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 9
1.1 Bài toán 9
1.2 Trình bày dữ liệu bài toán 9
1.3 Tiền xử lý dữ liệu 12
1.4 Trực quan hoá dữ liệu 12
1.5 Xây dựng mô hình học máy 12
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 13
2.1 Decision Tree 13
2.1.1 Giới thiệu thuật toán Decision Tree 13
2.1.2 Cấu trúc của mô hình Decision Tree 13
2.1.3 Quá trình xây dựng mô hình Decision Tree 13
2.2 Random Forest 14
2.2.1 Giới thiệu thuật toán Random Forest 14
2.2.2 Cấu trúc của mô hình Random Forest 15
2.2.3 Quá trình xây dựng mô hình Random Forest 15
2.3 K-Nearest Neighbors 16
2.3.1 Giới thiệu thuật toán K-Nearest Neighbors 16
2.3.2 Cấu trúc của mô hình K-Nearest Neighbors 16
2.3.3 Quá trình xây dựng mô hình K-Nearest Neighbors 17
2.4 Naive Bayes 18
2.4.1 Giới thiệu thuật toán Naive Bayes 18
2.4.2 Cấu trúc của mô hình Naive Bayes 18
2.4.3 Quá trình xây dựng mô hình Naive Bayes 19
2.5 XGBoost 20

2.5.1 Giới thiệu thuật toán XGBoost 20
2.5.2 Cấu trúc của mô hình XGBoost 20
2.5.3 Quá trình xây dựng mô hình XGBoost 20
CHƯƠNG 3: GIẢI PHÁP 22
3.1. Tiền xử lý dữ liệu 22
3.2. Trực quan hóa dữ liệu 28
3.3. Đánh giá mô hình 35
KẾT LUẬN 45
TÀI LIỆU THAM KHẢO 46

DANH MỤC CÁC HÌNH VẼ

Hình 3.1: Xử lý giá trị ngoại lai 23

Hình 3.2: Feature selection 24

Hình 3.3: Data Split 25

Hình 3.4: Cân bằng dữ liệu 26

Hình 3.5: Scale dữ liệu 27

Hình 3.6: Phân phối của nhãn dữ liệu 28

Hình 3.7: Phát hiện ngoại lai 29

Hình 3.8: Phân bố tuổi 30

Hình 3.9: Phân bố giới tính 31

Hình 3.10: Quan hệ giữa hút thuốc và ung thư phổi 32

Hình 3.11: Quan hệ giữa gia đình và ung thư phổi 33

Hình 3.12: Ma trận tương quan 34

DANH MỤC BẢNG BIỂU

Bảng 1.1 Tổng quan về bộ dữ liệu 10

Bảng 3.1 Xử lý giá trị khuyết thiếu, trùng lặp 22

Bảng 3.2 Danh sách models 35

Bảng 3.3 Lần 1: (chưa tiền xử lý, tham số mặc định) 36

Bảng 3.4 Lần 2: (sau tiền xử lý, SMOTE) 38

Bảng 3.5 Lần 2: (sau tiền xử lý, RandomOverSampler) 39

Bảng 3.6 Lần 3: (best parameters, Smote) 41

Bảng 3.7 Lần 3: (tùy chỉnh tham số, SMOTE) 42

Bảng 3.8 Lần 3: (best parameters, RandomOverSampler) 43

Bảng 3.9 Lần 3: (tùy chỉnh tham số, RandomOverSampler) 44

CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN

1.1 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.

1.2 Trình bày dữ liệu bài toán

Link dữ liệu trên Kaggle:

Lung Cancer Dataset | Kaggle

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

IMMUNE WEAKNESS	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

Bảng 1.1 Tổng quan về bộ dữ liệu

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:

1.3 Tiền xử lý dữ liệu

a) Xử lý giá trị khuyết thiếu, trùng lặp
b) Xử lý giá trị ngoại lai
c) Feature selection
d) Phân tách dữ liệu
e) Xử lý mất cân bằng nhãn
f) Xử lý chuẩn hóa dữ liệu

1.4 Trực quan hoá dữ liệu

a) Phân bố nhãn
b) Phát hiện ngoại lai
c) Phân bố tuổi
d) Phân bố giới tính
e) Quan hệ giữa hút thuốc và ung thư phổi
f) Quan hệ giữa gia đình và ung thư phổi
g) Ma trận tương quan

1.5 Xây dựng mô hình học máy

a) Xây dựng mô hình Decision Tree
b) Xây dựng mô hình Random Forest
c) Xây dựng mô hình KNN
d) Xây dựng mô hình Naive Bayes
e) Xây dựng mô hình XGBoost

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Decision Tree

2.1.1 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).

