# HỌC VIỆN NGÂN HÀNG

# KHOA HỆ THỐNG THÔNG TIN QUẢN LÝ

# ĐỀ TÀI: ỨNG DỤNG MỘT SỐ THUẬT TOÁN HỌC MÁY HỖ TRỢ QUÁ TRÌNH RA QUYẾT ĐỊNH PHÁT HÀNH THẺ TIN DỤNG

**Giảng viên hướng dẫn: TS. Bùi Thị Hồng Nhung**

**NHÓM 9**

**MÃ HỌC PHẦN: 231IS30A02**


**Nhóm sinh viên thực hiện:**

| | |
|---|---|
| Nguyễn Thị Mai Anh | 23A4040005 – K23HTTTA |
| Lê Hà Chi | 23A4040018 – K23HTTTA |
| Nguyễn Thị Duyên | 23A4040025 – K23HTTTA |
| Đặng Thị Hạnh | 23A4040034 – K23HTTTA |
| Phạm Việt Phương | 23A4040118 – K23HTTTA |


**Hà Nội, 12/2023**

## DANH SÁCH SINH VIÊN THỰC HIỆN

| Họ và tên | Mã sinh viên | Đóng góp |
|---|---|---|
| Nguyễn Thị Mai Anh | 23A4040005 | 20% |
| Lê Hà Chi | 23A4040018 | 20% |
| Nguyễn Thị Duyên | 23A4040025 | 20% |
| Đặng Thị Hạnh | 23A4040034 | 20% |
| Phạm Việt Phương | 23A4040118 | 20% |

# LỜI CẢM ƠN

Kính gửi TS. Bùi Thị Hồng Nhung,

Chúng em, nhóm sinh viên lớp Hệ thống thông tin quản lý xin gửi lời cảm ơn chân thành nhất đến cô trong học phần Khai phá dữ liệu.

Trong quá trình thực hiện bài làm, với sự hướng dẫn của cô, chúng em đã cố gắng vận dụng những kiến thức đã học kết hợp những sự tìm hiểu của cá nhân các thành viên với mong muốn hoàn thành bài làm một cách tốt nhất. Song, do lượng kiến thức là rất lớn, khả năng tìm hiểu còn hạn chế và còn chưa có nhiều kinh nghiệm thực tế nên bài làm của nhóm có thể còn tồn tại những thiếu sót. Chúng em rất mong sẽ nhận được ý kiến đóng góp của cô để bài làm được hoàn thiện hơn!

Một lần nữa, chúng em xin chân thành cảm ơn cô đã luôn quan tâm và giúp đỡ chúng em trong suốt thời gian qua. Chúng em xin chúc cô luôn mạnh khỏe, hạnh phúc và thành công trong công việc.

Trân trọng,
Nhóm 9

# MỤC LỤC


CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 1
1.1. Đặt vấn đề 1
1.2. Mục tiêu 1
1.3. Một số kết quả nghiên cứu tại Việt Nam [1] 2
1.4. Đối tượng và phương pháp nghiên cứu 2
1.5. Nội dung nghiên cứu 3
1.6. Ý nghĩa đề tài 3
1.7. Chuẩn bị dữ liệu [11] 3
1.8. Bố cục 7
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 8
2.1.Tổng quan quá trình khai phá dữ liệu [12] 8
2.2. Bài toán phân lớp trong quá trình khai phá dữ liệu 9
2.2.1. Khái niệm phân lớp 9
2.2.2. Quá trình phân lớp dữ liệu 9
2.2.3. Các loại phân lớp 9
2.2.4. Một số thuật toán phân lớp dữ liệu 9
2.3. Bài toán luật kết hợp trong quá trình khai phá dữ liệu 13
2.3.1. Khái niệm luật kết hợp [13] 13
2.3.2. Thuật toán Apriori 14
2.4. Cơ sở lý thuyết về tín dụng trong ngân hàng [4] 14
2.4.1. Tín dụng là gì 14
2.4.2. Những điều cần biết về thẻ tín dụng 14
2.4.3. Quy trình phát hành thẻ tín dụng của ngân hàng [10] 15
CHƯƠNG 3: XÂY DỰNG VÀ THỰC NGHIỆM MÔ HÌNH DỰ BÁO VỚI DỮ LIỆU ÍT THUỘC TÍNH 16
3.1. Mục đích thực nghiệm 16
3.2. Cách thức thực hiện thuật toán CART 16
3.3. Cách thức thực hiện thuật toán ID3 19
3.4. Thuật toán *phân lớp Naive Bayes* 23
CHƯƠNG 4: XÂY DỰNG VÀ THỰC NGHIỆM MÔ HÌNH DỰ BÁO VỚI DỮ LIỆU ĐẦY ĐỦ THUỘC TÍNH 25
4.1. Phân tích dữ liệu 25

4.2. Tiền xử lý và biến đổi dữ liệu 34
4.3. Xây dựng mô hình thuật toán phân lớp 38
4.3.1. Phân chia tập dữ liệu 38
4.3.2. Huấn luyện dữ liệu với thuật toán Decision Tree Classifier – CART 38
4.3.3. Huấn luyện dữ liệu với thuật toán Decision Tree Classifier - ID3 39
4.3.4. Huấn luyện dữ liệu với thuật toán KNN Classifier 41
4.3.5. Huấn luyện dữ liệu với thuật toán Gaussian Naive Bayes 43
4.4. Xây dựng mô hình luật kết hợp với thuật toán Apriori 44
4.5. Đánh giá chung về các mô hình 46
4.6. Phương pháp cải tiến mô hình 46
4.6.1. Phương pháp cải tiến thuật toán phân lớp 46
4.6.2. Phương pháp cải tiến thuật toán luật kết hợp 48
CHƯƠNG 5: TỔNG KẾT 51
5.1. Kết quả đạt được 51
5.2 Điểm hạn chế 51
5.3 Hướng phát triển 52
Tài liệu tham khảo 53

# DANH MỤC HÌNH ẢNH, BẢNG BIỂU

**Hình 2.1:** Quá trình khai phá tri thức

**Hình 3.1:** Cây quyết định sau khi thực hiện thuật toán CART

**Hình 3.2:** Cây quyết định sau khi thực hiện thuật toán ID3

**Hình 4.1:** Biểu đồ đánh giá thuộc tính theo giới tính

**Hình 4.2:** Biểu đồ xét theo thuộc tính ô tô xét tỷ lệ nam, nữ

**Hình 4.3:** Biểu đồ xét theo thuộc tính bất động sản

**Hình 4.4:** Biểu đồ xét theo thuộc tính thu nhập có việc làm

**Hình 4.5:** Biểu đồ xét theo thuộc tính trình độ học vấn là THPT

**Hình 4.6:** Biểu đồ xét theo thuộc tính tình trạng hôn nhân đã kết hôn

**Hình 4.7:** Biểu đồ xét theo tỷ lệ người ở nhà riêng và có căn hộ

**Hình 4.8:** Biểu đồ tương quan giữa trình độ học vấn, người có việc làm, nghề nghiệp

**Hình 4.9:** Biểu đồ thể hiện mối tương quan giữa nghề nghiệp và trình độ học vấn

**Bảng 3.1:** Tập dữ liệu mẫu xây dựng mô hình bằng thuật toán CART

**Bảng 3.2:** Bản tính Gini D lần 1

**Bảng 3.3:** Bản tính Gini D lần 3

**Bảng 3.4:** Tập dữ liệu mẫu xây dựng mô hình bằng thuật toán ID3

**Bảng 3.5:** 14 bản ghi ngẫu nhiên lấy từ bộ dữ liệu

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

## 1.1. Đặt vấn đề

Trong nền kinh tế thị trường, hệ thống ngân hàng đóng vai trò quan trọng như hệ thần kinh của cơ thể. Hoạt động của hệ thống ngân hàng quốc gia ổn định, lành mạnh và hiệu quả sẽ giúp các nguồn lực tài chính được luân chuyển, phân bổ và sử dụng hiệu quả, kích thích tăng trưởng kinh tế bền vững. Tuy nhiên, trong kinh tế thị trường, rủi ro trong kinh doanh là không thể tránh khỏi, đặc biệt là rủi ro trong hoạt động ngân hàng. Rủi ro ngân hàng có thể gây ra phản ứng dây chuyền, lây lan và ngày càng phức tạp. Khi ngân hàng rơi vào trạng thái tài chính khó khăn nghiêm trọng thì nguyên nhân thường phát sinh từ hoạt động tín dụng của ngân hàng. Trước áp lực tăng nợ xấu, nhiều ngân hàng đã tích cực trích lập nâng bộ đệm dự phòng rủi ro, nhưng vẫn chưa theo kịp tốc độ tăng của nợ xấu. Theo số liệu của Ngân hàng Nhà nước Việt Nam, tính đến tháng 12 năm 2023, tổng nợ xấu nội bảng của các ngân hàng thương mại (NHTM) là 350.862 tỷ đồng, chiếm tỷ lệ 3,3% tổng dư nợ tín dụng. Đây là mức nợ xấu cao nhất kể từ năm 2017. Tính riêng nợ xấu nhóm 3 (nợ có khả năng mất vốn) là 240.090 tỷ đồng, chiếm tỷ lệ 2,1% tổng dư nợ tín dụng. Nợ xấu nhóm 4 (nợ có khả năng mất vốn cao) là 96.433 tỷ đồng, chiếm tỷ lệ 0,8% tổng dư nợ tín dụng. Nợ xấu nhóm 5 (nợ có khả năng mất vốn rất cao) là 14.339 tỷ đồng, chiếm tỷ lệ 0,1% tổng dư nợ tín dụng.

Tình trạng này kéo dài và trở nên nghiêm trọng do một số nguyên nhân chủ yếu là do sự suy giảm của nền kinh tế: Nền kinh tế Việt Nam đang trong giai đoạn phục hồi sau đại dịch COVID-19, nhưng vẫn còn nhiều khó khăn, thách thức. Điều này đã ảnh hưởng đến hoạt động sản xuất kinh doanh của doanh nghiệp và khả năng trả nợ của người dân. Thứ hai là do sự tăng trưởng tín dụng nóng: Tăng trưởng tín dụng của các NHTM trong thời gian qua khá cao, nhưng chất lượng tín dụng chưa được cải thiện. Điều này đã làm gia tăng rủi ro nợ xấu.

Vì vậy để khắc phục tình trạng nợ xấu, cần có các giải pháp đồng bộ từ phía Chính phủ, Ngân hàng Nhà nước và các NHTM. Nhóm đã lựa chọn đề tài dự đoán phát hành thẻ tín dụng với mong muốn tìm ra các phương pháp giúp kiểm soát rủi ro, cải thiện tình trạng trên.

## 1.2. Mục tiêu

Với đề tài "Dự báo phát hành thẻ tín dụng", nhóm muốn hướng đến giúp các ngân hàng thương mai phê duyệt thẻ tín dụng, đưa ra quyết định một cách chuẩn xác nhất, tránh sai số và tổn thất nhất có thể. Để đạt được mục tiêu này, nhóm sẽ tập trung vào hai hướng chính:

- Giúp nhân viên tín dụng cải thiện hiệu suất làm việc: Nhóm sẽ phát triển các mô hình học máy để giúp nhân viên tín dụng tự động hóa các công đoạn xử lý dữ liệu, từ đó tiết kiệm thời gian và công sức.
- Nâng cao chất lượng của hệ thống xếp hạng tín dụng: Nhóm sẽ sử dụng các thuật toán học máy để phân tích dữ liệu khách hàng một cách toàn diện, từ đó đưa ra các đánh giá chính xác hơn về khả năng trả nợ của khách hàng.

Họ là những người thường xuyên phải tiếp xúc với nhiều loại dữ liệu khổng lồ và đa dạng. Ngoài ra, họ còn là những người có kiến thức nghiệp vụ tốt và có kinh nghiệm trong việc xử lý dữ liệu. Tuy nhiện, việc xử lý luồng dữ liệu đó sẽ tốn nhiều thời gian và công sức với họ. Vì vây việc cải thiện hiệu suất làm việc của nhân viên tín dụng sẽ giúp họ dành nhiều thời gian hơn cho việc phân tích các trường hợp phức tạp, từ đó đưa ra các quyết định phê duyệt thẻ tín dụng chính xác hơn. Việc nâng cao chất lượng của hệ thống xếp hạng tín dụng sẽ giúp ngân hàng phân loại khách hàng tốt hơn, phản ánh thực chất hơn tình trạng tín dụng của khách hàng. Từ đó quyết định phát hành thẻ tín dụng cho khách hàng.

### 1.3. Một số kết quả nghiên cứu tại Việt Nam [1]

Chấm điểm tín dụng và phân loại các khoản vay là một vấn đề quan trọng đối với các ngân hàng và tổ chức tài chính. Để giải quyết vấn đề này, nhiều phương pháp và thuật toán đã được nghiên cứu và phát triển.

Một số thuật toán phổ biến được Ngân hàng Trung ương (NHTW) nghiên cứu bao gồm hồi quy logistic (Logistic Regression), rừng ngẫu nhiên (Random Forest) và cây quyết định. Các thuật toán này sử dụng các biến dữ liệu khách hàng, chẳng hạn như lịch sử tín dụng, thu nhập, tài sản, v.v. để dự đoán khả năng trả nợ của khách hàng.

Ngoài ra, các mô hình Machine Learning chấm điểm tín dụng dựa trên trí tuệ nhân tạo (AI) kết hợp giữa lịch sử tín dụng của khách hàng và tiềm năng của Big Data cũng đã được phát triển. Sử dụng một nguồn lớn các thông tin được khai thác từ AI để cải thiện các quyết định tín dụng và mang lại sự thẩm định tốt hơn so với một chuyên gia phân tích là con người.

Các Ngân hàng Thương mại cũng nỗ lực tìm ra phương án kiểm soát rủi ro, phát triển hệ thống chấm điểm tín dụng tối ưu. VietinBank là một trong những NHTM tiên phong trong việc ứng dụng AI vào quản lý rủi ro tín dụng. Ngân hàng đã nghiên cứu và triển khai thành công giải pháp DB Lytix in Sybase IQ, sử dụng thuật toán phân lớp cây quyết định và thuật toán Naive Bayes để phân tích dữ liệu khách hàng.[2] Ngoài ra Ngân hàng Vietcombank cũng là một trong những NHTM tích cực ứng dụng nhiều công cụ, ứng dụ ng điển hình như Học máy, Big Data, AI, Blockchain, Công nghệ IoT (Internet of Thing), Điện toán đám mây, Data Warehouse, Data Lake,.. vào để phục vụ Khai phá nguồn dữ liệu khổng lồ từ khách hàng. [3]

### 1.4. Đối tượng và phương pháp nghiên cứu

Đề tài tập trung vào nghiên cứu các đối tượng là các hoạt động tín dụng tại ngân hàng, nhất là hoạt động phê duyệt thẻ tín dụng. Tiếp đến là các thuật toán phân lớp phổ biến có thể áp dụng cho việc phê duyệt thẻ tín dụng như: cây quyết định, KNN, Navie Bayes.

Để đạt được mục tiêu nghiên cứu, đề tài sẽ sử dụng các phương pháp nghiên cứu định tính, định lượng, phương pháp so sánh, đối chiếu, suy luận logic, phân tích hệ thống,... Các phương pháp này sẽ giúp làm sáng tỏ và cụ thể hóa nội dung nghiên cứu, đồng thời phê phán và tiếp thu có chọn lọc những kết quả nghiên cứu có liên quan đến đề tài.

### 1.5. Nội dung nghiên cứu

Nghiên cứu cơ sở lý thuyết về các lĩnh vực liên quan như khai phá dữ liệu, hoạt động phê duyệt thẻ tín dụng trong ngân hàng. Các thuật toán phân lớp phổ biến: Decision Tree, KNN, Naive Bayes, Bổ sung thêm các thuật toán phân cụm và luật kết hợp.

Xây dựng dựa trên Python 3.9.5, một số công cụ hỗ trợ: openpyxl 3.0.10, pandas 1.5.0, numpy 1.23.3, matplotlib 3.6.0, seaborn 0.12.0, scikit-learn 1.1.2

### 1.6. Ý nghĩa đề tài

Giải pháp trên sử dụng sức mạnh của tổ chức quản lý, vận hành và chuyển đổi dữ liệu để phân tích, đánh giá khả năng thanh toán của khách hàng, từ đó xem xét phát hành thẻ tín dụng. Đây là bước khởi đầu cho chiến lược thu hút và chăm sóc khách hàng của các doanh nghiệp nói chung, đặc biệt là các ngân hàng. Cụ thể, đề tài này có thể mang lại những lợi ích sau cho các ngân hàng thương mại:

Tăng cường hiệu quả hoạt động kinh doanh: Dự báo phát hành thẻ tín dụng giúp ngân hàng thương mại xác định được lượng khách hàng tiềm năng có khả năng sử dụng thẻ tín dụng của ngân hàng. Từ đó, ngân hàng có thể xây dựng các chiến lược kinh doanh phù hợp để thu hút và phát triển khách hàng, tăng cường hiệu quả hoạt động kinh doanh.

Giảm thiểu rủi ro: Dự báo phát hành thẻ tín dụng giúp ngân hàng thương mại sàng lọc được những khách hàng có khả năng trả nợ cao, từ đó hạn chế rủi ro nợ xấu.

Nâng cao chất lượng dịch vụ: Dự báo phát hành thẻ tín dụng giúp ngân hàng thương mại hiểu rõ hơn về nhu cầu và mong muốn của khách hàng. Từ đó, ngân hàng có thể phát triển các sản phẩm và dịch vụ thẻ tín dụng phù hợp với từng nhóm khách hàng, nâng cao chất lượng dịch vụ, đáp ứng nhu cầu của khách hàng.

Tăng cường khả năng cạnh tranh: Dự báo phát hành thẻ tín dụng giúp ngân hàng hiểu rõ hơn về thị trường thẻ tín dụng, từ đó có thể đưa ra các chiến lược kinh doanh phù hợp để cạnh tranh với các ngân hàng khác.

### 1.7. Chuẩn bị dữ liệu [11]

| Thuộc tính | Mô tả | Ghi chú |
|---|---|---|
| ID | Mã khách hàng | |
| CODE_GENDER | Giới tính | |
| FLAG_OWN_CAR | Có ô tô không | |
| FLAG_OWN_REALTY | Có sở hữu bất động sản không | |

| | | |
|---|---|---|
| CNT_CHILDREN | Số lượng con cái | |
| AMT_INCOME_TOTAL | Tổng thu nhập | |
| NAME_INCOME_TYPE | Loại thu nhập | |
| NAME_EDUCATION_TYPE | Trình độ học vấn | |
| NAME_FAMILY_STATUS | Tình trạng hôn nhân | |
| NAME_HOUSING_TYPE | Loại chỗ ở | |
| DAYS_BIRTH | Ngày sinh nhật | Đếm ngược so với ngày hiện tại (0), -1 có nghĩa là ngày hôm qua |
| DAYS_EMPLOYED | Ngày bắt đầu làm việc | Đếm ngược so với ngày hiện tại (0). Nếu dương, có nghĩa là người đó hiện đang thất nghiệp. |
| FLAG_MOBIL | Có điện thoại di động không | |
| FLAG_WORK_PHONE | Có điện thoại cơ quan không | |
| FLAG_PHONE | Có điện thoại không | |
| FLAG_EMAIL | Có email không | |
| OCCUPATION_TYPE | Nghề nghiệp | |

| | | |
|---|---|---|
| CNT_FAM_MEMBERS | Quy mô gia đình | |
| STATUS | Được phê duyệt cấp thẻ tín dụng hay không | Với giá trị 1 là được phê duyệt và giá trị 0 là không được phê duyệt |

Việc chọn các trường dữ liệu để dự đoán phát hành thẻ tín dụng thường dựa trên sự quan trọng và mức độ ảnh hưởng của từng trường đối với quyết định cuối cùng. Dưới đây là giải thích tại sao mỗi trường được chọn:

1. CODE_GENDER (Giới tính):

Giới tính có thể ảnh hưởng đến quyết định về tín dụng do các đặc điểm kinh tế và chi tiêu có thể khác nhau giữa nam và nữ.

2. FLAG_OWN_CAR (Sở hữu ô tô):

Người có ô tô có thể được xem xét khả năng thanh toán nợ tốt hơn do có thể sở hữu tài sản có giá trị.

3. FLAG_OWN_REALTY (Sở hữu bất động sản):

Việc sở hữu bất động sản cũng có thể là một chỉ số tích cực về khả năng thanh toán nợ.

4. CNT_CHILDREN (Số lượng con cái):

Số lượng con có thể ảnh hưởng đến khả năng chi tiêu và trách nhiệm tài chính của một người.

