gọi có thời gian ngắn hơn so với mức trung bình, có thể có một số cuộc gọi rất dài kéo mức trung bình lên cao.
* Campaign (Số lần liên hệ trong chiến dịch)
* Trung bình (avg\_campaign): 2.82
* Trung vị (median\_campaign): 2.0
* Nhận xét: Số lần liên hệ trung bình trong chiến dịch là khoảng 2.82, trong khi trung vị là 2. Điều này cho thấy đa số khách hàng được liên hệ khoảng 2 lần trong chiến dịch.
* Pdays (Số ngày kể từ lần liên hệ cuối cùng trong chiến dịch trước)
* Trung bình (avg\_pdays): 34.17
* Trung vị (median\_pdays): -1.0
* Nhận xét: Giá trị trung vị là -1 cho thấy phần lớn khách hàng chưa từng được liên hệ trong chiến dịch trước. Giá trị trung bình là 34.17 bị kéo cao bởi một số khách hàng có pdays dương.
* Previous (Số lần liên hệ trước đó)
* Trung bình (avg\_previous): 0.47
* Trung vị (median\_previous): 0.0
* Nhận xét: Số lần liên hệ trước đó trung bình là 0.47, trong khi trung vị là 0. Điều này cho thấy hầu hết khách hàng chưa từng được liên hệ trước đó.

## 1.4 Công cụ sử dụng

* Công cụ khai thác:
* Google Colab
* Jupyter Notebook
* Phiên bản Python: 3.11.5
* Thư viện kèm theo:
* import numpy as np
* import pandas as pd
* import seaborn as sns
* import math
* import matplotlib.pyplot as plt
* from sklearn import preprocessing
* from sklearn.preprocessing import StandardScaler, LabelEncoder
* from sklearn.model\_selection import train\_test\_split

# CHƯƠNG 2: TRỰC QUAN HÓA DỮ LIỆU

## 2.4 Thống kê tỉ lệ khách hàng đăng ký gửi tiền có kỳ hạn theo tình trạng hôn nhân

- Sử dụng hàm barplot trong Seaborn để xây dựng đồ thị dạng cột thống kê tỷ lệ khách hàng đăng ký gửi tiền có kỳ hạn theo tình trạng hôn nhân (term\_deposit = 1)

A screenshot of a computer code

Description automatically generated

- Kết quả thu được:

A blue rectangular bars with white text

Description automatically generated

- Kết luận: Từ biểu đồ ta dễ dàng nhận thấy nhóm khách hàng đã kết hôn có tỷ lệ gửi tiền ngân hàng cao nhất ( gần 12%). Tương tự, nhóm khách hàng độc thân có tỷ lệ gửi tiền thấp (xấp xỉ 8%). Điều này dễ thấy ở ngoài thực tế, những người có gia đình hay tích góp còn những người độc thân hay có xu hướng sống tự do và chưa có mục tiêu tích lũy.

## 2.5 So sánh tỉ lệ đăng ký gửi tiền có kì hạn giữa các nhóm khách hàng có khoản vay nhà và có khoản vay cá nhân

- Phân loại khách hàng thành 4 nhóm:

* Không có khoản vay nhà (0) và không có khoản vay cá nhân (0)
* Có có khoản vay nhà (1) và không có khoản vay cá nhân (0)
* Không có khoản vay nhà (0) và có khoản vay cá nhân (1)
* Có có khoản vay nhà (1) và có khoản vay cá nhân (1)

Sau đó sử bụng Bar Plot để hiển thị tỉ lệ đăng ký gửi tiền có kì hạn cho từng nhóm khách hàng trên.

A screenshot of a computer code

Description automatically generated

Hình 2. 1 Code Trực quan hóa dữ liệu 2.5

* Kết quả thu được:

A screenshot of a graph

Description automatically generated

Hình 2. 2 Kết quả Trực quan hóa dữ liệu 2.5

- Nhận xét: Từ biểu đồ ta dễ dàng nhận thấy nhóm khách hàng không có khoản vay nhà và không có khoản vay cá nhân có tỉ lệ đăng ký gửi tiền có kì hạn cao nhất (gần 14%). Tương tự, nhóm khách hàng có khoản vay nhà và có khoản vay cá nhân có tỉ lệ đăng ký gửi tiền có kì hạn thấp nhất (khoảng 6%). Điều này cũng dễ hiểu so với thực tế khi khách hàng có các vay thì xác suất người đó đi gửi tiền sẽ giảm đi.

## 2.6 Phân tích tần suất đăng ký gửi tiền có kỳ hạn theo từng tháng trong năm

- Chuyển dữ liệu cột month thành dạng Cetegorical để đảm bảo thứ tự các tháng từ tháng 1 đến tháng 12. Sau đó sử dụng Cat Plot của thư viện Seaborn để hiển thị tỉ lệ khách hàng đăng ký gửi tiền có kỳ hạn trong từng tháng trong năm.A close-up of text

Description automatically generated

Hình 2. 3 Code Trực quan hóa dữ liệu 2.6

- Kết quả thu được:

A line graph with blue lines

Description automatically generated

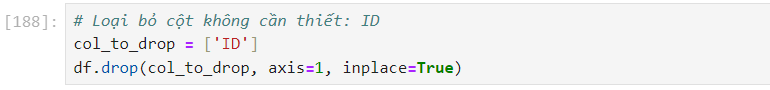
- Nhận xét: Từ biểu đồ ta có thể thấy khách hàng có xu hướng gửi tiền có kì hạn cao vào tháng 3 (tháng sau tết), và các tháng về cuối năm (9, 10, 12). Tỉ lệ khách hàng gửi tiền thấp vào các tháng 1,2 (đầu năm mới) và các tháng về giữa năm (5, 6, 7, 8).

# CHƯƠNG 3: TIỀN XỬ LÝ DỮ LIỆU

## 3.1 Làm sạch dữ liệu

### 3.1.1 Loại bỏ thuộc tính không cần thiết

- Loại bỏ thuộc tính ID vì không cần thiết



Hình 3. 1 Tiền xử lý: Loại bỏ thuộc tính ID

### 3.1.2 Nhận diện dữ liệu bị nhiễu

- Vẽ biểu bồ Box Plot của các thuộc tính định lượng và xem xét các giá trị nhiễu nằm ngoài khoảng chặn trên / chặn dưới.

A white background with black and red dots

Description automatically generated with medium confidence

Hình 3. 2 Tiền xử lý: Code vẽ biểu đồ BoxPlot xác định Outlier

- Kết quả thu được:

A group of graphs showing different sizes and shapes

Description automatically generated with medium confidence

Hình 3. 3 Tiền xử lý: Kết quả vẽ biểu đồ BoxPlot xác định Outlier

- Thống kê số lượng và tỉ lệ phần trăm giá trị nhiễu của từng thuộc tính bằng phương pháp tứ phân vị:

A computer screen shot of a computer code

Description automatically generated

Hình 3. 4 Tiền xử lý: Thống kê Outlier bằng phương pháp Box plot

- Thống kê số lượng và tỉ lệ phần trăm giá trị nhiễu của từng thuộc tính bằng phương pháp Z-score:

A screenshot of a computer code

Description automatically generated

Hình 3. 5 Tiền xử lý: Thống kê Outlier bằng phương pháp Z-score

- Nhận xét: Ta thấy rằng, tỉ lệ ngoại lệ khi thống kê bằng phương pháp tứ phân vị chiếm tỉ lệ khá lớn so với tập dữ liệu ban đầu (vd: cột previous chiếm tới hơn 15% giá trị ngoại lệ). Chứng tỏ dữ liệu của các thuộc tính định lượng ban đầu có một số giá trị có tần suất xuất hiện rất cao. Trong khi đó, thống kê bằng phương pháp Z-score lại cho kết quả khá thấp (vd: cột previous chỉ chiếm khoảng 1% giá trị ngoại lệ). Ta có thể giả sử rằng dữ liệu ban đầu tuân theo phân phối chuẩn.

