**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**

****

**BÀI TẬP LỚN**

**NHẬP MÔN KHOA HỌC DỮ LIỆU**

**ỨNG DỤNG HỌC MÁY TRONG**

**DỰ ĐOÁN BỆNH UNG THƯ PHỔI**

NGÀNH: KHOA HỌC MÁY TÍNH

SINH VIÊN: **DƯƠNG VIỆT HÙNG**

MÃ SINH VIÊN: **10122185**

LỚP: **124221KS**

NGƯỜI HƯỚNG DẪN: **PGS. TS. NGUYỄN MINH TIẾN**

**HƯNG YÊN – 2025**

**NHẬN XÉT**

**Nhận xét của giáo viên hướng dẫn**

**GIÁO VIÊN HƯỚNG DẪN**

**Nguyễn Minh Tiến**

**LỜI CAM ĐOAN**

Tôi xin cam đoan bài tập lớn “Ứng dụng học máy trong dự đoán bệnh ung thư phổi” là kết quả thực hiện của bản thân tôi.

Những phần sử dụng tài liệu tham khảo trong bài tập lớn đã được nêu rõ trong phần tài liệu tham khảo. Các kết quả trình bày trong bài tập lớn và chương trình xây dựng được hoàn toàn là kết quả do bản thân tôi thực hiện.

Nếu vi phạm lời cam đoan này, tôi xin chịu hoàn toàn trách nhiệm trước khoa và nhà trường.

*Hưng Yên, ngày 30 tháng 05 năm 2025*

Sinh viên Hùng

Dương Việt Hùng

**LỜI CẢM ƠN**

Để có thể hoàn thành bài tập lớn này, lời đầu tiên tôi xin phép gửi lời cảm ơn tới bộ môn Khoa học máy tính, Khoa Công nghệ thông tin – Trường Đại học Sư phạm Kỹ thuật Hưng Yên đã tạo điều kiện thuận lợi cho tôi thực hiện bài tập lớn môn học này.

Đặc biệt tôi xin chân thành cảm ơn thầy Nguyễn Minh Tiến đã rất tận tình hướng dẫn, chỉ bảo tôi trong suốt thời gian thực hiện bài tập lớn vừa qua.

Tôi cũng xin chân thành cảm ơn tất cả các Thầy, các Cô trong Trường đã tận tình giảng dạy, trang bị cho tôi những kiến thức cần thiết, quý báu để giúp tôi thực hiện được bài tập lớn này.

Mặc dù tôi đã có cố gắng, nhưng với trình độ còn hạn chế, trong quá trình thực hiện đề tài không tránh khỏi những thiếu sót. Tôi hy vọng sẽ nhận được những ý kiến nhận xét, góp ý của các Thầy cô về những kết quả triển khai trong bài tập lớn.

Tôi xin trân trọng cảm ơn!

**MỤC LỤC**

[CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN 8](#_Toc15587)

[1.1 Bài toán 8](#_Toc5980)

[1.2 Trình bày dữ liệu bài toán 8](#_Toc15671)

[1.3 Tiền xử lý dữ liệu 12](#_Toc7359)

[1.4 Trực quan hoá dữ liệu 12](#_Toc30733)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 13](#_Toc28904)

[2.1 Pandas 13](#_Toc15541)

[2.2 Matplotlib 15](#_Toc29220)

[2.3 Sklearn 17](#_Toc4999)

[2.4 Synthetic Minority Over-sampling Technique (SMOTE) 19](#_Toc1470)

[2.5 Random Oversampling Technique 20](#_Toc6569)

[2.6 Data Scaling 21](#_Toc5411)

[2.7 HyperParameter Tuning 22](#_Toc22574)

[2.8 Cross Validation 22](#_Toc7443)

[2.9 Machine Learning 24](#_Toc22790)

[2.9.1 Decision Tree 24](#_Toc10046)

[2.9.2 Naive Bayes 25](#_Toc4243)

[2.9.3 XGBoost 26](#_Toc2572)

[2.9.4 Random Forest 27](#_Toc10475)

[2.9.5 KNN 28](#_Toc26461)

[2.10 Confusion Matrix 29](#_Toc25606)

[2.11 Evaluated Metrics 30](#_Toc27264)

[CHƯƠNG 3: GIẢI PHÁP 32](#_Toc13410)

[3.1. Tiền xử lý dữ liệu 32](#_Toc1720)

[3.2. Trực quan hóa dữ liệu 36](#_Toc26611)

[3.3. Đánh giá mô hình 40](#_Toc17152)

[KẾT LUẬN 43](#_Toc24398)

[TÀI LIỆU THAM KHẢO 44](#_Toc17654)

**DANH MỤC CÁC HÌNH VẼ**

[Hình 1.1 Tổng quan về bộ dữ liệu Lung Cancer 10](#_Toc6288)

[Hình 1.2 Thông tin cơ bản về kiểu dữ liệu của từng đặc trưng 10](#_Toc3201)

[Hình 2.1: Synthetic Minority Oversampling Techique 20](#_Toc8656)

[Hình 2.2: Confusion matrix 30](#_Toc19761)

[Hình 3.1: Kiểm tra giá trị khuyết thiếu 33](#_Toc3342)

[Hình 3.2: Kiểm tra giá trị trùng lặp 33](#_Toc15275)

[Hình 3.3: Xử lý giá trị ngoại lai 34](#_Toc2221)

[Hình 3.4: Feature selection 34](#_Toc27850)

[Hình 3.5: Data Split 35](#_Toc7674)

[Hình 3.6: Cân bằng dữ liệu 35](#_Toc29955)

[Hình 3.7: Scale dữ liệu 36](#_Toc26249)

[Hình 3.8: Phân phối của nhãn dữ liệu 37](#_Toc19144)

[Hình 3.9: Phát hiện ngoại lai 38](#_Toc23941)

[Hình 3.10: Phân bố tuổi 38](#_Toc26893)

[Hình 3.11: Phân bố giới tính 39](#_Toc13479)

[Hình 3.12: Quan hệ giữa hút thuốc và ung thư phổi 39](#_Toc6901)

[Hình 3.13: Quan hệ giữa gia đình và ung thư phổi 40](#_Toc4519)

[Hình 3.14: Ma trận tương quan 40](#_Toc25074)

[Hình 3.15: Lần 1: (chưa tiền xử lý, tham số mặc định) 41](#_Toc94)

[Hình 3.16: Lần 2: (sau tiền xử lý, SMOTE) 41](#_Toc12767)

[Hình 3.17: Lần 2: (sau tiền xử lý, ROS) 42](#_Toc1795)

[Hình 3.18: Lần 3: (Tùy chỉnh tham số, SMOTE) 42](#_Toc23252)

[Hình 3.19: Lần 3: (Tùy chỉnh tham số, ROS) 43](#_Toc29978)

# GIỚI THIỆU BÀI TOÁN

## Bài toán

Chẩn đoán và dự đoán bệnh ung thư phổi là một trong những ứng dụng nổi bật và đầy tiềm năng của học máy (machine learning) trong lĩnh vực y học hiện đại. Ung thư phổi hiện là một trong những nguyên nhân gây tử vong hàng đầu trên toàn cầu, đặc biệt tại các quốc gia đang phát triển, nơi mà điều kiện tầm soát và chẩn đoán còn hạn chế. Tại Việt Nam, số ca mắc mới và tử vong do ung thư phổi vẫn ở mức cao, phần lớn do bệnh được phát hiện ở giai đoạn muộn. Việc ứng dụng học máy mang lại khả năng phân tích và xử lý dữ liệu để hỗ trợ phát hiện sớm và phân loại nguy cơ mắc bệnh với độ chính xác cao. Các thuật toán tiên tiến như rừng ngẫu nhiên (Random Forest), XGBoost đã chứng minh khả năng vượt trội trong việc nhận diện tổn thương phổi, phân biệt khối u lành tính và ác tính, cũng như dự đoán tiến triển của bệnh theo thời gian.

Ngoài ra, học máy còn hỗ trợ cá nhân hóa phương pháp điều trị thông qua phân tích đặc điểm di truyền của từng bệnh nhân (genomics), từ đó tối ưu hóa phác đồ hóa trị, xạ trị hoặc liệu pháp miễn dịch. Tuy nhiên, để các mô hình này phát huy hiệu quả trong thực tế lâm sàng, cần đảm bảo nguồn dữ liệu chất lượng cao, đa dạng và được chú thích chính xác, đồng thời đáp ứng các yêu cầu đạo đức và bảo mật thông tin y tế. Tại Việt Nam, những thách thức về hạ tầng công nghệ, nguồn nhân lực chuyên môn và hành lang pháp lý vẫn đang là rào cản lớn. Do đó, việc phát triển và triển khai hệ thống dự đoán ung thư phổi thông minh cần có sự phối hợp liên ngành giữa các cơ sở y tế, viện nghiên cứu, trường đại học và doanh nghiệp công nghệ nhằm xây dựng nền tảng dữ liệu y tế số hóa, thúc đẩy nghiên cứu ứng dụng AI trong y học, góp phần nâng cao chất lượng khám chữa bệnh và giảm tỷ lệ tử vong do ung thư phổi trong tương lai.

## Trình bày dữ liệu bài toán

Link dữ liệu trên Kaggle:

Lung Cancer Dataset [| Kaggle](https://www.kaggle.com/datasets/rabieelkharoua/alzheimers-disease-dataset/data)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **AGE** | **GENDER** | **SMOKING** | **FINGER\_DISCOLORATION** | **MENTAL\_STRESS** | **EXPOSURE\_TO\_POLLUTION** | **LONG\_TERM\_ILLNESS** | **ENERGY\_LEVEL** |
| 68 | 1 | 1 | 1 | 1 | 1 | 0 | 57.83 |
| 81 | 1 | 1 | 0 | 0 | 1 | 1 | 47.69 |
| 58 | 1 | 1 | 0 | 0 | 0 | 0 | 59.58 |
| 44 | 0 | 1 | 0 | 1 | 1 | 0 | 59.79 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **IMMUNEWEAKNESS** | **BREATHING\_ISSUE** | **ALCOHOL\_CONSUMPTION** | **THROAT\_DISCOMFORT** | **OXYGEN\_SATURATION** | **CHEST\_TIGHTNESS** | **FAMILY\_HISTORY** | **SMOKING\_FAMILY\_HISTORY** | **STRESS\_IMMUNE** | **PULMONARY\_DISEASE** |
| 0 | 0 | 1 | 1 | 95.98 | 1 | 0 | 0 | 0 | NO |
| 1 | 1 | 0 | 1 | 97.18 | 0 | 0 | 0 | 0 | YES |
| 0 | 1 | 1 | 0 | 94.97 | 0 | 0 | 0 | 0 | NO |
| 0 | 1 | 0 | 1 | 95.19 | 0 | 0 | 0 | 0 | YES |

Dữ liệu bài toán gồm các feature sau:

