ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KĨ THUẬT MÁY TÍNH

DATA MINING (CO3029)

Bài tập lớn - Học kì 251

DỰ ĐOÁN KHẢ NĂNG KHÁCH HÀNG ĐĂNG KÝ GỬI TIỀN CÓ KỲ HẠN

Advisor(s): Thầy Đỗ Thanh Thái

Student(s): Trần Đức Trí Cường - 2210443
Phạm Ngọc Long - 2211894
Bùi Trọng Văn - 2213915

THÀNH PHỐ HỒ CHÍ MINH, THÁNG 11 - 2025

Contents

Lời nói đầu

Nhóm xin chân thành cảm ơn thầy Đỗ Thanh Thái đã phụ trách giảng dạy, cung cấp một số kiến thức nền tảng làm bước đệm để nhóm hoàn thành bài tập lớn học kỳ 251.

Quá trình làm bài tập lớn cũng là quá trình tự học đối với từng thành viên của nhóm. Nhóm đã trau dồi được các kiến thức về khai phá dữ liệu, học máy, giải thuật, kĩ năng làm việc nhóm,... Bên cạnh đó, nhóm còn có cơ hội rèn luyện cách trình bày, báo cáo kết quả một cách khoa học và logic.

Dù gặp không ít khó khăn trong quá trình thực hiện, nhưng nhờ vào sự giúp đỡ, hỗ trợ lẫn nhau, nhóm đã có thể vượt qua và hoàn thành bài tập lớn một cách tốt nhất có thể. Đây là một trải nghiệm quý báu giúp nhóm không chỉ củng cố kiến thức chuyên môn mà còn phát triển thêm nhiều kỹ năng quan trọng cho công việc và học tập sau này.

Nhóm xin gửi lời cảm ơn chân thành đến thầy và hy vọng sẽ tiếp tục nhận được sự hướng dẫn, chia sẻ kiến thức từ thầy trong những bài tập lớn tiếp theo.

1 Danh sách thành viên & Phân chia công việc

STT	Họ và tên	MSSV	Nội dung	Mức độ hoàn thành
1	Trần Đức Trí Cường	2210443	Làm model, soạn report, slide	100%
2	Phạm Ngọc Long	2211894	Xử lí, so sánh, nhận xét dữ liệu, soạn report	100%
3	Bùi Trọng Văn	2213915	Xử lí, so sánh, nhận xét dữ liệu, soạn report	100%

2 Giới thiệu

2.1 Bối cảnh nghiên cứu

Trong bối cảnh ngân hàng bán lẻ cạnh tranh ngày càng gay gắt, các tổ chức tài chính phải liên tục triển khai các chiến dịch marketing trực tiếp (đặc biệt là telemarketing qua điện thoại) để giới thiệu sản phẩm tiền gửi có kỳ hạn cho khách hàng hiện hữu và khách hàng tiềm năng. Tuy nhiên, việc gọi điện "đại trà" cho toàn bộ danh sách khách hàng vừa tốn kém chi phí nhân sự, vừa gây phiền hà cho những khách hàng không có nhu cầu thực sự. Do đó, nhu cầu phân tích, xử lý dữ liệu, ứng dụng các mô hình học máy để dự đoán trước nhóm khách hàng có khả năng đăng ký tiền gửi có kỳ hạn là rất cấp thiết.

2.2 Mục tiêu nghiên cứu

Dự án này tập trung vào việc phân tích, xử lý dữ liệu, sử dụng các mô hình học máy để dự đoán biến mục tiêu `y` (khách hàng có đăng ký tiền gửi có kỳ hạn hay không, với hai giá trị `yes` và `no`) dựa trên tập dữ liệu `train.csv` của cuộc thi *Marketing Dataset* trên Kaggle. Cụ thể, các mục tiêu chính bao gồm:

- Phân tích, so sánh, khảo sát và tiền xử lý dữ liệu khách hàng từ tập `train.csv` (xử lý giá trị thiếu, mã hoá biến phân loại, chuẩn hoá/chuẩn chỉnh dữ liệu nếu cần)
- Xây dựng và so sánh nhiều mô hình(Logistic Regression, KNN, SVM, Decision Tree, Random Forest, ...)
- Đánh giá hiệu suất của các mô hình dự đoán

2.3 Bộ dữ liệu

Link Marketing dataset: https://www.kaggle.com/competitions/marketing-data/overview

Bộ dữ liệu được sử dụng trong dự án này xuất phát từ các chiến dịch marketing trực tiếp qua điện thoại của một ngân hàng tại Bồ Đào Nha . Dữ liệu bao gồm:

- Số lượng mẫu: 3,000 mẫu
- Số lượng biến: 21 biến (20 biến đầu vào và 1 biến mục tiêu `y`)
- **Biến mục tiêu:**
  - `y`: Khách hàng có đăng ký tiền gửi có kỳ hạn hay không (`yes` / `no`) ?
- **Nhóm biến đầu vào chính:**
  - `age`: tuổi khách hàng (số)
  - `job`: nghề nghiệp (admin., blue-collar, services, ...)
  - `marital`: tình trạng hôn nhân (single, married, divorced)
  - `education`: trình độ học vấn (basic, high.school, university.degree, ...)
  - `default`: có nợ xấu hay không (yes/no)
  - `housing`: có vay mua nhà hay không (yes/no)
  - `loan`: có vay tiêu dùng hay không (yes/no)
  - `contact`: kênh liên hệ (cellular, telephone)
  - `month`: tháng liên hệ cuối cùng (jan, feb, ..., dec)
  - `day_of_week`: ngày trong tuần liên hệ cuối cùng (mon, tue, ..., fri)
  - `duration`: thời lượng cuộc gọi cuối cùng (giây)
  - `campaign`: số lần liên hệ trong chiến dịch hiện tại với khách hàng này
  - `pdays`: số ngày kể từ lần liên hệ trước đó (999 nếu chưa từng liên hệ)

- `previous`: số lần liên hệ trước chiến dịch hiện tại
- `poutcome`: kết quả của chiến dịch marketing trước đó (success, failure, nonexistent)
- `emp.var.rate`: tỷ lệ biến động việc làm (chỉ báo hàng quý)
- `cons.price.idx`: chỉ số giá tiêu dùng (hàng tháng)
- `cons.conf.idx`: chỉ số niềm tin người tiêu dùng (hàng tháng)
- `euribor3m`: lãi suất Euribor 3 tháng (hàng ngày)
- `nr.employed`: số lượng lao động (chỉ báo hàng quý)

2.4 Ý nghĩa thực tiễn

Nghiên cứu và mô hình hoá bộ dữ liệu marketing ngân hàng này có ý nghĩa thực tiễn quan trọng:

- Giúp ngân hàng nhận diện tốt hơn nhóm khách hàng tiềm năng có khả năng đăng ký tiền gửi, từ đó tối ưu hoá danh sách gọi điện trong các chiến dịch telemarketing
- Góp phần giảm chi phí chiến dịch (ít cuộc gọi lãng phí hơn), tăng tỷ lệ chuyển đổi, nâng cao hiệu quả sử dụng nguồn lực nhân viên
- Cung cấp cơ sở dữ liệu định lượng để thiết kế chiến lược marketing cá nhân hoá, theo từng phân khúc khách hàng khác nhau

2.5 Cấu trúc báo cáo

Báo cáo được tổ chức thành các phần chính sau:

- Phần 1: Giới thiệu đề tài và bài toán dự đoán
- Phần 2: Khảo sát và tiền xử lý dữ liệu
- Phần 3: Xây dựng các mô hình học máy
- Phần 4: Đánh giá và so sánh mô hình
- Phần 5: Kết luận và hướng phát triển

3 Khảo sát và tiền xử lý dữ liệu

3.1 Tổng quan dữ liệu

Bộ dữ liệu train có 2,999 mẫu với 21 cột (20 features và 1 target). Sau khi check sơ bộ thì nhóm thấy:

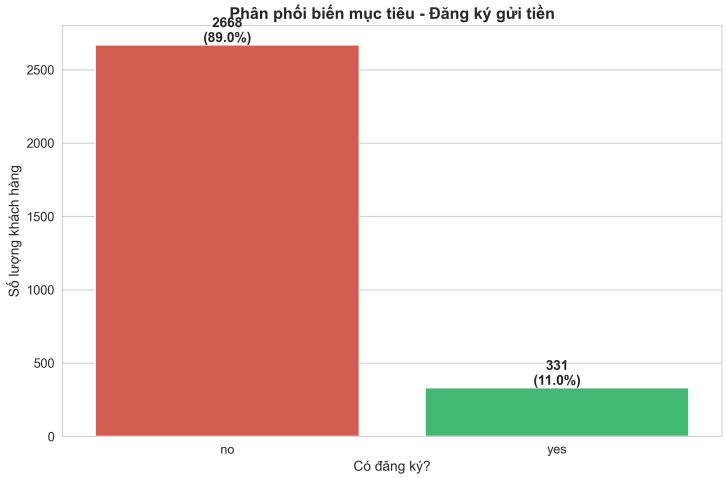
- **Không có missing data:** May mắn là tất cả các cột đều đầy đủ giá trị
- **Dấu phân cách:** File dùng dấu chấm phẩy (;) chứ không phải dấu phẩy như thường lệ
- **Kiểu dữ liệu:** Có cả số (int64, float64) và chữ (object)

3.2 Biến mục tiêu - vấn đề mất cân bằng

Biến `y` (có đăng ký hay không) bị lệch rất nhiều:

- `no`: 2,668 người (88.96%)
- `yes`: 331 người (11.04%)
- Tỷ lệ: 8.06:1 (no nhiều gấp 8 lần yes)

Dữ liệu mất cân bằng kiểu này cần xử lý bằng SMOTE, nếu không model sẽ thiên về predict toàn "no".