### 3.1.3 Xử lý dữ liệu bị nhiễu

- Giả sử rằng dữ liệu ban đầu của các thuộc tính định lượng tuân theo phân phối chuẩn , ta có thể áp dụng quy tắc 3σ cho phân phối chuẩn [1].

Trong phân phối chuẩn, giả sử μ là kỳ vọng và σ là độ lệch chuẩn. Quy tắc 3σ cho phân phối chuẩn nói rằng:

* 68% các điểm dữ liệu nằm trong khoảng μ ± 𝜎
* 95% các điểm dữ liệu nằm trong khoảng μ ± 2σ
* 99.7% các điểm dữ liệu nằm trong khoảng μ ± 3σ

Với một điểm dữ liệu x, z-score của nó được tính bởi:

Những điểm có z score nằm ngoài đoạn [−3,3] có thể được coi là các điểm ngoại lệ tương đương với việc các điểm nằm ngoài đoạn [μ−3σ,μ+3σ] được coi là các điểm ngoại lệ.

- Áp dụng kĩ thuật trên để xử lý giá trị nhiễu, ta sẽ xác định chặn dưới bằng μ - 3σ và chặn trên bằng μ + 3σ, đối với dữ liệu nằm ngoài chặn dưới ta cắt nó về giá trị chặn dưới, tương ứng với các điểm dữ liệu nằm ngoài chặn trên, ta cắt nó về giá trị chặn trên.

A screenshot of a computer program

Description automatically generated

Hình 3. 6 Tiền xử lý: Class ZscoreOutlierClipper

- Áp dụng vào dữ liệu của các thuộc tính trước đó, sau đó trực quan hóa dữ liệu trước và sau khi áp dụng kỹ thuật loại bỏ giá trị nhiễu cùng với giá trị skew (thể hiện độ nghiêng của đồ thị, giá trị skew càng gần 0 thì đồ thị càng đối xứng và đạt phân phối chuẩn)

A computer screen shot of a program code

Description automatically generated

Hình 3. 7 Tiền xử lý: Code Áp dụng kỹ thuật Zscore để xử lý giá trị nhiễu

- Kết quả:

A graph and diagram of a graph

Description automatically generated

Hình 3. 8 Tiền xử lý: Kết quả áp dụng kỹ thuật Zscore để xử lý giá trị nhiễu – Cột age

A graph of two people

Description automatically generated

Hình 3. 9 Tiền xử lý: Kết quả áp dụng kỹ thuật Zscore để xử lý giá trị nhiễu – Cột balance

A graph and chart of a graph

Description automatically generated with medium confidence

Hình 3. 10 Tiền xử lý: Kết quả áp dụng kỹ thuật Zscore để xử lý giá trị nhiễu – Cột day

A screenshot of a graph

Description automatically generated

Hình 3. 11 Tiền xử lý: Kết quả áp dụng kỹ thuật Zscore để xử lý giá trị nhiễu – Cột duration

A screenshot of a graph

Description automatically generated

Hình 3. 12 Tiền xử lý: Kết quả áp dụng kỹ thuật Zscore để xử lý giá trị nhiễu – Cột campaign

A screenshot of a graph

Description automatically generated

Hình 3. 13 Tiền xử lý: Kết quả áp dụng kỹ thuật Zscore để xử lý giá trị nhiễu – Cột pdays

A screenshot of a graph

Description automatically generated

Hình 3. 14 Tiền xử lý: Kết quả áp dụng kỹ thuật Zscore để xử lý giá trị nhiễu – Cột previous

- Nhận xét : Nhìn chung, tỉ lệ giá trị nhiễu đã giảm, giá trị skew của các thuộc tính đã giảm tương đối nhiều so giá skew ban đầu khi chưa áp dụng kĩ thuật. Tuy nhiên giá trị skew của một số cột vẫn tương đối cao, ta sẽ xử lý các cột đó ở các bước sau.

### 3.1.4 Nhận diện dữ liệu bị thiếu

- Đếm số lượng giá trị null ở từng thuộc tính trong dữ liệu ban đầu

A white rectangular object with a white border

Description automatically generated

Hình 3. 15 Tiền xử lý: Nhận diện giá trị bị thiếu

- Nhận xét: Dữ liệu ban đầu không có giá trị null nên ta không cần áp dụng kĩ thuật xử lý dữ liệu bị thiếu trên tập dữ liệu này.

### 3.1.5 Ép kiểu thuộc tính dữ liệu

- Xem xét kiểu dữ liệu của các thuộc tính:

A screenshot of a computer

Description automatically generated

Hình 3. 16 Tiền xử lý: Xem xét kiểu dữ liệu của các thuộc tính

- Có thể thấy một số thuộc tính phải có giá trị nguyên (age, campain. previous, pdays) nhưng đang có kiểu dữ liệu là float nên ta có thể chuyển kiểu dữ liệu của các thuộc tính này về kiểu số nguyên

A white rectangular object with a black border

Description automatically generated

Hình 3. 17 Tiền xử lý: Ép kiểu thuộc tính dữ liệu

### 3.1.6 Xử lý ý nghĩa dữ liệu

3.1.6.1 Cột “job”

A white screen with a black border

Description automatically generated with medium confidence

Hình 3. 18 Tiền xử lý: Đếm giá trị cột job

- Ta thấy dữ liệu cột job đang có dạng danh mục thể hiện cho từng loại hình công việc, ta thực hiện chuyển dữ liệu của cột job thành dạng số bằng phương pháp LabelEncoder.

A screenshot of a computer program

Description automatically generated

Hình 3. 19 Tiền xử lý: Xử lý ý nghĩa dữ liệu cột job

3.1.6.2 Cột “education”

A screenshot of a graph

Description automatically generated

Hình 3. 20 Tiền xử lý: Phân tích cột education

- Ta thấy dữ liệu cột education đang ở dạng danh mục có thứ bậc gồm 4 loại dữ liệu:

* + Tertiary: Cao đẳng, đại học
  + Secondary: Trung học
  + Primary: Tiểu học
  + Unknown: Không biết

- Khi biểu diễn tỉ lệ đăng ký khách hàng đăng ký gửi tiền có kỳ hạn theo từng mức độ học vấn, ta nhận thấy rằng, khách hàng có mức đồ học vấn càng cao thì tỉ lệ người đó đăng ký gửi tiền càng tăng. Như vậy ta thực hiển chuyển kiểu dữ liệu của cột education thành dạng số có thứ bậc tương ứng từ thấp đến cao, đối với dữ liệu “unknown” chiếm tỉ lệ khá thấp và tỉ lệ đăng ký gửi tiền khá gần với tỉ lệ của dữ liệu “secondary”, ta có thể gộp 2 loại này thành một để đơn giản hóa việc tính toán:

A screen shot of a computer

Description automatically generated

Hình 3. 21 Tiền xử lý: Xử lý ý nghĩa dữ liệu cột education

3.1.6.3 Cột “month”

A white rectangular object with a black border

Description automatically generated

Hình 3. 22 Tiền xử lý: Đếm giá trị cột month

- Ta thấy dữ liệu cột month đang ở dạng danh mục với các giá trị dạng chuỗi đại diện cho các tháng trong năm. Ta thực hiện chuyển kiểm dữ liệu cột “month” thành kiểu số tương tứng từ tháng 1 đến tháng 12.