5. AMT_INCOME_TOTAL (Tổng thu nhập):

Tổng thu nhập là một yếu tố quan trọng để đánh giá khả năng thanh toán nợ và chi tiêu hàng tháng.

6. NAME_INCOME_TYPE (Loại thu nhập):

Loại thu nhập có thể phản ánh ổn định và tính đều đặn của nguồn thu nhập, ảnh hưởng đến khả năng thanh toán.

7. NAME_EDUCATION_TYPE (Trình độ học vấn):

Trình độ học vấn có thể liên quan đến thu nhập và khả năng quản lý tài chính.

8. NAME_FAMILY_STATUS (Tình trạng hôn nhân):

Tình trạng hôn nhân có thể ảnh hưởng đến trách nhiệm tài chính và ổn định gia đình.

9. NAME_HOUSING_TYPE (Loại chỗ ở):

Loại chỗ ở có thể phản ánh mức độ ổn định và chi phí sống, ảnh hưởng đến khả năng thanh toán.

10. DAYS_BIRTH (Ngày sinh):

Tuổi có thể liên quan đến kinh nghiệm làm việc và ảnh hưởng đến khả năng thanh toán nợ.

11. DAYS_EMPLOYED (Số ngày làm việc):

Thời gian làm việc có thể liên quan đến ổn định nghề nghiệp và thu nhập.

12. FLAG_MOBIL (Sở hữu điện thoại di động):

Việc sở hữu điện thoại di động có thể liên quan đến tích hợp vào xã hội và mạng lưới liên lạc, ảnh hưởng đến đánh giá rủi ro tín dụng.

13. FLAG_WORK_PHONE (Sở hữu điện thoại di động cơ quan):

Việc sở hữu điện thoại di động công việc có thể là một chỉ số của mức độ liên kết với thị trường lao động và có thể ảnh hưởng đến ổn định công việc và thu nhập.

14. FLAG_PHONE (Sở hữu điện thoại):

Việc sở hữu điện thoại có thể phản ánh sự tiện ích và kết nối xã hội, ảnh hưởng đến khả năng liên lạc và đánh giá rủi ro tín dụng.

15. FLAG_EMAIL (Sở hữu email):

Sở hữu email có thể liên quan đến mức độ kết nối và sự hiện đại, có thể ảnh hưởng đến quyết định tín dụng.

16.OCCUPATION_TYPE (Ngành nghề): Ngành nghề có thể ảnh hưởng đến thu nhập và ổn định nghề nghiệp, ảnh hưởng đến khả năng thanh toán nợ.

17. CNT_FAM_MEMBERS (Số thành viên trong gia đình):

Số lượng thành viên trong gia đình có thể ảnh hưởng đến chi tiêu và khả năng thanh toán.

18. STATUS (Trạng thái): Trường này quan trọng vì nó chứa thông tin về quá trình phê duyệt cấp thẻ tín dụng. Giá trị 1 thường đại diện cho trường hợp được phê duyệt, trong khi giá trị 0 biểu thị cho trường hợp không được phê duyệt. Thông tin này là quan trọng để dự đoán và đánh giá hiệu suất của mô hình, giúp ta hiểu rõ hơn về yếu tố quyết định và tìm ra các mẫu quan trọng trong quá trình xét duyệt hồ sơ cấp thẻ tín dụng.

### 1.8. Bố cục

Đề tài có bố cục bao gồm 4 chương như sau:

Chương 1: Tổng quan về đề tài

Chương 2: Cơ sở lý thuyết

Chương 3: Xây dựng thực nghiệm mô hình dự báo phê duyệt thẻ tín dụng

Chương 4: Kết Luận

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1.Tổng quan quá trình khai phá dữ liệu [12]

Khám phá tri thức từ cơ sở dữ liệu (Knowledge Discovery in Databases – KDD) "KDD là quá trình tự động trích rút các tri thức tiềm ẩn trong một lượng dữ liệu lớn" (Fayyad, Platetsky-Shapiro, Smyth (1996)).

*Khai phá dữ liệu* là một tập hợp các kỹ thuật được sử dụng để tự động khai thác và tìm ra các mối quan hệ lẫn nhau của dữ liệu trong một tập hợp dữ liệu khổng lồ và phức tạp, đồng thời cũng tìm ra các mẫu tiềm ẩn trong tập dữ liệu đó.

Khai phá dữ liệu là một bước trong 7 bước của quá trình KDD và KDD được xem như 7 quá trình khác nhau theo thứ tự sau:

*1. Làm sạch dữ liệu* (data cleaning & preprocessing): Loại bỏ nhiễu và các dữ liệu không cần thiết.

*2. Tích hợp dữ liệu:* (data integration): quá trình hợp nhất dữ liệu thành những kho dữ liệu (data warehouses & data marts) sau khi đã làm sạch và tiền xử lý (data cleaning & preprocessing).

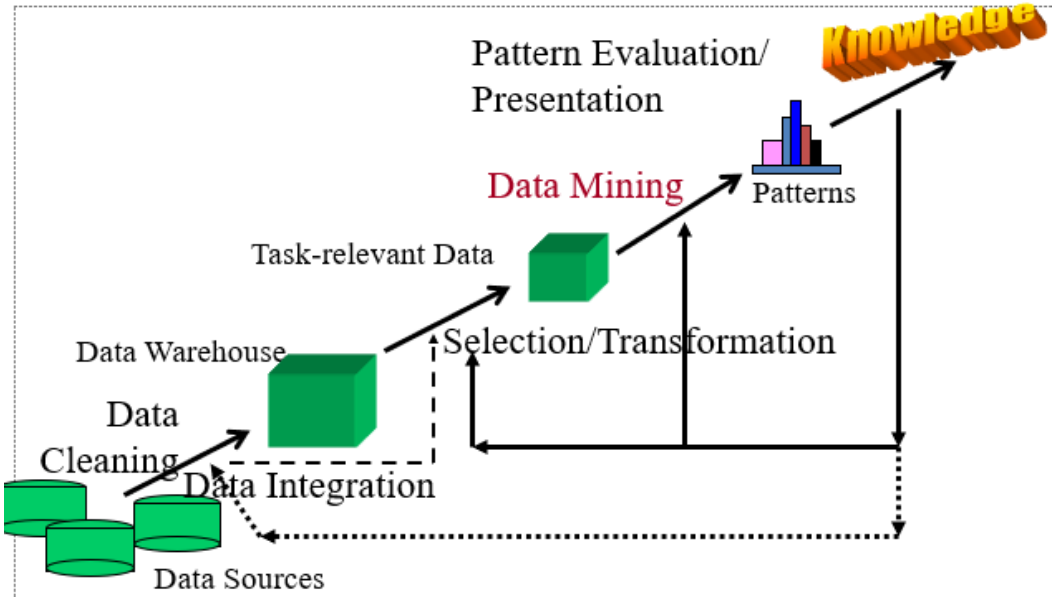
*3. Trích chọn dữ liệu* (data selection): trích chọn dữ liệu từ những kho dữ liệu và sau đó chuyển đổi về dạng thích hợp cho quá trình khai thác tri thức. Quá trình này bao gồm cả việc xử lý với dữ liệu nhiễu (noisy data), dữ liệu không đầy đủ (incomplete data), .v.v.

*4. Chuyển đổi dữ liệu*: Các dữ liệu được chuyển đổi sang các dạng phù hợp cho quá trình xử lý

*5. Khai phá dữ liệu* (data mining): Là một trong các bước quan trọng nhất, trong đó sử dụng những phương pháp thông minh để chắt lọc ra những mẫu dữ liệu.

*6. Ước lượng mẫu* (knowledge evaluation): Quá trình đánh giá các kết quả tìm được thông qua các độ đo nào đó.

*7. Biểu diễn tri thức* (knowledge presentation): Quá trình này sử dụng các kỹ thuật để biểu diễn và thể hiện trực quan cho người dùng.




Hình 2.1: Quá trình khai phá tri thức

## 2.2. Bài toán phân lớp trong quá trình khai phá dữ liệu

### *2.2.1. Khái niệm phân lớp*

Phân lớp dữ liệu là quá trình học có giám sát trên một tập dữ liệu đầu vào nhằm xây dựng một mô hình để có thể dự đoán xu hướng cho các dữ liệu mới.

***Đầu vào:*** Tập các dữ liệu có dạng (x,y) = (x1,x2,…,xn, y)

- x là biến độc lập (Independent variable) mô tả các thuộc tính của một đối tượng.
- y là biến phụ thuộc (Dependent variable) cần tìm hiểu, phân loại. y còn gọi là thuộc tính nhãn.

***Đầu ra***: Một mô hình có khả năng phân loại đúng đắn cho tập các dữ liệu đầu vào.

### *2.2.2. Quá trình phân lớp dữ liệu*

***Bước học*** (bước huấn luyện): Xây dựng mô hình

- Xác định tập dữ liệu huấn luyện gồm các mẫu đã được gán nhãn y.
- Chạy một thuật toán phân lớp trên tập dữ liệu huấn luyện.
- Mô hình được biểu diễn dưới dạng các luật phân lớp, các cây quyết định hoặc các công thức toán.

***Bước phân loại***: Sử dụng mô hình: để gán nhãn thích hợp cho các dữ liệu chưa được gán nhãn.

- Ước lượng độ chính xác của mô hình:
- Xác định tập dữ liệu kiểm thử gồm các mẫu đã được gán nhãn y (dữ liệu kiểm thử và dữ liệu huấn luyện phải khác nhau để tránh tình trạng quá khớp over-fitting)
- Chạy mô hình với tập dữ liệu kiểm thử thu được nhãn y’
- So sánh y và y’ để xác định độ chính xác của mô hình.
- Nếu mô hình chính xác, sử dụng nó để dự đoán nhãn cho các dữ liệu cần gán nhãn.

### *2.2.3. Các loại phân lớp*

- *Phân lớp nhị phân/đa lớp*
  - |C| = 2: phân lớp nhị phân.
    Giá trị của thuộc tính nhãn gồm: Có mua hay không mua máy tính?
  - |C| > 2: phân lớp đa lớp.
    Giá trị của thuộc tính nhãn gồm: Thể thao, Chính trị, Văn hóa, Sức khỏe
- *Phân lớp đơn nhãn/đa nhãn*

Đơn nhãn: mỗi mẫu được gán duy nhất vào một lớp. Tất cả các mẫu dữ liệu chỉ có một và chỉ một giá trị nhãn.

Đa nhãn: một mẫu có thể được gán nhiều hơn một lớp. Một bài báo có thể vừa được gán nhãn Thể thao vừa được gán nhãn Sức khỏe.

### *2.2.4. Một số thuật toán phân lớp dữ liệu*

a. Thuật toán cây quyết định **[8]**

Nhóm thuật toán dựa trên cây là nhóm thuật toán học máy có giám sát, sử dụng cấu trúc cây để phân lớp (Classification) hoặc hồi quy (Regression) các đối tượng dựa trên các thuộc tính của chúng.

***Đầu vào***: Dữ liệu đầu vào D, D là các phân vùng của dữ liệu, C là lớp với nhãn tương ứng với các vùng dữ liệu Di (I = 1,...,m).

***Đầu ra:*** Cây quyết định

***Thuật toán***: Thuật toán này bắt đầu bằng việc khởi tạo một nút gốc. Nút gốc này đại diện cho toàn bộ tập dữ liệu huấn luyện. Sau đó, thuật toán sẽ lặp lại các bước sau cho đến khi tất cả các nút trong cây đều thuần nhất:

1. Nếu tất cả các đối tượng trong nút hiện tại đều thuộc cùng một lớp, thì nút hiện tại là nút lá và được gán nhãn lớp của các đối tượng đó.
2. Ngược lại, thuật toán sẽ sử dụng một độ đo để xác định thuộc tính tốt nhất để phân chia nút hiện tại.
3. Ứng với mỗi giá trị của thuộc tính được chọn, thuật toán sẽ tạo ra một nút con mới. Mỗi nút con sẽ chứa tất cả các đối tượng trong nút hiện tại có giá trị thuộc tính đó.
4. Lặp lại các bước 1-3 cho mỗi nút chưa thuần nhất

**Một số thuật toán trong cây quyết định:**

● ***ID3***

Thuật toán ID3 là một thuật toán xây dựng cây quyết định, sử dụng phương pháp tham lam tìm kiếm từ trên xuống thông qua không gian của các nhánh có thể không có backtracking. Thuật toán ID3 sử dụng Entropy và Information Gain để chọn thuộc tính phân tách tốt nhất các đối tượng vào các lớp

CT1: Công thức tính lượng thông tin kỳ vọng để phân lớp một phần tử trong tập dữ liệu D- Entropy của D:

$$Entropy(D) = Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

(m là số lượng các lớp trong D)

CT2: Với pi là xác suất một phần tử dữ liệu trong D thuộc vào lớp Ci:

$$p_i = \frac{|D_i|}{|D|}$$

(Di là tập các phần tử dữ liệu trong D thuộc vào lớp Ci.)

CT3: Lượng thông tin kỳ vọng để phân lớp chính xác các tập con: ($Info_A(D)$ càng nhỏ thì độ đồng nhất của các tập con càng cao)

$$Info_A(D) = -\sum_{j=1}^{v} \frac{|Dj|}{|D|} \times Info(D_j)$$

CT4: Công thức đo độ lợi thông tin: (Gain(A) càng lớn càng tốt)

$$Gain(A) = Info(D) - Info_A(D)$$

● ***C4.5***

Thuật toán C4.5 là một thuật toán xây dựng cây quyết định, là một cải tiến của thuật toán ID3. C4.5 sử dụng Gain Ratio thay vì Information Gain để chọn thuộc tính phân tách tốt nhất, tránh hiện tượng ưu tiên những thuộc tính có nhiều giá trị với giá trị thông tin phân tán Split Information

CT1: Công thức độ đo của sự phân tán của các giá trị của thuộc tính A:

$$\text{SplitInfo}_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} * \log_2\left(\frac{|D_j|}{|D|}\right)$$

CT2: Thuộc tính A được chọn làm nút hiện tại để chia nhỏ tập dữ liệu khi GainRatio(A) là giá trị lớn nhất và nó được tính bằng:

$$\text{GainRatio}(A) = \frac{Gain(A)}{SplitInfo_A(D)}$$

- ***CART*** [7]

Thuật toán CART (Classification and Regression Tree) là một thuật toán xây dựng cây quyết định, có thể áp dụng cho cả bài toán phân loại và hồi quy. Sử dụng Gini Index thay vì Information Gain hoặc Gain Ratio để chọn thuộc tính phân tách tốt nhất, tránh hiện tượng ưu tiên những thuộc tính có nhiều giá trị và sử dụng các phân tách nhị phân, tức mỗi nút chỉ có hai nhánh, giúp giảm thiểu độ phức tạp của cây.

CT1: Gini Index của một tập dữ liệu A:

$$\text{Gini}(D) = 1 - \sum_{i=1}^{m} p_i^2$$

CT2: Gini Index của thuộc tính A trên một tập dữ liệu D:

$$\text{Gini}_A(D) = \frac{|D_1|}{|D|}\text{Gini}(D_1) + \frac{|D_2|}{|D|}\text{Gini}(D_2)$$

CT3: Delta Gini

$$\Delta\text{GiniA} = \text{GiniD} - \text{Gini}_A(D)$$

b. Thuật toán K Nearest Neighbors (KNN) [5]

Thuật toán k-Nearest Neighbors (viết tắt là KNN) là một thuật toán học máy có giám sát, sử dụng phương pháp học dựa trên các điểm lân cận để phân loại hoặc hồi quy các đối tượng mới. Thuật toán KNN không học một điều gì từ dữ liệu huấn luyện (đây cũng là lý do thuật toán này được xếp vào loại lazy learning - máy lười học), mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới, mà chỉ so sánh đối tượng mới với các đối tượng trong dữ liệu huấn luyện để tìm ra k đối tượng gần nhất. Kết quả phân loại hoặc hồi quy của đối tượng mới được quyết định bằng cách bỏ phiếu đa số hoặc tính trung bình của k đối tượng gần nhất.

***Đầu vào***:

- Tập dữ liệu D gồm n phần tử trong không gian m chiều. Tập dữ liệu này bao gồm các điểm dữ liệu được mô tả bằng m thuộc tính. Các thuộc tính này có thể là định lượng hoặc định tính.
- Số cụm k. Số cụm k là số cụm mà thuật toán KNN sẽ sử dụng để phân loại dữ liệu.
- Dữ liệu cần phân lớp. Dữ liệu cần phân lớp là một điểm dữ liệu mới mà thuật toán KNN sẽ sử dụng để dự đoán nhãn lớp.

***Đầu ra***: Lớp hoặc giá trị đầu ra của đối tượng mới

***Thuật toán***:

1. Ta có D là tập các điểm dữ liệu đã được gắn nhãn và A là dữ liệu chưa được phân loại.
2. Đo khoảng cách (Euclidian, Manhattan, Minkowski, Minkowski hoặc Trọng số) từ dữ liệu mới A đến tất cả các dữ liệu khác đã được phân loại trong D.
3. Chọn K (K là tham số mà bạn định nghĩa) khoảng cách nhỏ nhất.
4. Kiểm tra danh sách các lớp có khoảng cách ngắn nhất và đếm số lượng của mỗi lớp xuất hiện.
5. Lấy đúng lớp (lớp xuất hiện nhiều lần nhất).
6. Lớp của dữ liệu mới là lớp mà bạn đã nhận được ở bước 5.

***3 cách cơ bản để tính khoảng cách 2 điểm dữ liệu x, y có k thuộc tính:***

*Euclidian*: d(x,y) = $\sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$

*Manhattan:* d(x,y) = $\sum_{i=1}^{n} |x_i - y_i|$

*Minkowski:* d(x,y) = $(\sum_{i=1}^{n} |x_i - y_i|^q)^{1/q}$

c.Thuật toán phân lớp Naive Bayes [7]

Thuật toán Naive Bayes là một thuật toán học máy có giám sát, sử dụng định lý Bayes để dự đoán xác suất của một đối tượng thuộc vào một lớp nào đó. Nói một cách dễ hiểu, bộ phân loại Naive Bayes giả định rằng sự hiện diện của một đối tượng cụ thể trong một lớp là độc lập, không liên quan đến sự hiện diện của bất kỳ đối tượng nào khác.

***Đầu vào***:

- Tập dữ liệu D gồm n phần tử trong không gian m chiều. Tập dữ liệu này bao gồm các điểm dữ liệu được mô tả bằng m thuộc tính. Các thuộc tính này có thể là định lượng hoặc định tính.
- Dữ liệu cần phân lớp. Dữ liệu cần phân lớp là một điểm dữ liệu mới mà thuật toán Naive Bayes sẽ sử dụng để dự đoán nhãn lớp.

***Đầu ra***: Lớp của đối tượng mới, được dự đoán bằng cách tính xác suất hậu nghiệm (posterior probability) của mỗi lớp khi biết các thuộc tính đầu vào, và chọn lớp có xác suất cao nhất.

***Thuật toán***: Bộ phân lớp Naive bayes hay bộ phân lớp Bayes (simple byes classifier) hoạt động như sau:

1. Gọi D là tập dữ liệu huấn luyện, trong đó mỗi phần tử dữ liệu X được biểu diễn bằng một vector chứa n giá trị thuộc tính A1, A2,...,An = {x1,x2,...,xn}
2. Giả sử có m lớp C1, C2,..,Cm. Cho một phần tử dữ liệu X, bộ phân lớp sẽ gán nhãn cho X là lớp có xác suất hậu nghiệm lớn nhất. Cụ thể, bộ phân lớp Bayes sẽ dự đoán X thuộc vào lớp Ci nếu và chỉ nếu:

$$P(Ci|X) > P(Cj|X) \quad (1 <= i, j <= m, i != j)$$

Giá trị này sẽ tính dựa trên định lý Bayes.

3. Để tìm xác suất lớn nhất, ta nhận thấy các giá trị P(X) là giống nhau với mọi lớp nên không cần tính. Do đó ta chỉ cần tìm giá trị lớn nhất của P(X|Ci) * P(Ci). Chú ý rằng P(Ci) được ước lượng bằng |Di|/|D|, trong đó Di là tập các phần tử dữ liệu thuộc lớp Ci. Nếu xác suất tiền nghiệm P(Ci) cũng không xác định được thì ta coi chúng bằng nhau P(C1) = P(C2) = ... = P(Cm), khi đó ta chỉ cần tìm giá trị P(X|Ci) lớn nhất.
4. Khi số lượng các thuộc tính mô tả dữ liệu là lớn thì chi phí tính toàn P(X|Ci) là rất lớn, dó đó có thể giảm độ phức tạp của thuật toán Naive Bayes giả thiết các thuộc tính độc lập nhau. Khi đó ta có thể tính:

$$P(X|Ci) = P(x1|Ci)...P(xn|Ci)$$

Gaussian Naive Bayes là một biến thể của thuật toán Naive Bayes, sử dụng phân phối chuẩn (Gaussian distribution) để tính xác suất của các thuộc tính có giá trị thực. Công thức của Gaussian Naive Bayes là:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

Trong đó:

- $P(x_i|y)$: xác suất của thuộc tính thứ i khi biết lớp y (likelihood).
- $x_i$: giá trị của thuộc tính thứ i của đối tượng mới.
- $\mu_y$: giá trị trung bình của thuộc tính thứ i trong tập dữ liệu huấn luyện thuộc lớp y.
- $\sigma_y$ : độ lệch chuẩn của thuộc tính thứ i trong tập dữ liệu huấn luyện thuộc lớp y.