1. **AGE**: Tuổi của đối tượng (số nguyên). Tuổi tác có thể ảnh hưởng đến nguy cơ mắc ung thư phổi, đặc biệt ở những người lớn tuổi.
2. **GENDER**: Giới tính (0 hoặc 1, có thể đại diện cho nữ hoặc nam). Giới tính có thể liên quan đến tỷ lệ mắc bệnh do khác biệt về lối sống hoặc sinh học.
3. **SMOKING**: Thói quen hút thuốc (0: không hút, 1: hút thuốc). Hút thuốc là yếu tố nguy cơ chính gây ung thư phổi.
4. **FINGER\_DISCOLORATION**: Tình trạng đổi màu ngón tay (0: không, 1: có). Đây có thể là dấu hiệu của các vấn đề sức khỏe liên quan đến phổi hoặc tuần hoàn.
5. **MENTAL\_STRESS**: Mức độ căng thẳng tinh thần (0: không, 1: có). Căng thẳng có thể ảnh hưởng gián tiếp đến sức khỏe tổng thể và hệ miễn dịch.
6. **EXPOSURE\_TO\_POLLUTION**: Tiếp xúc với ô nhiễm (0: không, 1: có). Ô nhiễm không khí (như PM2.5, PM10, NO2) là yếu tố nguy cơ môi trường.
7. **LONG\_TERM\_ILLNESS**: Bệnh mãn tính lâu dài (0: không, 1: có). Các bệnh mãn tính có thể làm tăng nguy cơ ung thư phổi.
8. **ENERGY\_LEVEL**: Mức năng lượng (giá trị liên tục, có thể là phần trăm). Mức năng lượng thấp có thể liên quan đến các triệu chứng của bệnh phổi.
9. **IMMUNE\_WEAKNESS**: Suy yếu hệ miễn dịch (0: không, 1: có). Hệ miễn dịch yếu có thể làm tăng nguy cơ mắc bệnh nghiêm trọng.
10. **BREATHING\_ISSUE**: Vấn đề về hô hấp (0: không, 1: có). Khó thở là triệu chứng phổ biến liên quan đến ung thư phổi.
11. **ALCOHOL\_CONSUMPTION**: Tiêu thụ rượu bia (0: không, 1: có). Uống rượu có thể ảnh hưởng gián tiếp đến sức khỏe phổi.
12. **THROAT\_DISCOMFORT**: Khó chịu ở cổ họng (0: không, 1: có). Đây có thể là triệu chứng sớm của các vấn đề về đường hô hấp.
13. **OXYGEN\_SATURATION**: Độ bão hòa oxy trong máu (giá trị liên tục, thường là phần trăm). Giá trị thấp có thể chỉ ra vấn đề về phổi.
14. **CHEST\_TIGHTNESS**: Cảm giác tức ngực (0: không, 1: có). Tức ngực là triệu chứng tiềm năng của ung thư phổi hoặc bệnh phổi khác.
15. **FAMILY\_HISTORY**: Tiền sử gia đình mắc ung thư phổi (0: không, 1: có). Yếu tố di truyền có thể làm tăng nguy cơ.
16. **SMOKING\_FAMILY\_HISTORY**: Tiền sử gia đình có người hút thuốc (0: không, 1: có). Tiếp xúc thụ động với khói thuốc là yếu tố nguy cơ.
17. **STRESS\_IMMUNE**: Căng thẳng ảnh hưởng đến hệ miễn dịch (0: không, 1: có). Căng thẳng kéo dài có thể làm suy yếu khả năng miễn dịch.
18. **PULMONARY\_DISEASE**: Bệnh phổi hiện có (NO: không, YES: có). Đây là biến mục tiêu (target variable) trong tập dữ liệu, chỉ ra liệu đối tượng có mắc bệnh phổi (ung thư phổi) hay không.

Dữ liệu bài toán là 1 file csv gồm 5000 rows × 18 columns

Tương ứng với có 18 features và mỗi feature có 5000 dữ liệu đầu vào.

- Sau khi mô tả dữ liệu ta có:

## Tiền xử lý dữ liệu

1. Xử lý giá trị khuyết thiếu, trùng lặp
2. Xử lý giá trị ngoại lai
3. Feature selection
4. Phân tách dữ liệu
5. Xử lý mất cân bằng nhãn
6. Xử lý chuẩn hóa dữ liệu

## Trực quan hoá dữ liệu

1. Phân bố nhãn
2. Phát hiện ngoại lai
3. Phân bố tuổi
4. Phân bố giới tính
5. Quan hệ giữa hút thuốc và ung thư phổi
6. Quan hệ giữa gia đình và ung thư phổi
7. Ma trận tương quan

## Xây dựng mô hình học máy

1. Xây dựng mô hình Decision Tree
2. Xây dựng mô hình Random Forest
3. Xây dựng mô hình KNN
4. Xây dựng mô hình Naive Bayes
5. Xây dựng mô hình XGBoost

# CƠ SỞ LÝ THUYẾT

## Decision Tree

### Giới thiệu thuật toán Decision Tree

Cây quyết định là một thuật toán học máy thuộc nhóm mô hình dự đoán, sử dụng cấu trúc dạng cây để đưa ra quyết định dựa trên các đặc trưng của dữ liệu. Thuật toán này chia tập dữ liệu thành các vùng dựa trên các điều kiện quyết định, dẫn đến kết quả cuối cùng (lớp hoặc giá trị số).

### Cấu trúc của mô hình Decision Tree

Cấu trúc cây:

* Nút gốc: Đại diện cho toàn bộ tập dữ liệu, được chia thành các nhánh dựa trên một đặc trưng.
* Nút nội bộ: Đại diện cho các điều kiện quyết định dựa trên đặc trưng và ngưỡng.
* Nút lá: Đại diện cho kết quả cuối cùng (lớp trong phân loại hoặc giá trị trong hồi quy).

Kết quả:

* Phân loại: Dự đoán lớp dựa trên nhánh dẫn đến nút lá.
* Hồi quy: Dự đoán giá trị số tại nút lá

### Quá trình xây dựng mô hình Decision Tree

Chọn đặc trưng và ngưỡng: Tại mỗi nút, chọn đặc trưng và ngưỡng tối ưu để phân tách dữ liệu dựa trên chỉ số như Gini Index hoặc Entropy:

Gini Index:

D

Entropy:

Phân tách đệ quy: Chia tập dữ liệu thành các tập con dựa trên đặc trưng và ngưỡng được chọn, lặp lại cho đến khi đạt điều kiện dừng (ví dụ: độ sâu tối đa, số mẫu tối thiểu tại nút lá).

Dự đoán: Kết quả tại nút lá được sử dụng để đưa ra dự đoán cuối cùng..

**Ưu điểm của Decision Tree**

* Dễ hiểu và giải thích: Cấu trúc cây trực quan, dễ diễn giải các quyết định.
* Xử lý dữ liệu đa dạng: Hoạt động tốt với cả dữ liệu số và phân loại, không yêu cầu chuẩn hóa dữ liệu.
* Nhanh: Thời gian huấn luyện và dự đoán thường nhanh với tập dữ liệu vừa và nhỏ.

**Nhược điểm của Decision Tree**

* Dễ overfitting: Cây quá sâu có thể học cả nhiễu trong dữ liệu, dẫn đến hiệu suất kém trên dữ liệu mới.
* Nhạy cảm với dữ liệu: Thay đổi nhỏ trong dữ liệu có thể dẫn đến cấu trúc cây khác biệt lớn.
* Hiệu suất hạn chế: Thường kém chính xác hơn các mô hình phức tạp hơn như Random Forest hoặc Gradient Boosting..

**Đánh giá hiệu suất**

Hiệu suất của cây quyết định được đánh giá bằng các chỉ số như Accuracy, Precision, Recall, và F1 Score (cho phân loại) hoặc Mean Squared Error (cho hồi quy). Cây quyết định thường được sử dụng khi cần mô hình đơn giản, dễ giải thích, hoặc làm nền tảng cho các thuật toán ensemble như Random Forest.

## Random Forest

### Giới thiệu thuật toán Random Forest

Random Forest là một thuật toán học máy thuộc nhóm ensemble learning, kết hợp nhiều cây quyết định (Decision Trees) để cải thiện độ chính xác và giảm thiểu hiện tượng overfitting. Random Forest sử dụng hai kỹ thuật chính:

* Bagging (Bootstrap Aggregating): Tạo nhiều tập dữ liệu con bằng cách lấy mẫu ngẫu nhiên có thay thế.
* Lựa chọn ngẫu nhiên đặc trưng: Tại mỗi nút của cây quyết định, chỉ một tập hợp con ngẫu nhiên của các đặc trưng được xem xét để phân tách.

Random Forest có thể được sử dụng cho cả bài toán phân loại và hồi quy, nhưng phổ biến hơn trong phân loại.

### Cấu trúc của mô hình Random Forest

Nhiều cây quyết định: Mỗi cây được huấn luyện trên một tập dữ liệu con (bootstrap sample) và một tập đặc trưng ngẫu nhiên.

Kết quả tổng hợp:

* Phân loại: Sử dụng bỏ phiếu đa số (majority voting) để chọn lớp dự đoán.
* Hồi quy: Lấy trung bình kết quả từ các cây.

Công thức tổng hợp (phân loại):

Trong đó:

### Quá trình xây dựng mô hình Random Forest

Lấy mẫu dữ liệu: Sử dụng kỹ thuật Bootstrap để tạo **T** tập dữ liệu con từ tập dữ liệu gốc.

Xây dựng cây quyết định:

* Tại mỗi nút, chọn ngẫu nhiên một tập hợp con các đặc trưng
* Tìm đặc trưng và ngưỡng tối ưu để phân tách dựa trên chỉ số như Gini Index hoặc Entropy:

Tổng hợp kết quả: Kết hợp dự đoán từ tất cả các cây bằng bỏ phiếu đa số (phân loại) hoặc trung bình (hồi quy).

**Ưu điểm của Random Forest**

* Giảm thiểu overfitting: Sự kết hợp nhiều cây độc lập giúp mô hình tổng quát hóa tốt hơn.
* Xử lý dữ liệu phức tạp: Hoạt động tốt với dữ liệu phi tuyến tính và dữ liệu có nhiều đặc trưng.
* Độ chính xác cao: Thường cho kết quả tốt hơn các mô hình đơn lẻ như cây quyết định.

**Nhược điểm của Random Forest**

* Tốn tài nguyên tính toán: Yêu cầu nhiều bộ nhớ và thời gian khi số lượng cây lớn.
* Khó giải thích: Kết quả từ nhiều cây khó diễn giải hơn so với các mô hình tuyến tính như Logistic Regression.

**Đánh giá hiệu suất**

Hiệu suất được đánh giá bằng các chỉ số như Accuracy, Precision, Recall, và F1 Score. Random Forest thường được chọn khi cần độ chính xác cao và khả năng xử lý dữ liệu phức tạp.

## K-Nearest Neighbors

### Giới thiệu thuật toán K-Nearest Neighbors

K-Nearest Neighbors (KNN) là một thuật toán học máy thuộc nhóm học có giám sát (supervised learning), được sử dụng cho cả bài toán phân loại (classification) và hồi quy (regression), nhưng phổ biến hơn trong phân loại. KNN là một thuật toán dựa trên khoảng cách, phân loại hoặc dự đoán giá trị của một mẫu dữ liệu mới dựa trên K điểm dữ liệu gần nhất trong tập huấn luyện. KNN sử dụng các thước đo khoảng cách như Euclidean, Manhattan, hoặc Minkowski để xác định mức độ gần gũi giữa các điểm dữ liệu.