A screenshot of a computer

Description automatically generated

Hình 3. 23 Tiền xử lý: Xử lý ý nghĩa dữ liệu – Cột month

### 3.1.7 Nhận diện dữ liệu trùng lặp

A close-up of a computer screen

Description automatically generated

Hình 3. 24 Tiền xử lý: Nhận diện dữ liệu trùng lặp

- Ta thấy dữ liệu ban đầu không chứa các giá trị trùng nên ta không cần phải xử lý cho trường hợp này.

## 3.2 Tích hợp dữ liệu

- Vẽ biểu đồ Scatter và biểu đồ Line thể hiện độ tương quan giữa thuộc tính quyết định “term\_deposit” và các thuộc tính còn lại trong dữ liệu.

A computer screen shot of a computer

Description automatically generated

Hình 3. 25 Tiền xử lý: Code Thể hiện mức tương quan của term\_deposit và các thuộc tính còn lại

- Kết quả:

A screenshot of a graph

Description automatically generated

A screenshot of a graph

Description automatically generated

A graph of a graph with a red line

Description automatically generated with medium confidence

A graph with a red line

Description automatically generated

Hình 3. 26 Tiền xử lý: Kết quả Thể hiện mức tương quan của term\_deposit và các thuộc tính còn lại

- Nhật xét: Từ đồ thị ta có thể dễ dàng nhìn thấy

* Các thuộc tính tương quan thuận với thuộc tính quyết định term\_deposit như là: education, balance, month, duration, previous. Tức giá trị các thuộc tính trên càng tăng thì tỉ lệ đăng ký gửi tiền càng cao.
* Các thuộc tính tương quan nghịch với thuộc tính quyết định term\_deposit như là: default, housing, loan, campain. Tức giá trị các thuộc tính trên càng tăng thì tỉ lệ đăng ký gửi tiền càng thấp.
* Các thuộc tính các tính có mức tương quan thấp (gần bằng 0) với thuộc tính quyết định term\_deposit như là: age, marital, pdays, job.

## 3.3 Biến đổi và thu giảm chiều dữ liệu

### 3.3.1 Thu giảm chiều dữ liệu

- Thể hiện mức độ tương quan giữa các thuộc tính trong tập dữ liệu bằng biểu đồ Heatmap.

A screenshot of a computer

Description automatically generated

Hình 3. 27 Tiền xử lý: Biểu đồ thể hiện mức độ tương quan giữa các thuộc tính

- Ta thấy rằng, 2 thuộc tính pdays và previous có mức độ tương quan khá cao (0.74). Điều này xảy ra là bởi trong dữ liệu ban đầu, đối với khách hàng lần đầu khảo sát sẽ có dữ liệu cột pdays là -1 tương ứng dữ liệu cột previous sẽ là 0.Ta thực hiện loại bỏ một trong hai cặp thuộc tính có mức độ tương quan cao (ở đây ta xóa cột previous)

A screenshot of a computer code

Description automatically generated

Hình 3. 28 Tiền xử lý: Loại bỏ thuộc tính có độ tương quan cao

- Kết quả sau khi loại bỏ thuộc tính có mức độ tương quan cao (cột previous)

A blue and white grid with black squares

Description automatically generated

### 3.3.2 Chuẩn hóa dữ liệu

- Thống kê các thuộc tính định lượng sau tập dữ liệu sau các bước tiền xử lý trước đó

A screenshot of a computer

Description automatically generated

Hình 3. 29 Tiền xử lý: Thống kê thuộc tính

- Ta thấy dữ liệu các cột balance và duration có độ lệch chuẩn lớn và chênh lệch giữa giá trị lớn nhất và giá trị nhỏ nhất là khá cao. Ta sử dụng phương pháp Min-Max Nomalization để đưa giá trị cột balance và duration về khoảng [-1,1].

A screenshot of a computer

Description automatically generated

Hình 3. 30 Tiền xử lý: Code Chuẩn hóa dữ liệu

- Sau khi chuẩn hóa, dữ liệu cột balance và duration đã có độ lệch chuẩn bé hơn 1.

A screenshot of a graph

Description automatically generated

Hình 3. 31 Tiền xử lý: Kết quả chuẩn hóa dữ liệu

## 3.4 Rời rạc hóa dữ liệu

- Cột month: Chuyển đổi dữ liệu cột month thành 4 nhóm nhỏ tương ứng với từng khoảng thời gian của 4 quý trong năm nhằm phân dữ liệu tháng dựa trên chu kỳ thời gian hay vòng đời của dữ liệu nhằm phản ánh tốt hơn cho việc dự đoán.

A screenshot of a computer

Description automatically generated

Hình 3. 32 Tiền xử lý: Rời rạc hóa cột month

- Cột age:

* Tạo các nhóm tuổi chia giá trị cột age thành 5 nhóm tuổi nhỏ có số lượng khách hàng tương đối đều nhau. Sử dụng hàm qcut của thư viện Pandas.

A computer screen shot of a computer code

Description automatically generated

Hình 3. 33 Tiền xử lý: Rời rạc hóa cột age

* Kết quả đạt được:

A white background with colorful text

Description automatically generated

Hình 3. 34 Tiền xử lý: Kết quả rời rạc hóa cột age

## 3.5 Tách dữ liệu huấn luyện và kiểm thử

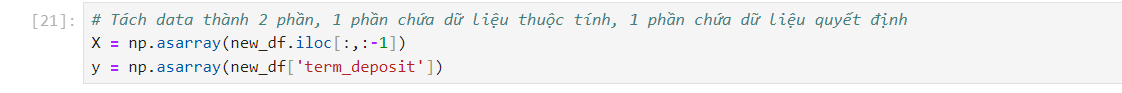
- Dữ liệu mới sau các bước tiền xử lý dữ liệu đã thực hiện ở trên.

A screenshot of a computer

Description automatically generated

Hình 3. 35 Tiền xử lý: Dữ liệu mới sau xử lý

* Tiến hành tách dữ liệu làm 2 phần X và y trong đó: X chứa các thuộc tính độc lập, y chứa thuộc tính quyết định (term\_deposit).



Hình 3. 36 Tiền xử lý: Tách dữ liệu thành X, y

* Từ biến X, y ta tiến hành tách dữ liệu làm 2 phần theo tỉ lệ 8:2. 80% dữ liệu cho phần huấn luyện (train), 20% dữ liệu cho phần kiểm định (test). Lưu kết quả vào các biến tương ứng X\_train. X\_test, y\_train, y\_test.