2.1.2 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.

2.1.3 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:

$$\text{Gini} = 1 - \sum_{i=1}^{C} p_i^2 \text{D}$$

Entropy:

$$\text{Entropy} = -\sum_{i=1}^{C} p_i \log_2 (p_i)$$

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.

2.2 Random Forest

2.2.1 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.

2.2.2 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):

$$\hat{y} = \text{mode}\{\widehat{y_1}, \widehat{y_2}, ..., \widehat{y_T}\}$$

Trong đó:

- $\widehat{y_i}$:Dự đoán từ cây thứ i.
- T:Số lượng cây trong rừng.

2.2.3 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 $\left(\text{thường là } \sqrt{p} \text{ hoặc } \log_2(p), \text{ với } p \text{ là số đặc trưng.}\right)$
- Tìm đặc trưng và ngưỡng tối ưu để phân tách dựa trên chỉ số như Gini Index hoặc Entropy:

$$\text{Gini} = 1 - \sum_{i=1}^{C} p_i^2$$

$$\text{Entropy} = -\sum_{i=1}^{C} p_i \log_2(p_i)$$

$Trop_i$ là tỷ lệ mẫu thuộc lớp i tại nút.

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.

2.3 K-Nearest Neighbors

2.3.1 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.

2.3.2 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):

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

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

2.3.3 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.

2.4 Naive Bayes

2.4.1 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).

2.4.2 Cấu trúc của mô hình Naive Bayes

Dựa trên Định lý Bayes:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}$$

- 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:

$$P(X|C) = \prod_{i=1}^{n} P\left(X_i|C\right)$$

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

$$\widehat{C} = arg\max_{C} P\left(C\right) \prod_{i=1}^{n} P\left(X_i|C\right)$$

2.4.3 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.

2.5 XGBoost

2.5.1 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.

2.5.2 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:

$$L = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i\right) + \sum_{k=1}^{K} \Omega\left(f_k\right)$$

Trong đó:

- $l(y_i, \hat{y}_i)$: Hàm mất mát (ví dụ: log loss cho phân loại, mean squared error cho hồi quy).
- $\Omega(f_k)$: 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).

2.5.3 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.

CHƯƠNG 3: GIẢI PHÁP

3.1. Tiền xử lý dữ liệu

a) Xử lý giá trị khuyết thiếu, trùng lặp

Để đánh giá chất lượng dữ liệu ban đầu, tôi sử dụng hàm isnull() kết hợp với sum() trong thư viện pandas nhằm thống kê số lượng giá trị khuyết thiếu trên từng cột dữ liệu. Tôi cũng sử dụng hàm duplicated() để kiểm tra sự xuất hiện của các dòng dữ liệu bị lặp lại trong tập dữ liệu.