### Cấu trúc của mô hình K-Nearest Neighbors

Không có giai đoạn huấn luyện phức tạp: KNN là một thuật toán lười học (lazy learning), nghĩa là nó không xây dựng mô hình rõ ràng trong giai đoạn huấn luyện mà lưu trữ toàn bộ tập dữ liệu huấn luyện.

Dự đoán:

* Phân loại: Dựa trên bỏ phiếu đa số (majority voting) của K láng giềng gần nhất.
* Hồi quy: Lấy trung bình (hoặc trung bình có trọng số) của giá trị K láng giềng gần nhất. Công thức khoảng cách Euclidean (phổ biến nhất):

Trong đó x và y là hai điểm dữ liệu, n là số đặc trưng.

### Quá trình xây dựng mô hình K-Nearest Neighbors

Chuẩn bị dữ liệu:

* Chuẩn hóa dữ liệu (ví dụ: Min-Max Scaling hoặc Standard Scaling) để đảm bảo các đặc trưng có cùng thang đo, tránh ảnh hưởng của các đặc trưng có giá trị lớn.

Chọn số láng giềng (K):

* Giá trị K là một siêu tham số (hyperparameter), thường được chọn bằng cách thử nghiệm hoặc sử dụng kỹ thuật như cross-validation.
* K nhỏ có thể dẫn đến nhiễu, trong khi K lớn có thể làm mất tính cục bộ của dữ liệu..

Dự đoán:

* Tính khoảng cách từ điểm dữ liệu mới đến tất cả các điểm trong tập huấn luyện.
* Chọn K điểm gần nhất.
* Tổng hợp kết quả: sử dụng bỏ phiếu đa số (phân loại) hoặc trung bình (hồi quy).

**Ưu điểm của K-Nearest Neighbors**

* Đơn giản và dễ hiểu: Thuật toán trực quan, dễ triển khai.
* Không cần giả định về dữ liệu: Hoạt động tốt với dữ liệu phi tuyến tính và không yêu cầu phân phối cụ thể của dữ liệu.
* Linh hoạt: Có thể áp dụng cho cả phân loại và hồi quy.

**Nhược điểm của K-Nearest Neighbors**

* Tốn tài nguyên tính toán: Yêu cầu tính toán khoảng cách cho mọi điểm dữ liệu trong tập huấn luyện, đặc biệt khi tập dữ liệu lớn.
* Nhạy cảm với dữ liệu nhiễu: Dữ liệu nhiễu hoặc ngoại lai có thể làm sai lệch kết quả.
* Phụ thuộc vào siêu tham số K: Việc chọn K không phù hợp có thể ảnh hưởng đến hiệu suất.

**Đánh giá hiệu suất**

Hiệu suất của KNN được đánh giá bằng các chỉ số như Accuracy, Precision, Recall, và F1 Score (cho phân loại) hoặc Mean Squared Error (cho hồi quy). KNN thường được chọn khi tập dữ liệu không quá lớn và cần một mô hình đơn giản, dễ triển khai.

## Naive Bayes

### Giới thiệu thuật toán Naive Bayes

Naive Bayes là một thuật toán học máy thuộc nhóm học có giám sát (supervised learning), chủ yếu được sử dụng cho bài toán phân loại (classification). Naive Bayes sử dụng xác suất để dự đoán lớp của một mẫu dữ liệu mới, dựa trên xác suất có điều kiện của các đặc trưng. Có ba biến thể chính:

* Gaussian Naive Bayes: Dành cho dữ liệu liên tục, giả định đặc trưng tuân theo phân phối chuẩn.
* Multinomial Naive Bayes: Phù hợp cho dữ liệu rời rạc, thường dùng trong phân loại văn bản.
* Bernoulli Naive Bayes: Dành cho dữ liệu nhị phân (0/1).

### Cấu trúc của mô hình Naive Bayes

Dựa trên Định lý Bayes:

* P(C|X): Xác suất lớp C khi biết đặc trưng X (xác suất hậu nghiệm).
* P(X|C): Xác suất của đặc trưng X trong lớp C (likelihood).
* P(C): Xác suất tiên nghiệm của lớp C.
* P(X): Xác suất của đặc trưng X (thường được bỏ qua vì không ảnh hưởng đến so sánh lớp).

Giả định độc lập: Mỗi đặc trưng X được giả định độc lập với nhau, nên:

Dự đoán: Chọn lớp có xác suất hậu nghiệm cao nhất:

### Quá trình xây dựng mô hình Naive Bayes

Tính xác suất tiên nghiệm:

* Tính P(C) dựa trên tỷ lệ các lớp trong tập huấn luyện

Tính xác suất có điều kiện:

* Đối với Gaussian Naive Bayes: Ước lượng trung bình và phương sai của mỗi đặc trưng trong mỗi lớp.
* Đối với Multinomial Naive Bayes: Tính xác suất xuất hiện của các đặc trưng (ví dụ: tần suất từ trong văn bản).
* Đối với Bernoulli Naive Bayes: Tính xác suất đặc trưng có giá trị 1 trong mỗi lớp.

Dự đoán:

* Với mỗi mẫu dữ liệu mới, tính xác suất hậu nghiệm cho từng lớp.
* Chọn lớp có xác suất cao nhất.

**Ưu điểm của Naive Bayes**

* Đơn giản và nhanh: Thuật toán dễ triển khai, tính toán nhanh, đặc biệt với dữ liệu lớn.
* Hiệu quả với dữ liệu văn bản: Hoạt động tốt trong các bài toán như phân loại email rác, phân tích cảm xúc.
* Yêu cầu dữ liệu nhỏ: Có thể hoạt động tốt với tập dữ liệu huấn luyện nhỏ.

**Nhược điểm của Naive Bayes**

* Giả định độc lập không thực tế: Giả định các đặc trưng độc lập có thể làm giảm độ chính xác trong một số trường hợp.
* Nhạy cảm với dữ liệu mất cân bằng: Nếu một lớp chiếm ưu thế, mô hình có thể thiên vị lớp đó.
* Xử lý dữ liệu liên tục phức tạp: Gaussian Naive Bayes có thể kém hiệu quả nếu dữ liệu không tuân theo phân phối chuẩn.

**Đánh giá hiệu suất**

Hiệu suất của Naive Bayes được đánh giá bằng các chỉ số như Accuracy, Precision, Recall, và F1 Score. Thuật toán này thường được chọn khi cần một mô hình đơn giản, nhanh, và hiệu quả cho các bài toán phân loại, đặc biệt với dữ liệu văn bản hoặc dữ liệu có đặc trưng rời rạc.

## XGBoost

### Giới thiệu thuật toán XGBoost

XGBoost (Extreme Gradient Boosting) là một thuật toán học máy thuộc nhóm ensemble learning, sử dụng kỹ thuật gradient boosting để xây dựng một mô hình mạnh từ nhiều cây quyết định (decision trees). XGBoost cải tiến so với các phương pháp boosting truyền thống bằng cách tối ưu hóa hiệu suất và tốc độ tính toán, đồng thời giảm thiểu hiện tượng overfitting.

### Cấu trúc của mô hình XGBoost

Nhiều cây quyết định: Các cây được xây dựng tuần tự, mỗi cây tập trung vào việc giảm thiểu lỗi của mô hình tổng hợp.

Kết quả tổng hợp:

* Phân loại: Tổng hợp dự đoán từ các cây bằng cách sử dụng hàm softmax (cho phân loại đa lớp) hoặc sigmoid (cho phân loại nhị phân).
* Hồi quy: Tổng hợp dự đoán bằng cách cộng giá trị đầu ra của các cây.

Hàm mất mát:

Trong đó:

* : Hàm mất mát (ví dụ: log loss cho phân loại, mean squared error cho hồi quy).
* : Hạng điều chuẩn để kiểm soát độ phức tạp của cây f, thường bao gồm số nút lá và tổng bình phương trọng số lá.

Gradient Boosting: Mỗi cây được xây dựng để tối ưu hóa gradient của hàm mất mát, sử dụng đạo hàm bậc một và bậc hai (Hessian).

### Quá trình xây dựng mô hình XGBoost

Khởi tạo mô hình:

* Bắt đầu với một dự đoán ban đầu (thường là trung bình cho hồi quy hoặc xác suất bằng nhau cho phân loại).

Xây dựng cây tuần tự:

* Tính gradient và Hessian của hàm mất mát để xác định hướng tối ưu hóa.
* Chọn đặc trưng và ngưỡng phân tách tối ưu tại mỗi nút dựa trên tiêu chí tối ưu hóa hàm mất mát.
* Áp dụng điều chuẩn (regularization) để tránh cây quá phức tạp.

Tổng hợp kết quả:

* Cộng kết quả từ tất cả các cây để tạo ra dự đoán cuối cùng.
* Đối với phân loại, áp dụng hàm kích hoạt (softmax/sigmoid) để chuyển đổi thành xác suất.

**Ưu điểm của XGBoost**

* Độ chính xác cao: Hiệu quả vượt trội trong nhiều bài toán nhờ khả năng tối ưu hóa gradient và điều chuẩn.
* Xử lý dữ liệu phức tạp: Hoạt động tốt với dữ liệu phi tuyến tính, dữ liệu mất cân bằng, và dữ liệu có nhiều đặc trưng.
* Tính linh hoạt: Hỗ trợ nhiều loại hàm mất mát và có thể tùy chỉnh thông qua các siêu tham số.
* Tối ưu hóa hiệu suất: Tăng tốc tính toán thông qua xử lý song song và kỹ thuật như histogram-based splitting.

**Nhược điểm của XGBoost**

* Tốn tài nguyên tính toán: Yêu cầu nhiều bộ nhớ và thời gian huấn luyện khi tập dữ liệu lớn hoặc số cây lớn.
* Khó giải thích: Kết quả từ nhiều cây phức tạp hơn so với các mô hình đơn giản như Decision Tree.
* Nhạy cảm với siêu tham số: Cần điều chỉnh cẩn thận các tham số như learning rate, max depth, hoặc subsample để đạt hiệu suất tối ưu.

**Đánh giá hiệu suất**

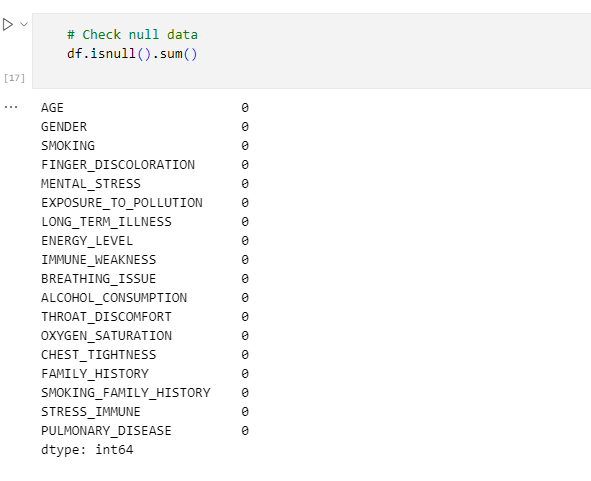
Hiệu suất của XGBoost được đánh giá bằng các chỉ số như Accuracy, Precision, Recall, và F1 Score (cho phân loại) hoặc Mean Squared Error (cho hồi quy). XGBoost thường được chọn khi cần độ chính xác cao và khả năng xử lý dữ liệu phức tạp, đặc biệt trong các bài toán cạnh tranh hoặc dữ liệu thực tế.