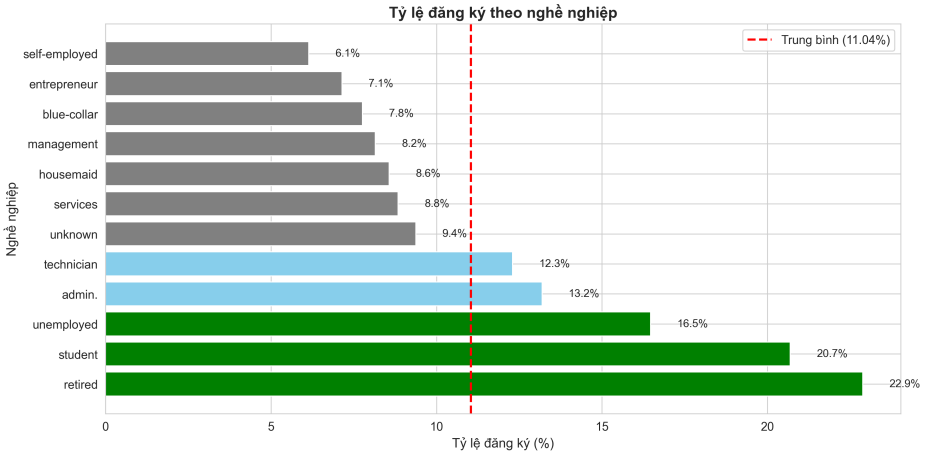
Hình 1: Phân phối biến mục tiêu - mất cân bằng nghiêm trọng (8:1)

3.3 Các biến phân loại quan trọng

3.3.1 Nghề nghiệp (job)

Nhóm phát hiện ra nghề nghiệp ảnh hưởng khá nhiều đến việc đăng ký:

- **Top 3 đăng ký nhiều nhất:**
  - Retired (về hưu): 22.88%
  - Student (sinh viên): 20.69%
  - Unemployed (thất nghiệp): 16.47%
- **Đăng ký ít:** Blue-collar (7.76%), self-employed (6.14%)
- Người về hưu và sinh viên có tỷ lệ đăng ký cao gấp đôi mức trung bình

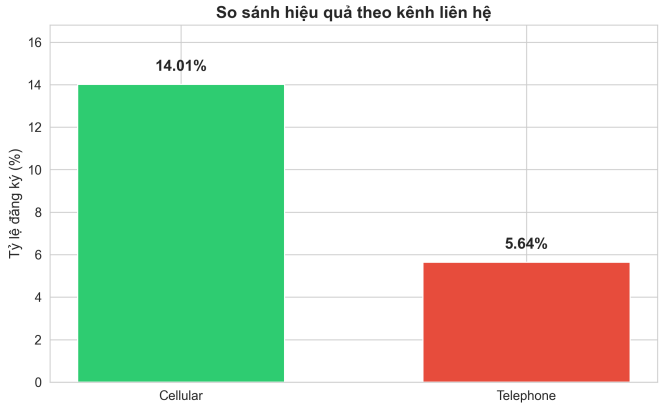


Hình 2: Tỷ lệ đăng ký theo nghề nghiệp - retired và student cao nhất

3.3.2 Kênh liên hệ (contact)

Gọi điện thoại di động có vẻ tốt hơn gọi bàn:

- **Cellular**: 14.01% đăng ký
- **Telephone**: 5.64%
- Cellular hiệu quả hơn **2.48 lần**

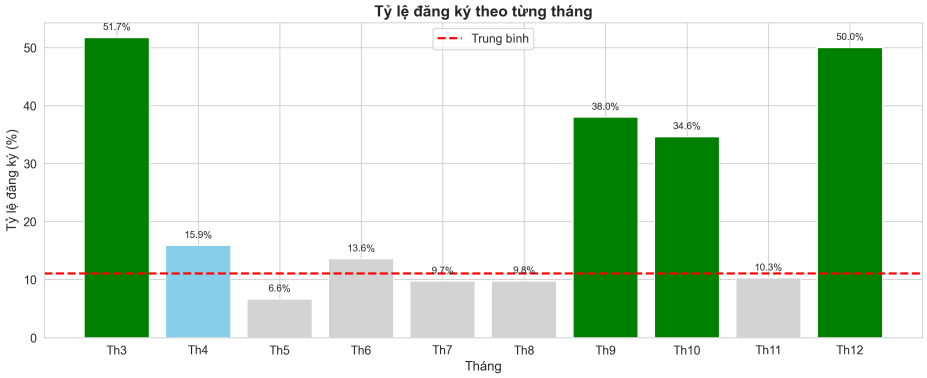


Hình 3: So sánh hiệu quả giữa cellular và telephone

3.3.3 Tháng liên hệ (month)

Tháng gọi cũng có mức độ ảnh hưởng khá lớn:

- **Tháng tốt:** March (51.72%), December (50%), September (38%), October (34.62%)
- **Tháng trung bình:** April (15.89%), June (13.58%)
- **Tháng tệ:** May (6.64%), July (9.73%), August (9.76%)

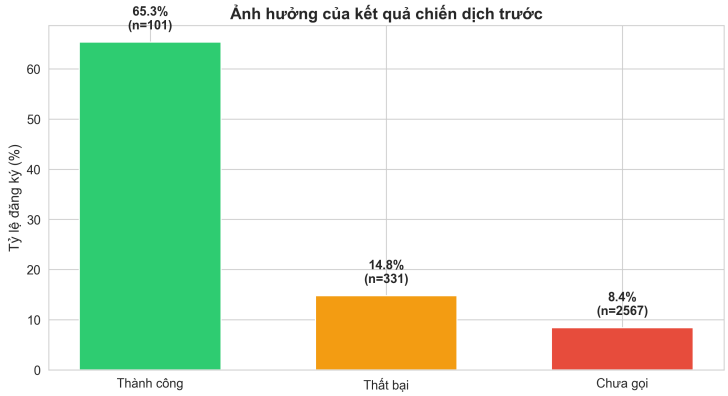


Hình 4: Hiệu quả theo từng tháng - tháng 3, 9, 12 tốt nhất

3.3.4 Kết quả lần gọi trước (poutcome)

Đây là biến rất quan trọng:

- **Success** (lần trước thành công): 65.35% đăng ký tiếp
- **Failure** (lần trước thất bại): 14.80%
- **Nonexistent** (chưa gọi bao giờ): 8.41%
- Khách đã thành công trước có tỷ lệ đăng ký cao gấp gần 8 lần so với khách mới



Hình 5: Impact của kết quả chiến dịch trước - success rate 65%

3.3.5 Giá trị 'unknown'

Một số cột có giá trị 'unknown':

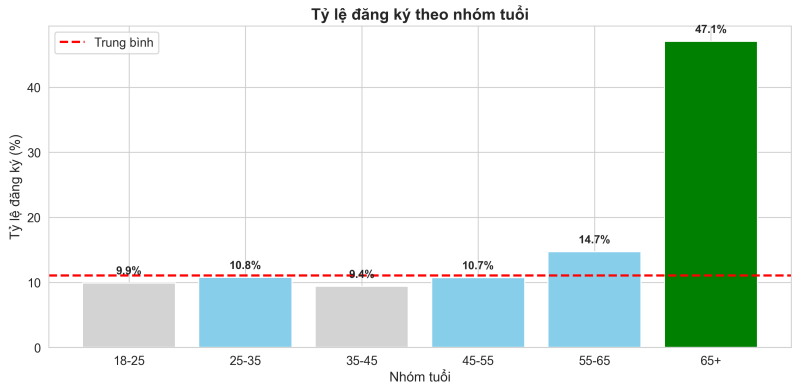
- `default`: 596 người (19.87%)
- `education`: 138 người (4.60%)
- `housing`, `loan`: 77 người (2.57%)
- **Xử lý:** Giữ nguyên 'unknown' như 1 category riêng

3.4 Các biến số

3.4.1 Tuổi (age)

Tuổi cũng ảnh hưởng khá nhiều:

- Trung bình: 39.9 tuổi, trung vị: 38.0 tuổi
- **Chia theo nhóm tuổi:**
  - 18-25: 9.92%
  - 25-35: 10.77%
  - 35-45: 9.38%
  - 45-55: 10.73%
  - 55-65: 14.73%
  - **65+**: 47.06%
- Người già (65+) có tỷ lệ đăng ký cao gấp gần 5 lần mức trung bình