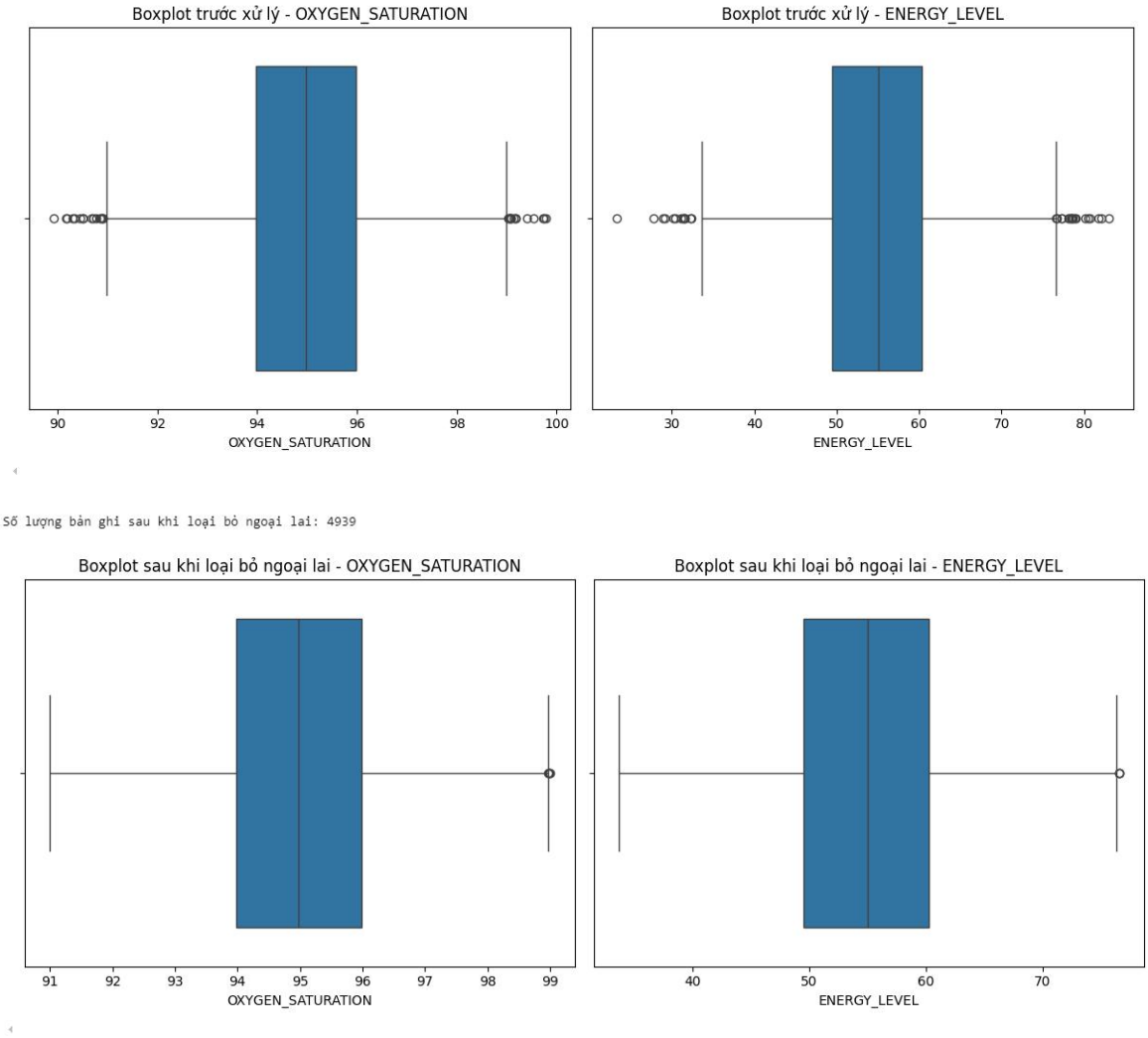
=> Kết quả là không có dữ liệu khuyết thiếu, trùng lặp

Tên Cột	Số Giá Trị Thiếu	Phần Trăm Thiếu (%)
AGE	0	0.0
ALCOHOL_CONSUMPTION	0	0.0
PULMONARY_DISEASE	0	0.0
STRESS_IMMUNE	0	0.0
SMOKING_FAMILY_HISTORY	0	0.0
FAMILY_HISTORY	0	0.0
CHEST_TIGHTNESS	0	0.0
OXYGEN_SATURATION	0	0.0
THROAT_DISCOMFORT	0	0.0
BREATHING_ISSUE	0	0.0
GENDER	0	0.0
IMMUNE_WEAKNESS	0	0.0
ENERGY_LEVEL	0	0.0
LONG_TERM_ILLNESS	0	0.0
EXPOSURE_TO_POLLUTION	0	0.0
MENTAL_STRESS	0	0.0
FINGER_DISCOLORATION	0	0.0
SMOKING	0	0.0