# GIẢI PHÁP

## Tiền xử lý dữ liệu

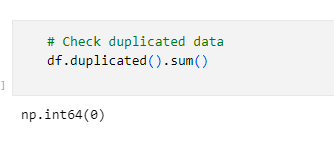
1. **Xử lý giá trị khuyết thiếu, trùng lặp**

Kiểm tra các giá trị khuyết thiếu



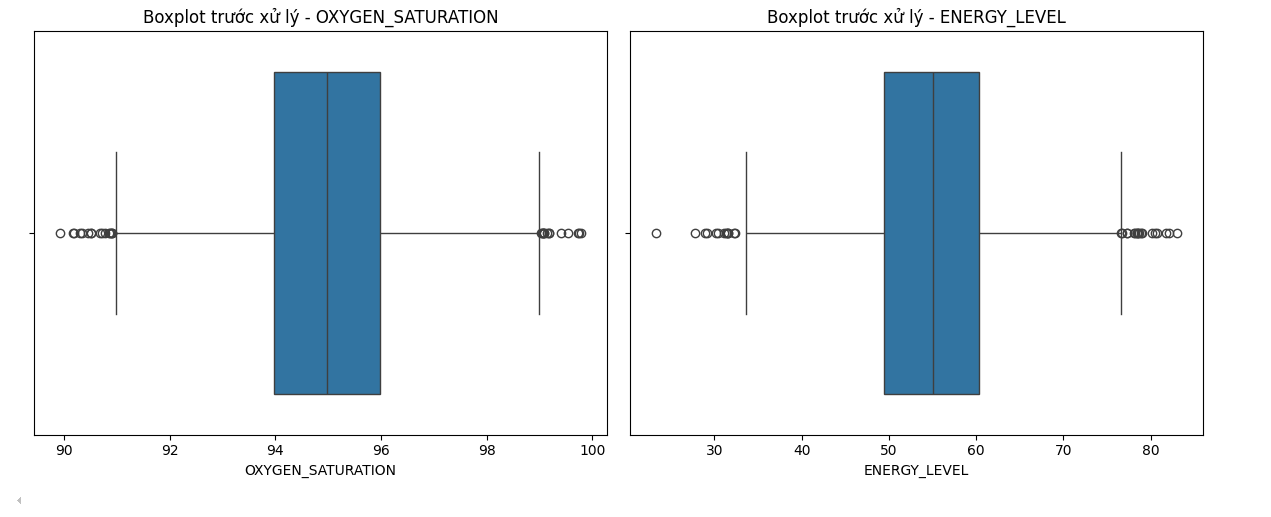
Hình 3.1: Kiểm tra giá trị khuyết thiếu

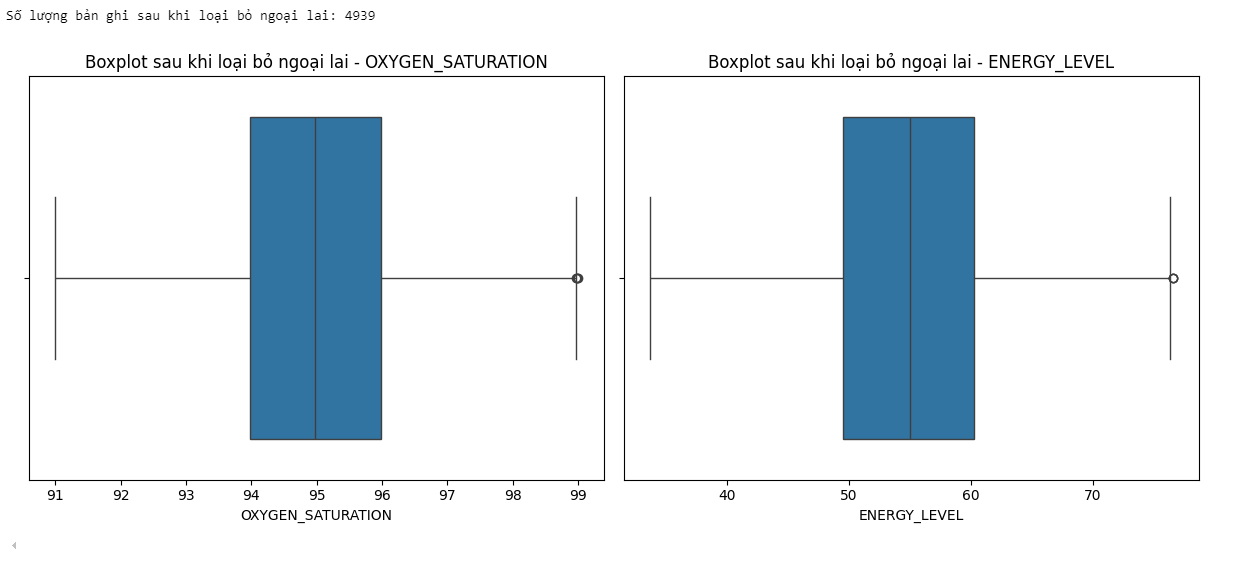
Kiểm tra các giá trị trùng lặp



Hình 3.2: Kiểm tra giá trị trùng lặp

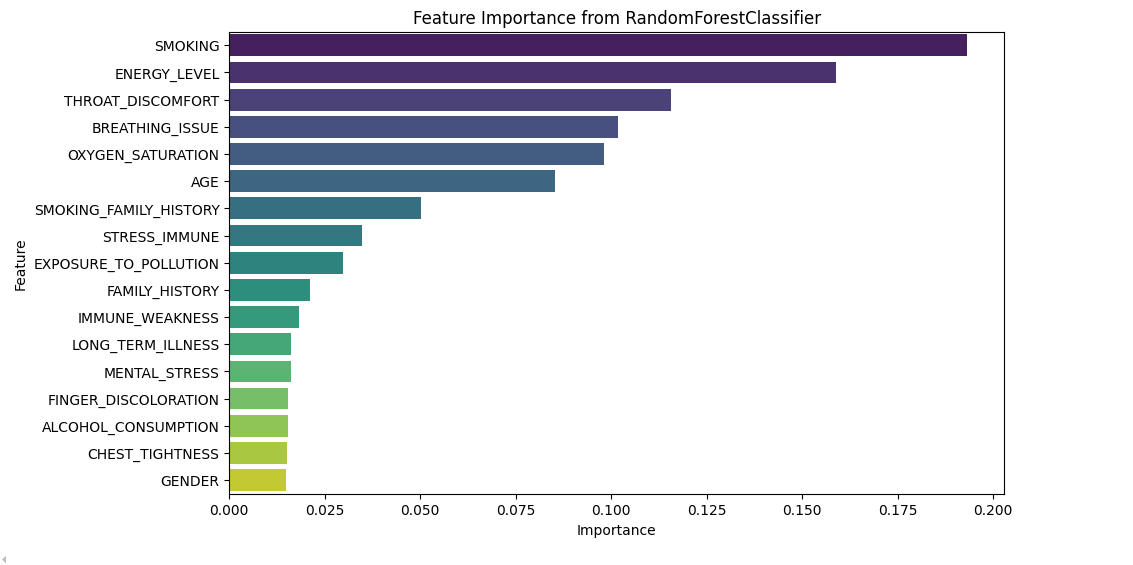
1. **Xử lý giá trị ngoại lai**





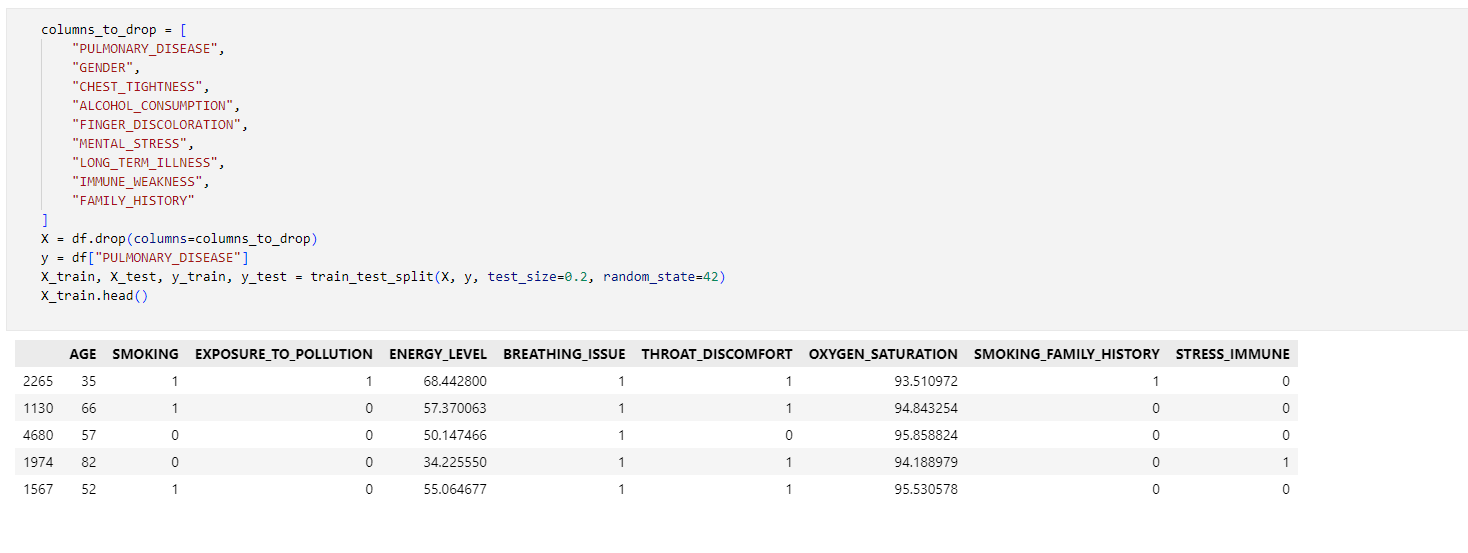