A screenshot of a computer

Description automatically generated

Hình 3. 37 Tiền xử lý: Tách train & test

# CHƯƠNG 4: ÁP DỤNG CÁC GIẢI THUẬT KHAI THÁC DỮ LIỆU

# 4.5 Giải thuật K-Nearest Neighbor

### 4.5.1 Khái niệm

- Thuật toán K-Nearest Neighbor (KNN) là một thuật toán phân supervised-learning đơn giản nhất (mà hiệu quả trong một vài trường hợp) trong Machine Learning. Khi huấn luyện, thuật toán này không học một điều gì từ dữ liệu training (đây là lý do thuật toán này được xếp vào loại lazy-learning), mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới. KNN có thể được áp dụng cho cả 2 bài toán là Phân lớp (Classification) và Hồi quy (Regression).

- Với KNN, trong bài toán Phân lớp, nhãn của một điểm dữ liệu mới được suy ra trực tiếp từ K điểm dữ liệu gần nó nhất trong tập traning set. Nhãn của một dữ liệu test có thể được quyết định bằng cách bầu chọn giữa các điểm gần nhất (số lần xuất hiện), hoặc cũng có thể được suy ra bằng các đánh trọng số khách nhau cho mỗi trong điểm trong các điểm gần nhất đó rồi suy ra nhãn. Trong bài toán Hồi quy, đầu ra của một điểm dữ liệu được tính bằng chính đầu ra của điểm gần nhất (trong trường hợp K=1), hoặc là trung bình có trọng số của các điểm gần nhất. Một cách ngắn gọn, KNN là thuật toán đi tìm đầu ra của một điểm dữ liệu mới bằng cách chỉ dựa trên thông tin của K điểm dữ liệu trong traning set gần nó nhất (K-lân cận), không quan tâm đến việc có một vài điểm dữ liệu trong những điểm gần nhất này là nhiễu.

### 4.5.2 Lý do lựa chọn

- K-nearest Neighbor là một trong những thuật toán học có giám sát đơn giản nhất được ứng dụng cho các bài toán Phân lớp và Hồi quy. Hoàn toàn đáp ứng được yêu cầu khai thác dữ liệu của đồ án.

- Thời gian huấn luyện của mô hình là bằng 0, KNN được gọi là Lazy Learning vì nó không thực hiện quá trình huấn luyện nào trên dữ liệu huấn luyện. Thay vào đó, nó chỉ lưu trữ dữ liệu huấn luyện trong bộ nhớ và thực hiện tính toán khi có dữ liệu mới.

- Việc dữ đoán kết quả của dữ liệu mới rất đơn giản vì thuật toán này không yêu cầu đào tạo trước khi đưa ra dự đoán. Dữ liệu mới có thể được thêm vào một cách liền mạch mà không làm thay đổi độ chính xác của thuật toán.

- KNN là một mô hình không có tham số (nonparametric). Quá trình tính toán được thực hiện độc lập với mỗi điểm dữ liệu khác nhau. Hơn nữa, thuật toán KNN yêu cầu rất ít tham số, chỉ cần 2 tham số là K – số điểm dữ liệu và một độ đo khoảng cách. Điều này là ít hơn khi so sánh với các thuật toán Machine Learning khác.

### 4.5.3 Áp dụng

Các bước tính toán của thuật toán KNN:

- Đối với thuật toán KNN không đánh trọng số (tức xem các điểm lân cận là như nhau):

* Thực hiện tính toán khoảng cách giữa điểm dữ liệu mới và tất cả các điểm dữ liệu huấn luyện. Độ đo khoảng các thường được sử dụng là khoảng cách Euclidean – là đường thẳng nối giữa 2 điểm tính bằng công thức:
* Tiếp đó, chọn ra k điểm có khoảng cách gần nhất với dữ liệu mới.
* Đếm số lượng xuất hiện của các nhãn giữa k điểm lân cận đó, chọn ra nhãn có số lần xuất hiện lớn nhất là nhãn dự đoán của dữ liệu mới.

- Đối với thuật toán KNN đánh trọng số, không chỉ xem xác điểm lân cận gần nhất mà còn xét xem khoảng cách của chúng tới điểm được xét. Theo cách tiếp cận này, các điểm gần dữ liệu mới hơn thì sẽ được đánh trọng số cao hơn, cách đơn giản nhất là lấy nghịch đảo của khoảng cách giữa điểm đó với điểm dữ liệu mới:

* 2 bước đầu tiên thực hiện tương tự thuật toán KNN không đánh trọng số.
* Tiếp theo ta gán trọng số cho k điểm lân cận bằng nghịch đảo khoảng các giữa chúng với điểm dữ liêu mới.
* Khi đó, sự đóng góp của một điểm dữ liệu thay vì được tính bằng 1 sẽ được tính bằng trọng số của chúng. Nhãn nào có trọng số lớn nhất được chọn làm nhãn dự đoán của dữ liệu mới.

- Chạy thuật toán:

#### 4.5.3.1 Tính toán thủ công

- Bước 1: Định nghĩa một hàm tính khoảng cách Euclidean giữa 2 điểm dữ liệu:

A screenshot of a computer

Description automatically generated

- Bước 2: Đối với KNN không trọng số, định nghĩa một hàm với 2 tham số như sau:

A white screen with black text

Description automatically generated

Hình 4. 1 Thủ công - KNN không trọng số

Tham số thứ nhất point – là điểm dữ liệu mới, k là số lượng điểm dữ liệu gần nhất muốn xét.

* Đầu tiên ta tính khoảng cách từ điểm dữ liệu mới đến toàn bộ dữ liệu train,
* Sau đó lấy ra vị trí của k điểm dữ liệu có khoảng cách gần nhất.
* Tiếp theo ta lấy ra nhãn của k điểm đó, kết quả trả về nhãn dữ liệu có tần suất xuất hiện cao nhất làm nhãn dự đoán cho điểm dữ liệu mới.

- Bước 3: Đối với KNN có trọng số, định nghĩa hàm với 2 trọng số như sau:

A screenshot of a computer program

Description automatically generated

Hình 4. 2 Thủ công - KNN có tham số

* 3 bước đầu thực hiện tương tự hàm KNN không trọng số, tuy nhiên ở bước tiếp theo, ta tạo một mảng weights là trọng số của k điểm lân cận bằng cách lấy nghịch đảo khoảng cách của điểm dữ liệu đó tới điểm dữ liệu mới (+ 1e-5 là để tránh thường hợp chia cho 0) .
* Sau đó tiến hành lọc ra các nhãn duy nhất từ k điểm dữ liệu và tạo một mảng chứa trọng số của các nhãn dữ liệu đó.
* Lần lượt tính trọng số cho mỗi nhãn dữ liệu lưu vào mảng weight\_votes bằng việc tính tổng trọng số của các điểm dữ liệu có nhãn là nhãn dữ liệu đang xét.
* Kết quả trả về nhãn dữ liệu có trọng số cao nhất làm nhãn dự đoán cho dữ liệu mới.

- Bước 4: Thực hiện dự đoán trên tập dữ liệu test với 2 hàm KNN có trọng số và không có trọng số vừa tạo. Ở đây em chỉ thực hiện trên 100 dữ liệu test đầu tiên vì quá trình tính toán từ toàn bộ dữ liệu test, mỗi dữ liệu lại tính khoảng cách tới toàn bộ dữ liệu train là rất lâu. Thực tế, người ta sẽ thường tối ưu việc tìm các điểm gần nhất bằng một số cấu trúc dữ liệu KD-Tree hay Ball-Tree để tăng tốc độ truy vấn các điểm gần nhất. Tuy nhiên trong phạm vi giải thuật thì em sẽ không đề cập đến các khái niệm nâng cao trên.