Bảng 3.1 Xử lý giá trị khuyết thiếu, trùng lặp

b) Xử lý giá trị ngoại lai

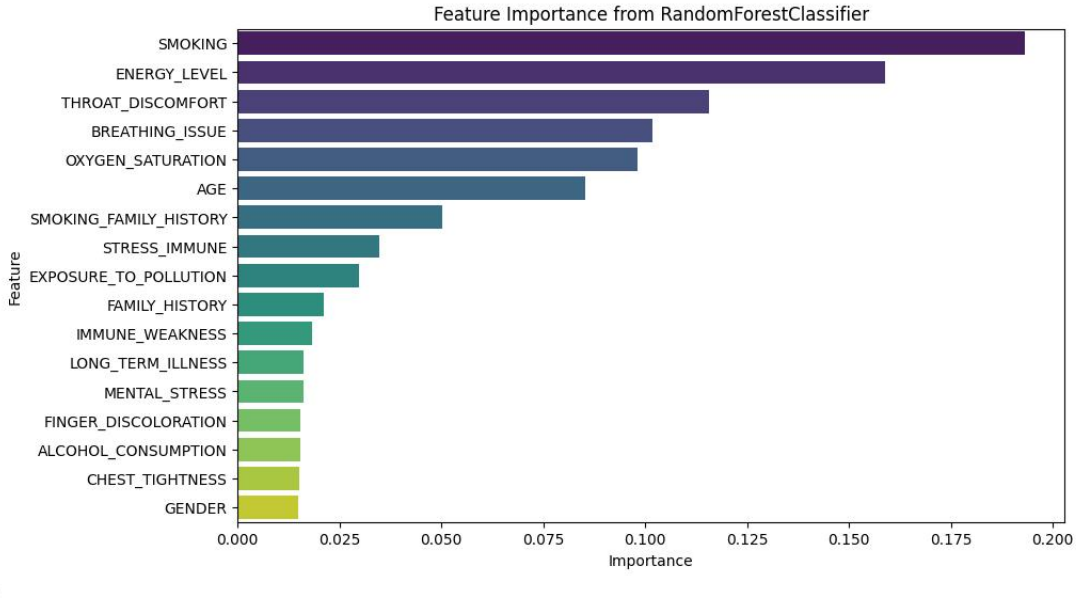
Trong quá trình tiền xử lý dữ liệu, tôi tiến hành kiểm tra và xử lý các giá trị ngoại lai nhằm đảm bảo độ tin cậy và tính đại diện của tập dữ liệu. Việc tồn tại các giá trị ngoại lai có thể gây nhiễu cho quá trình phân tích thống kê và huấn luyện mô hình, đặc biệt là trong các thuật toán nhạy cảm với khoảng cách như KNN.



Hình 3.1: Xử lý giá trị ngoại lai

c) **Feature selection**

Để cải thiện hiệu suất mô hình và giảm độ phức tạp tính toán, tôi thực hiện chọn đặc trưng (feature selection) nhằm xác định các biến đầu vào có mức độ ảnh hưởng lớn đến biến mục tiêu. Trong bước này, tôi sử dụng Random Forest — một thuật toán cây quyết định dạng ensemble, vốn có khả năng đánh giá mức độ quan trọng của từng đặc trưng.



Hình 3.2: Feature selection

d) **Data Split**

Sau khi loại bỏ một số cột không cần thiết thông qua columns_to_drop, tôi tiến hành tách biến đầu vào (X) và biến mục tiêu (y). Cụ thể:

Biến mục tiêu: PULMONARY_DISEASE

Biến đầu vào còn lại gồm các đặc trưng như: AGE, SMOKING, EXPOSURE_TO_POLLUTION, ENERGY_LEVEL, BREATHING_ISSUE, THROAT_DISCOMFORT,OXYGEN_SATURATION,SMOKING_FAMILY_HISTORY, STRESS_IMMUNE

Sau đó, tôi sử dụng hàm train_test_split từ thư viện sklearn.model_selection để chia tập dữ liệu thành 2 phần: 80% cho tập huấn luyện (X_train, y_train) 20% cho tập kiểm tra (X_test, y_test).

```
columns_to_drop = [
    "PULMONARY_DISEASE",
    "GENDER",
    "CHEST_TIGHTNESS",
    "ALCOHOL_CONSUMPTION",
    "FINGER_DISCOLORATION",
    "MENTAL_STRESS",
    "LONG_TERM_ILLNESS",
    "IMMUNE_WEAKNESS",
    "FAMILY_HISTORY"
]
X = df.drop(columns=columns_to_drop)
y = df["PULMONARY_DISEASE"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
X_train.head()
```

	AGE	SMOKING	EXPOSURE_TO_POLLUTION	ENERGY_LEVEL	BREATHING_ISSUE	THROAT_DISCOMFORT	OXYGEN_SATURATION	SMOKING_FAMILY_HISTORY	STRESS_IMMUNE
2265	35	1	1	68.442800	1	1	93.510972	1	0
1130	66	1	0	57.370063	1	1	94.843254	0	0
4680	57	0	0	50.147466	1	0	95.858824	0	0
1974	82	0	0	34.225550	1	1	94.188979	0	1
1567	52	1	0	55.064677	1	1	95.530578	0	0