Hình 3.3: Xử lý giá trị ngoại lai

1. **Feature selection**



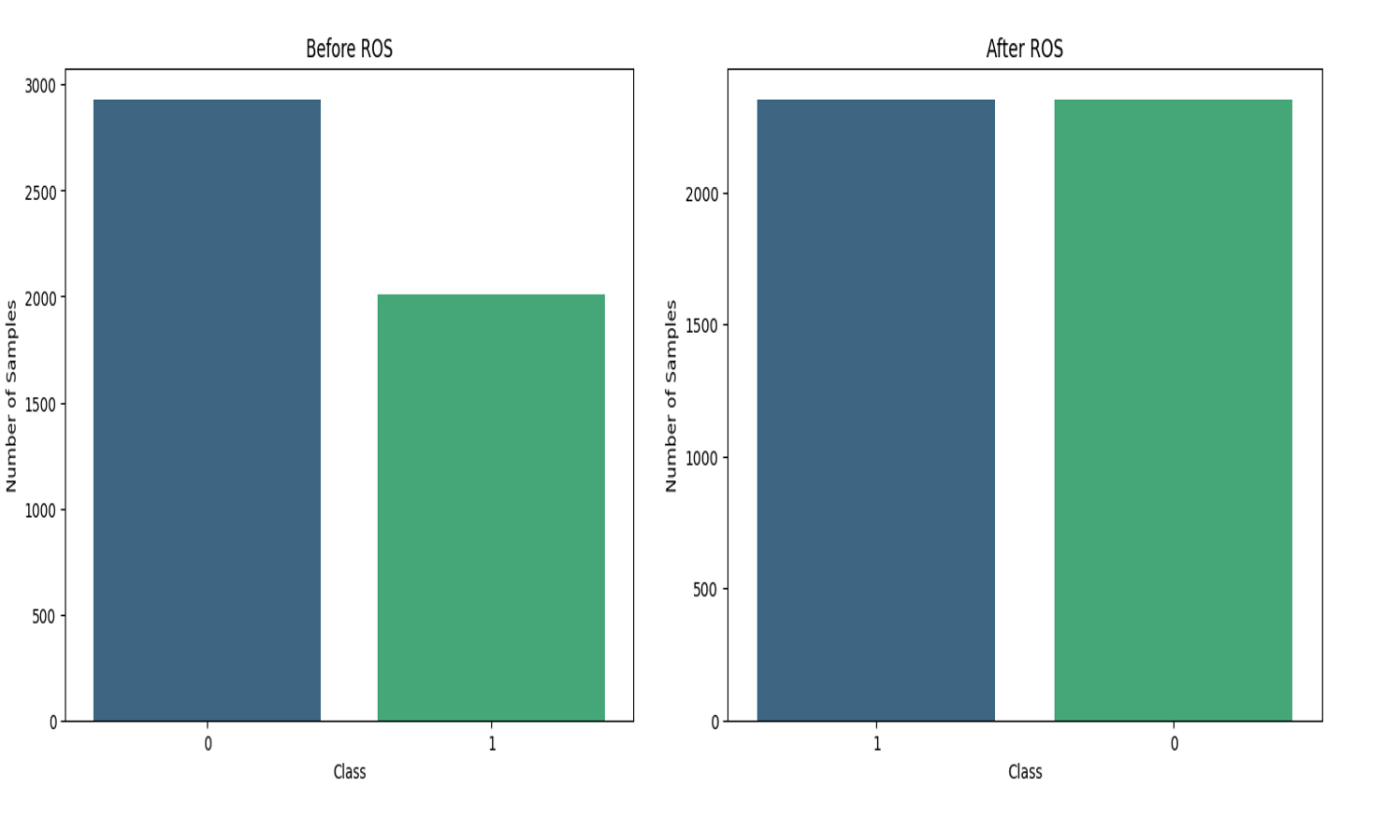
Hình 3.4: Feature selection

1. **Data Split**



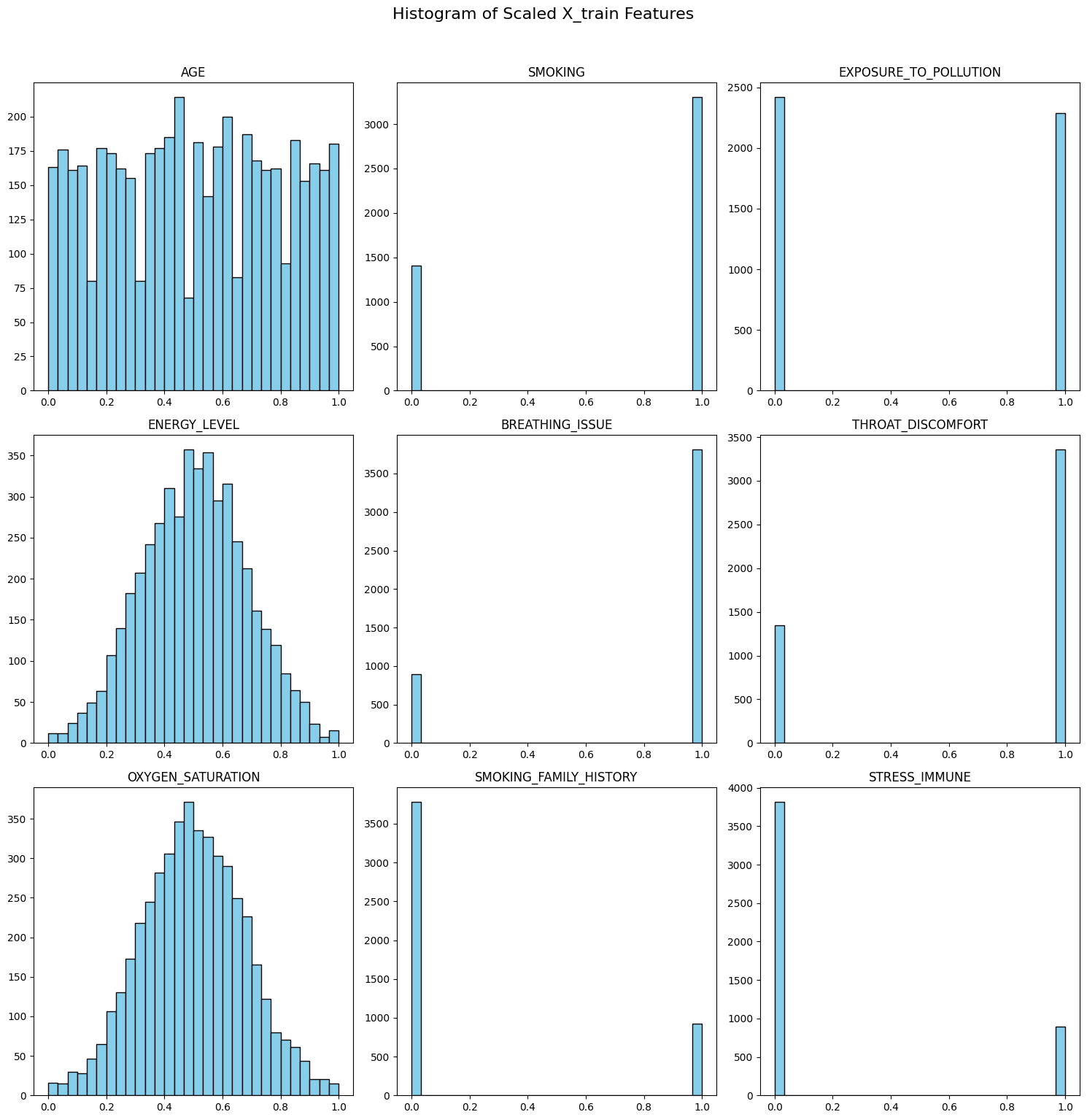
Hình 3.5: Data Split

1. **Cân bằng dữ liệu**



Hình 3.6: Cân bằng dữ liệu

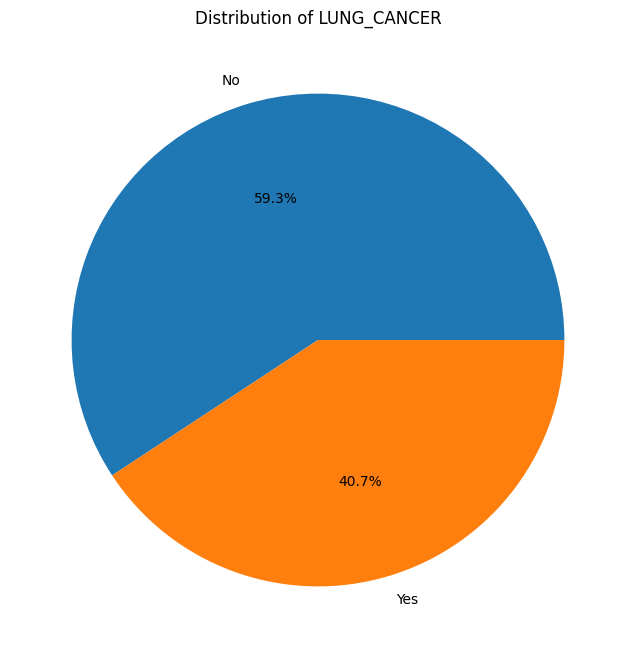
1. **Scale dữ liệu**



Hình 3.7: Scale dữ liệu

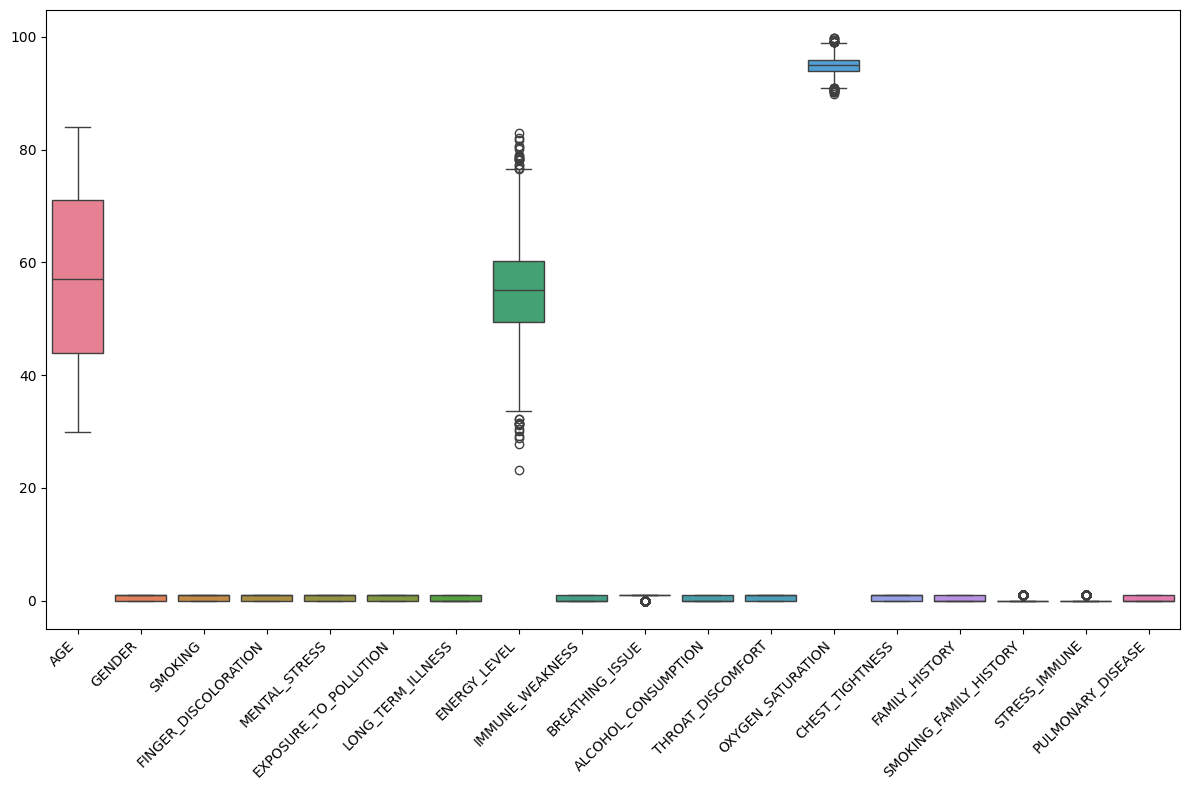
## Trực quan hóa dữ liệu

1. Phân phối nhãn dữ liệu