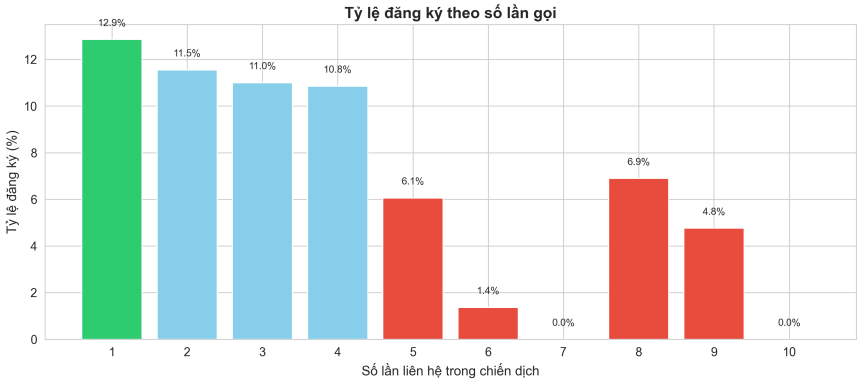


Hình 6: Tỷ lệ đăng ký theo nhóm tuổi - 65+ cao nhất (47%)

3.4.2 Số lần gọi trong chiến dịch (campaign)

Số lần gọi càng nhiều thì tỷ lệ càng giảm:

- **Lần 1**: 12.85%
- **Lần 2**: 11.55%
- **Lần 3**: 11.00%
- **Lần 4**: 10.85%
- **Từ lần 5 trở đi**: giảm mạnh (dưới 7%)
- Gọi nhiều quá làm khách phiền, phản tác dụng



Hình 7: Tỷ lệ đăng ký giảm dần theo số lần gọi - lần đầu tốt nhất

3.4.3 Thời lượng cuộc gọi (duration)

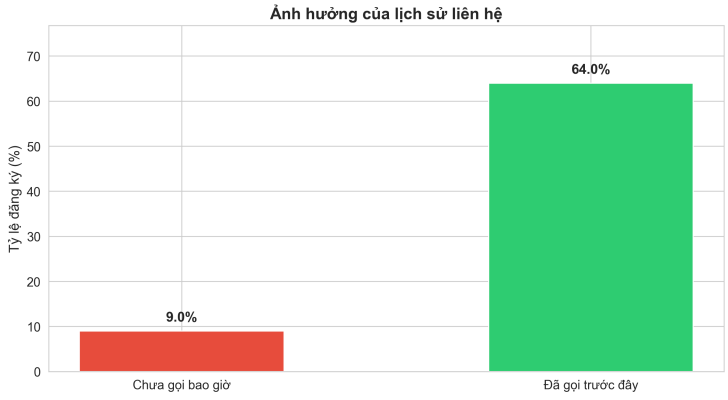
Duration có tương quan mạnh với kết quả:

- Trung bình (no): 216 giây
- Trung bình (yes): 570 giây
- Chênh lệch: 354 giây
- **Vấn đề Data Leakage:**
  - Duration chỉ biết *sau khi* gọi xong
  - Không thể biết trước khi gọi
  - **Quyết định:** Bỏ biến này để tránh data leakage

3.4.4 Lịch sử liên hệ (pdays, previous)

Khách cũ dễ đăng ký hơn khách mới rất nhiều:

- **Chưa gọi bao giờ** (pdays=999): 9.00% (2,888 người)
- **Đã gọi trước đây** (pdays $\neq$ 999): 63.96% (111 người)
- Khách cũ có tỷ lệ thành công cao gấp **7.1 lần**



Hình 8: Khách đã gọi trước đây có tỷ lệ đăng ký cao gấp 3 lần

3.5 Tương quan giữa các biến

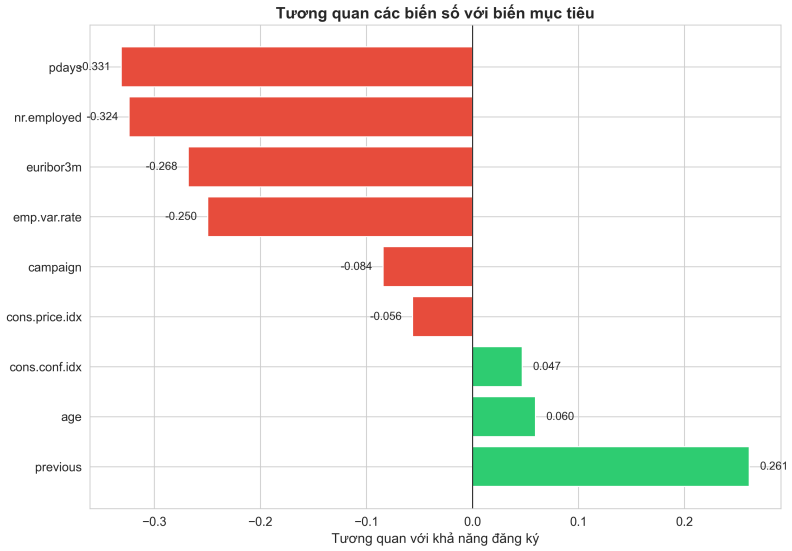
3.5.1 Tương quan với target

Top 5 biến số có tương quan mạnh nhất (bỏ duration):

1. `previous`: +0.2611 (tương quan dương)
2. `age`: +0.0596
3. `cons.conf.idx`: +0.0470
4. `cons.price.idx`: -0.0565
5. `campaign`: -0.0842

Các biến kinh tế có tương quan âm mạnh:

- `nr.employed`: -0.3237
- `pdays`: -0.3312 (không liên hệ gần đây)
- `euribor3m`: -0.2679
- `emp.var.rate`: -0.2496



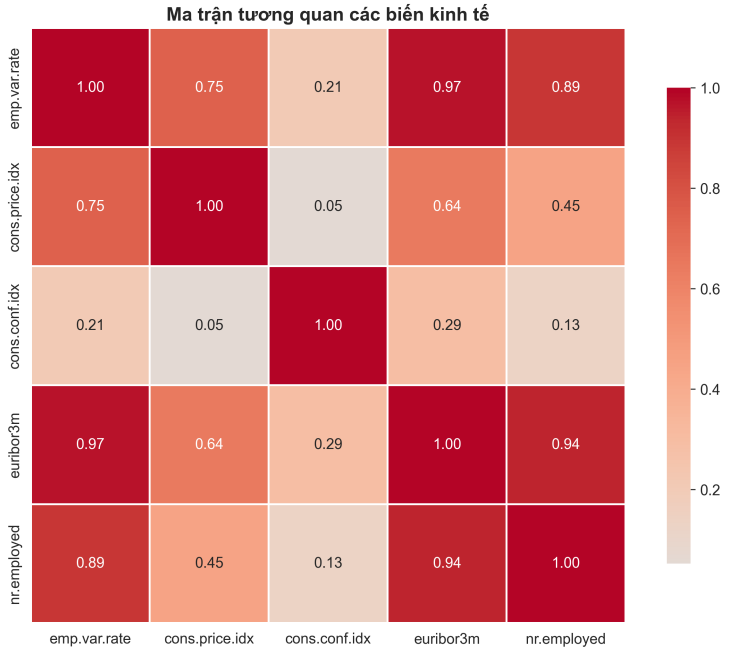
Hình 9: Tương quan các biến số với biến mục tiêu

3.5.2 Vấn đề multicollinearity

Các biến kinh tế tương quan với nhau quá cao (>0.7):

- `emp.var.rate` ↔ `euribor3m`: 0.970
- `euribor3m` ↔ `nr.employed`: 0.940
- `emp.var.rate` ↔ `nr.employed`: 0.892
- `emp.var.rate` ↔ `cons.price.idx`: 0.746

Giải pháp: Bỏ 3 biến (`emp.var.rate`, `cons.price.idx`, `cons.conf.idx`), chỉ giữ `euribor3m` và `nr.employed`.