Hình 3.3: Data Split

e) Cân bằng dữ liệu

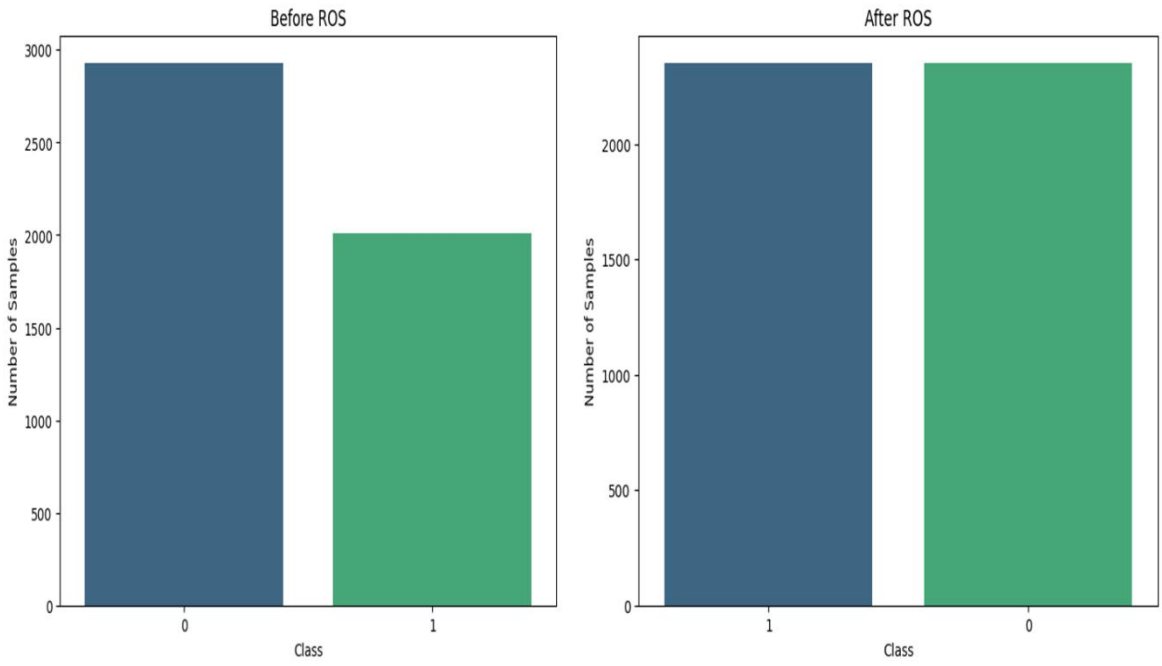
Trong các bài toán phân loại, đặc biệt là trong lĩnh vực y tế như chẩn đoán bệnh, tập dữ liệu thường có hiện tượng mất cân bằng lớp (class imbalance) – tức là số lượng mẫu thuộc một lớp (ví dụ: không mắc bệnh) chiếm tỷ lệ áp đảo so với lớp còn lại (mắc bệnh). Việc này có thể dẫn đến mô hình thiên lệch, chỉ dự đoán tốt lớp chiếm đa số và bỏ qua lớp thiểu số, gây sai lệch nghiêm trọng trong thực tế.

Qua phân tích biến mục tiêu PULMONARY_DISEASE, tôi nhận thấy hiện tượng mất cân bằng dữ liệu giữa hai lớp là 59,3% ở lớp 0 và 40.7% ở lớp 1.

Phương pháp xử lý:

Tôi đã áp dụng cả phương pháp SMOTE (Synthetic Minority Oversampling Technique) để tăng số lượng mẫu của lớp thiểu số bằng cách sinh ra các mẫu mới nhân tạo dựa trên các điểm lân cận trong không gian đặc trưng và Random OverSampling, trong đó các mẫu thuộc lớp thiểu số sẽ được nhân bản ngẫu nhiên cho đến khi đạt số lượng tương đương với lớp chiếm đa số.