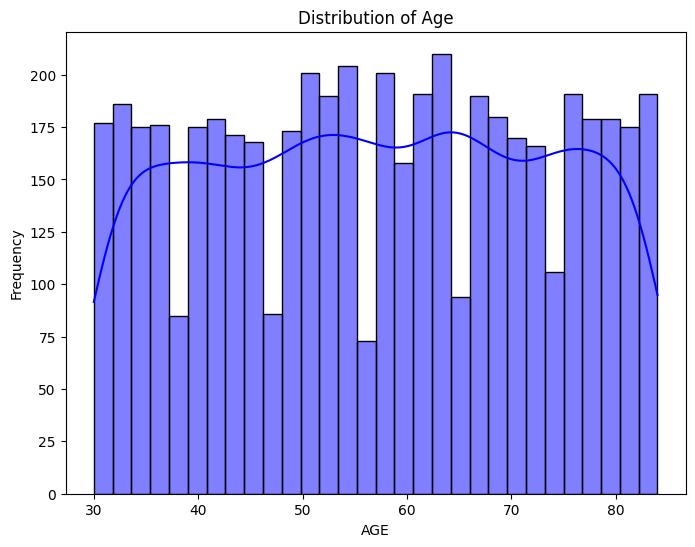
Hình 3.8: Phân phối của nhãn dữ liệu

1. Phát hiện ngoại lai



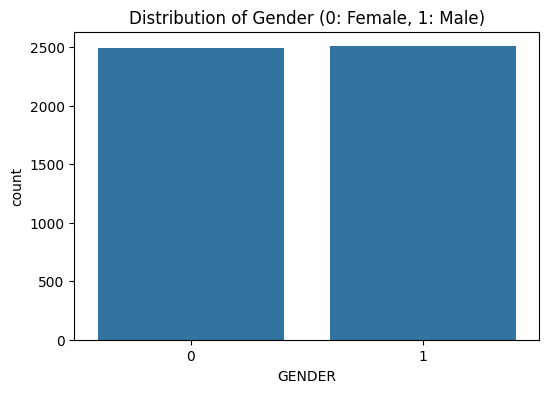
Hình 3.9: Phát hiện ngoại lai

1. Phân bố tuổi



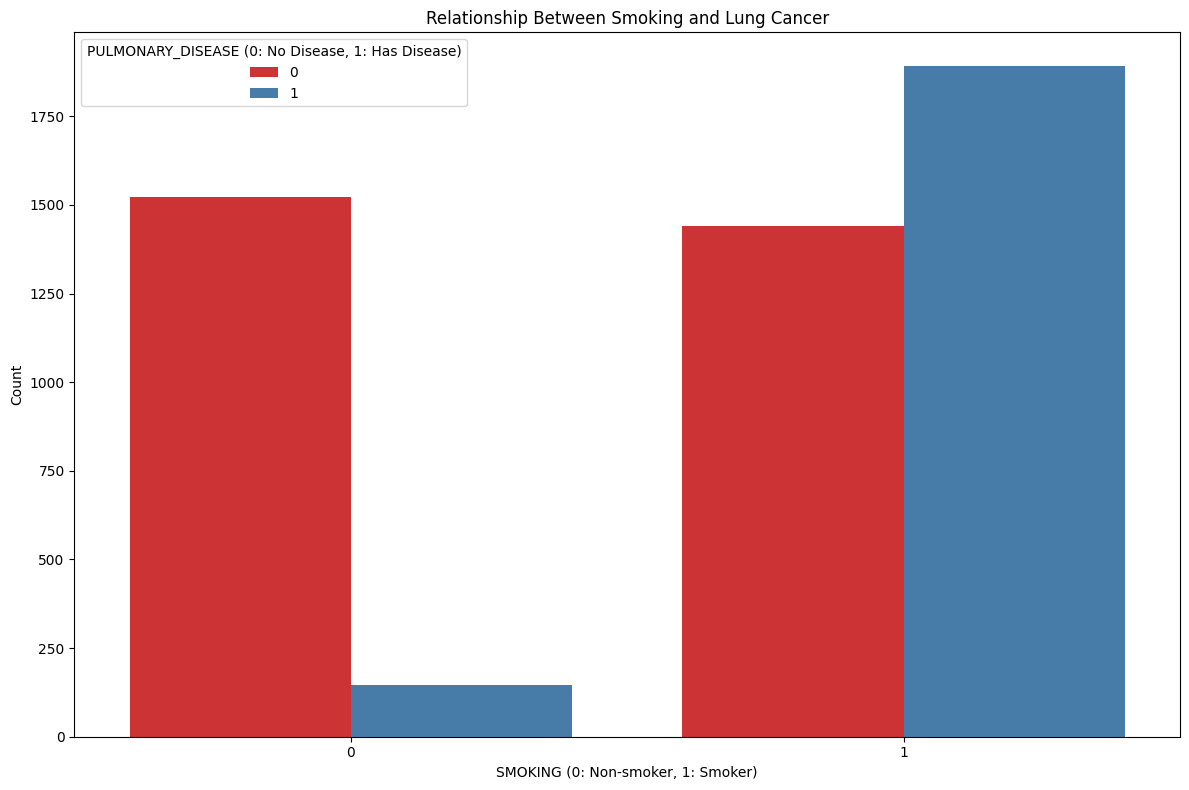
Hình 3.10: Phân bố tuổi

1. Phân bố giới tính



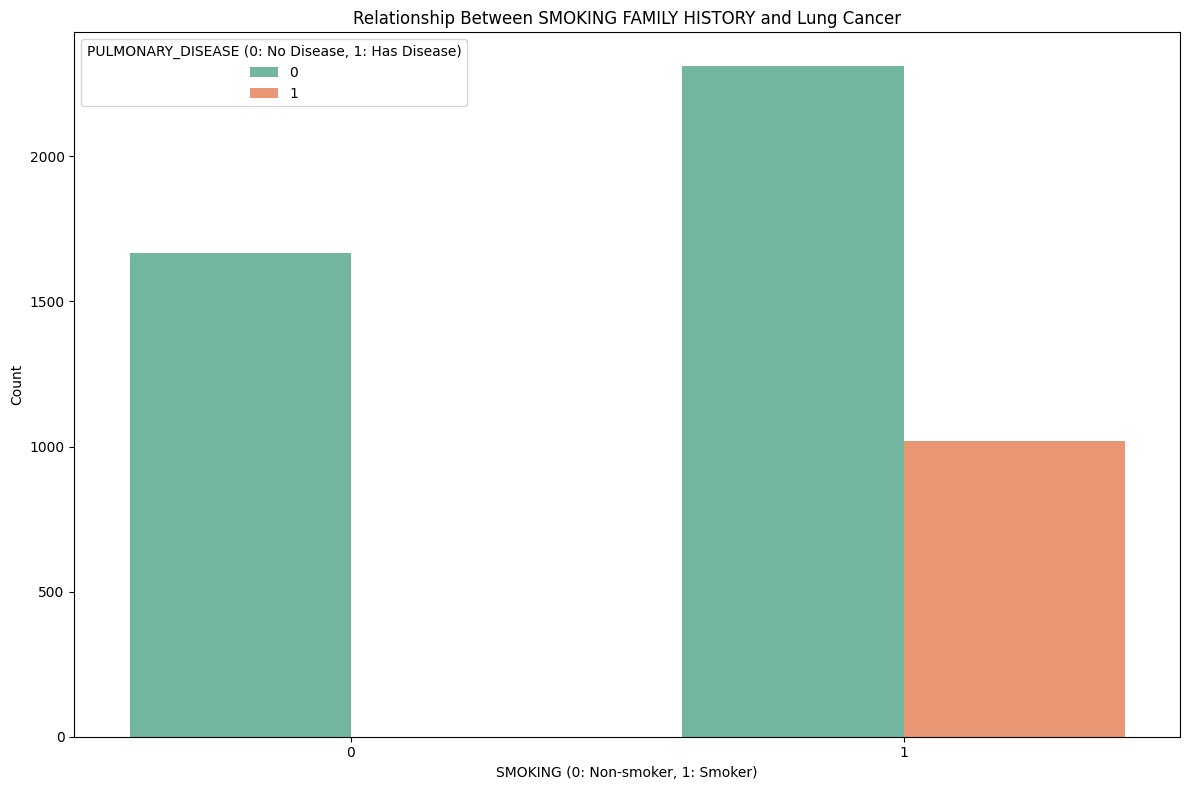
Hình 3.11: Phân bố giới tính

1. Quan hệ giữa hút thuốc và ung thư phổi



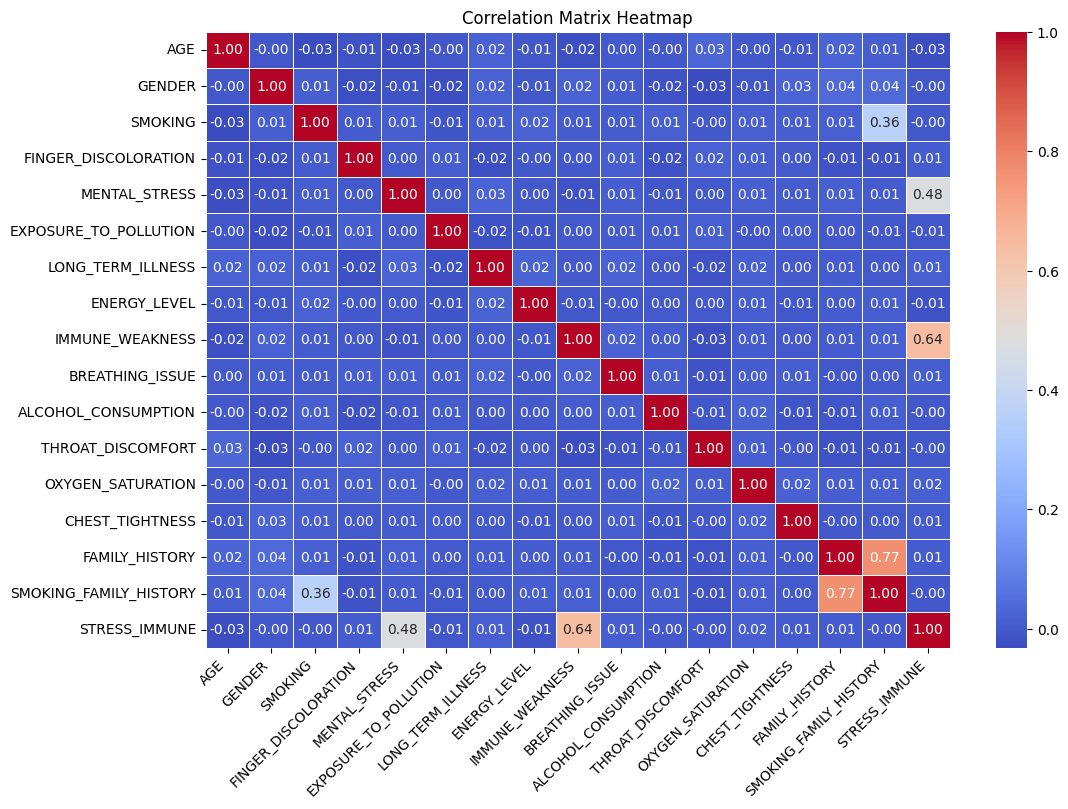
Hình 3.12: Quan hệ giữa hút thuốc và ung thư phổi