Để tính giá trị trung bình và độ lệch chuẩn của mỗi thuộc tính, ta có thể sử dụng các công thức sau:∈

$$\mu_y = \frac{1}{n_y}\sum_{x \in D_y} x_i$$

$$\sigma_y = \sqrt{\frac{1}{n_y}\sum_{x \in D_y} (x_i - \mu_y)^2}$$

Trong đó:

- $n_y$: số lượng đối tượng trong tập dữ liệu huấn luyện thuộc lớp y.
- $D_y$: là tập con của tập dữ liệu huấn luyện chứa các đối tượng thuộc lớp y.
- $x_i$: giá trị của thuộc tính thứ i của đối tượng trong $D_y$

## 2.3. Bài toán luật kết hợp trong quá trình khai phá dữ liệu

### *2.3.1. Khái niệm luật kết hợp* [13]

Khai phá luật kết hợp với các quy tắc kết hợp là các câu lệnh If/then đơn giản giúp các mối quan hệ trong cơ sở dữ liệu. Hầu hết các thuật toán học máy liên quan tập dữ liệu số và có xu hướng toán học. Tuy nhiên, khai phá luật kết hợp với dữ liệu không phải số (non-numeric), dữ liệu định danh (categorical data). Khai phá quy tắc luật kết hợp là một thủ tục nhằm mục đích tìm các mẫu xảy ra phổ biến, đồng thời, có liên kết với nhau trong các cơ sở dữ liệu khác nhau như cơ sở dữ liệu quan hệ, cơ sở dữ liệu giao dịch và các dạng kho lưu trữ khác.

### *2.3.2. Thuật toán Apriori*

Cho CSDL D tập mục I, với:

Minsup: độ hỗ trợ tối thiểu

Minconf: độ tin cậy tối thiểu. Cần phát hiện các tập luật kết hợp có dạng X⇒Y thỏa mãn:

- sup(X⇒Y) ≥ minsup
- conf(X⇒Y) ≥ minconf

Dữ liệu vào: Tập các giao dịch D, ngưỡng support tối thiểu minsup

Dữ liệu ra: L - tập mục phổ biến trong D

Phương pháp:

$L_1$= {large 1- itemset} // tìm tất cả các tập mục phổ biến nhận được $L_1$

**for** (k=2; $L_{k-1} \neq \emptyset$; $k++$) **do**

**begin**

$C_k$ = apriori-gen($L_{k-1}$) // sinh ra tập ứng cử viên từ $L_{k-1}$

for (mỗi một giao dịch T ∈ D) **do**

**begin**

$C_T$ = subset ($C_k$, T); lấy tập con của T là ứng cử viên trong $C_k$

**for** (mỗi một ứng viên c ∈ $C_T$ ) **do**

c.count++; // tăng bộ đếm tần suất 1 đơn vị

**end**

$L_k$ = {c ∈ $C_k$ | c.count ≥ minsup * |D|};

**end**

**return** $\cup k\ L_k$

## 2.4. Cơ sở lý thuyết về tín dụng trong ngân hàng [4]

### *2.4.1. Tín dụng là gì*

Tín dụng là khái niệm thể hiện mối quan hệ giữa người cho vay và người vay. Người cho vay cung cấp nguồn tài chính cho người vay trong một thời hạn nhất định và thường kèm theo lãi suất. Người vay có nghĩa vụ trả lại số tiền hoặc giá trị hàng hoá đã vay khi đến hạn trả nợ. Tín dụng có thể được thực hiện bằng nhiều hình thức, như thẻ tín dụng, vay tiêu dùng, vay mua nhà, vay kinh doanh,...

Tín dụng ngân hàng là mối quan hệ tín dụng giữa ngân hàng, các tổ chức tín dụng (TCTD), với các cá nhân, tổ chức có nhu cầu vay tiền. Trong đó, các TCTD chuyển giao tài sản cho bên đi vay sử dụng trong một thời gian nhất định theo thoả thuận, và bên đi vay có trách nhiệm hoàn trả cả vốn gốc và lãi cho TCTD khi đến hạn thanh toán.

### *2.4.2. Những điều cần biết về thẻ tín dụng*

Có thể hiểu đơn giản, thẻ tín dụng là loại thẻ cho phép khách hàng thanh toán mà không cần có tiền trong thẻ. Nói cách khác thì đây chính là một hình thức vay tiền của ngân hàng để thanh toán trước và tới kỳ hạn thanh toán chủ thẻ có nhiệm vụ phải trả tiền lại đầy đủ cho ngân hàng.

Thông thường, thời gian miễn lãi kéo dài khoảng từ 45 - 50 ngày (tùy từng ngân hàng). Nếu chủ thẻ trả đủ nợ trong thời gian miễn lãi này thì sẽ không bị tính lãi.

Điều kiện mở thẻ tín dụng:

Đối với cá nhân:

- Là người đủ 18 tuổi trở lên có năng lực hành vi dân sự đầy đủ
- Người từ đủ 15 tuổi đến dưới 18 tuổi: Không bị mất/hạn chế năng lực hành vi dân sự.

Đối với tổ chức, doanh nghiệp: Là pháp nhân được thành lập và hoạt động hợp pháp. Tổ chức được lập văn bản ủy quyền cho cá nhân sử dụng thẻ của tổ chức đó hoặc cho phép cá nhân sử dụng thẻ phụ.

### *2.4.3. Quy trình phát hành thẻ tín dụng của ngân hàng* [10]

Thông thường quá trình làm thẻ tín dụng tại các ngân hàng sẽ thực hiện theo các bước sau:

Bước 1: Khách hàng đến phòng giao dịch, chi nhánh của ngân hàng yêu cầu mở thẻ.

Bước 2: Cung cấp thông tin và hồ sơ theo yêu cầu của nhân viên ngân hàng.

Bước 3: Nhân viên ngân hàng kiểm tra hồ sơ, đối chiếu thông tin của khách hàng với các thông tin lưu trữ tại ngân hàng.

Bước 4: Nhân viên thông báo kết quả phát hành thẻ.

- Có thể từ chối phát hành thẻ trong các trường hợp: Thông tin đối chiếu hồ sơ không hợp lệ hoặc không đủ điều kiện phát hành thẻ...
- Chấp thuận phát hành thẻ trong các trường hợp: Thông tin đối chiếu hợp lệ và khách hàng đủ điều kiện phát hành thẻ.

# CHƯƠNG 3: XÂY DỰNG VÀ THỰC NGHIỆM MÔ HÌNH DỰ BÁO VỚI DỮ LIỆU ÍT THUỘC TÍNH

## 3.1. Mục đích thực nghiệm

Minh họa thuật toán phân lớp với dữ liệu ít bằng công thức tay nhằm mục đích giúp người học hiểu rõ hơn về cách thức hoạt động của thuật toán.

Khi viết code, người học thường chỉ tập trung vào việc triển khai thuật toán mà không chú ý đến cách thức hoạt động của thuật toán. Việc minh họa thuật toán bằng công thức tay sẽ giúp người học hiểu rõ hơn về các bước mà thuật toán thực hiện, từ đó có thể hiểu rõ hơn về cách thức hoạt động của thuật toán và có thể điều chỉnh thuật toán một cách hiệu quả hơn.

Các bước mà thuật toán thực hiện:

Cách thức mà thuật toán sử dụng dữ liệu để đưa ra quyết định phân lớp

Cách thức mà thuật toán tính toán độ chính xác

=> Ngoài ra, việc minh họa thuật toán phân lớp với dữ liệu ít bằng công thức tay cũng có thể giúp người học hiểu rõ hơn về các yếu tố ảnh hưởng đến độ chính xác của thuật toán. Ví dụ, độ chính xác của thuật toán kNN phụ thuộc vào số lượng k, độ đa dạng của dữ liệu và độ nhiễu của dữ liệu.

## 3.2. Cách thức thực hiện thuật toán CART

Trong tập dữ liệu thu thập được từ Kaggle, nhóm lựa chọn ra 14 bản ghi với các thuộc tính CODE_GENDER, FLAG_OWN_REALITY, AMT_INCOME_TOTAL, AGE và STATUS để làm cơ sở dữ liệu xây dựng mô hình bằng thuật toán CART. Các bản ghi này đã được tiền xử lý và biến đổi

| CODE_GENDER | FLAG_OWN_REALITY | AMT_INCOME_TOTAL | AGE | STATUS |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 2 | 1 | 1 |
| 0 | 0 | 1 | 2 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 2 | 1 |
| 1 | 0 | 0 | 2 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 2 | 1 |
| 0 | 1 | 1 | 0 | 1 |

*Bảng 3.1: Tập dữ liệu mẫu xây dựng mô hình bằng thuật toán CART*

$$\textbf{GINI(D)} = 1 - \sum_{i=1}^{m} p_i^2 = 1 - \left[\left(\frac{1}{14}\right)^2 + \left(\frac{13}{14}\right)^2\right] = 0.13$$

$$\textbf{GINI}_A\textbf{(D)} = \frac{|D1|}{|D|}\textbf{Gini(D}_1\textbf{)} + \frac{|D2|}{|D|}\textbf{Gini(D}_2\textbf{)}$$

Có 4 biến độc lập:

**CG: CODE_GENDER**

**FOR: FLAG_OWN_REALITY**

**AIT: AMT_INCOME_TOTAL**

**A: AGE**

| CÁCH PHÂN | NÚT TRÁI | NÚT PHẢI | $GINI_A(D)$ |
|---|---|---|---|
| 1 | CG0 | CG1 | $\frac{11}{14} \times \left[1 - \left(\frac{0}{11}\right)^2 - \left(\frac{11}{11}\right)^2\right] + \frac{3}{14} \times \left[1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2\right]$=0.1 |
| 2 | FOR 0 | FOR 1 | $\frac{6}{14} \times \left[1 - \left(\frac{1}{6}\right)^2 - \left(\frac{5}{6}\right)^2\right] + \frac{8}{14} \times \left[1 - \left(\frac{0}{8}\right)^2 - \left(\frac{8}{8}\right)^2\right]$=0.12 |
| 3 | AIT0 | AIT12 | $\frac{9}{14} \times \left[1 - \left(\frac{1}{9}\right)^2 - \left(\frac{8}{9}\right)^2\right] + \frac{5}{14} \times \left[1 - \left(\frac{0}{5}\right)^2 - \left(\frac{5}{5}\right)^2\right]$=0.13 |
| 4 | AIT1 | AIT02 | $\frac{4}{14} \times \left[1 - \left(\frac{0}{4}\right)^2 - \left(\frac{4}{4}\right)^2\right] + \frac{10}{14} \times \left[1 - \left(\frac{1}{10}\right)^2 - \left(\frac{9}{10}\right)^2\right]$=0.13 |
| 5 | AIT2 | AIT01 | $\frac{1}{14} \times \left[1 - \left(\frac{1}{1}\right)^2 - \left(\frac{0}{1}\right)^2\right] + \frac{13}{14} \times \left[1 - \left(\frac{1}{13}\right)^2 - \left(\frac{12}{13}\right)^2\right]$=0.13 |
| 6 | A0 | A12 | $\frac{3}{14} \times \left[1 - \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2\right] + \frac{11}{14} \times \left[1 - \left(\frac{1}{11}\right)^2 - \left(\frac{10}{11}\right)^2\right]$=0.13 |
| 7 | A1 | A02 | $\frac{7}{14} \times \left[1 - \left(\frac{1}{7}\right)^2 - \left(\frac{6}{7}\right)^2\right] + \frac{7}{14} \times \left[1 - \left(\frac{0}{7}\right)^2 - \left(\frac{7}{7}\right)^2\right]$=0.12 |
| 8 | A2 | A01 | $\frac{4}{14} \times \left[1 - \left(\frac{0}{4}\right)^2 - \left(\frac{4}{4}\right)^2\right] + \frac{10}{14} \times \left[1 - \left(\frac{1}{10}\right)^2 - \left(\frac{9}{10}\right)^2\right]$=0.13 |

*Bảng 3.2: Bản tính Gini D lần 1*

Với cách chia 1 có $GINI_A(D)$ nhỏ nhất nên chọn CODE_GENDER làm nút gốc.

- Xét nhánh CODE_GENDER 0: ta thấy STATUS đều mang giá trị 1 nên CODE_GENDER 0 là nút lá.
- Xét nhánh CODE_GENDER 1:

| FLAG_OWN_REALITY | AMT_INCOME_TOTAL | AGE | STATUS |
|---|---|---|---|
| 0 | 2 | 1 | 1 |
| 0 | 0 | 2 | 1 |

| | | | |
|---|---|---|---|
| 0 | 0 | 1 | 0 |

Ta có bảng cách chia:

| CÁCH PHÂN | NÚT TRÁI | NÚT PHẢI | $GINI_A(D)$ |
|---|---|---|---|
| 1 | AIT0 | AIT12 | $\frac{2}{3}\times\left[1-\left(\frac{1}{2}\right)^2-\left(\frac{1}{2}\right)^2\right]+\frac{1}{3}\times\left[1-\left(\frac{0}{1}\right)^2-\left(\frac{1}{1}\right)^2\right]=0.33$ |
| 2 | AIT1 | AIT02 | $\frac{3}{3}\times\left[1-\left(\frac{1}{3}\right)^2-\left(\frac{2}{3}\right)^2\right]=0.44$ |
| 3 | AIT2 | AIT01 | $\frac{1}{3}\times\left[1-\left(\frac{1}{1}\right)^2-\left(\frac{0}{1}\right)^2\right]+\frac{2}{3}\times\left[1-\left(\frac{1}{2}\right)^2-\left(\frac{1}{2}\right)^2\right]=0.33$ |
| 4 | A0 | A12 | $\frac{3}{3}\times\left[1-\left(\frac{1}{3}\right)^2-\left(\frac{2}{3}\right)^2\right]=0.44$ |
| 5 | A1 | A02 | $\frac{2}{3}\times\left[1-\left(\frac{1}{2}\right)^2-\left(\frac{1}{2}\right)^2\right]+\frac{1}{3}\times\left[1-\left(\frac{0}{1}\right)^2-\left(\frac{1}{1}\right)^2\right]=0.33$ |
| 6 | A2 | A01 | $\frac{1}{3}\times\left[1-\left(\frac{1}{1}\right)^2-\left(\frac{0}{1}\right)^2\right]+\frac{2}{3}\times\left[1-\left(\frac{1}{2}\right)^2-\left(\frac{1}{2}\right)^2\right]=0.33$ |
| 7 | FOR0 | FOR1 | $\frac{3}{3}\times\left[1-\left(\frac{1}{3}\right)^2-\left(\frac{2}{3}\right)^2\right]=0.44$ |

*Bảng 3.2: Bản tính Gini D lần 2*

Chọn cách chia 1: ta có AMT_INCOME_TOTAL là nút điều kiện, với 2 nhánh con là AMT_INCOME_TOTAL 0 và AMT_INCOME_TOTAL 12

- Xét nhánh con AMT_INCOME_TOTAL 12: STATUS chỉ nhận giá trị 1 nên AMT_INCOME_TOTAL 12 là nút lá.
- Xét nhánh con AMT_INCOME_TOTAL 0:

| FLAG_OWN_REALITY | AGE | STATUS |
|---|---|---|
| 0 | 2 | 1 |
| 0 | 1 | 0 |

Có bảng các cách chia:

| CÁCH PHÂN | NÚT TRÁI | NÚT PHẢI | $GINI_A(D)$ |
|---|---|---|---|
| 1 | A0 | A12 | $\frac{2}{2}\times\left[1-\left(\frac{1}{2}\right)^2-\left(\frac{2}{2}\right)^2\right]=0.5$ |
| 2 | A1 | A02 | $\frac{1}{2}\times\left[1-\left(\frac{1}{1}\right)^2-\left(\frac{0}{1}\right)^2\right]+\frac{1}{2}\times\left[1-\left(\frac{0}{1}\right)^2-\left(\frac{1}{1}\right)^2\right]=0$ |

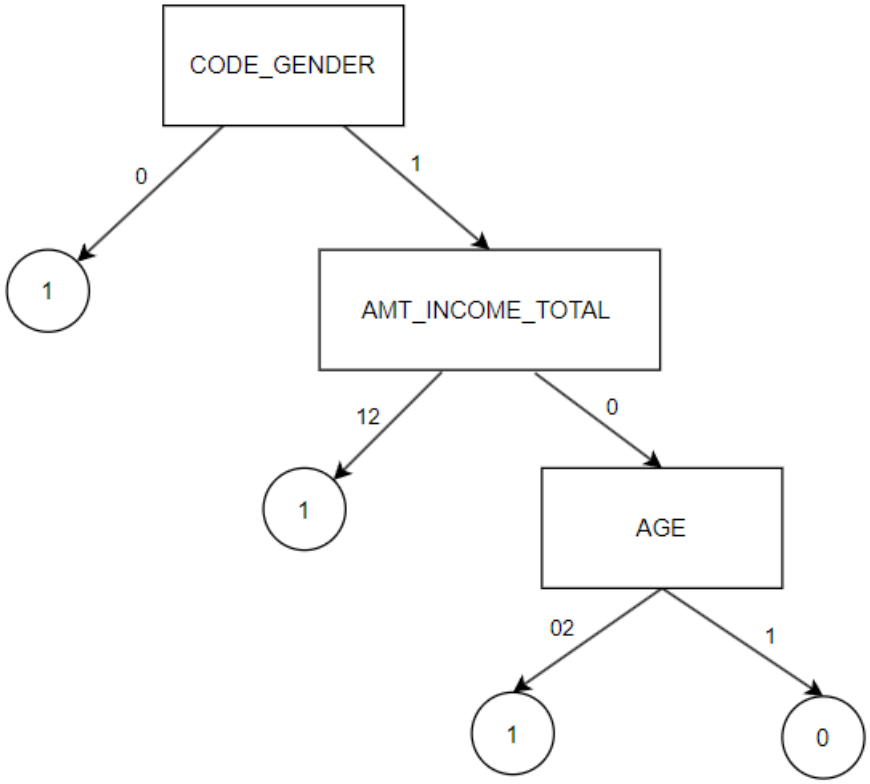
| | | | |
|---|---|---|---|
| 3 | A2 | A01 | $\frac{1}{2}\times\left[1-\left(\frac{1}{1}\right)^2-\left(\frac{0}{1}\right)^2\right]+\frac{1}{2}\times\left[1-\left(\frac{0}{1}\right)^2-\left(\frac{1}{1}\right)^2\right]=0$ |
| 4 | FOR 0 | FOR 1 | $\frac{2}{2}\times\left[1-\left(\frac{1}{2}\right)^2-\left(\frac{1}{2}\right)^2\right]=0.5$ |

*Bảng 3.3: Bản tính Gini D lần 3*

Chọn cách chia 2, ta có AGE là nút quyết định

- Xét nhánh AGE 1: thu được nút lá STATUS = 0
- Xét nhánh AGE 02: thu được nút lá STATUS = 1

⇨ Cây quyết định thu được:




*Hình 3.1: Cây quyết định sau khi thực hiện thuật toán CART*

## 3.3. Cách thức thực hiện thuật toán ID3

| CODE_GENDER | FLAG_OWN_REALITY | AMT_INCOME_TOTAL | AGE | STATUS |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 2 | 1 | 1 |
| 0 | 0 | 1 | 2 | 1 |
| 0 | 1 | 1 | 0 | 1 |

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 2 | 1 |
| 1 | 0 | 0 | 2 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 2 | 1 |
| 0 | 1 | 1 | 0 | 1 |

*Bảng 3.5: Tập dữ liệu mẫu xây dựng mô hình bằng thuật toán ID3*

Lượng thông tin kỳ vọng để phân lớp một phần tử trong tập dữ liệu D- Entropy của D:

$$\mathbf{Info(D)} = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

Với $p_i$ là xác suất một phần tử dữ liệu trong D thuộc vào lớp $C_i$.

$$\mathbf{p_i} = \frac{|Di|}{|D|}$$

$D_i$ là tập các phần tử dữ liệu trong D thuộc vào lớp $C_i$.

m là số lượng các lớp trong D

$$\mathbf{Info_A(D)} = \sum_{j=1}^{v} \frac{|Dj|}{|D|} \times \mathbf{Info(D_j)}$$

Công thức đo độ lợi thông tin:

$$\mathbf{Gain(A) = Info(D) - Info_A(D)}$$

Với bảng giá trị trên, D có 14 phần tử, trong đó 13 phần tử mang nhãn 1, và 1 phần tử mang nhãn 0.