=> Kết quả:



Hình 3.4: Cân bằng dữ liệu

f) Scale dữ liệu

Trong tập dữ liệu ban đầu, các đặc trưng số học như AGE, ENERGY_LEVEL, OXYGEN_SATURATION có đơn vị đo lường khác nhau. Việc để nguyên các giá trị này có thể dẫn đến mô hình thiên lệch về phía các đặc trưng có giá trị lớn hơn, làm giảm hiệu quả học của mô hình.

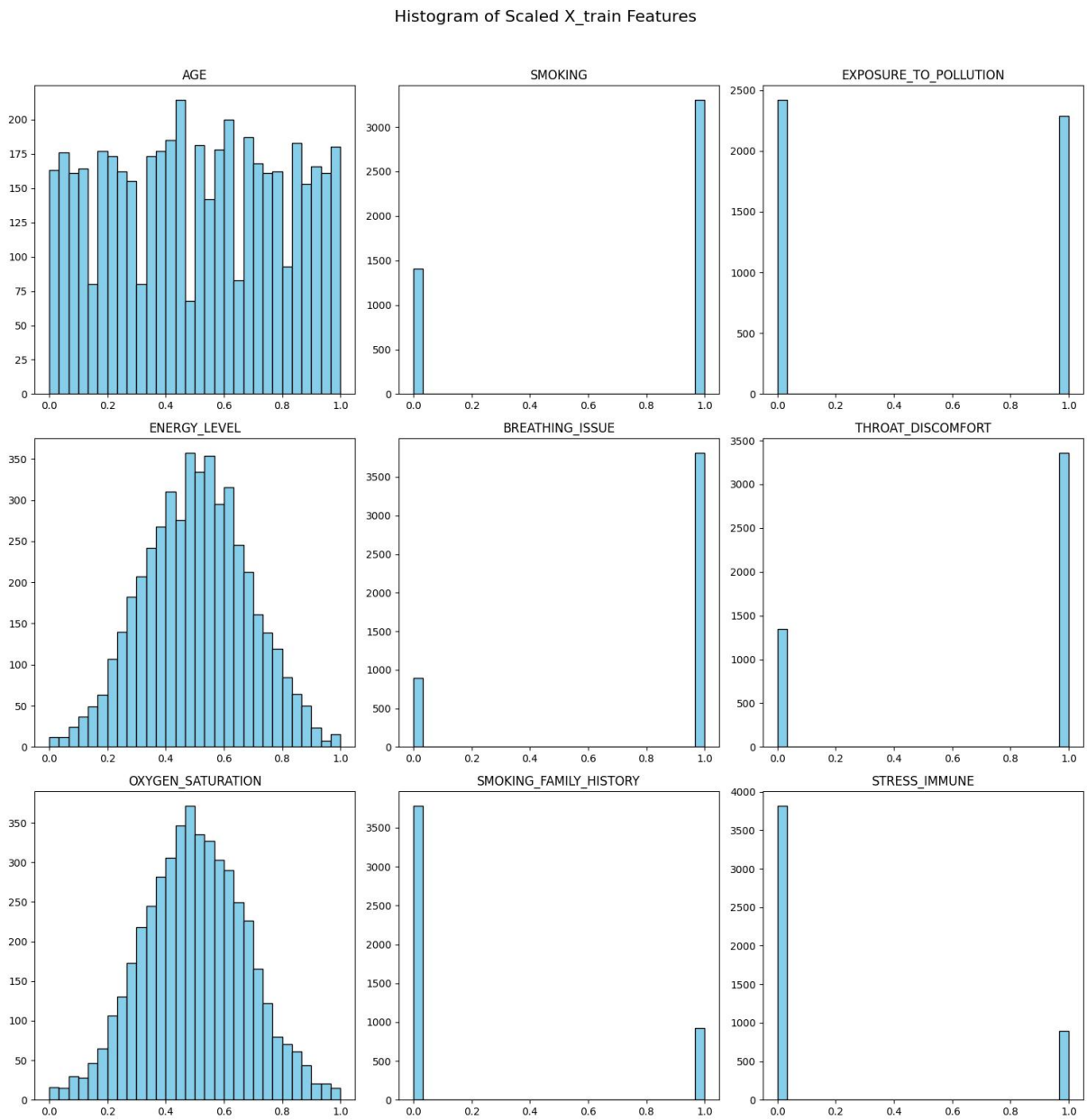
Giải pháp: Min-Max Scaling

Tôi sử dụng phương pháp Min-Max Scaling thông qua MinMaxScaler từ thư viện sklearn.preprocessing để đưa tất cả các giá trị về trong khoảng [0, 1].