1. Quan hệ giữa gia đình và ung thư phổi



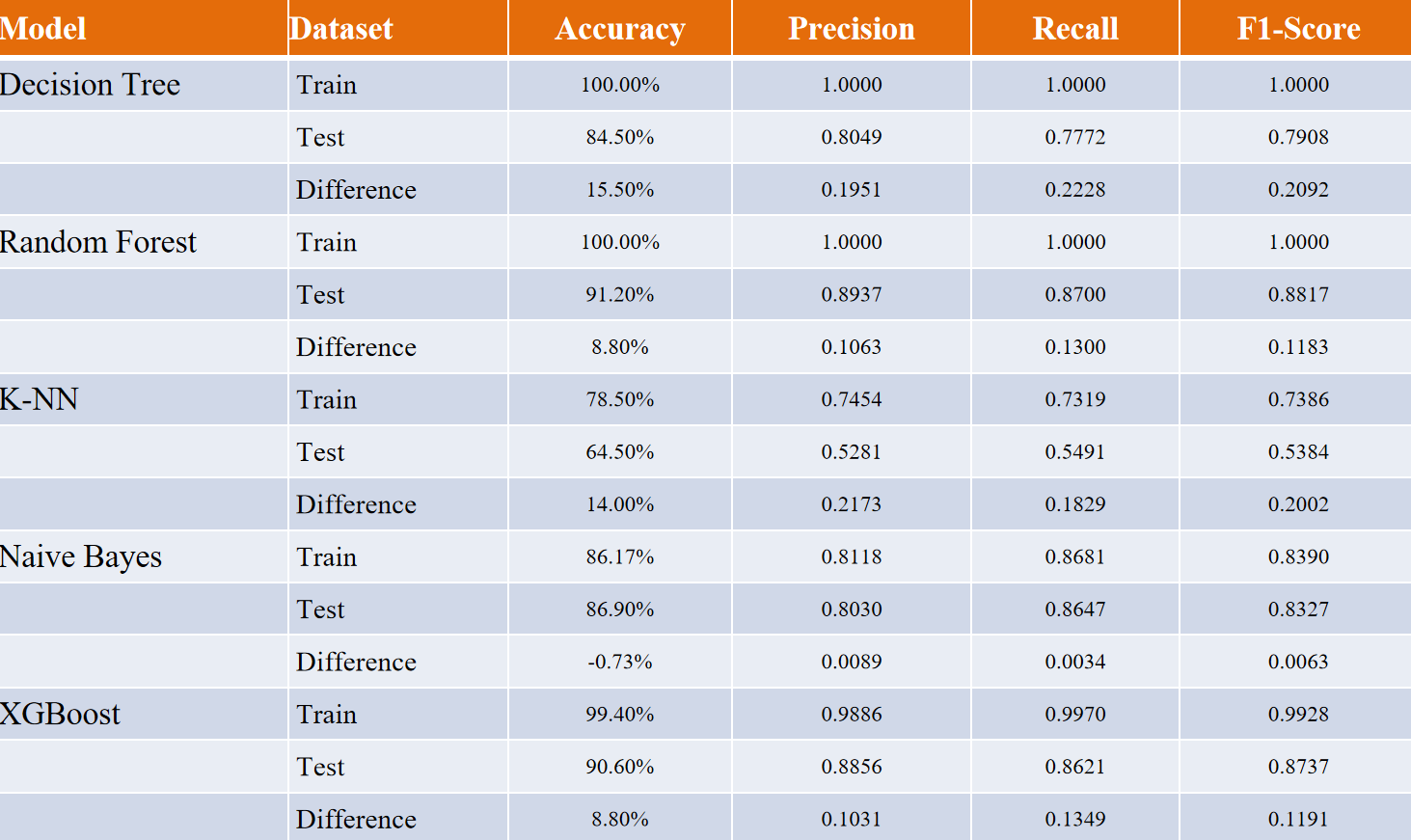
Hình 3.13: Quan hệ giữa gia đình và ung thư phổi

1. Ma trận tương quan

Hình 3.14: Ma trận tương quan

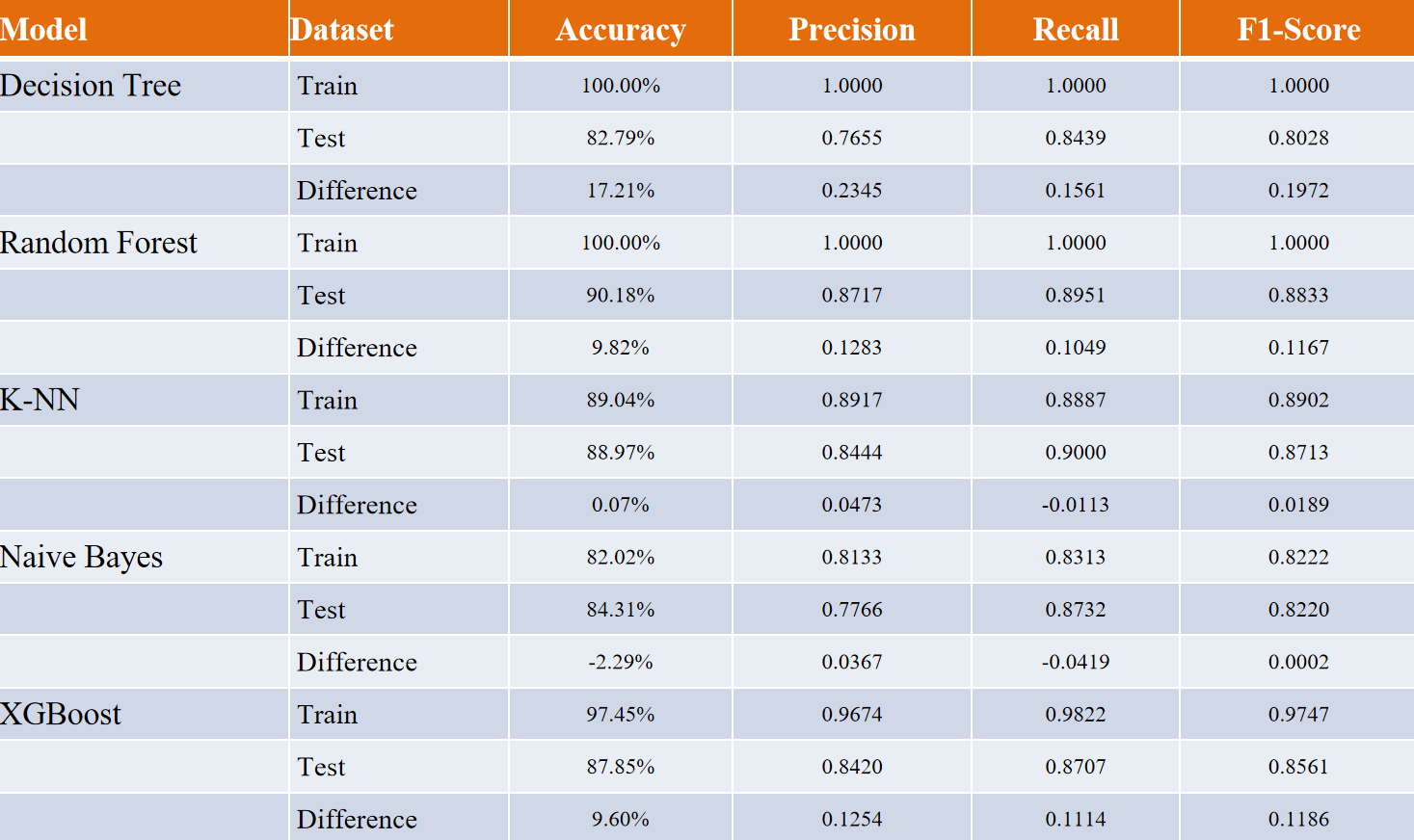
## Đánh giá mô hình

Lần 1: (chưa tiền xử lý, tham số mặc định)



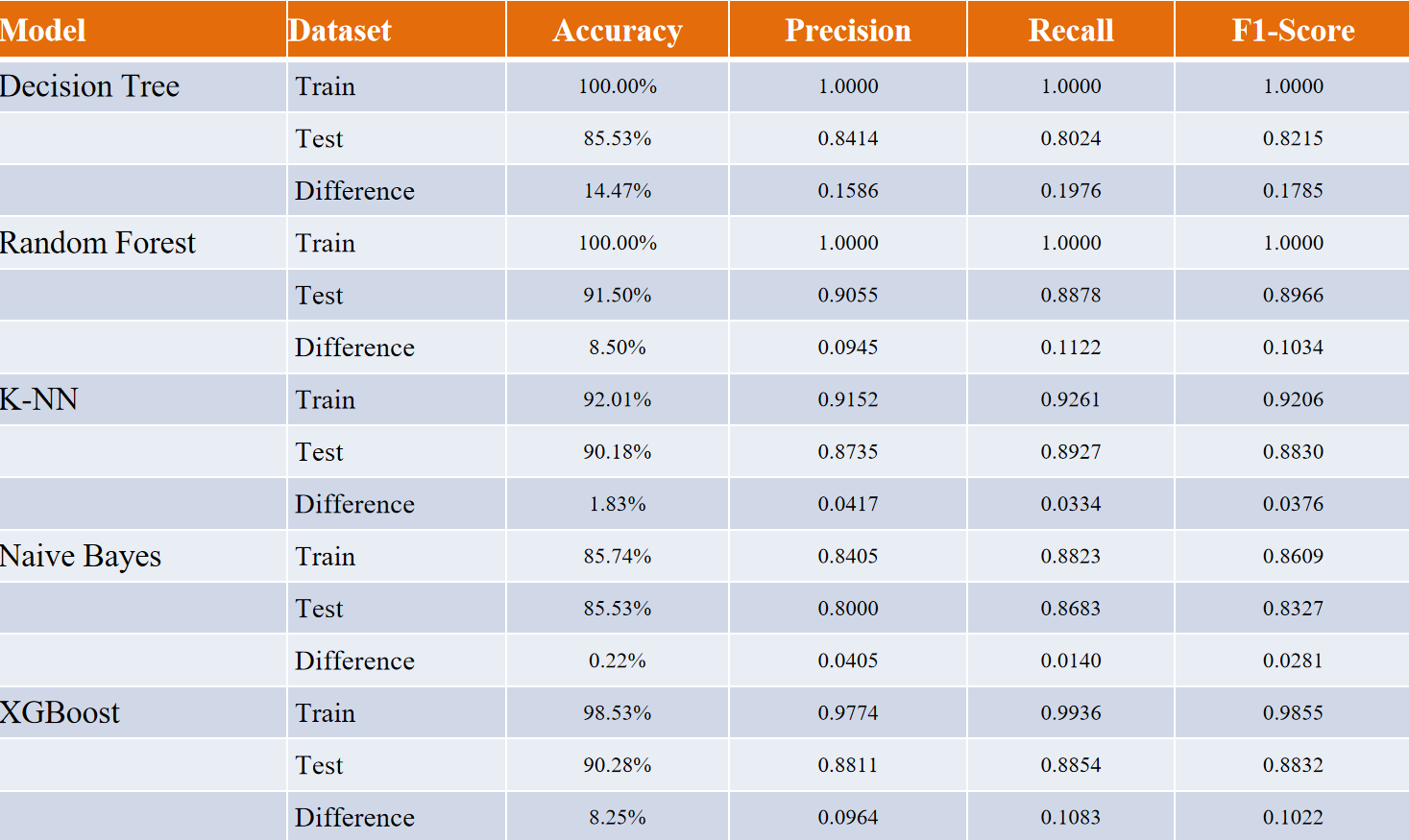
Hình 3.15: Lần 1: (chưa tiền xử lý, tham số mặc định)