Hình 10: Ma trận tương quan các biến kinh tế - multicollinearity cao

3.6 Tổng kết phần khảo sát

Sau khi phân tích kỹ, nhóm rút ra được:

1. **Features quan trọng:** poutcome (kết quả lần trước), pdays, previous
2. **Khách hàng tiềm năng:** Người già (65+), về hưu, sinh viên
3. **Cách liên hệ tốt:** Dùng cellular, gọi tháng 3/9/12, không gọi quá nhiều lần
4. **Cần xử lý:** Dữ liệu mất cân bằng (8:1), data leakage (duration), multicollinearity (biến kinh tế)

3.7 Tiền xử lý dữ liệu

Dựa vào những gì phân tích được, nhóm làm các bước sau:

3.7.1 Bước 1: Bỏ biến gây data leakage

- Bỏ `duration` vì chỉ biết sau khi gọi xong

3.7.2 Bước 2: Xử lý multicollinearity

- Bỏ 3 biến kinh tế: `emp.var.rate`, `cons.price.idx`, `cons.conf.idx`
- Giữ lại: `euribor3m`, `nr.employed`

3.7.3 Bước 3: Tạo features mới

- **Nhóm tuổi:** Chia `age` thành 6 nhóm [18-25, 25-35, 35-45, 45-55, 55-65, 65+]
- **Log transform:** Dùng log1p cho `campaign` và `previous` để giảm skewness

3.7.4 Bước 4: Encode categorical features

Nhóm dùng cách encode kết hợp:

- **One-Hot Encoding** cho 4 biến quan trọng:
  - `poutcome`: failure, nonexistent, success
  - `contact`: cellular, telephone
  - `month`: jan, feb, ..., dec
  - `age_group`: 18-25, 25-35, ..., 65+
  - → Tạo ra 20 cột
- **Label Encoding** cho 7 biến còn lại:
  - `job`, `marital`, `education`, `default`, `housing`, `loan`, `day_of_week`
  - → 7 cột

3.7.5 Bước 5: Encode target

- Đổi `y` từ ['yes', 'no'] sang [1, 0]

3.7.6 Kết quả

Sau khi xử lý xong:

- **Tổng cộng 36 features:**
  - 20 cột từ one-hot
  - 7 cột từ label encoding
  - 6 cột số gốc: age, campaign, pdays, previous, euribor3m, nr.employed
  - 2 cột mới: campaign_log, previous_log
- **Shape:** (2999, 37) kể cả target

3.8 Chuẩn bị dữ liệu cho model

3.8.1 Chia train-test

- Train: 80% (2,399 mẫu)
- Test: 20% (600 mẫu)
- Dùng `stratify=Y` để tỷ lệ yes/no đều nhau ở 2 tập

3.8.2 Chuẩn hóa (Standardization)

- Dùng `StandardScaler` đưa tất cả features về cùng scale (mean=0, std=1)
- Cần thiết vì một số model (KNN, SVM, Neural Network) nhạy cảm với scale
- Fit trên train, rồi transform cả train và test

3.8.3 Xử lý imbalance với SMOTE

Dùng SMOTE chỉ cho training set:

- **Trước SMOTE:**
  - Class 0 (no): 2,134 mẫu (89%)

  - Class 1 (yes): 265 mẫu (11%)
- **Sau SMOTE:**
  - Class 0 (no): 2,134 mẫu (50%)
  - Class 1 (yes): 2,134 mẫu (50%)
  - Tổng: 4,268 mẫu
- **Tại sao chỉ SMOTE train:**
  - Test phải giữ nguyên phân phối thực để đánh giá đúng
  - SMOTE tạo thêm mẫu giả cho class thiểu số

3.9 Tổng kết

Tóm lại quy trình của nhóm:

1. Bỏ duration (data leakage) và 3 biến kinh tế (multicollinearity)
2. Tạo features mới: age_group, log transform cho campaign và previous
3. Encode: One-hot cho 4 biến quan trọng, Label cho 7 biến còn lại
4. Chia train-test 80-20
5. StandardScaler chuẩn hóa
6. SMOTE cân bằng train set

Kết quả cuối:

- 36 features sạch sẽ
- Train set cân bằng (4,268 mẫu)
- Test set giữ nguyên (600 mẫu)

4 Xây dựng các mô hình học máy

Nhóm sử dụng 9 mô hình học máy sau để dự đoán cho bài toán đã giới thiệu ở trên:

- KNN
- Logistic Regression
- SVM
- Decision Tree
- Random Forest
- XGBoost
- Gradient Boosting
- Naivebayes
- MLP

Các mô hình được đánh giá bởi các thông số sau:

- Accuracy
- Precision
- Recall
- F1-score
- Confussion Matrix

4.1 Mô hình KNN

4.1.1 Giới thiệu

Mô hình K–Nearest Neighbors (KNN) là một thuật toán phân loại dựa trên "láng giềng gần nhất". Ý tưởng chính là, để dự đoán nhãn của một mẫu mới, mô hình sẽ tìm k điểm dữ liệu gần nhất trong tập huấn luyện (theo một khoảng cách, thường dùng Euclid) và gán nhãn theo đa số phiếu của các láng giềng này. KNN là mô hình phi tham số, đơn giản, dễ cài đặt, nhưng có thể tốn thời gian khi dự đoán nếu số lượng mẫu lớn.

4.1.2 Kết quả huấn luyện mô hình

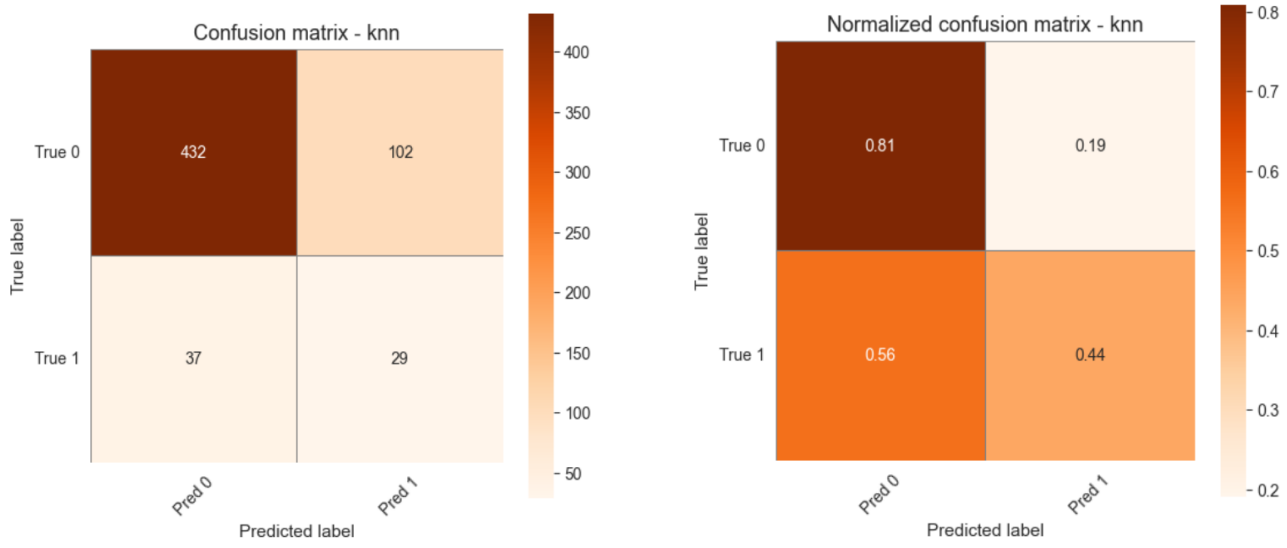
Configuration: {'n_neighbors': 15, 'weights': 'distance', 'algorithm': 'auto', 'leaf_size': 30, 'p': 1, 'metric': 'manhattan'}

Kết quả trên tập test:

- Accuracy : 0.7683
- Precision: 0.8441
- Recall : 0.7683
- F1-score : 0.7990

	Precision	Recall	F1-score	Support
0	0.92	0.81	0.86	534
1	0.22	0.44	0.29	66
Accuracy			0.77	600
Macro avg	0.57	0.62	0.58	600
Weighted avg	0.84	0.77	0.80	600

Bảng 1: Detailed metrics per class



Hình 11: Confusion matrix of KNN

4.2 Mô hình Logistic Regression

4.2.1 Giới thiệu

Logistic Regression là một mô hình tuyến tính dùng cho bài toán phân loại nhị phân. Mô hình giả định rằng logit (log odds) của xác suất thuộc lớp dương là một hàm tuyến tính của các biến đầu vào. Đầu ra của mô hình là xác suất thông qua hàm sigmoid, từ đó ta chọn ngưỡng (thường là 0.5) để phân loại. Ưu điểm của Logistic Regression là dễ huấn luyện, dễ diễn giải hệ số và thường cho kết quả ổn định trên nhiều bộ dữ liệu.

4.2.2 Kết quả huấn luyện mô hình

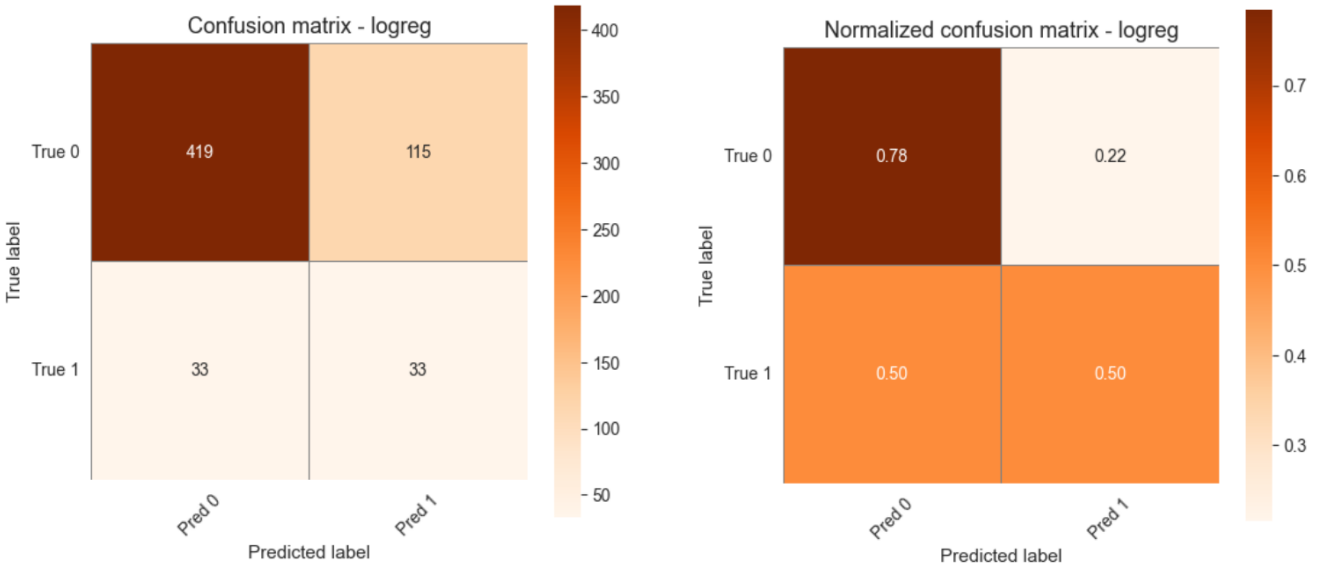
Configuration: {'penalty': 'l2', 'dual': False, 'tol': 0.0001, 'C': 15.0, 'solver': 'saga', 'max_iter': 5000, 'class_weight': 'balanced'}