A computer code on a white background

Description automatically generated

Hình 4. 3 Thủ công – KNN dự đoán

Kết quả cho thấy với 100 đữ liệu test đầu tiên, cả 2 hàm dự đoán KNN có trọng số và không có trọng số đều đạt độ chính xác là 95%. Một kết quả khá tốt.

#### 4.5.3.2 Sử dụng thư viện

- Bước 1: Import thư viện neighbors của sklearn. Ta thực hiện tạo mô hình KNN không có trọng số như sau:

KNeighborsClassifier(n\_neighbors = 10, p = 2)

Trong đó:

* n\_neighbors: là số lượng điểm dữ liệu lân cận muốn xét tương tự k, ở đây là 10.
* p: độ đo khoảng cách được tính, ở đây p=2 tức là tính theo khoảng cách Euclidean giống các phần ở trên.

A white background with colorful text

Description automatically generated

Hình 4. 4 Thư viện – KNN không trọng số

- Bước 2: Ta thực hiện tạo mô hình KNN có trọng số như sau:

KNeighborsClassifier(n\_neighbors = 10, p = 2, weights = 'distance')

Trong đó:

* 2 tham số đầu tiên là tương tự hàm trước đó
* weights: là trọng số được chọn, ở đây là distance, tức lấy nghịch đảo khoảng cách giữa 2 điểm.

A close-up of a computer screen

Description automatically generated

Hình 4. 5 Thư viện – KNN có trọng số

- Bước 3: Thực hiện dự đoán dữ liệu test trên 2 mô hình đã tạo, và đánh giá thời gian chạy thuật toán trên 2 mô hình trên các tập dữ liệu train, test tương ứng.

A screen shot of a computer code

Description automatically generated

Hình 4. 6 Thư viện – KNN dự đoán

Kết quả cho thấy, độ chính xác của 2 mô hình là khá khả quan (trên 90%) trên cả 2 dữ liệu train, test

* Độ chính xác của mô hình KNN không trọng số trên dữ liệu train là 90.83%, trên dữ liệu test là 90.72%.
* Độ chính xác của mô hình KNN có trọng số trên dữ liệu train là 100% (đặc trưng của thuật toán KNN), trên dữ liệu test là 90.69%. Dẫn đếnh hiện tượng Overfitting.
* Tuy nhiên thời gian chạy của mô hình KNN có trọng số (2.36) lâu hơn nhiều so với mô hình KNN không trọng số (0.68s).

- Bước 4: Vẽ ma trận nhầm lẫn của 2 mô hình KNN trên dữ liệu test.

A screenshot of a computer program

Description automatically generated

Hình 4. 7 Thư viện – Bước 4

Kết quả thu được:

A screenshot of a computer

Description automatically generated

A comparison of a graph

Description automatically generated with medium confidence

Hình 4. 8 Thư viện – KNN Ma trận nhầm lẫn

# 4.6 Giải thuật Perceptron Learning Algorithm

### 4.6.1 Khái niệm

- Perceptron Learning Algorithm (PLA) là một trong thuật toán Classification cho trường hợp đơn giản nhất: chỉ có 2 lớp cần phân loại (bài toán chỉ 2 lớp được gọi là binary classification) và cũng chỉ hoạt động trong một trường hợp rất cụ thể. Tuy nhiên nó là nền tảng cho một mảng lớp quan trọng của Machine Learning là Neural Network và sau này là Deep Learning.

- Giả sử chúng ta có 2 tập hợp dữ liệu đã được gán nhãn được minh họa trong hình bên trái dưới đây. Hai lớp của chúng ta là tập các diểm màu xanh và tập các điểm màu đỏ. Thuật toán PLA được hiểu là, chúng ta cần tìm một lãnh thổ cho mỗi lớp sao cho, với mỗi một điểm dữ liệu mới, ta chỉ cần xác định xem nó nằm trong lãnh thổ của lớp nào rồi quyết định nó thuộc lớp đó.

A couple of blue and red dots

Description automatically generated with medium confidence

- Để tìm lãnh thổ của mỗi lớp, ta cần đi tìm biên giới (boundary) giữa 2 lãnh thổ này. Biên giới đơn giản nhất trong không gian hai chiều là một đường thẳng, trong không gian ba chiều là một mặt phẳng, trông không gian nhiều chiều là một siêu mặt phẳng (hyperplane).

- Bài toán Perceptron được phát biểu như sau: Cho hai lớp được gán nhãn, hãy tìm một đường phẳng sao cho toàn bộ các điểm thuộc lớp thứ nhất nằm về một phía, toàn bộ các điểm thuộc lớp thú 2 nằm về phía còn lại của đường thẳng đó. Với giả định rằng tồn tại một đường phẳng như thế. Nếu tồn tại một đường phảng phân chia hai lớp thì ta gọi hai lớp đó là linearly separable. Các thuật toán Phân lớp tạo ra các boundary là các đường phẳng này được gọi chung là Linear Classifier.

### 4.6.2 Lý do lựa chọn

- Thuật toán PLA được sử dụng để phân lớp trong trường hợp đơn giản nhất để phân loại, phù hợp với dữ liệu đồ án gồm 2 lớp để phân loại là 0 – Trường hợp khách hàng không đăng kí gửi tiền có kỳ hạn và 1 – Trường hợp khách hàng đăng kí gửi tiền có kì hạn.

- PLA là thuật toán đơn giản và dễ hiểu, dễ dàng triển khai bằng tính toán thủ công mà không cần các thư viện phức tạp. PLA hoạt động rất tốt với các bài toán phân lớp mà dữ liệu được phân tách tuyến tính, và có chi phí tính toán thấp khi so sánh với các thuật toán phức tạp hơn như Support Vector Machines hoặc các mô hình Neural Network.

- Mục đính chính của việc lựa chọn thuật toán PLA là để làm nền tảng cho nhiều mô hình Neural Network phức tạp hơn. Thuật toán PLA hay còn được gọi là Neural Network 1 lớp, việc tìm hiêu thuật toán này giúp ta có cái nhìn khái quát về Neural Network trước khi triển khai thuật toán Neural Network đa lớp được trình bày trong phần tiếp theo của đồ án.

### 4.6.3 Áp dụng

- Một số ký hiệu :

* là ma trận chứa các điểm dữ liệu với d là số thuộc tính của dữ liệu, và N là số lượng điểm dữ liệu. Vỗi cột là mội điểm dữ liệu trong không gian d chiều.
* là một vector hàng chứa nhãn tương ứng với từng điểm dữ liệu. Với nếu thuộc phân lớp thứ nhất và nếu thuộc phân lớp thứ 2.
* Tại một thời điểm, giả sử ta tìm được boundary là đường phẳng có phương trình:

+

=

Với w là ma trận chứa các hệ số và là điểm dữ liệu mở rộng của X bằng các thêm phần tử = 1 lên trước vector X tương tự như trong thuật toán Logistic Regression.

* Với w là một nghiệm của thuật toán PLA, với mội điểm dữ liệu mới x chưa được gán nhãn, ta có thể xác định phân lớp của nó bằng công thức:

label(x) = sgn()

Trong đó sgn là hàm các định dấu của một số, với giả sử rằng sgn(0)=1.

* Hàm mất mát của thuật toán PLA được xây dựng như sau:

Trong đó M là tập hợp các điểm bị phân lớp sai.