Lần 2: (sau tiền xử lý, SMOTE)



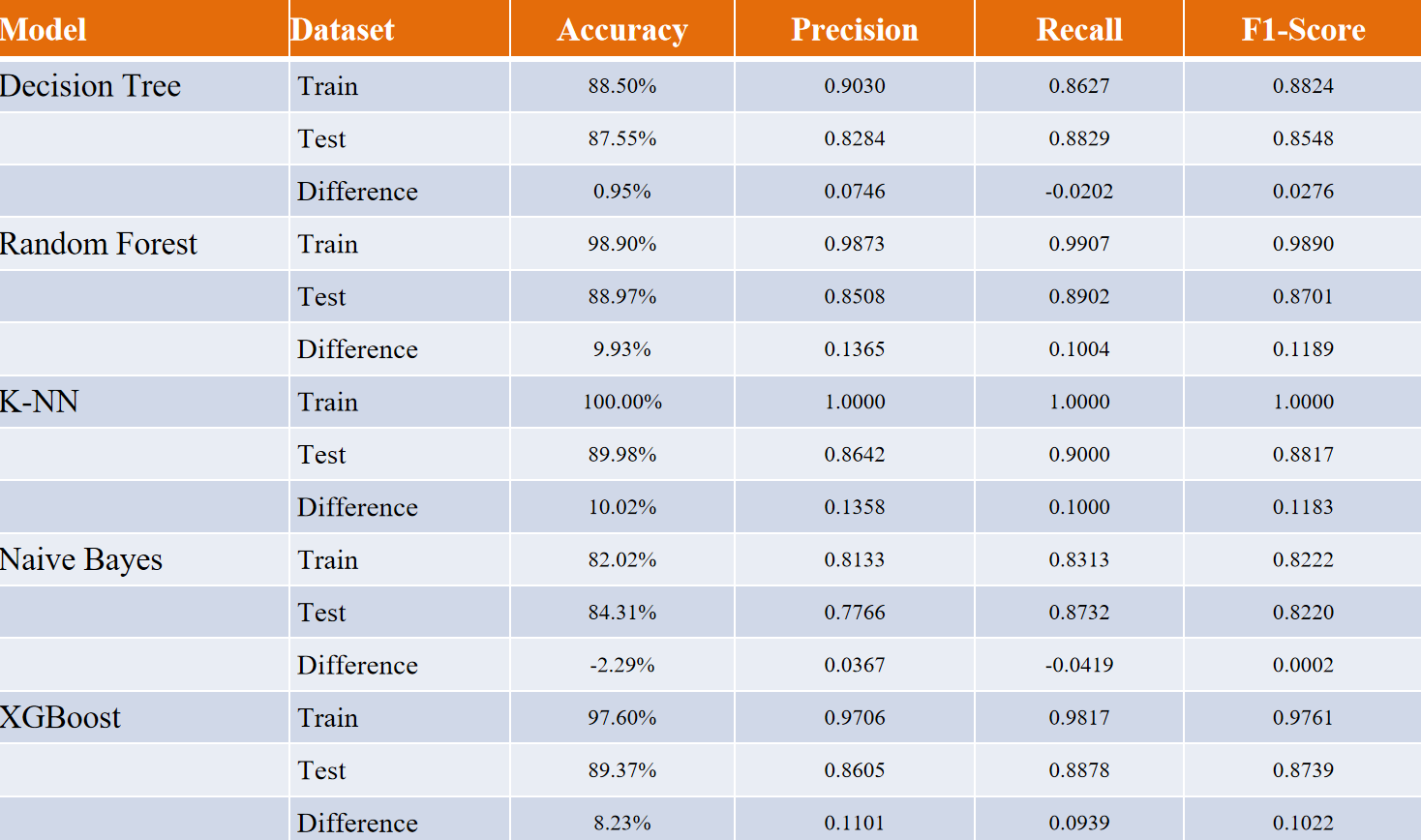
Hình 3.16: Lần 2: (sau tiền xử lý, SMOTE)

Lần 2: (sau tiền xử lý, ROS)



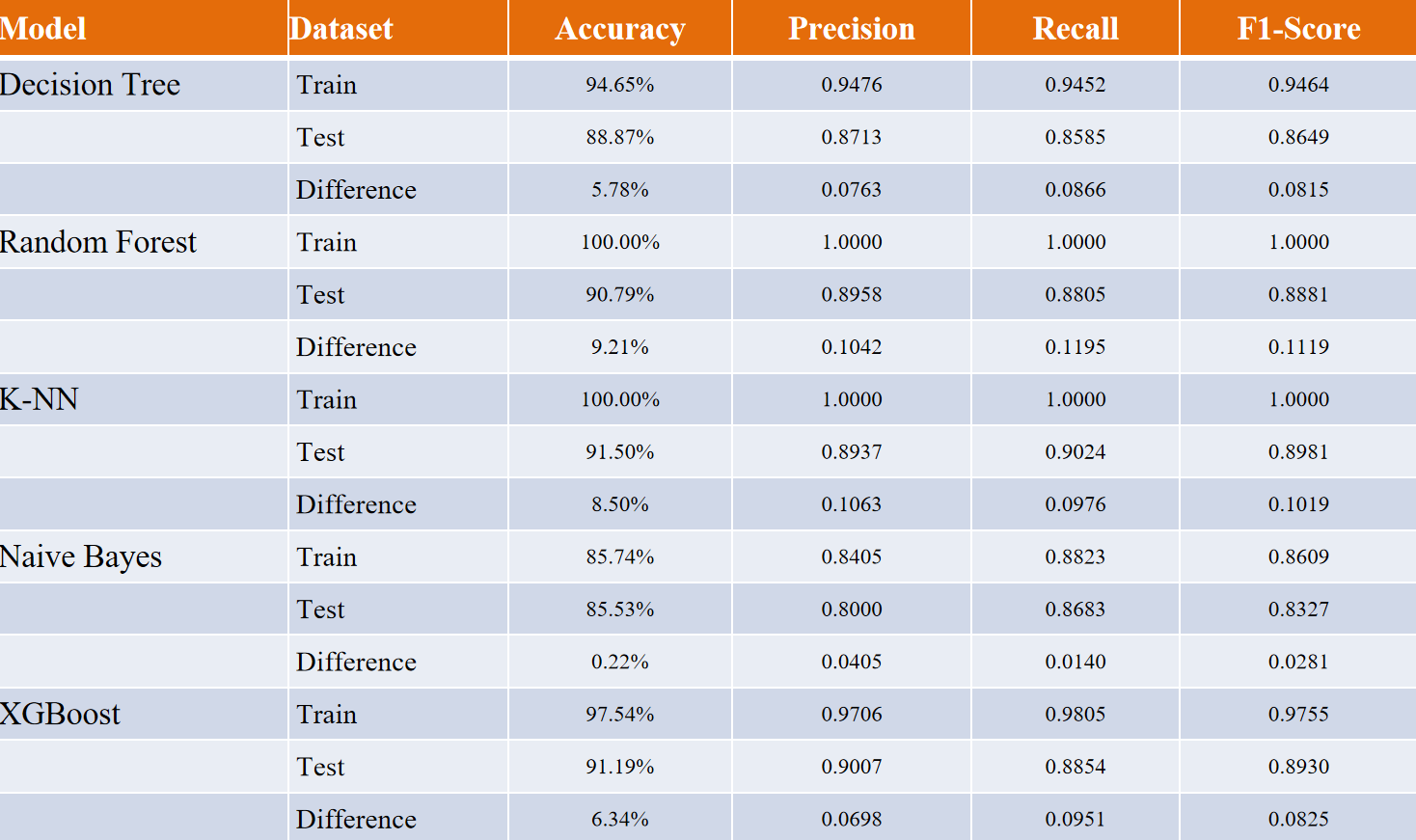
Hình 3.17: Lần 2: (sau tiền xử lý, ROS)

Lần 3: (Tùy chỉnh tham số, SMOTE)



Hình 3.18: Lần 3: (Tùy chỉnh tham số, SMOTE)

Lần 3: (Tùy chỉnh tham số, ROS)



Hình 3.19: Lần 3: (Tùy chỉnh tham số, ROS)

# KẾT LUẬN

1. **Thành tựu:**

* Dự án đã đạt được những kết quả đáng kể trong việc phân loại nguy cơ ung thư phổi bằng cách sử dụng các mô hình học máy như Random Forest và XGBoost. Các mô hình này thể hiện hiệu quả cao trong việc phân tích các đặc trưng như tuổi, thói quen hút thuốc, tiếp xúc với ô nhiễm, độ bão hòa oxy, và các triệu chứng như khó thở hoặc tức ngực, từ đó dự đoán chính xác khả năng mắc bệnh phổi.
* Khả năng xử lý các bộ dữ liệu lớn với nhiều đặc trưng đa dạng (bao gồm yếu tố môi trường, lối sống, và tiền sử gia đình) đã cung cấp những hiểu biết sâu sắc, hỗ trợ xây dựng các chiến lược phòng ngừa và điều trị hiệu quả hơn.

1. **Hạn chế:**

* **Hạn chế dữ liệu:** Tập dữ liệu có thể thiếu thông tin chi tiết về các yếu tố khác như mức độ tiếp xúc với ô nhiễm (ví dụ: nồng độ PM2.5, PM10) hoặc các yếu tố di truyền cụ thể, đặc biệt ở các khu vực có hệ thống thu thập dữ liệu y tế kém phát triển. Ngoài ra, dữ liệu có thể không đại diện cho toàn bộ dân số do thiếu sự đa dạng về địa lý hoặc nhân khẩu học.
* **Khả năng tổng quát hóa:** Các mô hình được huấn luyện trên dữ liệu từ một nhóm đối tượng cụ thể có thể không đạt hiệu quả cao khi áp dụng cho các nhóm khác với đặc điểm lối sống, môi trường, hoặc yếu tố di truyền khác biệt.

1. **Hướng phát triển:**

* **Cải thiện dữ liệu:** Cần thu thập và tích hợp các bộ dữ liệu đa dạng hơnbao gồm dữ liệu thời gian thực từ các thiết bị y tế (như máy đo độ bão hòa oxy), thông tin về mức độ ô nhiễm môi trường chi tiết hơn, và dữ liệu di truyền dài hạn để nâng cao tính toàn diện và khả năng áp dụng của mô hình.
* **Tăng độ chính xác:** Áp dụng các kỹ thuật học máy tiên tiến như học sâu (deep learning) để nâng cao độ chính xác trong việc phát hiện sớm các dấu hiệu ung thư phổi, đặc biệt ở những trường hợp có triệu chứng không rõ ràng.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Dataset link: *https://www.kaggle.com/datasets/shantanugarg274/lung-cancer-prediction-dataset* |
| [2] | Scikit-learn Developers. Scikit-learn: Machine Learning in Python |
| [3] | P. Vinothini and P. Rajalakshmi, “Predicting Lung Cancer using Machine Learning Algorithms,” International Journal of Scientific & Engineering Research |
| [4] | “Lung cancer detection using machine learning: A review,” Computer Methods and Programs in Biomedicine, vol. 208, p. 106288, 2021 |