Kết quả trên tập test:

- Accuracy : 0.7533
- Precision: 0.8495
- Recall : 0.7533
- F1-score : 0.7903

	Precision	Recall	F1-score	Support
0	0.93	0.78	0.85	534
1	0.22	0.50	0.31	66
Accuracy			0.75	600
Macro avg	0.57	0.64	0.58	600
Weighted avg	0.85	0.75	0.79	600

Bảng 2: Detailed metrics per class



Hình 12: Confusion matrix of Logistic Regression

4.3 Mô hình SVM

4.3.1 Giới thiệu

Support Vector Machine (SVM) là mô hình phân loại tìm một siêu phẳng (hyperplane) có khoảng cách biên (margin) lớn nhất để tách các lớp dữ liệu. Với hạt nhân (kernel), SVM có thể ánh xạ dữ liệu sang không gian đặc trưng chiều cao hơn để xử lý những bài toán không tuyến tính. SVM thường hoạt động tốt trên dữ liệu có số chiều vừa phải và phân tách tương đối rõ ràng giữa các lớp.

4.3.2 Kết quả huấn luyện mô hình

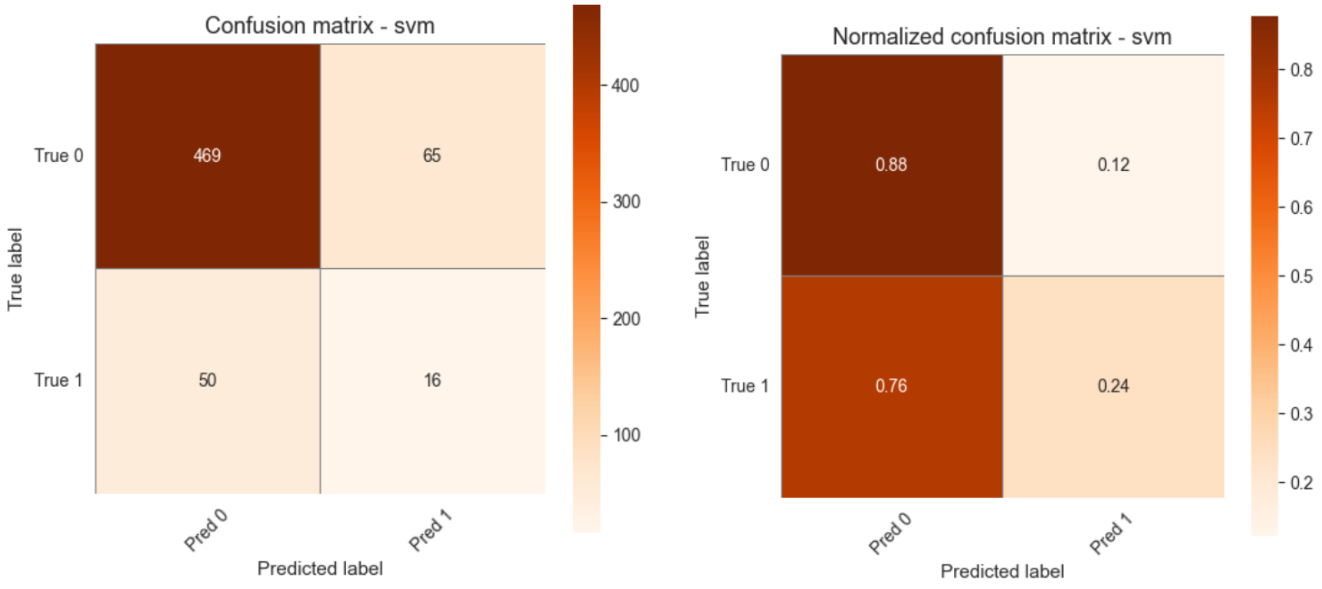
Configuration: {'C': 15.0, 'kernel': 'rbf', 'degree': 3, 'gamma': 'scale', 'coef0': 0.0, 'class_weight': 'balanced', 'cache_size': 1000}

Kết quả trên tập test:

- Accuracy : 0.8083
- Precision: 0.8260
- Recall : 0.8083
- F1-score : 0.8167

	Precision	Recall	F1-score	Support
0	0.90	0.88	0.89	534
1	0.20	0.24	0.22	66
Accuracy			0.81	600
Macro avg	0.55	0.56	0.55	600
Weighted avg	0.83	0.81	0.82	600

Bảng 3: Detailed metrics per class



Hình 13: Confusion matrix of SVM

4.4 Mô hình Decision Tree

4.4.1 Giới thiệu

Decision Tree là mô hình dự đoán dựa trên cấu trúc cây, trong đó mỗi nút trong cây là một điều kiện tách dữ liệu trên một thuộc tính, và mỗi lá cây tương ứng với một nhãn dự đoán. Mô hình được xây dựng bằng cách chọn những phép tách làm giảm độ hỗn loạn (entropy, Gini) nhiều nhất. Decision Tree dễ trực quan hoá và giải thích, nhưng dễ bị quá khớp (overfitting) nếu không được cắt tỉa (pruning) hoặc giới hạn độ sâu.

4.4.2 Kết quả huấn luyện mô hình

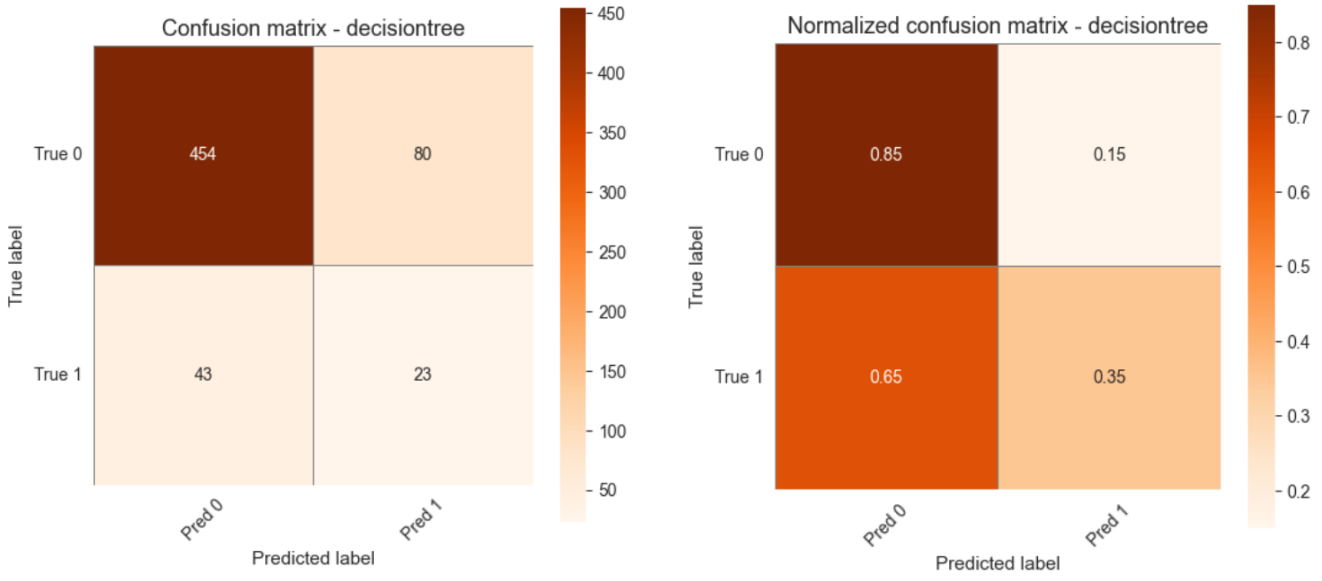
Configuration: {'criterion': 'gini', 'splitter': 'best', 'max_depth': 35, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_features': 'sqrt', 'class_weight': 'balanced'}

Kết quả trên tập test:

- Accuracy : 0.7950
- Precision: 0.8376
- Recall : 0.7950
- F1-score : 0.8138

	Precision	Recall	F1-score	Support
0	0.91	0.85	0.88	534
1	0.22	0.35	0.27	66
Accuracy			0.80	600
Macro avg	0.57	0.60	0.58	600
Weighted avg	0.84	0.80	0.81	600

Bảng 4: Detailed metrics per class



Hình 14: Confusion matrix of Decision Tree

4.5 Mô hình Random Forest

4.5.1 Giới thiệu

Random Forest là một mô hình tập hợp (ensemble) của nhiều cây quyết định. Mỗi cây được huấn luyện trên một mẫu bootstrap của dữ liệu và tại mỗi nút chỉ xem xét một tập con ngẫu nhiên các thuộc tính khi tách. Dự đoán cuối cùng được lấy bằng cách bỏ phiếu đa số từ các cây thành phần. Nhờ cơ chế lấy trung bình qua nhiều cây, Random Forest thường giảm được hiện tượng overfitting và cho hiệu suất tốt, ổn định.