Ta có: $Info(D) = -\frac{13}{14}\log_2(\frac{13}{14}) - \frac{1}{14}\log_2(\frac{1}{14}) = 0.37$

Tính lượng thông tin kỳ vọng của CODE_GENDER gồm 2 giá trị: {"0": 11 nhãn 1, 0 nhãn 0} và {"1": 2 nhãn 1, 1 nhãn 0)

$Info_{CODE\_GENDER} = \frac{11}{14} \times (-\frac{11}{11}\log_2\frac{11}{11} - \frac{0}{11}\log_2\frac{0}{11}) + \frac{3}{14} \times (-\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3}) = 0.197$

Tương tự ta có:

$Info_{FLAG\_OWN\_REALITY} = 0.279$

$Info_{AMT\_INCOME\_TOTAL} = 0.324$

$Info_{FLAG\_OWN\_REALITY}$ = 0.296

Từ đó ta có:

$Gain_{(CODE\_GENDER)}$ = Info(D) - $Info_{CODE\_GENDER}$ = 0.173

$Gain_{(FLAG\_OWN\_REALITY)}$ = 0.091

$Gain_{(AMT\_INCOME\_TOTAL)}$ = 0.046

$Gain_{(AGE)}$ = 0.074

⇨ Chọn CODE_GENDER làm thuộc tính phân nhánh.

- Xét nhánh CODE_GENDER = 0: tất cả mẫu đều có thuộc tính STATUS = 1 nên gán nhãn 1 cho nút lá.
- Xét nhánh CODE_GENDER = 1:
  Tính entropy của $D_{CODE\_GENDER = 1}$ : Info($D_{CODE\_GENDER = 1}$) = 0.918

  Tính lượng thông tin kỳ vọng của các thuộc tính:

  $Info_{FLAG\_OWN\_REALITY}$ = 0.918

  $Info_{AMT\_INCOME\_TOTAL}$ = 0.667

  $Info_{AGE}$ = 0.667

  Từ đó ta có:

  $Gain_{(FLAG\_OWN\_REALITY)}$ = Info($D_{CODE\_GENDER = 1}$) - $Info_{FLAG\_OWN\_REALITY}$ = 0

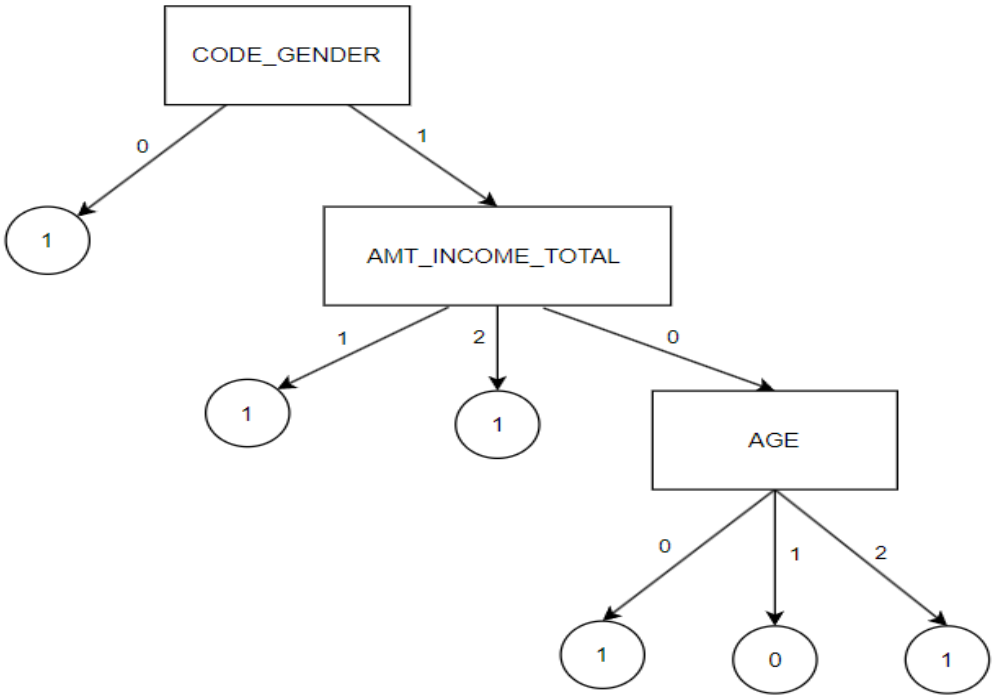
  $Gain_{(AMT\_INCOME\_TOTAL)}$ = 0.251

  $Gain_{(AGE)}$ = 0.251

  ⇨ Chọn AMT_INCOME_TOTAL làm thuộc tính phân nhánh

- Xét nhánh AMT_INCOME_TOTAL = 2: STATUS chỉ nhận giá trị 1 nên gán nhãn 1 cho nút lá
- Xét nhánh AMT_INCOME_TOTAL = 1: tập $D_{AMT\_INCOME\_TOTAL = 1}$ rỗng nên ta gán giá trị phổ biến nhất trong tập D cho STATUS => gán nhãn 1 cho nút lá
- Xét nhánh AMT_INCOME_TOTAL = 0: vì FLAG_OWN_REALITY chỉ nhận 1 giá trị nên có Gain = 0, ta chọn AGE làm thuộc tính phân nhánh
- Xét nhánh AGE = 0: tập $D_{AGE = 0}$ rỗng nên ta gán cho nó giá trị phổ biến nhất, tức là STATUS = 1
- Xét nhánh AGE = 1: gán nhãn 0 cho STATUS
- Xét nhánh AGE = 2: gán nhãn 1 cho STATUS

⇨ Cây quyết định thu được:




*Hình 3.2: Cây quyết định sau khi thực hiện thuật toán ID3*

3.3. Cách thức thực hiện thuật toán KNN

Ta có 13 dòng dữ liệu mẫu lấy từ bộ dữ liệu như sau:

| CODE_GENDER | FLAG_OWN_REALTY | AMT_INCOME_TOTAL | AGE | STATUS |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 2 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 2 | 0 |
| 0 | 1 | 1 | 2 | 0 |
| 0 | 1 | 1 | 2 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 |

*Bảng 3.4: 14 bản ghi ngẫu nhiên lấy từ bộ dữ liệu*

Áp dụng thuật toán KNN với k = 6 và tập dữ liệu đầu vào là: Giới tính, tài sản, thu nhập tuổi

Xét khoảng cách từ A(1,1,0,2) đến các điểm trong dữ liệu

d(A,C1) = $\sqrt{(1-1)^2 + (1-1)^2 + (0-1)^2 + (2-0)^2} = \sqrt{5}$

d(A,C2) = $\sqrt{(1-1)^2 + (1-1)^2 + (0-1)^2 + (2-0)^2} = \sqrt{5}$

d(A,C3) = $\sqrt{(1-1)^2 + (1-0)^2 + (0-0)^2 + (2-2)^2} = 1$

d(A,C4) = $\sqrt{(1-0)^2 + (1-1)^2 + (0-1)^2 + (2-1)^2} = \sqrt{3}$

d(A,C5) = $\sqrt{(1-0)^2 + (1-1)^2 + (0-1)^2 + (2-1)^2} = \sqrt{3}$

d(A,C6) = $\sqrt{(1-0)^2 + (1-1)^2 + (0-1)^2 + (2-1)^2} = \sqrt{3}$

d(A,C7) = $\sqrt{(1-0)^2 + (1-1)^2 + (0-1)^2 + (2-2)^2} = \sqrt{2}$

d(A,C8) = $\sqrt{(1-0)^2 + (1-1)^2 + (0-1)^2 + (2-2)^2} = \sqrt{2}$

d(A,C9) = $\sqrt{(1-0)^2 + (1-1)^2 + (0-1)^2 + (2-2)^2} = \sqrt{2}$

d(A,C10) = $\sqrt{(1-1)^2 + (1-1)^2 + (0-1)^2 + (2-1)^2} = \sqrt{2}$

d(A,C11) = $\sqrt{(1-1)^2 + (1-0)^2 + (0-1)^2 + (2-1)^2} = \sqrt{3}$

d(A,C12) = $\sqrt{(1-1)^2 + (1-1)^2 + (0-0)^2 + (2-1)^2} = 1$

d(A,C13) = $\sqrt{(1-1)^2 + (1-1)^2 + (0-0)^2 + (2-1)^2} = 1$

Với k = 6, ta lấy 5 điểm có khoảng cách gần nhất với A là C1, C2, C4, C5, C6, C11

Trong 5 điểm này có 5 trạng thái là 1, và có 1 trạng thái là 0

=> A được phân loại trạng thái là 1

### 3.4. Thuật toán *phân lớp Naive Bayes*

X= (GENDER = 0, REALITY = 1, INCOME = 2, AGE = 1, STATUS = 0)
C1 = {X'|X'.CREDIT_CARD = YES)
C2 = {X"|X".CREDIT_CARD = NO)
P(CREDIT_CARD = YES) = 11/14 = 0.786
P(CREDIT_CARD = NO) = 3/14 = 0.214
P(GENDER = 0 | CREDIT_CARD = YES) = 7/11 = 0.63

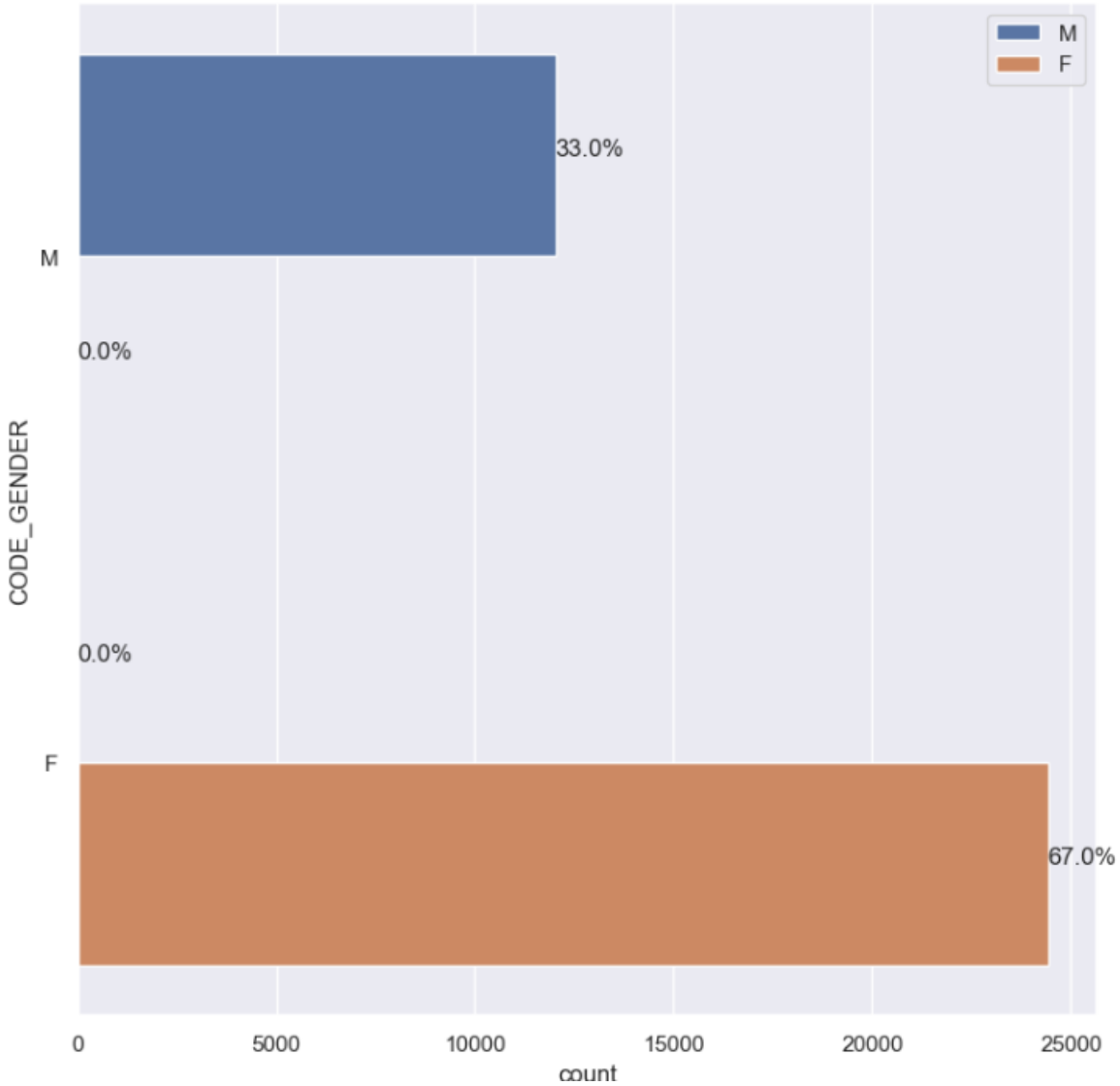
P(GENDER = 0 | CREDIT_CARD = NO) = 3/3 = 1
P(REALITY = 1 | CREDIT_CARD = YES) = 11/11 = 1
P(REALITY = 1 | CREDIT_CARD = NO) = 0/3 = 0
P(INCOME = 2 | CREDIT_CARD = YES) = 1/11 = 0.09
P(INCOME = 2 | CREDIT_CARD = NO) = 0/3 = 0
P(AGE = 1 | CREDIT_CARD = YES) = 3/11 = 0.27
P(AGE = 1 | CREDIT_CARD = NO) = 1/3= 0.33
P(STATUS = 0 | CREDIT_CARD = YES) = 2/11 = 0.18
P(STATUS = 0 | CREDIT_CARD = NO) = 3/3 = 1
P(X|CREDIT_CARD = YES) = P(GENDER = 0 | CREDIT_CARD = YES)
* P(REALITY = 1 | CREDIT_CARD = YES)
* P(INCOME = 2 | CREDIT_CARD = YES)
* P(AGE = 1 | CREDIT_CARD = YES)
* P(STATUS = 0 | CREDIT_CARD = YES)
= 0.63*1*0.09*0.27*0.18 = 0.00276
P(X|CREDIT_CARD = NO) = P(GENDER = 0 | CREDIT_CARD = NO)
* P(REALITY = 1 | CREDIT_CARD = NO)
* P(INCOME = 2 | CREDIT_CARD = NO)
* P(AGE = 1 | CREDIT_CARD = NO)
* P(STATUS = 0 | CREDIT_CARD = NO)
= 1*0*0*0.33*1= 0
P(X|CREDIT_CARD = YES)* P(CREDIT_CARD = YES) = 0.00276*0.786 = 0.00217
P(X|CREDIT_CARD = NO)* P(CREDIT_CARD = NO) = 0
⇨ X ∈ C1

# CHƯƠNG 4: XÂY DỰNG VÀ THỰC NGHIỆM MÔ HÌNH DỰ BÁO VỚI DỮ LIỆU ĐẦY ĐỦ THUỘC TÍNH

## 4.1. Phân tích dữ liệu

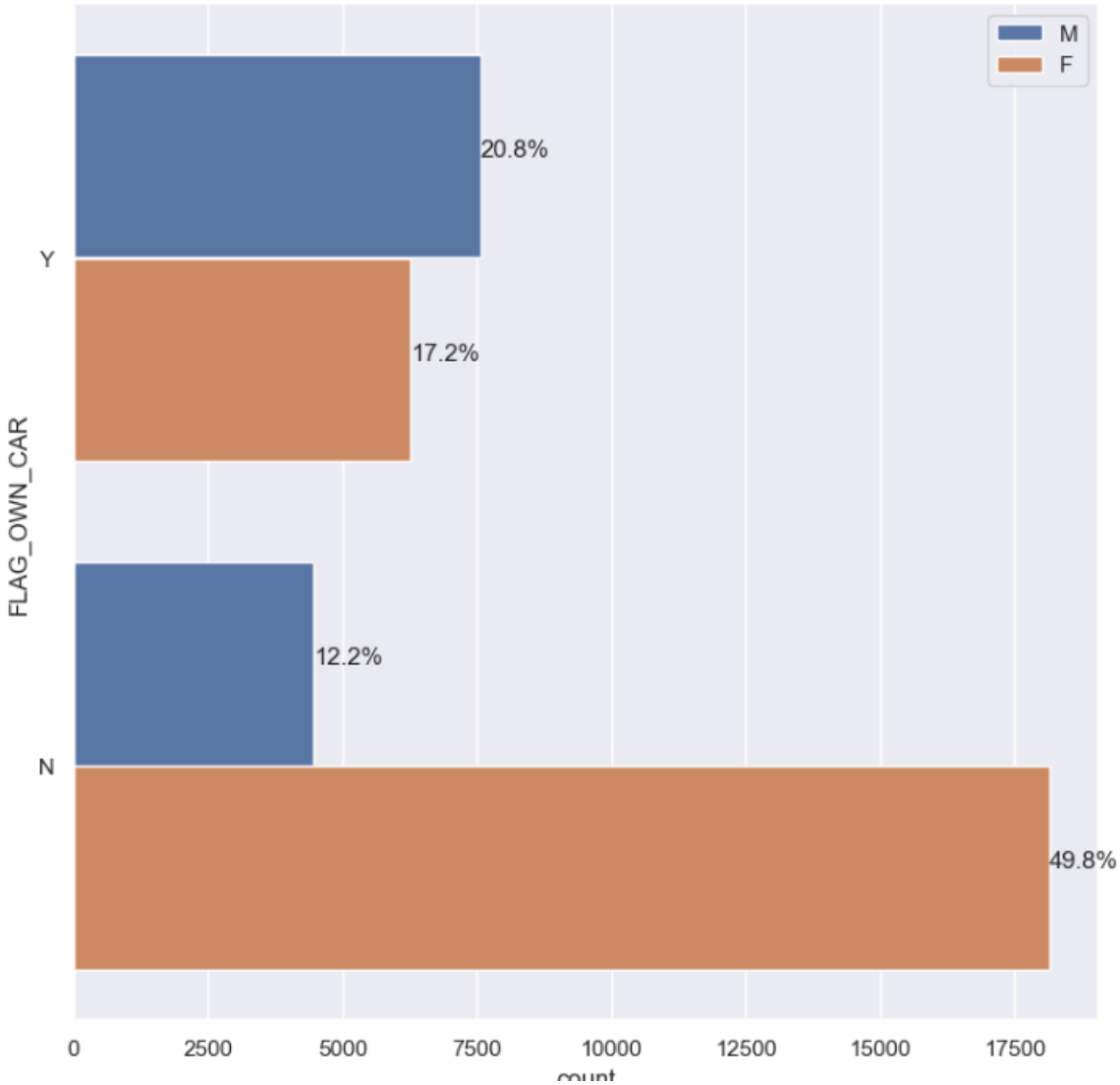
Từ tập dữ liệu thu được, ta tiến hành vẽ các biểu đồ đánh giá thuộc tính theo giới tính và thu được một số phân tích như sau:

- Trong số những người đăng ký có 33% là nam giới, 67% là nữ giới:



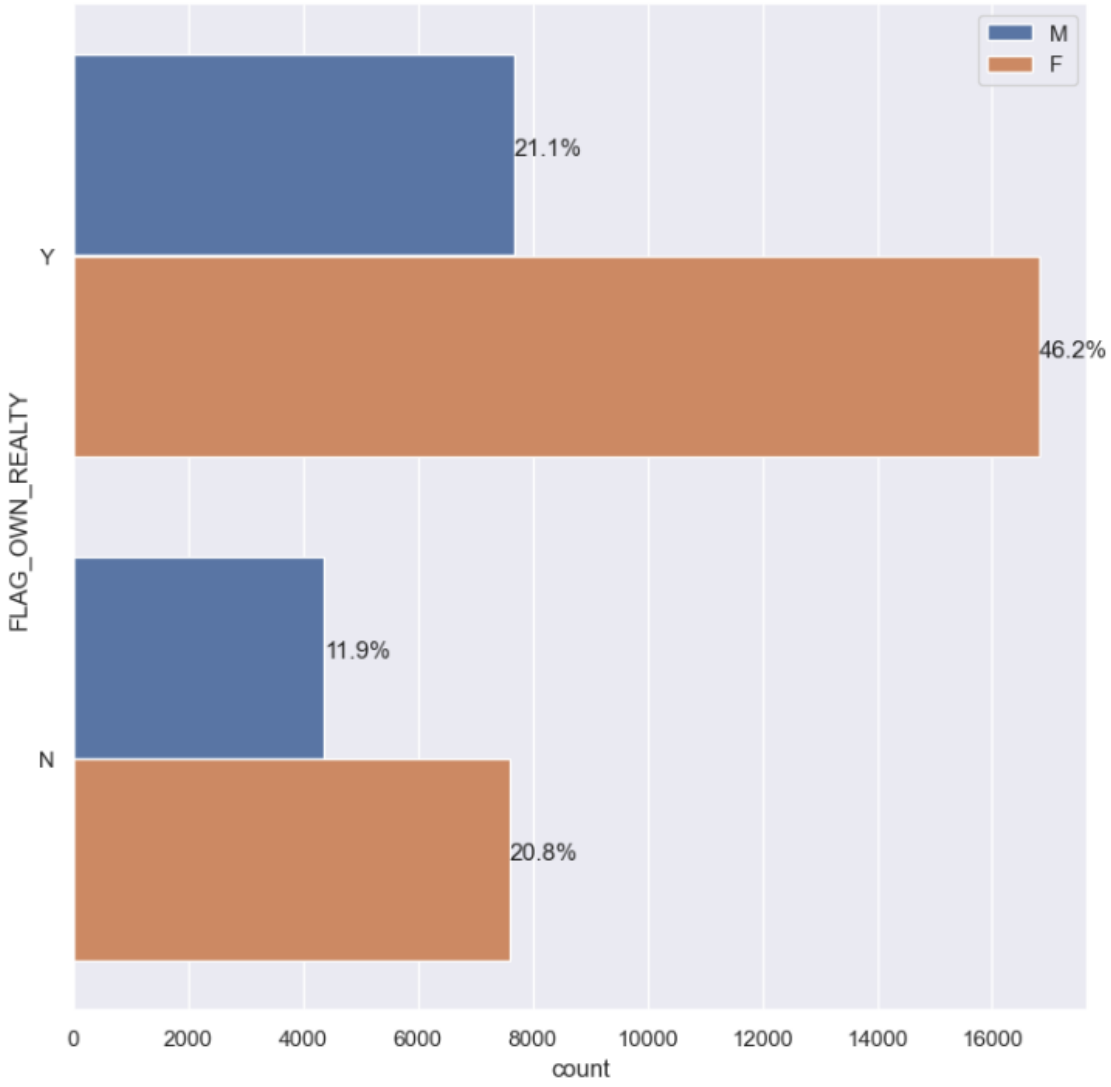

*Hình 4.1: Biểu đồ đánh giá thuộc tính theo giới tính*

- Xét thuộc tính có ô tô thì tỷ lệ có ô tô là 38% với nam giới chiếm 20,8% và nữ là 20,8%. Tỷ lệ không có ô tô là 62% với nam giới là 12,2% và nữ giới là 49,8%.