Ta có thể sử dụng Gradient Descent hoặc Stochastic Gradient Descent (SGD) để tối ưu hàm mất mát này. Với ưu điểm của SGD cho các bài toán large-scale, chúng ta sẽ làm theo thuật toán này. Các bước thực hiện của thuật toán PLA:

* Bước 1: Chọn ngẫu nhiên một vector hệ số với các phần tử gần bằng 0.
* Bước 2: Duyện ngẫu nhiên qua từng điểm dữ liệu
  + Nếu được phân lớp đúng, tức sgn() =
  + Ngược lại, ta tính đạo hàm của hàm mất mát theo w

∇w =−

Sau đó tiến hành cập nhật w theo công thức:

w = w +

Với

* Bước 3: Kiểm tra xem có bao nhiêu điểm bị phân lớp sai. Nếu không có điểm nào, dừng thuật toán. Nếu còn, quay lại bước 2.

#### 4.6.3.1 Tính toán thủ công

- Bước 1: Định nghĩa hàm dự đoán nhãn của dữ liệu theo công thức đã trình bày ở trên:

label(x) = sgn()

A close-up of a white background

Description automatically generated

Hình 4. 9 PLA – Thủ công: Bước 1

- Bước 2: Định nghĩa hàm thuật toán Perceptron với 3 tham số như sau:

A screenshot of a computer program

Description automatically generated

Hình 4. 10 PLA – Thủ công: Bước 2

Tham số thứ nhất X– là điểm dữ liệu đầu vào, y là dữ liệu dự đoán, w\_init là hệ số khởi tạo ban đầu. Ở đây thuật toán có thay đổi nhỏ, ta sử dụng một cải tiến của thuật toán PLA là Pocket Algorithm để khắc phục nhược điểm không hội tụ với các dữ liệu gần linearly seperatable của PLA:

* Đầu tiên ta định nghĩa giới hạn số lần lặp (iteration) là 1000 lần, tỉ lệ học (eta) = 1, khởi tạo số biến đếm số lượng điểm dữ liệu tối thiểu được phân loại sai (minMisclassifications) ban đầu với giá trị vô cùng lớn. Và một biến để đếm số lượng điểm dữ liệu bị phân loại sai (misClassifications).
* Ta thực hiện vòng lặp thứ 1 cho đến khi số lượng điểm dữ liệu sai (misClassifications) = 0 hoặc đạt đến số lần lặp (iteration) = 1000.
* Ở mỗi vòng lặp thứ 1, ta thực hiện vòng lặp thứ 2 qua toàn bộ dữ liệu X theo theo một thứ tự ngẫu nhiên. Thực hiện dự đoán nhãn của điểm dữ liệu đó bằng hàm predict() đã định nghĩa trước đó với w được cập nhật gần nhất.
* Nếu điểm dữ liệu bị dự đoán sai, ta thực hiện cập nhật hệ số w theo công thức:

w = w +

và biến đếm số lượng điểm bị phân loại sau lên một.

* Trước khi kết thúc mỗi vòng lặp thứ 1. Ta thực hiện cập nhật lại biến lưu số lượng điểm dữ liệu tối thiểu được phân loại sai (minMisclassifications) nếu misClassifications < minMisclassifications

- Bước 3: Ta biến đổi các dữ liệu train, test ban đầu để phù hợp với đầu vào của thuật toán.

A screenshot of a computer code

Description automatically generated

Hình 4. 11 PLA – Thủ công: Bước 3

Với dữ liệu dự đoán, thực hiện chuyển giá trị của dữ liệu từ (0,1) thành (-1, 1) tương ứng. Với dữ liệu input, thực hiện thêm một phần tử 1 ở trước mỗi điểm dữ liệu.

- Bước 4: Thực hiện khởi tạo giá trị hệ số w ban đầu và chạy thuật toán

A screen shot of a computer code

Description automatically generated

Hình 4. 12 PLA – Thủ công: Bước 4

Kết quả mỗi lần chạy thuật toán:

A white background with black and white clouds

Description automatically generated

Kết quả cuối cùng của thuật toán:

A close up of a computer code

Description automatically generated

Hình 4. 13 PLA – Thủ công: Bước 4 – Kết quả

Từ kết quả, ta thấy rằng:

* Số lượng điểm dữ liệu bị phân loại sai nhỏ nhất là 4471.
* Vector hệ số cuối cùng
* Thời gian chạy thuật toán là 222.56s, một thời gian chạy khá lâu trên tập dữ liệu traning.

- Bước 5: Thực hiện đánh giá độ chính xác của mô hình trên tập dữ liệu train và test.

A screen shot of a computer code

Description automatically generated

Hình 4. 14 PLA – Thủ công: Bước 5

Từ kết quả ta thấy, độ chính xác của mô hình trên 2 tập dữ liệu train và test là khá khả quan (>90%). Với độ chính xác trên tập train là 90.53% và trên tập test có nhỉnh hơn là 90.64%.

#### 4.6.3.2 Sử dụng thư viện

- Bước 1: Import thư viện Perceptron của sklearn.linear\_model. Ta thực hiện tạo mô hình PLA với số lần lặp là 1000 và hệ số học eta =1 như sau:

Perceptron(max\_iter=1000, eta0=1)

A white background with a black and white flag

Description automatically generated with medium confidence

Hình 4. 15 PLA – Thư viện: Bước 1

* Bước 2: Thực hiện dự đoán dữ liệu train và test trên 2 mô hình đã tạo, và đánh giá thời gian chạy thuật toán mô hình.

A computer screen shot of a computer code

Description automatically generated

Hình 4. 16 PLA - Thư viện: Bước 2

Từ kết quả ta thấy, độ chính xác của mô hình trên 2 tập dữ liệu train và test là khá quan(>90%) và khá tương đồng với thuật toán PLA được tính toán thủ công. Với độ chính xác trên tập train là 90.83% và trên tập test có nhỉnh hơn là 90.72%. Tuy nhiên thời gian chạy thuật toán PLA bằng thư viện được thực hiện nhanh hơn nhiều lần so với thuật toán PLA được tính toán thủ công với chỉ 0.02s

* Bước 3: Vẽ ma trận nhầm lẫn của 2 mô hình trên dữ liệu train và test.

A screen shot of a computer code

Description automatically generated

Hình 4. 17 PLA - Thư viện: Bước 3

Kết quả thu được:

A screenshot of a computer

Description automatically generated

A comparison of a graph

Description automatically generated with medium confidence

Hình 4. 18 PLA - Thư viện: Bước 4

# CHƯƠNG 5: ĐÁNH GIÁ CÁC THUẬT KHAI THÁC DỮ LIỆU

## 5.3 Đánh giá về thời gian chạy thuật toán

A graph with text and numbers

Description automatically generated

Hình 1. 6 Đánh giá – Thời gian chạy

* Từ biểu đồ ta có thể thấy, thời gian chạy các thuật toán tương ứng như sau:
* Naive Bayes: 0.04 giây
* Logistic Regression: 0.65 giây
* Decision Tree: 1.03 giây
* Random Forest: 33.72 giây
* K-Nearest Neighbor: 2.36 giây
* Perceptron: 0.05 giây
* Neural Network: 7.68 giây

- Nhận xét:

* + Các thuật toán đơn giản và tuyến tính như Naive Bayes, Logistic Regression và Perceptron có thời gian chạy rất ngắn, thích hợp cho các bài toán với dữ liệu lớn nhưng đơn giản.
  + Các thuật toán phức tạp hơn như Random Forest và Neural Network có thời gian chạy lâu hơn, nhưng thường cho kết quả chính xác hơn trên các bài toán phức tạp và phi tuyến tính.
  + Decision Tree và K-Nearest Neighbor nằm ở khoảng giữa, với Decision Tree là một lựa chọn tốt cho các bài toán cần giải thích rõ ràng và trực quan, trong khi KNN thường được sử dụng cho các bài toán với dữ liệu nhỏ hoặc vừa phải.