4.5.2 Kết quả huấn luyện mô hình

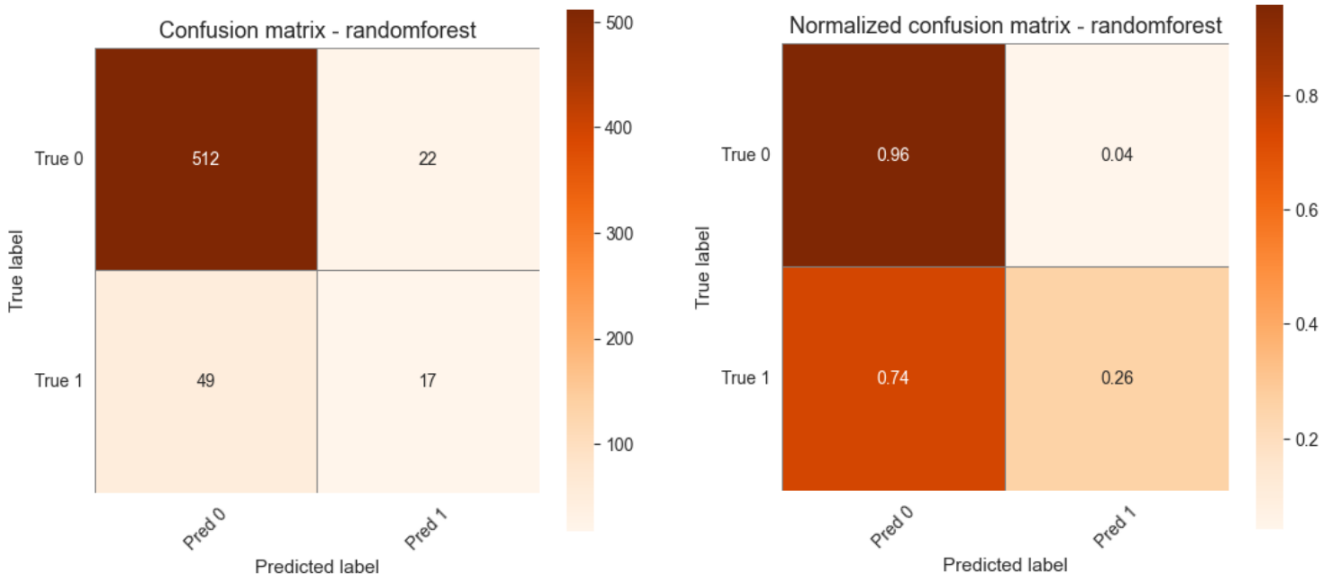
Configuration: {'n_estimators': 1500, 'criterion': 'gini', 'max_depth': 20, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_features': 'sqrt', 'bootstrap': True, 'max_samples': 0.9, 'class_weight': 'balanced_subsample'}

Kết quả trên tập test:

- Accuracy : 0.8817
- Precision: 0.8602
- Recall : 0.8817
- F1-score : 0.8679

	Precision	Recall	F1-score	Support
0	0.91	0.96	0.94	534
1	0.44	0.26	0.32	66
Accuracy			0.88	600
Macro avg	0.67	0.61	0.63	600
Weighted avg	0.86	0.88	0.87	600

Bảng 5: Detailed metrics per class



Hình 15: Confusion matrix of Random Forest

4.6 Mô hình XGBoost

4.6.1 Giới thiệu

XGBoost (Extreme Gradient Boosting) là một cài đặt tối ưu và mở rộng của thuật toán Gradient Boosting, tập trung vào hiệu năng tính toán và khả năng tổng quát hoá. Mô hình xây dựng dần dần một tập các cây quyết định nông, mỗi cây mới cố gắng sửa lỗi của mô hình hiện tại bằng cách tối thiểu hoá một hàm mất mát thông qua gradient. XGBoost hỗ trợ nhiều kỹ thuật regularization, xử lý giá trị thiếu và song song hoá, nên thường đạt kết quả rất tốt trong các bài toán thi trên Kaggle.

4.6.2 Kết quả huấn luyện mô hình

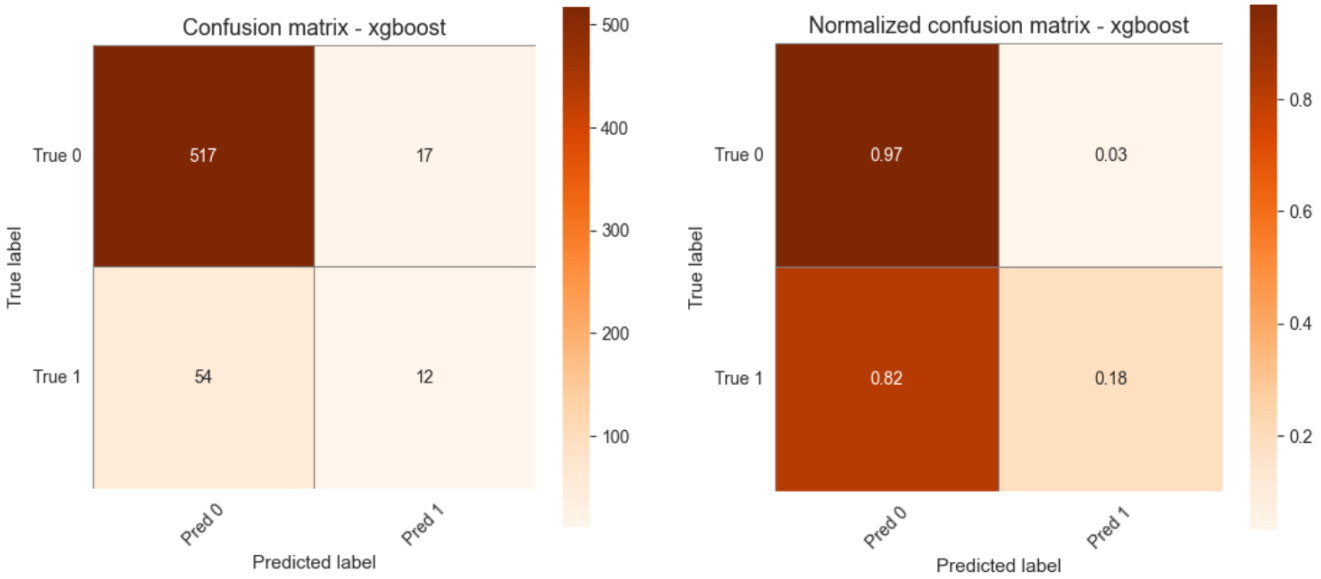
Configuration: {'n_estimators': 1500, 'learning_rate': 0.02, 'max_depth': 15, 'verbosity': 1, 'subsample': 0.85, 'colsample_bytree': 0.85, 'colsample_bylevel': 0.9, 'gamma': 0.1, 'reg_alpha': 0.05, 'reg_lambda': 1.5, 'min_child_weight': 2, 'scale_pos_weight': 1}

Kết quả trên tập test:

- Accuracy : 0.8817
- Precision: 0.8513
- Recall : 0.8817
- F1-score : 0.8606

	Precision	Recall	F1-score	Support
0	0.91	0.97	0.94	534
1	0.41	0.18	0.25	66
Accuracy			0.88	600
Macro avg	0.66	0.57	0.59	600
Weighted avg	0.85	0.88	0.86	600

Bảng 6: Detailed metrics per class



Hình 16: Confusion matrix of XGBoost

4.7 Mô hình Gradient Boosting

4.7.1 Giới thiệu

Gradient Boosting là một phương pháp ensemble xây dựng mô hình một cách tuần tự, trong đó mỗi mô hình con (thường là cây quyết định nông) được huấn luyện để giảm dần sai số còn lại (residual) của tổ hợp các mô hình trước đó. Mô hình cuối cùng là tổng có trọng số của các mô hình con. Gradient Boosting có khả năng mô tả tốt các quan hệ phi tuyến, nhưng cần điều chỉnh siêu tham số cẩn thận (learning rate, số lượng cây, độ sâu cây, ...) để tránh overfitting.