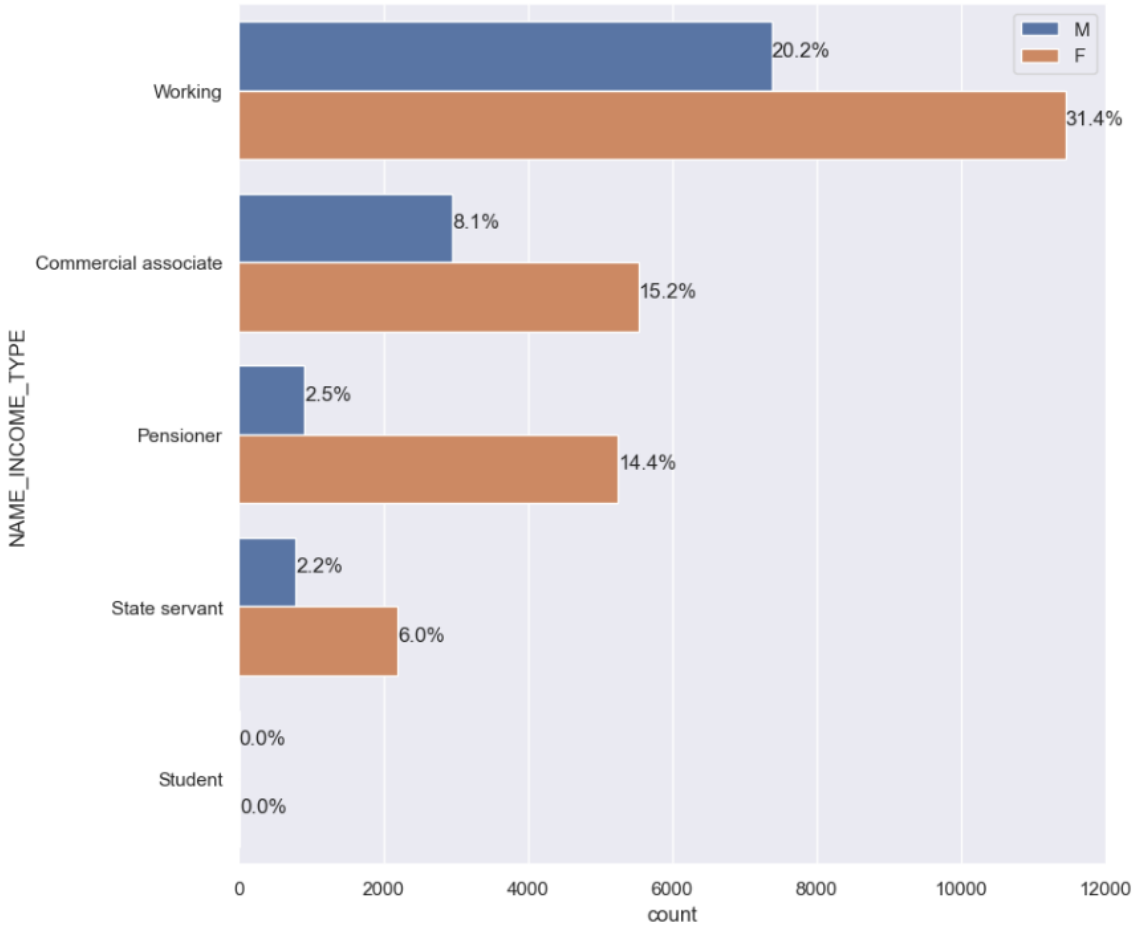

*Hình 4.2: Biểu đồ xét theo thuộc tính ô tô xét tỷ lệ nam, nữ*

- Tỷ lệ sở hữu tài sản bất động sản là 67,3% (trong đó nam là 21,1% và nữ là 46,2%) và không sở hữu là 32,7% (nam là 11,9% và nữ là 20,8%).



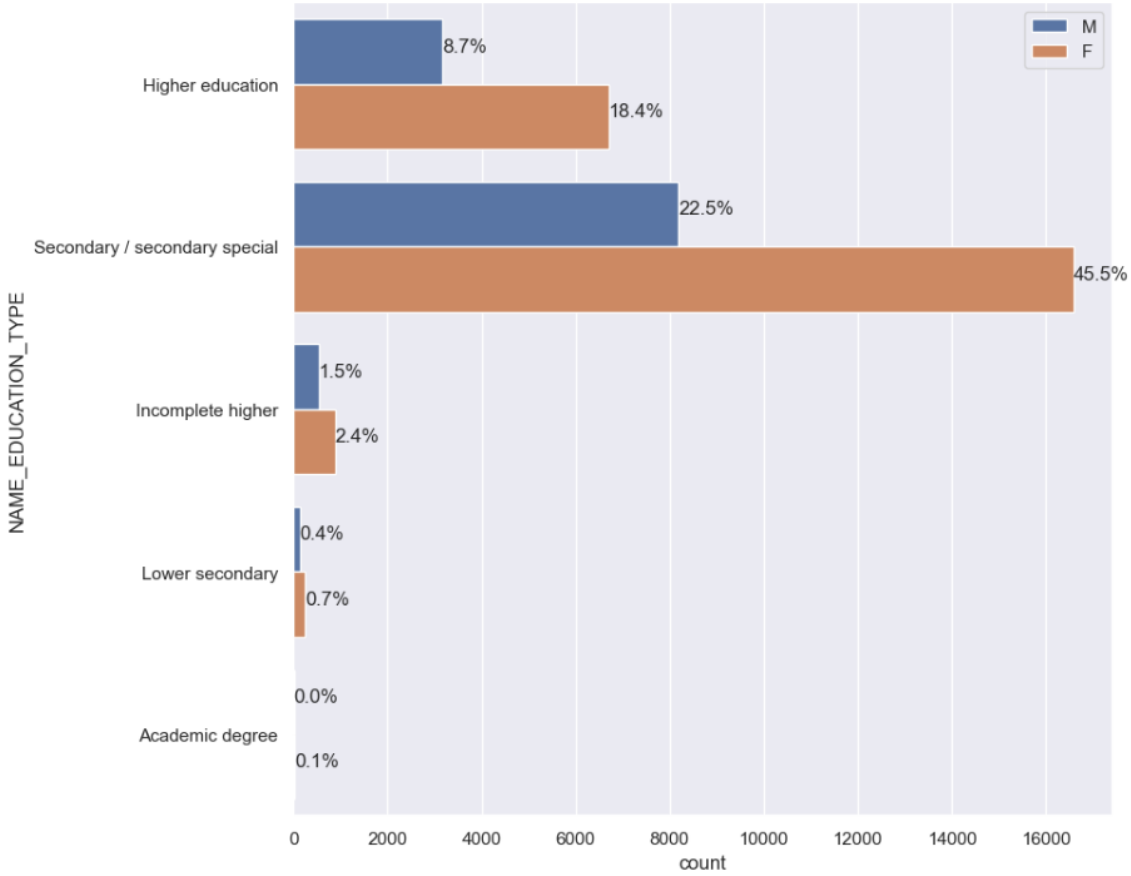

*Hình 4.3: Biểu đồ xét theo thuộc tính bất động sản*

- Tỷ lệ người có thu nhập từ việc làm là cao nhất với 51,6% (nam là 20,2% và nữ là 31,4%)



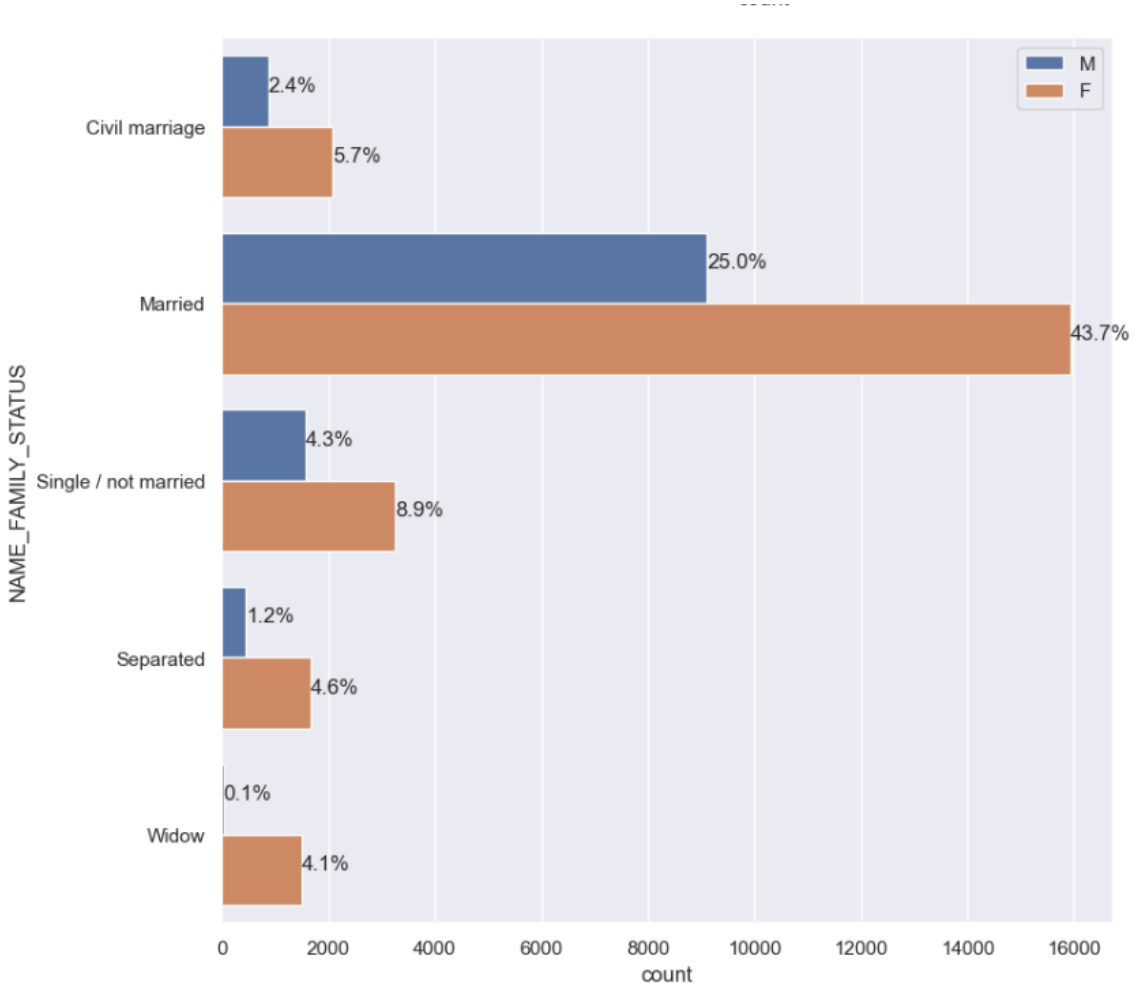

*Hình 4.4: Biểu đồ xét theo thuộc tính thu nhập có việc làm*

- Tỷ lệ người có trình độ học vấn là THPT cao nhất với 68% (nữ giới chiếm nhiều hơn với 45,5% và nam là 22,5%)



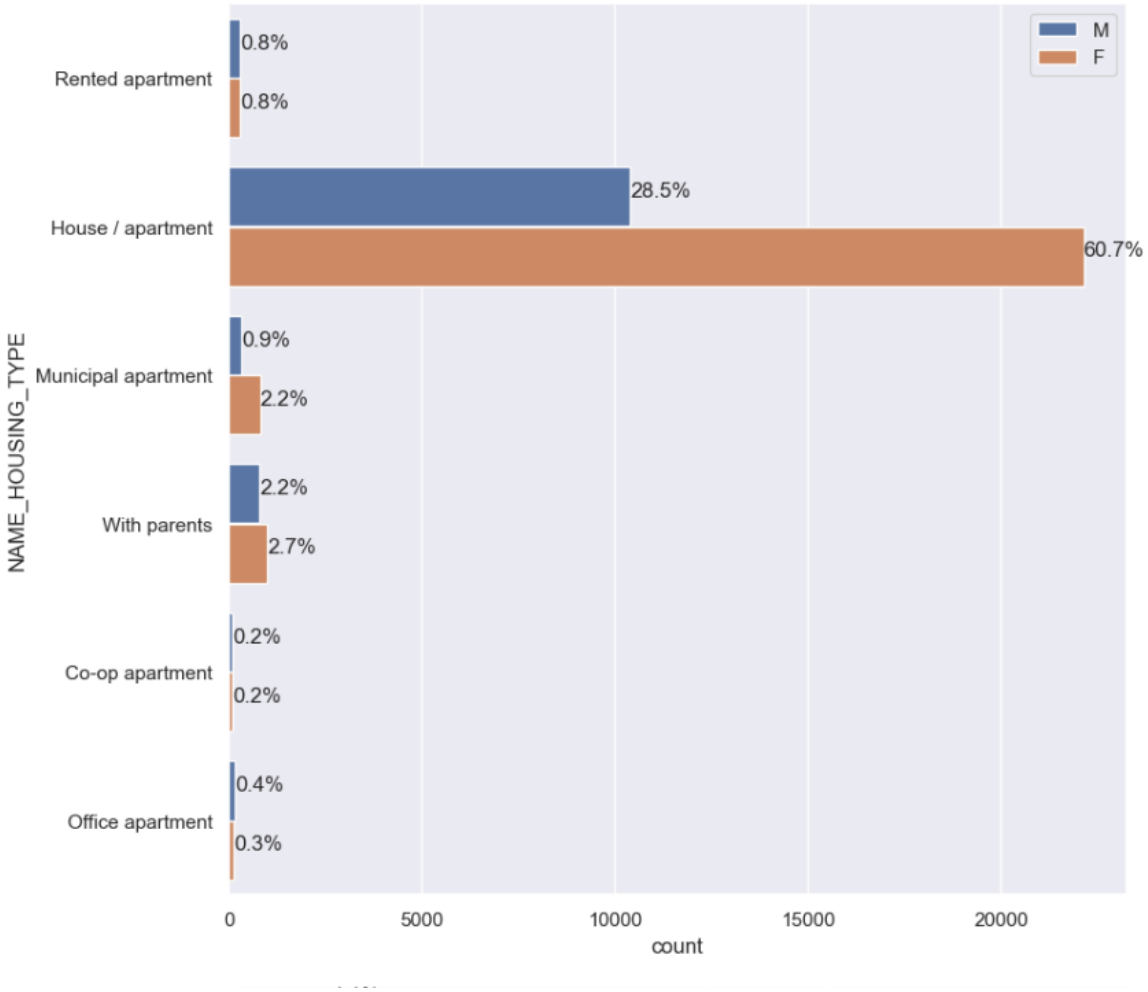

*Hình 4.5: Biểu đồ xét theo thuộc tính trình độ học vấn là THPT*

- Tình trạng hôn nhân: đã kết hôn chiếm tỷ lệ cao nhất với 68,7%




*Hình 4.6: Biểu đồ xét theo thuộc tính tình trạng hôn nhân đã kết hôn*

- Tỷ lệ người sở sống ở nhà riêng hoặc căn hộ cao nhất: 89,2%

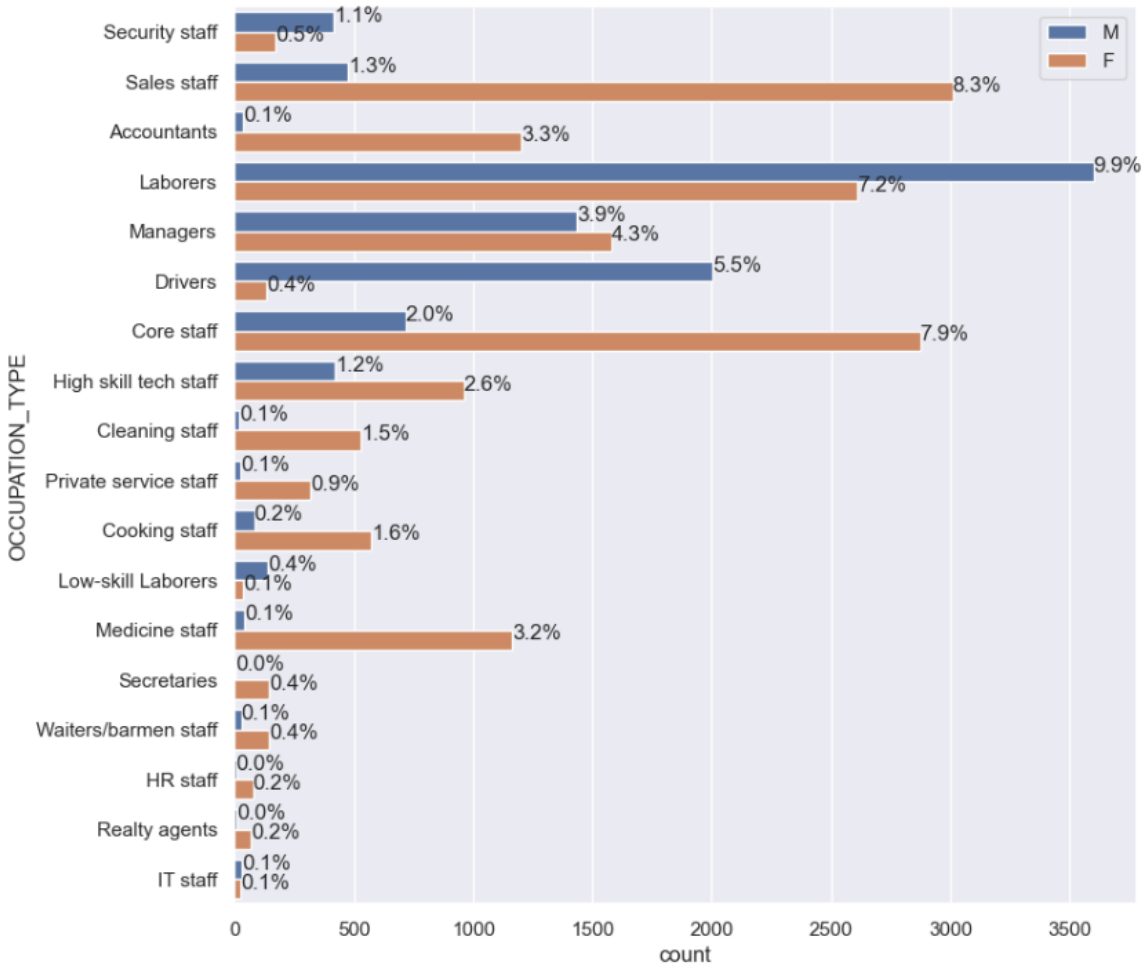



*Hình 4.7: Biểu đồ xét theo tỷ lệ người ở nhà riêng và có căn hộ*

- Top 3 công việc được làm nhiều nhất là Laborers, Core staff và Sales Staff

  Bên cạnh đó, giữ trình độ học vấn và nghề nghiệp có một số tương quan như sau:

  - Trình độ học vấn cao có tỉ lệ làm Managers và Core staff là lớn nhất lần lượt với 4.4% và 4.9%
  - Trình độ học vấn là Secondary/Secondary Special chiếm đa số và thường làm Sales staff(7.1%), Laborers(14.8%)

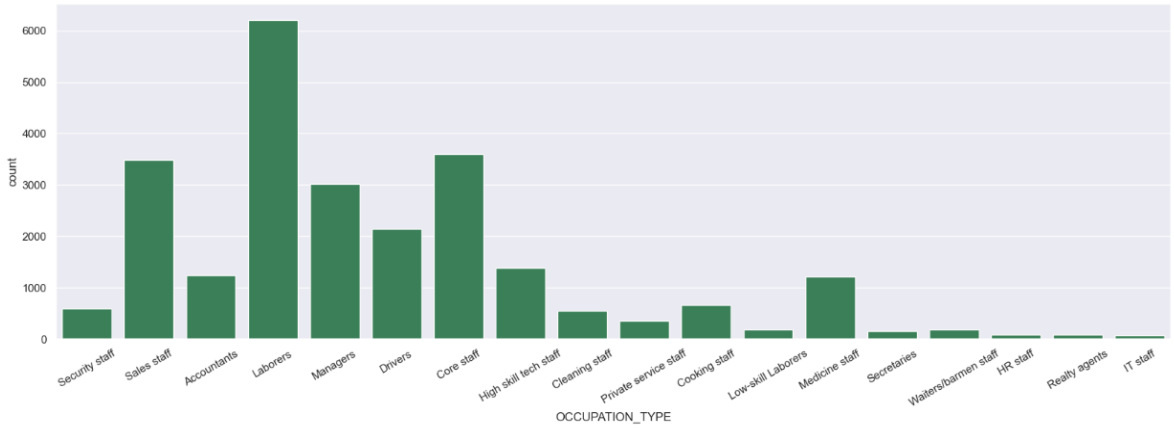



*Hình 4.8: Biểu đồ tương quan giữa trình độ học vấn, người có việc làm, nghề nghiệp*

- Biểu đồ sự phân bố nghề nghiệp

```
#Sự Phân bố nghề nghiệp

plt.figure(figsize=(20,6))
ax = sns.countplot(data=application_detail, x='OCCUPATION_TYPE', color='seagreen')
ax.set_xticklabels(ax.get_xticklabels(),rotation = 30)
plt.show()
```



*Hình: Biểu đồ phân bố nghề nghiệp*

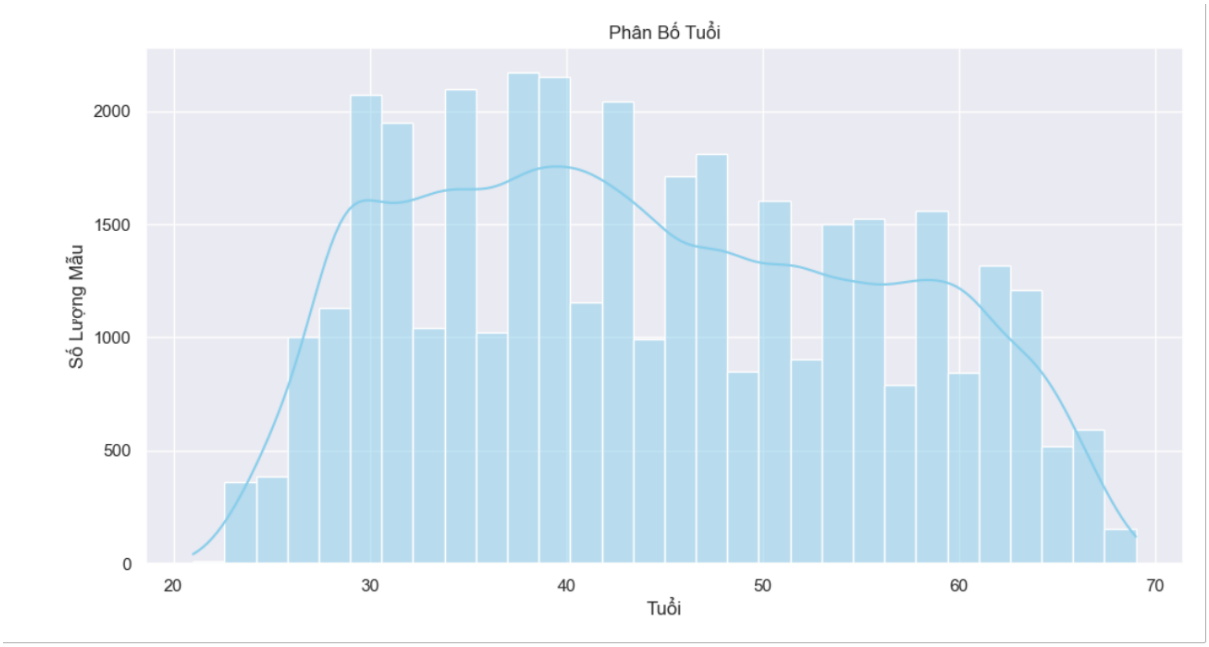
Biểu đồ cho thấy sự phân bố nghề nghiệp của tất cả các ngành. Trong đó, số lượng người làm "Laborers" (làm thuê) là cao nhất và ít nhất là các nghề "HR staff", "IT Staff"

- Biểu đồ phân bố tuổi đăng ký tín dụng

```
[73]: plt.figure(figsize=(12, 6))

      # Vẽ histogram phân bố tuổi
      ax = sns.histplot(application_detail['AGE'], bins=30, kde=True, color='skyblue')

      # Đặt tiêu đề và nhãn trục
      plt.title('Phân Bố Tuổi')
      plt.xlabel('Tuổi')
      plt.ylabel('Số Lượng Mẫu')
```




*Hình: Biểu đồ phân bố tuổi đăng ký tín dụng*

**Nhận xét:**

- Số lượng ứng viên theo tuổi: Số lượng ứng viên theo tuổi có thể được phân thành hai nhóm chính: nhóm có số lượng ứng viên lớn và nhóm có số lượng ứng viên nhỏ. Nhóm có số lượng ứng viên lớn bao gồm các ứng viên trong độ tuổi từ 20 đến 30 tuổi. Nhóm có số lượng ứng viên nhỏ bao gồm các ứng viên dưới 20 tuổi và trên 30 tuổi.

- Xu hướng phân bố tuổi: Xu hướng phân bố tuổi của các ứng viên là có sự tập trung vào các ứng viên trong độ tuổi từ 20 đến 30 tuổi. Điều này cho thấy rằng các ứng viên trong độ tuổi này có xu hướng tích cực tìm kiếm việc làm

## 4.2. Tiền xử lý và biến đổi dữ liệu

- Xử lý giá trị ngoại lai của thuộc tính CNT_CHILDREN

**Thay giá trị ngoại lai bằng giá trị xuất hiện nhiều nhất**

```
[54]: mode = application_detail['CNT_CHILDREN'].mode().iloc[0]
      # Xác định và thay thế giá trị ngoại lai trong cột 'CNT_CHILDREN' bằng giá trị mode
      outliers = application_detail['CNT_CHILDREN'] > 5  # Điều kiện cho giá trị ngoại lai
      application_detail.loc[outliers, 'CNT_CHILDREN'] = mode
```

```
[55]: application_detail['CNT_CHILDREN'].value_counts()
```

```
[55]: 0    25207
      1     7492
      2     3256
      3      419
      4       63
      5       20
      Name: CNT_CHILDREN, dtype: int64
```

- Biến đổi dữ liệu cho thuộc tính AMT_INCOME_TOTAL, DAYS_BIRTH và DAYS_EMPLOYED:

Sử dụng công thức tuổi = ngày sinh/-365.25 và số năm làm việc = ngày làm việc/-365.25. Sau đó, chia các thuộc tính tuổi: dưới 37 bằng 0, từ 37 đến 53 bằng 1 và trên 53 bằng 2. Năm kinh nghiệm: 0 đến 14 năm bằng 0, từ 14 năm đến 29 năm bằng 1 và trên 29 năm bằng 2.

***Biến đổi các thuộc tính thu nhập, tuổi, năm làm việc theo 3 mức***

```
[63]: # Chia giá trị cột Age thành 3 khoảng
      a = pd.cut(credit_approval_data['AGE'], 3)
      # Hiển thị những khoảng duy nhất
      a.unique()
```

```
:[63]: [(20.952, 37.0], (53.0, 69.0], (37.0, 53.0]]
       Categories (3, interval[float64, right]): [(20.952, 37.0] < (37.0, 53.0] < (53.0, 69.0]]
```

```
[64]: # Chia giá trị cột AMT_INCOME_TOTAL thành 3 khoảng
      b = pd.cut(credit_approval_data['AMT_INCOME_TOTAL'], 3)
      b.unique()
```

```
:[64]: [(25452.0, 543000.0], (543000.0, 1059000.0], (1059000.0, 1575000.0]]
       Categories (3, interval[float64, right]): [(25452.0, 543000.0] < (543000.0, 1059000.0] < (1059000.0, 1575000.0]]
```

```
[65]: # Chia giá trị cột YEARS_EMPLOYED thành 3 khoảng
      c = pd.cut(credit_approval_data['YEARS_EMPLOYED'], 3)
      c.unique()
```

```
:[65]: [(-0.044, 14.667], (14.667, 29.333], (29.333, 44.0]]
       Categories (3, interval[float64, right]): [(-0.044, 14.667] < (14.667, 29.333] < (29.333, 44.0]]
```

*Biến đổi các thuộc tính thu nhập, tuổi, năm làm việc theo 3 mức với công thực h=max-min/3*

```
]: for index,row in credit_approval_data.iterrows():
       if(row['AMT_INCOME_TOTAL']<=226740):
           credit_approval_data._set_value(index,'AMT_INCOME_TOTAL',0)
       elif(row['AMT_INCOME_TOTAL']<=450870):
           credit_approval_data._set_value(index,'AMT_INCOME_TOTAL',1)
       else:
           credit_approval_data._set_value(index,'AMT_INCOME_TOTAL',2)
```

```
]: for index,row in credit_approval_data.iterrows():
       if(row['YEARS_EMPLOYED']<=16):
           credit_approval_data._set_value(index,'YEARS_EMPLOYED',0)
       elif(row['YEARS_EMPLOYED']<=32):
           credit_approval_data._set_value(index,'YEARS_EMPLOYED',1)
       else:
           credit_approval_data._set_value(index,'YEARS_EMPLOYED',2)
```

```
]: for index,row in credit_approval_data.iterrows():
       if(row['AGE']<=37):
           credit_approval_data._set_value(index,'AGE',0)
       elif(row['AGE']<=53):
           credit_approval_data._set_value(index,'AGE',1)
       else:
           credit_approval_data._set_value(index,'AGE',2)
```

```
]: credit_approval_data.describe().T
```

- Biến đổi các thuộc tính định lượng thành định tính:
  Đếm các giá trị khác nhau của từng thuộc tính và đánh số lần lượt từ 0 đến hết.

```
: def Cat_to_Num(features):
      for feature in features:
          feature_list = list(np.unique(application_detail[feature]))
          feature_dict = {}
          for i in range(len(feature_list)):
                        feature_dict[feature_list[i]] = i
          application_detail.replace({feature : feature_dict}, inplace=True)
          print(feature, '-->', feature_dict)
```

```
: categorical_features = ['CODE_GENDER', 'FLAG_OWN_CAR', 'FLAG_OWN_REALTY', 'NAME_INCOME_TYPE', 'NAME_FAMILY_STATUS', 'OCCUPATION_T
  Cat_to_Num(categorical_features)
```

```
CODE_GENDER --> {'F': 0, 'M': 1}
FLAG_OWN_CAR --> {'N': 0, 'Y': 1}
FLAG_OWN_REALTY --> {'N': 0, 'Y': 1}
NAME_INCOME_TYPE --> {'Commercial associate': 0, 'Pensioner': 1, 'State servant': 2, 'Student': 3, 'Working': 4}
NAME_FAMILY_STATUS --> {'Civil marriage': 0, 'Married': 1, 'Separated': 2, 'Single / not married': 3, 'Widow': 4}
OCCUPATION_TYPE --> {'Accountants': 0, 'Cleaning staff': 1, 'Cooking staff': 2, 'Core staff': 3, 'Drivers': 4, 'HR staff': 5,
'High skill tech staff': 6, 'IT staff': 7, 'Laborers': 8, 'Low-skill Laborers': 9, 'Managers': 10, 'Medicine staff': 11, 'Priva
te service staff': 12, 'Realty agents': 13, 'Sales staff': 14, 'Secretaries': 15, 'Security staff': 16, 'Waiters/barmen staff':
17}
NAME_HOUSING_TYPE --> {'Co-op apartment': 0, 'House / apartment': 1, 'Municipal apartment': 2, 'Office apartment': 3, 'Rented a
partment': 4, 'With parents': 5}
NAME_EDUCATION_TYPE --> {'Academic degree': 0, 'Higher education': 1, 'Incomplete higher': 2, 'Lower secondary': 3, 'Secondary
/ secondary special': 4}
```

- Kiểm tra dữ liệu

```
application_detail.isnull().sum()
```

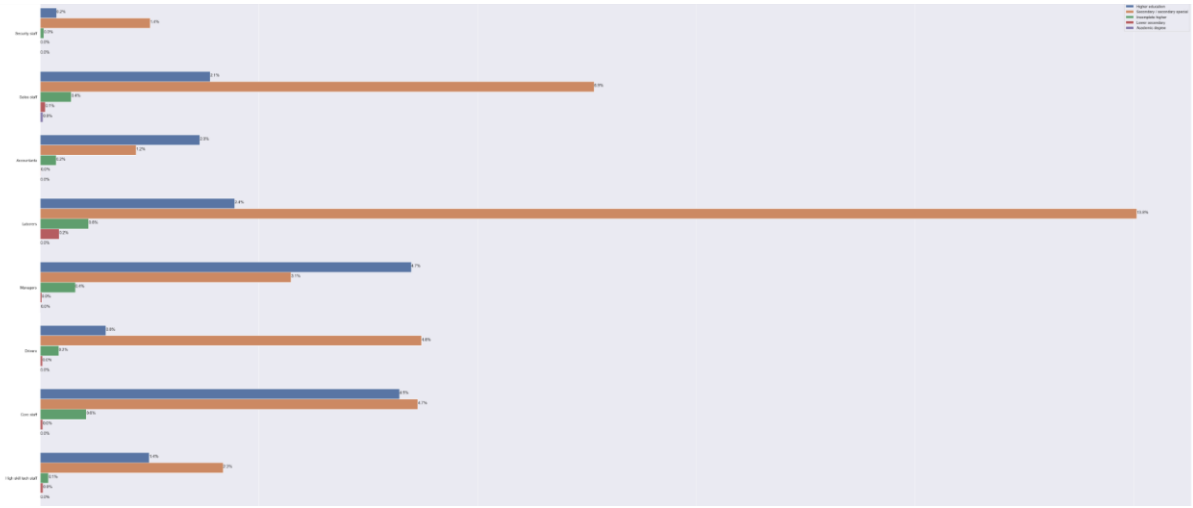
```
ID                          0
CODE_GENDER                 0
FLAG_OWN_CAR                0
FLAG_OWN_REALTY             0
CNT_CHILDREN                0
AMT_INCOME_TOTAL            0
NAME_INCOME_TYPE            0
NAME_EDUCATION_TYPE         0
NAME_FAMILY_STATUS          0
NAME_HOUSING_TYPE           0
DAYS_BIRTH                  0
DAYS_EMPLOYED               0
FLAG_MOBIL                  0
FLAG_WORK_PHONE             0
FLAG_PHONE                  0
FLAG_EMAIL                  0
OCCUPATION_TYPE         11323
CNT_FAM_MEMBERS             0
STATUS                      0
dtype: int64
```

Như vậy,ta thấy có 11323 giá trị thuộc tính nghề nghiệp bị thiếu, không thể xóa các bản ghi này đi. Vì vậy, ta sẽ điền các giá trị thiếu bằng cách tìm ra các nghề nghiệp được làm nhiều nhất theo giới tính và trình độ học vấn rồi điền vào giá trị trống

- Điền các giá trị thiếu cho thuộc tính nghề nghiệp

Với các giá trị bị thiếu của thuộc tính nghề nghiệp sẽ tính tỷ lệ nghề nghiệp theo giới tính và trình độ học vấn. Sau đó chọn ra tỷ lệ cao nhất để gán tương ứng với giới tính và học vấn và điền lại vào dataframe với các giá trị bị thiếu.

Trước hết, ta sẽ vẽ biểu đồ thể hiện mối tương quan giữa nghề nghiệp và trình độ học vấn:



Hình 4.7: Biểu đồ thể hiện mối tương quan giữa nghề nghiệp và trình độ học vấn

Từ biểu đồ trên có thể thấy những người có việc làm chủ yếu là nhóm người có trình độ học vấn là secondary/secondary special. Công việc phổ biến nhất là Leborers với chủ yếu là những người có trình độ học vấn là secondary/secondary special. Nhóm công việc Managers tập trung những người có trình độ học vấn Higher education nhất.