## 5.4 Đánh giá về độ chính xác tổng quát của thuật toán (Accuracy)

- Khái niệm: Accuracy (độ chính xác) là một thước đo đánh giá hiệu suất của một mô hình phân loại. Accuracy cho biết tỷ lệ phần trăm các dự đoán đúng của mô hình so với tổng số các dự đoán. Nói cách khác, nó là tỷ lệ giữa số lượng dự đoán đúng (bao gồm cả True Positives và True Negatives) trên tổng số các trường hợp được đánh giá.

- Công thức tính:

=

### 5.4.1 So sánh độ chính xác của tập huấn luyện

A graph of blue rectangular bars

Description automatically generated with medium confidence

Hình 1. 7 Đánh giá – Accuracy trên tập huấn luyện

* Từ biểu đồ ta có thể thấy, độ chính xác trên tập huấn luyện của các thuật toán tương ứng như sau:
* Naive Bayes: 89.95%
* Logistic Regression: 91.23%
* Decision Tree: 91.23%
* Random Forest: 90.72%
* K-Nearest Neighbor: 100.00%
* Perceptron: 90.83%
* Neural Network: 90.73%
* Nhận xét:
* K-Nearest Neighbor có độ chính xác tuyệt đối trên tập huấn luyện, điều này thường là dấu hiệu của overfitting, tức là mô hình học quá chi tiết về dữ liệu huấn luyện và có thể không tổng quát hóa tốt trên dữ liệu kiểm tra.
* Naive Bayes có độ chính xác thấp nhất trong số các thuật toán được so sánh, nhưng vẫn ở mức khá cao, cho thấy đây là một mô hình đơn giản và nhanh, phù hợp cho nhiều loại dữ liệu.
* Logistic Regression và Decision Tree có độ chính xác ngang nhau và cao, cho thấy hiệu suất tốt trên tập huấn luyện. Tuy nhiên, Decision Tree có thể dễ bị overfitting hơn nếu không được cắt tỉa hợp lý.
* Random Forest có độ chính xác cao và ổn định, thường cho thấy khả năng tổng quát hóa tốt hơn so với Decision Tree đơn lẻ.
* Perceptron và Neural Network có độ chính xác gần tương đương với Random Forest, nhưng Neural Network cần nhiều thời gian và tài nguyên hơn để huấn luyện như đã trình bày ở phần đánh giá trước đó.

### So sánh độ chính xác của tập thử nghiệm

A graph of blue rectangular bars

Description automatically generated with medium confidence

Hình 1. 8 Đánh giá – Accuracy trên tập thử nghiệm

* Từ biểu đồ ta có thể thấy, độ chính xác trên tập huấn luyện của các thuật toán tương ứng như sau:
* Naive Bayes: 90.19%
* Logistic Regression: 91.04%
* Decision Tree: 91.03%
* Random Forest: 90.69%
* K-Nearest Neighbor: 90.69%
* Perceptron: 90.72%
* Neural Network: 90.62%

- Nhận xét:

* Naive Bayes: Hiệu suất tốt và nhất quán giữa tập huấn luyện và tập kiểm định.
* Logistic Regression và Decision Tree: Hiệu suất cao và ổn định, mặc dù Decision Tree có thể bị overfitting nhẹ.
* Random Forest: Hiệu suất tốt và ổn định, ít bị overfitting.
* K-Nearest Neighbor: Bị overfitting nặng trên tập huấn luyện, không tổng quát hóa tốt.
* Perceptron và Neural Network: Hiệu suất ổn định và có khả năng tổng quát hóa, nhưng Neural Network có thể cần thêm tinh chỉnh để đạt độ chính xác cao hơn.

## 5.7 Nhận xét các thuật toán

### 5.7.5 K-nearest Neighbor

- Từ kết quả của các độ đo đánh giá, so sánh các thuật toán ở trên, ta có thể suy ra nhận xét về ưu, nhược điểm của thuật toán KNN trên tập dữ liệu như sau:

- Ưu điểm:

* Đơn giản và dễ hiểu: KNN là một thuật toán không tham số và dễ hiểu. Nó chỉ đơn giản dựa trên khoảng cách đến các điểm lân cận.
* Hiệu suất cao trên tập huấn luyện: KNN đạt độ chính xác tuyệt đối trên tập huấn luyện, cho thấy nó có khả năng học tốt các mẫu huấn luyện.
* Nhược điểm:
* Thời gian chạy: KNN có thời gian chạy tương đối dài (2.36 giây). Điều này là do KNN phải tính khoảng cách đến tất cả các điểm dữ liệu trong tập huấn luyện mỗi khi dự đoán, khiến nó không hiệu quả trên các tập dữ liệu lớn.
* Overfitting: Độ chính xác tuyệt đối trên tập huấn luyện (100.00%) và giảm đáng kể trên tập kiểm định (90.69%) cho thấy KNN bị overfitting. Nó học quá chi tiết về dữ liệu huấn luyện và không tổng quát hóa tốt trên dữ liệu mới.
* Nhạy cảm với nhiễu: KNN dễ bị ảnh hưởng bởi nhiễu trong dữ liệu vì nó xét các điểm gần kề, bao gồm cả các điểm nhiễu.

### 5.7.6 Perceptron Learning Algorithm

- Từ kết quả của các độ đo đánh giá, so sánh các thuật toán ở trên, ta có thể suy ra nhận xét về ưu, nhược điểm của thuật toán PLA trên tập dữ liệu như sau:

- Ưu điểm:

* Thời gian chạy: Perceptron có thời gian chạy rất ngắn (0.05 giây), cho thấy nó là một thuật toán hiệu quả về mặt thời gian.
* Hiệu suất ổn định: Độ chính xác trên tập huấn luyện (90.83%) và tập kiểm định (90.72%) tương đối gần nhau, cho thấy Perceptron không bị overfitting và tổng quát hóa tốt trên dữ liệu mới.
* Đơn giản và nhanh chóng: Là một thuật toán tuyến tính đơn giản, Perceptron dễ dàng triển khai và tính toán nhanh chóng.
* Nhược điểm:
* Khả năng biểu diễn hạn chế: Perceptron chỉ có thể phân loại các dữ liệu chỉ gồm 2 phân lớp và có thể tách biệt tuyến tính. Nó không hiệu quả với các bài toán phức tạp gồm nhiều phân lớp hoặc không tuyến tính.
* Hiệu suất thấp hơn trên một số dữ liệu phức tạp: Mặc dù độ chính xác của Perceptron ổn định, nó có thể không đạt hiệu suất cao như các mô hình phức tạp hơn trên các tập dữ liệu phức tạp hơn.

# CHƯƠNG 6: TỔNG KẾT

## 6.1 Kết quả đạt được

- Về cơ bản, trong đồ án lần này, nhóm chúng em đã triển khai được toàn bộ quá trình của một đồ án khai thác dữ liệu. Đi từ những bước cơ bản nhất như mô tả được dữ liệu cũng như mô tả được bài toán mà đồ án cần giải quyết được. Tiếp theo là biết cách sử dụng các công cụ, thư viện của Python để trực quan hóa dữ liệu trước tiền xử lý, giúp chúng em có được cái nhìn khái quát về tập dữ liệu trước khi đi sâu bên trong.