4.7.2 Kết quả huấn luyện mô hình

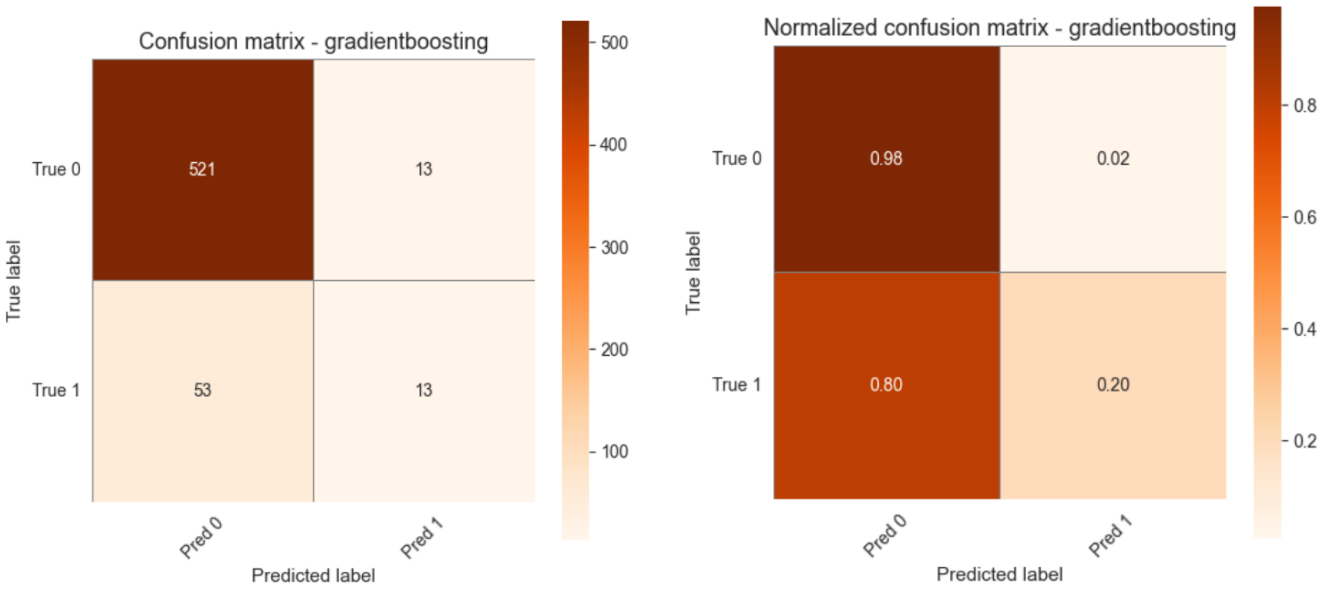
Configuration: {'loss': 'log_loss', 'learning_rate': 0.02, 'n_estimators': 1500, 'subsample': 0.85, 'max_depth': 15, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_features': 'sqrt', 'validation_fraction': 0.15, 'n_iter_no_change': 30, 'tol': 1e-06}

Kết quả trên tập test:

- Accuracy : 0.8900
- Precision: 0.8628
- Recall : 0.8900
- F1-score : 0.8681

	Precision	Recall	F1-score	Support
0	0.91	0.98	0.94	534
1	0.50	0.20	0.28	66
Accuracy			0.89	600
Macro avg	0.70	0.59	0.61	600
Weighted avg	0.86	0.89	0.87	600

Bảng 7: Detailed metrics per class



Hình 17: Confusion matrix of Gradient Boosting

4.8 Mô hình Naive Bayes

4.8.1 Giới thiệu

Naive Bayes là họ mô hình xác suất dựa trên định lý Bayes, với giả định mạnh mẽ rằng các thuộc tính là độc lập với nhau khi đã biết nhãn lớp (giả định "naive"). Mô hình ước lượng xác suất có điều kiện từ dữ liệu huấn luyện và suy ra xác suất hậu nghiệm cho từng lớp. Dù giả định đơn giản, Naive Bayes thường cho kết quả khá tốt, đặc biệt trên dữ liệu văn bản hoặc dữ liệu có số chiều cao, và có tốc độ huấn luyện, dự đoán rất nhanh.

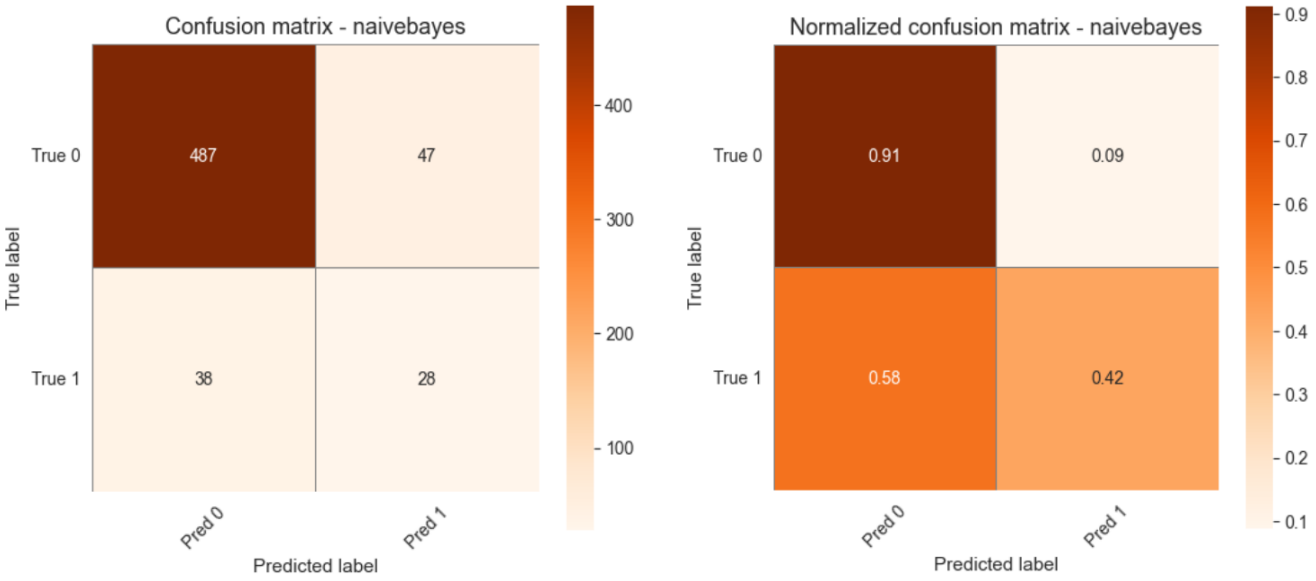
4.8.2 Kết quả huấn luyện mô hình

Kết quả trên tập test:

- Accuracy : 0.8583
- Precision: 0.8666
- Recall : 0.8583
- F1-score : 0.8623

	Precision	Recall	F1-score	Support
0	0.93	0.91	0.92	534
1	0.37	0.42	0.40	66
Accuracy			0.86	600
Macro avg	0.65	0.67	0.66	600
Weighted avg	0.87	0.86	0.86	600

Bảng 8: Detailed metrics per class



Hình 18: Confusion matrix of Naive Bayes

4.9 Mô hình MLP

4.9.1 Giới thiệu

Multilayer Perceptron (MLP) là một dạng mạng nơ-ron truyền thẳng (feedforward neural network) gồm một lớp vào, một hoặc nhiều lớp ẩn và một lớp đầu ra. Mỗi lớp gồm nhiều nút (neuron) kết nối đầy đủ với lớp tiếp theo, dùng các hàm kích hoạt phi tuyến (ReLU, sigmoid, ...) để học các quan hệ phức tạp giữa đầu vào và đầu ra. MLP có khả năng biểu diễn mạnh, nhưng cần số lượng dữ liệu đủ lớn, kỹ thuật regularization và tối ưu siêu tham số để đạt hiệu quả cao và tránh overfitting.