Bước 1: *Tính tỷ lệ nghề nghiệp theo giới tính và trình độ học vấn*

```
In [25]: #Khởi tạo danh sách mới với các trường Gender, education, occupation, Percentage
def occupation_by_education_by_gender(occupations, educations, genders, length):
    Gender = []
    Education = []
    Occupation = []
    Percentage = []
 # Lặp qua tất cả các giá trị của giới tính, trình độ học vấn và nghề nghiệp
    for gender in genders:
        for education in educations:
            for occupation in occupations:
                # Thêm giá trị giới tính, trình độ học vấn và nghề nghiệp vào danh sách tương ứng
                Gender.append(gender)
                Education.append(education)
                Occupation.append(occupation)
                # Lọc dữ liệu theo các tiêu chí
                temp_df = application_detail[
                    (application_detail['OCCUPATION_TYPE'] == occupation) &
                    (application_detail['CODE_GENDER'] == gender) &
                    (application_detail['NAME_EDUCATION_TYPE'] == education)
                ]
                #tính số lượng hàng và tỷ lệ phổ biến bằng cách chia lại cho 'length'
                length_filter = len(temp_df)
                Percentage.append(length_filter / length)
    # Tạo DataFrame từ dữ liệu đã thu thập
    percent_dict = {
        'Gender': Gender,
        'Education': Education,
        'Occupation': Occupation,
        'Popularity Percentage': Percentage
    }

    df = pd.DataFrame(percent_dict)
    # Lưu DataFrame vào một tệp Excel
    df.to_excel('Occupation_popularity_index.xlsx', index=False)
```

Bước 2*: Thay thế các giá trị nghề nghiệp bị thiếu với tỷ lệ nghề nghiệp cao nhất*

```
def create_tag_dict(data):
    # Lấy danh sách các tag duy nhất
    tag_list = list(data['Tag'].unique())

    # Khởi tạo danh sách nghề nghiệp phổ biến
    popular_occupation = []

    # Nhóm dữ liệu theo cột 'Tag'
    data_grouped = data.groupby('Tag')

    # Duyệt qua từng tag
    for tag in tag_list:
        # Lấy dữ liệu của nhóm hiện tại
        grouped_data = data_grouped.get_group(tag)

        # Tìm nghề nghiệp phổ biến nhất trong nhóm
        popular_occupation.append(
            grouped_data.loc[grouped_data['Popularity Percentage'].idxmax()]['Occupation']
        )

    # Tạo từ điển Tag_dict
    Tag_dict = {
        'Tag': tag_list,
        'Occupation': popular_occupation
    }

    return Tag_dict
```

Bước 3*: Thực hiện điền nghề nghiệp vào các giá trị còn thiếu*

```
application_detail['Tag'] = application_detail['CODE_GENDER']+ '_' + application_detail['NAME_EDUCATION_TYPE']
```

```
#thay thế các giá trị còn thiếu
def Replace_nan_occupation(df):
    for key in tag_dict:
        if df['Tag'] == key and pd.isnull(df['OCCUPATION_TYPE']):
            df['OCCUPATION_TYPE'] = tag_dict[key]
    return df
```

```
In [61]: credit_approval_data.head(10)
```

Out[61]:

| | CODE_GENDER | FLAG_OWN_CAR | FLAG_OWN_REALTY | CNT_CHILDREN | AMT_INCOME_TOTAL | NAME_INCOME_TYPE | NAME_EDUCATION_TYPE | NAME_FAM |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 | 1.0 | 4 | 1 | |
| 1 | 1 | 1 | 1 | 0 | 1.0 | 4 | 1 | |
| 2 | 1 | 1 | 1 | 0 | 0.0 | 4 | 4 | |
| 3 | 0 | 0 | 1 | 0 | 1.0 | 0 | 4 | |
| 4 | 0 | 0 | 1 | 0 | 1.0 | 0 | 4 | |
| 5 | 0 | 0 | 1 | 0 | 1.0 | 0 | 4 | |
| 6 | 0 | 0 | 1 | 0 | 1.0 | 0 | 4 | |
| 7 | 0 | 0 | 1 | 0 | 1.0 | 1 | 1 | |
| 8 | 0 | 0 | 1 | 0 | 1.0 | 1 | 1 | |
| 9 | 0 | 0 | 1 | 0 | 1.0 | 1 | 1 | |

*Dữ liệu sau khi tiền xử lý và biến đổi*

## 4.3. Xây dựng mô hình thuật toán phân lớp

### *4.3.1. Phân chia tập dữ liệu*

Từ tập dữ liệu ban đầu, chia ra với 30% để test và 70% để training. Trong đó thuộc tính phụ thuộc là STATUS và các thuộc tính còn lại là thuộc tính độc lập.

```
X = credit_approval_data.drop('STATUS', axis=1)
y = credit_approval_data['STATUS']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

*Hình : Phân chia tập dữ liệu*

### *4.3.2. Huấn luyện dữ liệu với thuật toán Decision Tree Classifier – CART*

- Khai báo, thực thi và kiểm thử mô hình:

```
#khai báo mô, thực thi, kiểm thử mô hình
my_tree = DecisionTreeClassifier()
my_tree.fit(X_train, y_train)
y_predict = my_tree.predict(X_test)
print("Giá trị mô hình dự đoán được: \n", y_predict)
```

=> Giá trị mô hình dự đoán được:

```
Giá trị mô hình dự đoán được:
 [0 1 1 ... 1 1 1]
```

- Độ chính xác của mô hình: Với độ chính xác là 97,605%, mô hình dự đoán chính xác 97,605% các trường hợp. Điều này cho thấy mô hình có khả năng dự đoán các trường hợp không cho mở thẻ tín dụng và cho mở thẻ tín dụng khá tốt.

```
#đưa ra độ chính xác
print('Accuracy score is: {:.5}'.format(accuracy_score(y_test, y_predict)))
```

```
Accuracy score is: 0.97605
```

- Báo cáo phân loại chi tiết hơn với độ precision, recall, f1-score:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.12 | 0.06 | 0.08 | 190 |
| 1 | 0.98 | 0.99 | 0.99 | 10748 |
| accuracy | | | 0.98 | 10938 |
| macro avg | 0.55 | 0.53 | 0.53 | 10938 |
| weighted avg | 0.97 | 0.98 | 0.97 | 10938 |

***Nhận xét***:

- Precision = 0.98: Trong tất cả những người xét mở thẻ tín dụng thì có 98% lượng người có thể mở thẻ

- Recall = 0.99: Trong số tất cả những người thực sự được mở thẻ, mô hình dự đoán kết quả này một cách chính xác lên đến 99% cho những người đó

- f1 score = 0.99: Vì giá trị này gần với 1, nó cho chúng ta biết rằng mô hình thực hiện công việc dự đoán chuẩn liệu người xét duyệt có được mở thẻ tín dụng hay không.

- support: Các giá trị này chỉ đơn giản cho chúng ta biết có bao nhiêu người xét duyệt thuộc về mỗi lớp trong tập dữ liệu thử nghiệm. Chúng ta có thể thấy rằng trong số những người xét duyệt trong bộ dữ liệu thử nghiệm, 190 người không được mở thẻ và 10748 người được mở thẻ.

***Nhìn chung***, mô hình có thể phân loại tốt lớp 1 nhưng không phân loại tốt lớp 0. Với lớp 0, precision là 0.12 và recall là 0.06, cho thấy mô hình có độ chính xác thấp trong việc xác định và truy xuất các mẫu liên quan của lớp này. Điều này có thể do lớp 0 là lớp thiểu số hoặc có nhiều nhiễu trong dữ liệu. Với lớp 1, precision là 0.98 và recall là 0.99, cho thấy mô hình có độ chính xác cao và hiệu quả trong việc xác định và truy xuất các mẫu của lớp này. Điều này có thể do lớp 1 là lớp đa số hoặc có ít nhiễu trong dữ liệu.

### *4.3.3. Huấn luyện dữ liệu với thuật toán Decision Tree Classifier - ID3*

Xét trường hợp random_state = 100, tức là số dòng dữ liệu được sử dụng để khởi tạo thuật toán là 100. Độ sâu của cây là max_depth = 3, và mỗi nút lá sẽ chứa ít nhất 5 mẫu (min_samples_leaf = 5).

- Khai báo, thực thi và kiểm thử mô hình:

```
# khai báo mô hình cây quyết định
model = tree.DecisionTreeClassifier(criterion = "entropy",
                                    random_state = 100,
                                    max_depth = 3,
                                    min_samples_leaf = 5)
#thực thi mô hình
model = model.fit(X_train,y_train)
# kiểm thử mô hình
y_pred = model.predict(X_test)
print("Giá trị nhãn mô hình dự đoán được:\n ", y_pred)
```

```
Giá trị nhãn mô hình dự đoán được:
  [1 1 1 ... 1 1 1]
```

- Độ chính xác của mô hình: Với độ chính xác là 98.95%, mô hình dự đoán chính xác 98.95% các trường hợp. Điều này cho thấy mô hình có khả năng dự đoán các trường hợp không cho mở thẻ tín dụng và cho mở thẻ tín dụng khá tốt.

```
# độ chính xác của mô hình cây
print("Accuracy: ", accuracy_score(y_test,y_pred)*100)
```

```
Accuracy:  98.94861949168038
```

- Báo cáo phân loại chi tiết hơn với ma trận nhầm lẫn 2 chiều:

```
# xác định ma trận nhầm lẫn
print("Confusion Matrix:\n", confusion_matrix(y_test,y_pred))
print("True Positives: ",confusion_matrix(y_test,y_pred)[0][0])
print("False Positives: ",confusion_matrix(y_test,y_pred)[0][1])
print("False Negatives: ",confusion_matrix(y_test,y_pred)[1][0])
print("True Negatives: ",confusion_matrix(y_test,y_pred)[1][1])
```

```
Confusion Matrix:
 [[    0   115]
 [    0 10823]]
True Positives:  0
False Positives:  115
False Negatives:  0
True Negatives:  10823
```

Nhận xét ma trận nhầm lẫn:

- True Positives (TP) = 0: Số người không được mở thẻ tín dụng mà hệ thống dự đoán là không được mở thẻ tín dụng là 0.Điều này có nghĩa là hệ thống dự đoán chính xác 100% những người không được mở thẻ tín dụng.
- False Positives (FP) = 115: Số người được mở thẻ tín dụng mà hệ thống dự đoán là không được mở thẻ tín dụng là 115. Điều này có nghĩa là hệ thống dự đoán sai 115 người được mở thẻ tín dụng.
- False Negatives (FN) = 0: Số người không được mở thẻ tín dụng mà hệ thống dự đoán là được mở thẻ tín dụng là 0. Điều này có nghĩa là hệ thống dự đoán chính xác 100% những người không được mở thẻ tín dụng.
- True Negatives (TN) = 10823: Số người được mở thẻ tín dụng mà hệ thống dự đoán là được mở thẻ tín dụng là 10823. Điều này có nghĩa là hệ thống dự đoán chính xác 100% những người được mở thẻ tín dụng.

=> Vậy kết quả ma trận này cho thấy hệ thống dự đoán khá tốt cho cả hai trường hợp: không được mở thẻ tín dụng và được mở thẻ tín dụng. Cụ thể, hệ thống dự đoán chính xác 100% những người không được mở thẻ tín dụng và 99.8% những người được mở thẻ tín dụng.

- Độ đo precision và recall:

```
from sklearn.metrics import precision_score, recall_score
# Giá trị precision
p = precision_score(y_test, y_pred)
#Giá trị recall
r = recall_score(y_test,y_pred)
print("giá trị độ đo precision:",p)
print("giá trị độ đo recall:",r)
```

```
giá trị độ đo precision: 0.9894861949168038
giá trị độ đo recall: 1.0
```

- Với độ đo precision là 0.9894861949168038, mô hình dự đoán chính xác 98.94% các trường hợp không cho mở thẻ tín dụng. Điều này cho thấy mô hình có khả năng phân biệt giữa các trường hợp không cho mở thẻ tín dụng và cho mở thẻ tín dụng khá tốt.
- Với độ đo recall là 1.0, mô hình dự đoán chính xác tất cả các trường hợp không cho mở thẻ tín dụng. Điều này cho thấy mô hình không bỏ sót bất kỳ trường hợp không cho mở thẻ tín dụng nào.

- Dự báo với khách hàng mới: Thực hiện dự báo với khách hàng có:

```
# sử dụng mô hình dự đoán khả năng mở thẻ tin dụng
#giới tính 1, ô tô 0, tài sản 0, con 0, thu nhập hàng năm 0, loại thu nhập 4, trình độ 1, hôn nhân 3, loại nhà 5
#điện thoại di động 1, đthoại cơ quan 1, đthoại 1, email 1, nghề nghiệp 8, quy mô gia đình 40, ngày sinh 0, ngày làm việc 0
X = [[1,0,0,0,0,4,1,3,5,1,1,1,1,8,40,0,0]]
y = model.predict(X)
if y==1:
    print("khách hàng có khả năng mở tín dụng")
else:
    print("khách hàng không có khả năng mở tín dụng")
```

```
khách hàng có khả năng mở tín dụng
```

- Đánh giá mô hình cây ID3:

**Nhìn chung**, kết quả này khá tốt. Mô hình có khả năng dự đoán chính xác cao cả các trường hợp không cho mở thẻ tín dụng và không bỏ sót bất kỳ trường hợp nào.

### *4.3.4. Huấn luyện dữ liệu với thuật toán KNN Classifier*

Xét trường hợp K = 2, tức là với mỗi điểm test data, ta chỉ xét 2 điểm training data gần nhất và lấy label của điểm đó để dự đoán cho điểm test này. Tính khoảng cách bằng công thức minkowski, p = 2 là khoảng cách Euclidean được sử dụng và ra kết quả như sau:

- Khai báo, thực thi và kiểm thử mô hình:

```
knn = KNeighborsClassifier(n_neighbors = 2, metric = 'minkowski' , p = 2)
#Thực thi mô hình
knn.fit(X_train,y_train)
#Kiểm thử mô hình
y_pred = knn.predict(X_test)
print(y_pred)
```

```
[1 1 1 ... 1 1 1]
Wall time: 32.5 s
```

- Độ chính xác của mô hình: Có thể thấy rằng, độ chính xác của mô hình là 97.88% với k = 2, chứng tỏ mô hình có khả năng dự đoán khá chính xác với mẫu mới

```
#Hiển thị độ chính xác
print("Accuracy of 2NN: %.2f %%" %(100*accuracy_score(y_test, y_pred)))
```

```
Accuracy of 2NN: 97.88 %
```

- Báo cáo phân loại chi tiết hơn với độ precision, recall, f1-score:

```
#in ra báo cáo phân loại chi tiết hơn, bao gồm độ chính xác (precision), độ nhớ (recall), F1-score, v.v. cho từng lớp.
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.10      0.13      0.11       115
           1       0.99      0.99      0.99     10823

    accuracy                           0.98     10938
   macro avg       0.55      0.56      0.55     10938
weighted avg       0.98      0.98      0.98     10938
```

***Nhận xét***:

- Precision = 0.99: Trong tất cả những người xét mở thẻ tín dụng thì có 99% lượng người có thể mở thẻ
- Recall = 0.99: Trong số tất cả những người thực sự được mở thẻ, mô hình dự đoán kết quả này một cách chính xác lên đến 99% cho những người đó
- f1 score = 0.99: Vì giá trị này gần với 1, nó cho chúng ta biết rằng mô hình thực hiện công việc dự đoán chuẩn liệu người xét duyệt có được mở thẻ tín dụng hay không.
- support: Các giá trị này chỉ đơn giản cho chúng ta biết có bao nhiêu người xét duyệt thuộc về mỗi lớp trong tập dữ liệu thử nghiệm. Chúng ta có thể thấy rằng trong số những người xét duyệt trong bộ dữ liệu thử nghiệm, 115 người không được mở thẻ và 10823 người được mở thẻ.

***Nhìn chung***, mô hình có thể phân loại tốt lớp 1 nhưng không phân loại tốt lớp 0. Với lớp 0, precision là 0.1 và recall là 0.13, cho thấy mô hình có độ chính xác thấp trong việc xác định và truy xuất các mẫu liên quan của lớp này. Điều này có thể do lớp 0 là lớp thiểu số hoặc có nhiều nhiễu trong dữ liệu. Với lớp 1, precision là 0.99 và recall là 0.99, cho thấy mô hình có độ chính xác cao và hiệu quả trong việc xác định và truy xuất các mẫu của lớp này. Điều này có thể do lớp 1 là lớp đa số hoặc có ít nhiễu trong dữ liệu.

- Cải thiện độ chính xác: Một cách có thể làm tăng độ chính xác là tăng số lượng điểm lân cận lên, ví dụ 5 điểm, và xem xem trong 5 điểm gần nhất, class nào chiếm đa số thì dự đoán kết quả là class đó và sau khi thay đổi, độ chính xác đã tăng đến 98.94%

```
#Cải thiện độ chính xác bằng cách tăng điểm lân cận
knn1 = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski' ,p = 2)
knn1.fit(X_train, y_train)
y_pred = knn1.predict(X_test)

print("Accuracy of 5NN: %.2f %%" %(100*accuracy_score(y_test, y_pred)))
```

```
Accuracy of 5NN: 98.94 %
```

- Dự báo với khách hàng mới: Thực hiện dự báo với khách hàng có:

  CODE_GENDER = 1(*Nam*)

  FLAG_OWN_CAR = 0 (*Không có ô tô*),

  FLAG_OWN_REALTY = 1 (*Có tài sản*),

  CNT_CHILDREN = 0 (*Không có con*)

  AMT_INCOME_TOTAL = 1 (*Thu nhập từ từ 2267401 đến 4508700)*

  NAME_INCOME_TYPE = 4 (*Loại thu nhập Working*)

NAME_EDUCATION_TYPE = 2 (*Trình độ học vấn Incomplete higher*)
NAME_FAMILY_STATUS = 3 (*Tình trạng hôn nhân Single / not married*)
NAME_HOUSING_TYPE = 4 (*Loại nhà Rented apartment*)
FLAG_MOBIL = 1 (*Có di động)*
FLAG_WORK_PHONE = 1 (*Có điện thoại công ty*)
FLAG_PHONE = 1 (*Có điện thoại*)
FLAG_EMAIL = 1 (*Có email*)
OCCUPATION_TYPE = 5 (*Nghề nghiệp: HR staff*)
CNT_FAM_MEMBERS = 1 (*Quy mô gia đình 1*)
AGE = 0 (*Dưới 37 tuổi*)
YEARS_EMPLOYED = 0 (*Kinh nghiệm làm việc 0 đến 16 năm*)

```
x = [[1, 0, 1, 0, 1, 4, 2, 3, 4, 1, 1, 1, 1, 5, 1, 0, 0]]
y = knn1.predict(x)
if y==1:
    print("Khách hàng có khả năng mở thẻ tín dụng")
else:
    print("Khách hàng không có khả năng mở thẻ tín dụng")
```

```
Khách hàng có khả năng mở thẻ tín dụng
```

### *4.3.5. Huấn luyện dữ liệu với thuật toán Gaussian Naive Bayes*

Khai báo mô hình và huấn luyện mô hình

```
# Khởi tạo mô hình Gaussian Naive Bayes
modelgb = GaussianNB()
# Huấn Luyện mô hình
modelgb.fit(X_train, y_train)
```

Kiểm thử mô hình

```
#Kiểm thử mô hình
y_pred = modelgb.predict(X_test)
```

Đánh giá mô hình

```
# Đánh giá mô hình
accuracy = metrics.accuracy_score(y_test, y_pred)
print("Độ chính xác:", accuracy)
```

```
Độ chính xác: 0.979246663009691
```

=> Độ chính xác mô hình là 97,94%

Báo cáo phân loại chi tiết hơn với độ precision, recall, f1-score:

```
y_predict = modelgb.predict(X_test)
print (classification_report(y_test, y_predict))
#precision (độ chính xác), recall (độ nhớ), và f1-score (kết hợp của precision và recall)
# Support là số lượng mẫu thực tế thuộc mỗi lớp trong tập kiểm tra.
```

```
              precision    recall  f1-score   support

           0       0.01      0.01      0.01       115
           1       0.99      0.99      0.99     10823

    accuracy                           0.98     10938
   macro avg       0.50      0.50      0.50     10938
weighted avg       0.98      0.98      0.98     10938
```

***Nhận xét***:

- Precision = 0.99: Trong tất cả những người xét mở thẻ tín dụng thì có 99% lượng người có thể mở thẻ
- Recall = 0.99: Trong số tất cả những người thực sự được mở thẻ, mô hình dự đoán kết quả này một cách chính xác lên đến 99% cho những người đó
- f1 score = 0.99: Vì giá trị này gần với 1, nó cho chúng ta biết rằng mô hình thực hiện công việc dự đoán chuẩn liệu người xét duyệt có được mở thẻ tín dụng hay không.
- support: Các giá trị này chỉ đơn giản cho chúng ta biết có bao nhiêu người xét duyệt thuộc về mỗi lớp trong tập dữ liệu thử nghiệm. Chúng ta có thể thấy rằng trong số những người xét duyệt trong bộ dữ liệu thử nghiệm, 115 người không được mở thẻ và 10823 người được mở thẻ.
- Macro Avg: Đơn giản là lấy trung bình của các chỉ số (precision, recall, f1-score) cho cả hai lớp mà không quan tâm đến kích thước của từng lớp.
- Weighted Avg: Lấy trung bình có trọng số dựa trên kích thước của từng lớp.

***Nhìn chung***, mô hình có thể phân loại tốt lớp 1 nhưng không phân loại tốt lớp 0. Với lớp 0, precision là 0.1 và recall là 0.01, cho thấy mô hình có độ chính xác thấp trong việc xác định và truy xuất các mẫu liên quan của lớp này. Điều này có thể do lớp 0 là lớp thiểu số hoặc có nhiều nhiễu trong dữ liệu. Với lớp 1, precision là 0.99 và recall là 0.99, cho thấy mô hình có độ chính xác cao và hiệu quả trong việc xác định và truy xuất các mẫu của lớp này. Điều này có thể do lớp 1 là lớp đa số hoặc có ít nhiễu trong dữ liệu.

### 4.4. Xây dựng mô hình luật kết hợp với thuật toán Apriori

- Cài đặt thư viện apyori

```
!pip install apyori
from apyori import apriori
```

```
Requirement already satisfied: apyori in d:\python\lib\site-packages (1.1.2)
```

- Tiền xử lý dữ liệu:

```
# Tiền xử lý dữ liệu
df2['CODE_GENDER'].replace('M','Male', inplace= True)
df2['CODE_GENDER'].replace( 'F', 'Female', inplace= True)

df2['FLAG_OWN_CAR'].replace('Y','Car', inplace= True)
df2['FLAG_OWN_CAR'].replace('N','No Car', inplace= True)


df2['STATUS'].replace(1,'Yes credit card', inplace= True)
df2['STATUS'].replace(0,'No credit card', inplace= True)
```

- Xóa các thuộc tính không ảnh hưởng tới luật:

```
column = ['ID','FLAG_OWN_REALTY','CNT_CHILDREN','FLAG_MOBIL','FLAG_WORK_PHONE','FLAG_PHONE','FLAG_EMAIL','CNT_FAM_MEMBERS','
df2 = df2.drop(columns =column )
```

- Tạo mảng record chứa 300 danh sách con, mỗi danh sách con chứa các giá trị chuỗi từ cột 1 đến cột 9 của một hàng trong df2