- Hiểu rõ quy trình tiền xử lý dữ liệu bao gồm các bước được triển khai trong chương 3:

* Làm sạch dữ liệu: Loại bỏ thuộc tính không cần thiết, Nhận diện và xử lý dữ liệu bị nhiễn, Nhận diện dữ liệu bị thiếu, Ép kiểu thuộc tính dữ liệu, Xử lý ý nghĩa dữ liệu, Nhận diện dữ liệu trùng lặp.
* Tích hợp dữ liệu.
* Biến đổi và thu giảm chiều dữ liệu: Thu giảm chiều dữ liệu, Chuẩn hóa dữ liệu.
* Rời rạc hóa dữ liệu.
* Tách dữ liệu huấn luyện và kiểm thử.

- Từ mục tiêu của bài toán dự đoán khả năng khách hàng đăng ký gửi tiền có kỳ hạn, chúng em đã đề xuất 7 thuật toán phân lớp từ cơ bản đến phức tạp trong Machine Learning. Quá trình triển khai các thuật toán được đi từ việc học tập, hiểu các khái niệm, đặc điểm của mỗi thuật toán cũng như công thức tính toán đến việc triển khai các thuật toán trên Python với 2 phương pháp: Tính toán thủ công và Sử dụng thư viện hỗ trợ:

* Naive Bayes
* Logistic Regression
* Decision Tree
* Random Forest
* K-Nearest Neighbor
* Perceptron Learning Algorithm
* Neural Network

- Kết hợp với việc triển khai các thuật toán, nhóm em đã đưa ra các độ đo để đánh giá, so sánh cũng như rút ra nhận xét về ưu, nhược điểm của các thuật toán trên tập dữ liệu:

* Ma trận nhầm lẫn
* Đánh giá về tỉ lệ TPR trên tỉ lệ FPR
* Đánh giá về thời gian chạy thuật toán
* Đánh giá về độ chính xác tổng quát của thuật toán (Accuracy)
* Đánh giá về độ chính xác của thuật toán (Precision)
* Đánh giá về độ nhậy của thuật toán (Recall)

## 6.2 Hướng phát triển

- Từ kết quả đạt được, về cơ bản nhóm em đã đưa ra được các thuật toán phân lớp để dự đoán khả năng gửi tiền có kì hạn của khách hàng trên tập dữ liệu đơn giản. Tuy nhiên vì thời gian học tập và nghiên cứu hạn chế nên nhóm em chỉ triển khai các thuật toán tổng quát và tương đối đơn giản để phục vụ đồ án.

- Để tiếp tục phát triển đồ án này, một số hướng nghiên cứu và cải tiến có thể được thực hiện trong tương lai:

* Cải thiện KNN: Áp dụng các kỹ thuật giảm nhiễu và chọn lựa số lượng lân cận (k) phù hợp để giảm thiểu overfitting và cải thiện khả năng tổng quát hóa của KNN.
* Tối ưu hóa Neural Network: Điều chỉnh các siêu tham số của mô hình Neural Network và sử dụng các kỹ thuật như dropout, batch normalization để cải thiện hiệu suất và giảm thiểu overfitting.
* Tăng cường dữ liệu (Data Augmentation): Sử dụng các phương pháp tăng cường dữ liệu để cải thiện khả năng học của các mô hình, đặc biệt là các mô hình phức tạp như Neural Network.
* Kết hợp nhiều mô hình (Ensemble Methods): Sử dụng các phương pháp kết hợp nhiều mô hình như boosting, bagging hoặc stacking để cải thiện hiệu suất và độ chính xác của các mô hình.
* Đánh giá trên nhiều tập dữ liệu: Kiểm tra và đánh giá các mô hình trên nhiều tập dữ liệu khác nhau để đảm bảo tính tổng quát hóa và hiệu suất ổn định của các mô hình.
* Triển khai hệ thống thực tế: Xây dựng một hệ thống dự đoán thực tế dựa trên mô hình đã huấn luyện và kiểm định, đồng thời theo dõi và cập nhật mô hình dựa trên dữ liệu mới thu thập được.

# BẢNG PHÂN CHIA CÔNG VIỆC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nội Dung** | **Nguyễn Hoàng Việt** | **Nguyễn Hoàng Phúc** | **Nguyễn Văn Phát** | **Lê Quang Nhân** |
| Chương 1.1 + 1.2 + 1.4 |  |  | X |  |
| Chương 1.3.1 |  | X |  |  |
| Chương 1.3.2 |  |  |  | X |
| Chương 2.1 + 2.2 |  | X |  |  |
| Chương 2.3+2.4 |  |  | X |  |
| Chương 2.5+2.6 | X |  |  |  |
| Chương 2.7+2.8 |  |  |  | X |
| Chương 3 | X |  |  |  |
| Chương 4.1 + 4.2 |  | X |  |  |
| Chương 4.3 + 4.4 |  |  | X |  |
| Chương 4.5 + 4.6 | X |  |  |  |
| Chương 4.7 |  |  |  | X |
| Chương 5.1 + 5.2 |  |  | X |  |
| Chương 5.3 + 5.4 | X |  |  |  |
| Chương 5.5 |  | X |  |  |
| Chương 5.6 |  |  |  | X |
| Chương 5.7 | X | X | X | X |
| Chương 6 | X | X | X | X |
| **Hoàn thành** | **100%** | **100%** | **100%** | **100%** |

# DANH MỤC TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | T. Vu, “Xử lý các giá trị ngoại lệ¶,” 2021. [Trực tuyến]. Available: https://machinelearningcoban.com/tabml\_book/ch\_data\_processing/process\_outliers.html. [Đã truy cập 28 4 2024]. |
| [2] | T. V. Huu, “Bài 6: K-nearest neighbors,” 8 1 2017. [Trực tuyến]. Available: https://machinelearningcoban.com/2017/01/08/knn/. |
| [3] | L. T. Illuri, “Understanding Weighted k-Nearest Neighbors (k-NN) Algorithm,” 26 9 2023. [Trực tuyến]. Available: https://medium.com/@lakshmiteja.ip/understanding-weighted-k-nearest-neighbors-k-nn-algorithm-3485001611ce. |
| [4] | “Mô hình phân lớp Naive Bayes,” [Trực tuyến]. Available: https://viblo.asia/p/mo-hinh-phan-lop-naive-bayes-vyDZO0A7lwj. |
| [5] | “Logistic Regression (Hồi quy Logistic),” [Trực tuyến]. Available: https://trituenhantao.io/machine-learning-co-ban/bai-6-logistic-regression-hoi-quy-logistic/. |
| [6] | T. V. Huu, “Bài 9: Perceptron Learning Algorithm,” 21 1 2017. [Trực tuyến]. Available: https://machinelearningcoban.com/2017/01/21/perceptron. |
| [7] | S. Tinani, “Understanding and Implementing the Perceptron and Pocket Algorithm,” 24 9 2018. [Trực tuyến]. Available: https://simran-tinani.medium.com/the-perceptron-learning-algorithm-is-one-of-the-simplest-machine-learning-algorithms-and-a-6ab04c5772ad. |