4.9.2 Kết quả huấn luyện mô hình

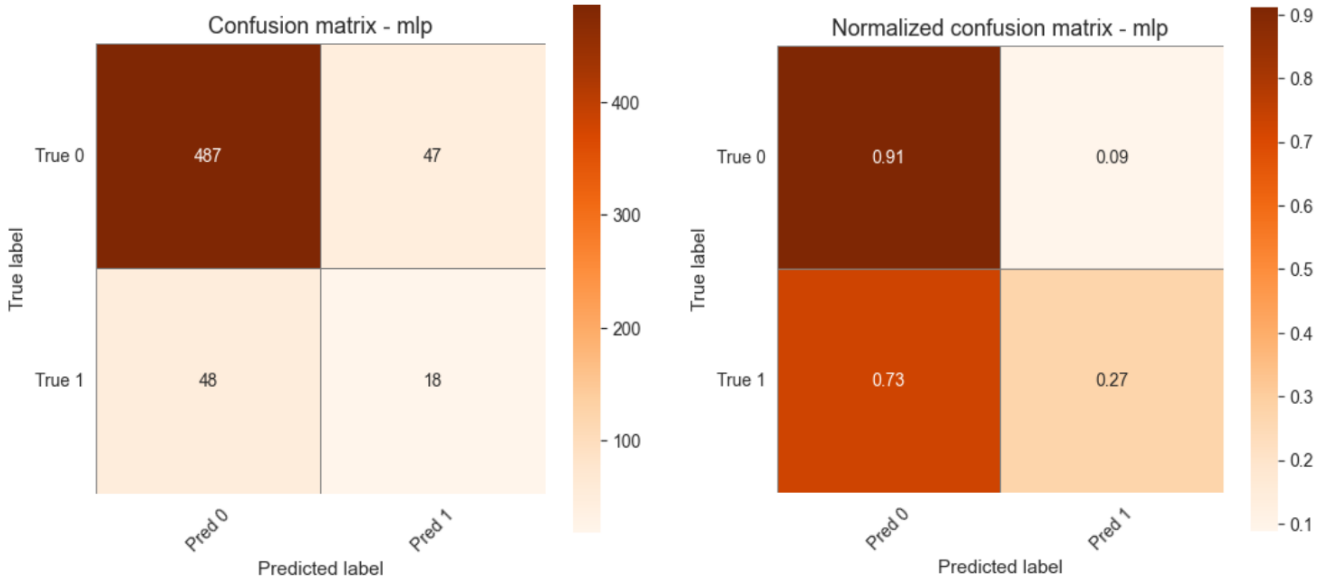
Configuration: {'hidden_layer_sizes': (300, 200, 100), 'activation': 'relu', 'solver': 'adam', 'max_iter': 5000, 'alpha': 0.0001, 'learning_rate': 'adaptive', 'early_stopping': True, 'validation_fraction': 0.2, 'n_iter_no_change': 30, 'batch_size': 64}

Kết quả trên tập test:

- Accuracy : 0.8417
- Precision: 0.8406
- Recall : 0.8417
- F1-score : 0.8411

	Precision	Recall	F1-score	Support
0	0.91	0.91	0.91	534
1	0.28	0.27	0.27	66
Accuracy			0.84	600
Macro avg	0.59	0.59	0.59	600
Weighted avg	0.84	0.84	0.84	600

Bảng 9: Detailed metrics per class



Hình 19: Confusion matrix of Multilayer Perceptron

5 Đánh giá và so sánh mô hình

Model	Accuracy	Precision	Recall	F1-Score	N_Features
GRADIENTBOOSTING	0.890000	0.862822	0.890000	0.868073	36
RANDOMFOREST	0.881667	0.860213	0.881667	0.867911	36
XGBOOST	0.881667	0.851349	0.881667	0.860604	36
NAIVEBAYES	0.858333	0.866648	0.858333	0.862253	36
MLP	0.841667	0.840611	0.841667	0.841136	36
SVM	0.808333	0.825987	0.808333	0.816747	36
DECISIONTREE	0.795000	0.837561	0.795000	0.813762	36
KNN	0.768333	0.844138	0.768333	0.799046	36
LOGREG	0.753333	0.849549	0.753333	0.790335	36

Bảng 10: So sánh hiệu quả các mô hình học máy

Dựa trên Bảng trên, có thể thấy nhóm mô hình **ensemble dựa trên cây** cho hiệu quả tốt nhất. Mô hình **Gradient Boosting** đạt Accuracy cao nhất (xấp xỉ 0.89), đồng thời có Precision, Recall và F1-score đều ở mức cao, nên được xem là mô hình toàn diện nhất. Tiếp theo là **Random Forest** và **XGBoost** với Accuracy khoảng 0.88 và F1-score đều trên 0.86, cho thấy ba mô hình ensemble này đều học tốt các quan hệ phi tuyến và tương tác phức tạp giữa các đặc trưng.

Nhóm mô hình **Naive Bayes** và **MLP** cho kết quả khá, với Accuracy trong khoảng 0.84 đến 0.86 và F1-score ở mức ổn định. Các mô hình còn lại như **SVM**, **Decision Tree**, **KNN** và **Logistic Regression** có Accuracy và F1-score thấp hơn, trong đó Logistic Regression đạt Accuracy thấp nhất. Tuy nhiên, một số mô hình đơn giản như Logistic Regression hay KNN vẫn có Precision tương đối cao, nghĩa là khi đã dự đoán dương tính thì thường đúng, nhưng Recall thấp cho thấy bỏ sót khá nhiều trường hợp khách hàng thực sự đăng ký.

Vì tất cả mô hình đều sử dụng cùng một tập 36 đặc trưng, sự khác biệt chủ yếu đến từ khả năng mô hình hóa của từng thuật toán, và kết quả cho thấy các mô hình ensemble là lựa chọn phù hợp hơn cho bài toán này.

6 Kết luận và hướng phát triển

6.1 Kết luận

Trong bài tập lớn này, nhóm đã xây dựng một quy trình khai phá dữ liệu hoàn chỉnh cho bài toán dự đoán khả năng khách hàng đăng ký tiền gửi có kỳ hạn dựa trên bộ dữ liệu marketing của ngân hàng. Quy trình bao gồm các bước: khảo sát và tiền xử lý dữ liệu, xây dựng nhiều mô hình học máy khác nhau và cuối cùng là đánh giá, so sánh hiệu quả của các mô hình trên cùng một tập dữ liệu.

Từ các kết quả thực nghiệm trình bày ở Phần 3, 4 và 5 có thể rút ra một số nhận xét chính như sau:

- Các bước tiền xử lý (làm sạch dữ liệu, mã hoá biến phân loại, chuẩn hoá/chuẩn chỉnh dữ liệu, xử lý dữ liệu mất cân bằng, ...) có ảnh hưởng rất lớn đến chất lượng dự đoán của mô hình. Việc xử lý đúng giúp mô hình ổn định hơn và cải thiện các chỉ số đánh giá.
- Những mô hình tuyến tính đơn giản (ví dụ như Logistic Regression) cho kết quả ổn định, dễ huấn luyện và dễ diễn giải, trong khi các mô hình phi tuyến phức tạp hơn (như cây quyết định, rừng ngẫu nhiên, SVM, ...) có khả năng khai thác mối quan hệ phi tuyến giữa các biến, nhờ đó cải thiện hiệu suất dự đoán trong nhiều trường hợp.
- Các đặc trưng liên quan đến lịch sử liên hệ và cường độ chiến dịch marketing (thời lượng cuộc gọi, số lần liên hệ, kết quả các chiến dịch trước, ...) thường mang nhiều thông tin hơn so với một số đặc trưng thông tin cá nhân, qua đó cho thấy tầm quan trọng của việc theo dõi hành vi tương tác của khách hàng.

6.2 Hướng phát triển

Trong phạm vi bài tập lớn, nhóm chỉ mới dừng lại ở một số mô hình cơ bản và cách tiền xử lý dữ liệu tương đối đơn giản. Dựa trên kết quả đạt được, nhóm đề xuất một số hướng phát triển trong tương lai:

- Áp dụng các thuật toán mạnh hơn cho bài toán phân loại như LightGBM, CatBoost hoặc các mô hình mạng nơ-ron đơn giản để xem xét khả năng cải thiện thêm hiệu suất dự đoán.
- Xây dựng mô hình dựa trên dữ liệu theo thời gian thực.
- Tích hợp với các công nghệ mới như Apache Spark để xử lý dữ liệu lớn, Google BigQuery làm Data Warehouse...
- Triển khai mô hình: Xây dựng một API hoặc một giao diện web đơn giản cho phép nhập thông tin khách hàng và trả về xác suất đăng ký tiền gửi, qua đó mô phỏng bước đầu việc đưa mô hình vào sử dụng trong môi trường thực tế.

Thông qua dự án này, nhóm đã tích lũy được nhiều kinh nghiệm thực tiễn trong việc khảo sát, tiền xử lý dữ liệu, xây dựng và đánh giá mô hình học máy. Những kỹ năng này sẽ là nền tảng vững chắc cho các nghiên cứu và dự án sâu hơn trong lĩnh vực trí tuệ nhân tạo và khai thác dữ liệu.

7 Tài liệu tham khảo

References

[1] Data Mining. Concepts and Techniques, 3rd Edition. https://ia800603.us.archive.org/2/items/datamining_201811/DS-book%20u5.pdf

[2] Marketing Dataset https://www.kaggle.com/competitions/marketing-data/overview

[3] Introduction to Machine Learning with Python https://www.nrigroupindia.com/e-book/Introduction%20to%20Machine%20Learning%20with%20Python%20(%20PDFDrive.com%20)-min.pdf

[4] Phạm Đình Khánh. *Deep AI Book - Random Forest.* https://phamdinhkhanh.github.io/deepai-book/ch_ml/RandomForest.html

[5] Machine Learning Cơ Bản. *Random Forest.* https://machinelearningcoban.com/tabml_book/ch_model/random_forest.html

[6] GeeksforGeeks. *XGBoost.* https://www.geeksforgeeks.org/xgboost/

[7] Viblo. *Tổng quan về Artificial Neural Network.* https://viblo.asia/p/tong-quan-ve-artificial-neural-network-1VgZvwYrlAw

[8] Viblo. *Linear Regression - Hồi quy tuyến tính trong Machine Learning.* https://viblo.asia/p/linear-regression-hoi-quy-tuyen-tinh-trong-machine-learning-4P856akRlY3

[9] GeeksforGeeks. *ML Linear Regression.* https://www.geeksforgeeks.org/ml-linear-regression/