```
records = []
for i in range(0, 300):
    records.append([str(df2.values[i, j]) for j in range(1, 10)])
```

- In ra 5 danh sách đầu tiên

```
# Hiển thị 5 bản ghi đầu tiên
print(records[0 : 5])
```

```
[['Car', '427500.0', 'Working', 'Higher education', 'Civil marriage', 'Rented apartment', 'IT staff', 'Yes credit card', '3
3.0'], ['Car', '427500.0', 'Working', 'Higher education', 'Civil marriage', 'Rented apartment', 'IT staff', 'Yes credit car
d', '33.0'], ['Car', '112500.0', 'Working', 'Secondary / secondary special', 'Married', 'House / apartment', 'Security staf
f', 'Yes credit card', '59.0'], ['No Car', '270000.0', 'Commercial associate', 'Secondary / secondary special', 'Single / no
t married', 'House / apartment', 'Sales staff', 'Yes credit card', '53.0'], ['No Car', '270000.0', 'Commercial associate',
'Secondary / secondary special', 'Single / not married', 'House / apartment', 'Sales staff', 'Yes credit card', '53.0']]
```

- Chạy mô hình luật kết hợp và đổi các luật về dạng danh sách

```
association_rules = apriori(records, min_support = 0.09, min_confidence = 0.2, min_lift = 3, min_length = 2)
```

```
# Đổi các luật về dạng danh sách
association_results = list(association_rules)
```

+ min_support = 0.09: Tỷ lệ hỗ trợ tối thiểu được đặt thành 0,09. Nghĩa là luật kết hợp chỉ được coi là có ý nghĩa nếu nó xuất hiện trong ít nhất 9% các bản ghi dữ liệu.
+ min_confidence = 0.2: Tỷ lệ tin cậy tối thiểu được đặt thành 0,2. Điều này có nghĩa là luật kết hợp chỉ được coi là có ý nghĩa nếu xác suất một điểm dữ liệu có thuộc tính đích nếu điểm dữ liệu đó có thuộc tính đầu vào là ít nhất 20%.
+ min_lift = 3: Tỷ lệ tăng tối thiểu được đặt thành 3. Điều này có nghĩa là một luật kết hợp chỉ được coi là có ý nghĩa nếu tỷ lệ tin cậy của luật kết hợp cao hơn tỷ lệ hỗ trợ của thuộc tính đích ít nhất 3 lần.
+ min_length = 2: Độ dài tối thiểu của luật kết hợp được đặt thành 2. Điều này có nghĩa là chỉ các luật kết hợp có ít nhất 2 thuộc tính sẽ được khai thác.

- Tổng số luật được tạo ra là 470:

```
print(len(association_results))
```

```
470
```

- Hiển thị luật, độ hỗ trợ, độ tin cậy và lift cho từng luật rõ ràng hơn:

```
for item in association_results:
    #Dòng đầu
    pair = item[0]
    items = [x for x in pair]
    print('Rule: ' + items[0] + " -> " + items[1])

    #Dòng thứ 2 chứa độ hỗ trợ
    print('Support: ' + str(item[1]))

    #Dòng thứ 3
    print('Confidence: ' + str(item[2][0][2]))
    print('Lift: ' + str(item[2][0][3]))
    print('=====================================')
```

- Kết quả 1:

```
=====================================
Rule: IT staff -> Yes credit card
Support: 0.09333333333333334
Confidence: 0.4516129032258065
Lift: 4.838709677419355
=====================================
```

- Đánh giá luật:

Luật này cho thấy những người có nghề nghiệp là IT staff có khả năng mở thẻ tín dụng. Giá trị hỗ trợ cho quy tắc này là 0.09. Con số này được tính bằng cách lấy số lượng người có nghề nghiệp là IT staff chia cho tổng số người. Mức độ tin cậy cho quy tắc là 0.45 cho thấy rằng trong số tất cả những người có nghề nghiệp là IT staff, 45% có thể mở thẻ tín dụng. Cuối cùng, mức nâng 4,83 cho chúng ta biết rằng những người mở thẻ làm nghề IT staff cao hơn 4,83 lần so với khả năng những người chỉ làm nghề IT Staff.

- Kết quả 2:

```
Rule: Working -> Yes credit card
Support: 0.053
Confidence: 0.6235294117647058
Lift: 3.464052287581699
```

- Đánh giá luật:

Luật này cho thấy những người có loại thu nhập là Working có khả năng mở thẻ tín dụng. Giá trị hỗ trợ cho quy tắc này là 0.053. Con số này được tính bằng cách lấy số lượng người loại thu nhập là Working chia cho tổng số người. Mức độ tin cậy cho quy tắc là 0.62 cho thấy rằng trong số tất cả những người có loại thu nhập là Working, 62% có thể mở thẻ tín dụng. Cuối cùng, mức nâng 3.46 cho chúng ta biết rằng những người mở thẻ có khả năng ở ở loại thu nhập Working cao hơn 4,76 lần so với khả năng những người chỉ ở loại thu nhập là Working.

### 4.5. Đánh giá chung về các mô hình

Sau khi tiến hành triển khai các kỹ thuật khai phá dữ liệu dựa trên bộ dữ liệu sưu tầm được, nhóm chúng em xin đưa ra một số nhận định như sau:

Thứ nhất, kỹ thuật phân lớp bằng thuật toán Cây quyết định (ID3, Cart), Bayes và KNN hoạt động tốt trên bộ dữ liệu, do bộ dữ liệu có nhiều thuộc tính định danh nên việc phân lớp sẽ trở nên dễ dàng hơn.

Thứ hai, kỹ thuật khai phá luật kết hợp với thuật toán Apriori mang lại hiệu quả tốt trong việc đưa ra một số nhóm đặc tính phổ biến của khách hàng có khả năng trả tín dụng, do kỹ thuật này hướng tới việc tìm ra những tập mục phổ biến. Kỹ thuật này cũng hoạt động tốt đối với bộ dữ liệu có những thuật toán định danh.

=> Nhìn chung, với bộ dữ liệu của Application Credit Record Hoa Kỳ, để dự đoán khả năng phát hành thẻ tín dụng của ngân hàng. Kỹ thuật khai phá luật kết hợp để tìm ra những nhóm thuộc tính phổ biến của những người có đủ khả năng trả tín dụng.

### 4.6. Phương pháp cải tiến mô hình

#### *4.6.1. Phương pháp cải tiến thuật toán phân lớp*

Một số cách có thể cải tiến hiệu suất:

- Kỹ thuật xử lý đặc trưng: Tạo ra thông tin mới hoặc tăng cường thông tin từ dữ liệu hiện có bằng cách sử dụng bộ dữ liệu lớn hơn với nhiều đặc trưng hơn. Hoặc có thể kết hợp để tạo ra các đặc trưng khác. Ví dụ như kết hợp tuổi và Nghề nghiệp

- Đối với mô hình cây ID3: có thể tối ưu hóa tham số mô hình: Sử dụng các chỉ số khác để lựa chọn thuộc tính phân chia (entropy, Gini…); độ sâu tối đa của cây (max_depth), số mẫu tối thiểu ở mỗi lá (min_samples_leaf).
- Thử nghiệm nhiều Mô hình: Thử nghiệm với các mô hình khác nhau để xem mô hình nào phù hợp tốt nhất với dữ liệu. Cụ thể, thử nghiệm các mô hình khác như C4.5, CART và Random Forest…
- Thay đổi tỉ lệ chia tập dữ liệu: Ví dụ: tỷ lệ phần trăm dữ liệu được sử dụng làm tập kiểm tra test_size = 0.3, khởi tạo bộ số ngẫu nhiên random_state = 42….

**Chạy kịch bản mới**

Kịch bản 1: Với bộ dữ liệu chia thành tập trên và test theo tỷ lệ 70/30

```
X = credit_approval_data.drop('STATUS', axis=1)
y = credit_approval_data['STATUS']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

Kết quả độ chính xác của các mô hình:

- Thuật toán Gaussian Naive Bayes

```
# Đánh giá mô hình
accuracy = metrics.accuracy_score(y_test, y_pred)
print("Độ chính xác:", accuracy)
```

```
Độ chính xác: 0.979246663009691
```

- Thuật toán Decision Tree Classifier – CART

```
#đưa ra độ chính xác
print('Accuracy score is: {:.5}'.format(accuracy_score(y_test, y_predict)))
```

```
Accuracy score is: 0.97605
```

- Thuật toán Decision Tree Classifier - ID3

```
# độ chính xác của mô hình cây
print("Accuracy: ", accuracy_score(y_test,y_pred)*100)
```

```
Accuracy:  98.94861949168038
```

- Thuật toán KNN Classifier

```
#Hiển thị độ chính xác
print("Accuracy of 2NN: %.2f %%" %(100*accuracy_score(y_test, y_pred)))
```

```
Accuracy of 2NN: 97.88 %
```

**Nhận xét:** Với tỷ lệ train/test là 70/30, độ chính xác của các mô hình khoảng 97%

Kịch bản 2: Với bộ dữ liệu chia thành tập test và train

**Kịch bản 2: chia bộ dữ liệu thành tập train và test theo tỷ lệ 80/20**

```
#Chia bộ dữ liệu theo tỷ lệ mới 80/20
X = credit_approval_data.drop('STATUS', axis=1)
y = credit_approval_data['STATUS']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

- Thuật toán Gaussian Naive Bayes

```
# Đánh giá mô hình
accuracy = metrics.accuracy_score(y_test, y_pred)
print("Độ chính xác:", accuracy)
```

```
Độ chính xác: 0.9777838727372463
```

- Thuật toán Decision Tree Classifier – CART

**Phân lớp bằng phương pháp sử dụng thuật toán CART**

```
#khai báo mô, thực thi, kiểm thử mô hình
my_tree = DecisionTreeClassifier()
my_tree.fit(X_train, y_train)
y_predict = my_tree.predict(X_test)
#đưa ra độ chính xác
print('Accuracy score is: {:.5}'.format(accuracy_score(y_test, y_predict)))
#in ra báo cáo phân loại chi tiết hơn bao gồm độ chính xác, độ nhớ recall, f1-score cho từng lớp
print(classification_report(y_test, y_predict))
```

```
Accuracy score is: 0.98738
```

- Thuật toán Decision Tree Classifier - ID3

```
# độ chính xác của mô hình cây
print("Accuracy: ", accuracy_score(y_test,y_pred)*100)
```

```
Accuracy:  98.99890290729567
```

- Thuật toán KNN Classifier

```
#Hiển thị độ chính xác
print("Accuracy of 2NN: %.2f %%" %(100*accuracy_score(y_test, y_pred)))
```

```
Accuracy of 2NN: 97.27 %
```

**Nhận xét:** Với tỷ lệ train/test là 80/20 độ chính xác của mô hình khoảng 98%

=>**Đánh giá:** Khi thay đổi tập test/train từ 70/30 thành 80/20, độ chính xác của mô hình tăng khoảng 1%

### *4.6.2. Phương pháp cải tiến thuật toán luật kết hợp*

Có nhiều phương pháp có thể được sử dụng để cải thiện độ chính xác của luật kết hợp trong dự đoán mở thẻ tín dụng. Một số phương pháp phổ biến bao gồm:

Chọn các đặc điểm phù hợp: Một ngân hàng có thể chọn các đặc điểm như tuổi, giới tính, thu nhập, và lịch sử tín dụng để xây dựng luật kết hợp. Các đặc điểm này đã được chứng minh là có liên quan đến việc mở thẻ tín dụng.

Cài đặt các tham số phù hợp: Một ngân hàng có thể điều chỉnh các tham số của thuật toán Apriori để giảm thiểu số lượng luật kết hợp không chính xác. Ví dụ, ngân hàng có thể tăng mức hỗ trợ tối thiểu để đảm bảo rằng chỉ các luật kết hợp xuất hiện thường xuyên mới được chấp nhận.

Sử dụng các phương pháp lọc: Một ngân hàng có thể sử dụng các phương pháp lọc, chẳng hạn như phương pháp kiểm tra độ tin cậy, để loại bỏ các luật kết hợp có độ tin cậy thấp.

Sử dụng các kỹ thuật học máy nâng cao: Một ngân hàng có thể sử dụng các kỹ thuật học máy nâng cao, chẳng hạn như học máy theo quy tắc, để xây dựng các luật kết hợp có độ chính xác cao hơn.

=> Việc lựa chọn phương pháp nào để cải thiện độ chính xác của luật kết hợp phụ thuộc vào nhiều yếu tố, bao gồm dữ liệu có sẵn, mục tiêu của ngân hàng, và nguồn lực có sẵn.

## Luật kết hợp 1

```
association_rules = apriori(records, min_support = 0.02, min_confidence = 0.0002, min_lift = 3, min_length = 2)
```

*Xem tổng số luật được tạo ra*

```
print(len(association_results))
```

```
895
```

**Nhận xét:** Khi thay đổi min_support từ 0.09 xuống 0.02, và min_confidence từ 0.02 xuống 0.002, só luật được tạo ra nhiều hơn

```
Rule: Higher education -> Yes credit card
Support: 0.071
Confidence: 0.27952755905511806
Lift: 3.038343033207805
```

Đánh giá kết quả: Luật này cho biết những người có trình độ học tập là Higher education có khả năng mở thẻ tín dụng

```
====================================
Rule: House / apartment -> Yes credit card
Support: 0.037
Confidence: 0.8043478260869565
Lift: 3.3654720756776424
```

Đánh giá kết quả: Luật này cho biết những người có loại nhà là House/apartment  có khả năng mở thẻ tín dụng

```
Rule: Pensioner -> Yes credit card
Support: 0.037
Confidence: 0.8043478260869565
Lift: 7.312252964426878
```

Đánh giá kết quả: Luật này cho biết những người loại thu nhập là Pensioner có khả năng mở thẻ tín dụng

## Luật kết hợp 2:

```
association_rules = apriori(records, min_support = 0.04, min_confidence = 0.0002, min_lift = 3, min_length = 2)
```

```
print(len(association_results))
```

```
535
```

Khi thay đổi min_support từ 0.04 xuống 0.02, và min_confidence từ 0.02 xuống 0.0002, só luật được tạo ra nhiều hơn so với ban đầu và ít hơn so với Kịch bản 1

```
Rule: 43.0 -> Yes credit card
Support: 0.05
Confidence: 0.9803921568627452
Lift: 13.071895424836603
---------------------------------------
Rule: Single / not married -> Yes credit card
Support: 0.05
Confidence: 0.9803921568627452
Lift: 7.906388361796332
```

Đánh giá kết quả: Luật này cho biết những người có độ tuổi là 43có khả năng mở thẻ tín dụng và những người Single/not married có khả năng mở thẻ tín dụng

# CHƯƠNG 5: TỔNG KẾT

## 5.1. Kết quả đạt được

Qua quá trình tìm hiểu, nghiên cứu, và triển khai xây dựng giải pháp cho công tác dự báo phát hành thẻ tín dụng, dự án đã đạt được những thành tựu đáng chú ý như sau:

### 5.1.1. Hiểu rõ quy trình phát hành thẻ tín dụng của ngân hàng và các lưu ý quan trọng:

Tiến hành nghiên cứu kỹ lưỡng về các bước và quy trình liên quan đến việc phát hành thẻ tín dụng trong ngân hàng.

Đặt ra các điểm lưu ý quan trọng trong quá trình này để tối ưu hóa quá trình dự báo.

### 5.1.2. Phân tích, đánh giá các giải pháp dự báo hiện có trong ngành ngân hàng:

Tiến hành nghiên cứu sâu rộng về các giải pháp dự báo được triển khai trong các ngân hàng hiện nay.

Đưa ra nhận định về hiệu suất, ưu điểm và nhược điểm của các giải pháp này.

### 5.1.3. Ứng dụng kiến thức để tiền xử lý và biến đổi dữ liệu:

Sử dụng kiến thức đúc kết để thực hiện quá trình tiền xử lý dữ liệu, biến đổi bộ dữ liệu để chuẩn bị cho việc xây dựng các mô hình dự báo.

### 5.1.4. Triển khai các giải thuật phân lớp với độ chính xác cao:

Áp dụng các giải thuật phân lớp như CART, KNN, Gaussian Naive Bayes vào bài toán dự báo phát hành thẻ tín dụng.

Đạt được kết quả với độ chính xác cao và tốc độ xử lý nhanh, giúp cải thiện hiệu suất dự báo.

### 5.1.5. Phân tích ưu nhược điểm của các giải thuật và lựa chọn thuật toán phù hợp:

Đưa ra phân tích chi tiết về ưu và nhược điểm của các giải thuật phân lớp sử dụng trong dự án.

Xác định cách lựa chọn thuật toán linh hoạt để đáp ứng các yêu cầu cụ thể trong quá trình dự báo.

Tổng cộng, dự án đã đạt được những kết quả tích cực thông qua quá trình nghiên cứu sâu và triển khai các giải pháp hiệu quả trong lĩnh vực dự báo phát hành thẻ tín dụng.

## 5.2 Điểm hạn chế

Tuy đã đạt được một số kết quả tích cực như trên, nhưng thực tế đề tài vẫn còn một số điểm hạn chế có thể kể đến như: bộ dữ liệu phục vụ việc xây dựng mới chỉ mang tính chất tham khảo, chưa phải dự liệu thật và các thuộc tính cũng chưa đầy đủ. Chưa kết hợp được nhiều thuật toán để đưa ra phương án tối ưu nhất.

### 5.3 Hướng phát triển

Việc dự báo phát hành thẻ tín dụng vẫn là một bài toán cần giải quyết của các ngân hàng hiện nay, để tiếp tục cài thiện đề tài và mang lại nhiều đóng góp hơn, nhóm đưa ra một số đề xuất như sau:

Một là, nghiên cứu đánh giá chi tiết hơn về các thủ tục, yêu cầu, quy trình phê duyệt thẻ tín dụng của ngân hàng và đặc biệt là các tiêu chí đánh giá.

Hai là, tìm hiểu và xây dựng các model được xây dựng trên nhiều các thuật toán khác hoặc kết hợp các thuật toán để có những nhận định đúng đắn nhất.

Ba là, tổng hợp và thu thập thêm nhiều dạng dữ liệu liên quan hơn đến việc phát hành thẻ tín dụng để phục vụ việc phân tích, đánh giá và xây dựng model.

# Tài liệu tham khảo

[1] Trang web chính thức của Ngân hàng Nhà nước Việt Nam. (2023, July 20). Retrieved from https://www.sbv.gov.vn/

[2] VietinBank. (2023, Tháng 12 27). VietinBank chính thức công bố kết quả kinh doanh năm 2023. https://investor.vietinbank.vn/NewsDetail.aspx?NewsID=78968

[3] Ngân hàng Nhà nước Việt Nam. (2020, Tháng 12 25). Xu hướng công nghệ số trong lĩnh vực ngân hàng và những yêu cầu đặt ra đối với hệ thống ngân hàng. Tạp chí Ngân hàng:https://tapchinganhang.gov.vn/xu-huong-cong-nghe-so-trong-linh-vuc-ngan-hang-va-nhung-yeu-cau-dat-ra-doi-voi-he-thong-ngan-hang-th.htm

[4] SKC Company. (2021, Tháng 9 29). Tín dụng ngân hàng là gì? Đặc điểm và vai trò như thế nào? Smartly Kapital. https://smartlykapital.vn/blogs/chia-se-bi-quyet-kinh-doanh/tin-dung-ngan-hang-la-gi-dac-diem-va-vai-tro-nhu-the-nao

[5] huytuong010101. (2020). Thuật Toán K-Nearest Neighbors (KNN) Siêu Cơ Bản. CodeLearn

[6] Cây quyết định (Decision Tree). (2019). Trong Machine Learning cơ bản. Truy cập ngày 28/12/2023.

[7] Tưởng Nguyễn Huy. (2022). ML From Scratch: Thuật toán phân loại Naive Bayes. Viblo

[8] Tek4. (2020, Tháng 12 10). Tổng quan về thuật toán cây quyết định. Tek4. Truy cập ngày 28 Tháng 12 năm 2023

[10] Kim Anh. (2020, 18 tháng 7). Thẻ tín dụng: Tất cả những điều cần biết trước khi sử dụng. Luật Việt Nam.

[11] Seanny, "Kaggle," 2019: https://www.kaggle.com/datasets/rikdifos/credit-card-approval-prediction.

[12] H. Khoa HTTTQL, "Tài liệu môn học Khai phá dữ liệu".

[13] Nhung, B. T. H., Hiến, V. D., & Huyền, G. T. T. (2022). Tài liệu học tập Khai phá dữ liệu trong lĩnh vực ngân hàng. Hồng Đức.