Lợi ích của Min-Max Scaling:

- Phù hợp với các thuật toán dựa trên khoảng cách như K-Nearest Neighbors (KNN).
- Bảo toàn phân bố ban đầu của dữ liệu (không làm thay đổi hình dạng phân phối như StandardScaler).
- Giúp mô hình hội tụ nhanh hơn và ổn định hơn.

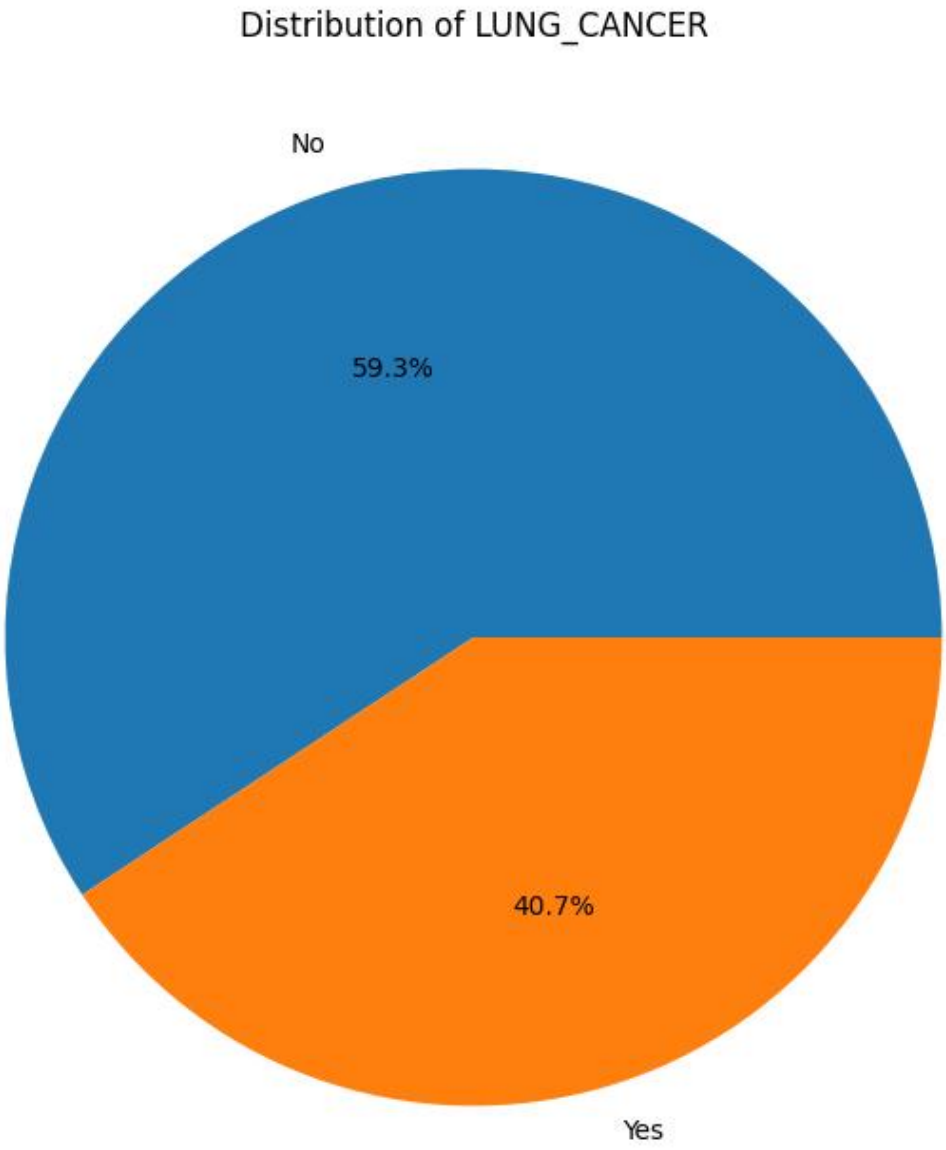
=> Kết quả:



Hình 3.5: Scale dữ liệu

3.2. Trực quan hóa dữ liệu

1. Phân phối nhãn dữ liệu

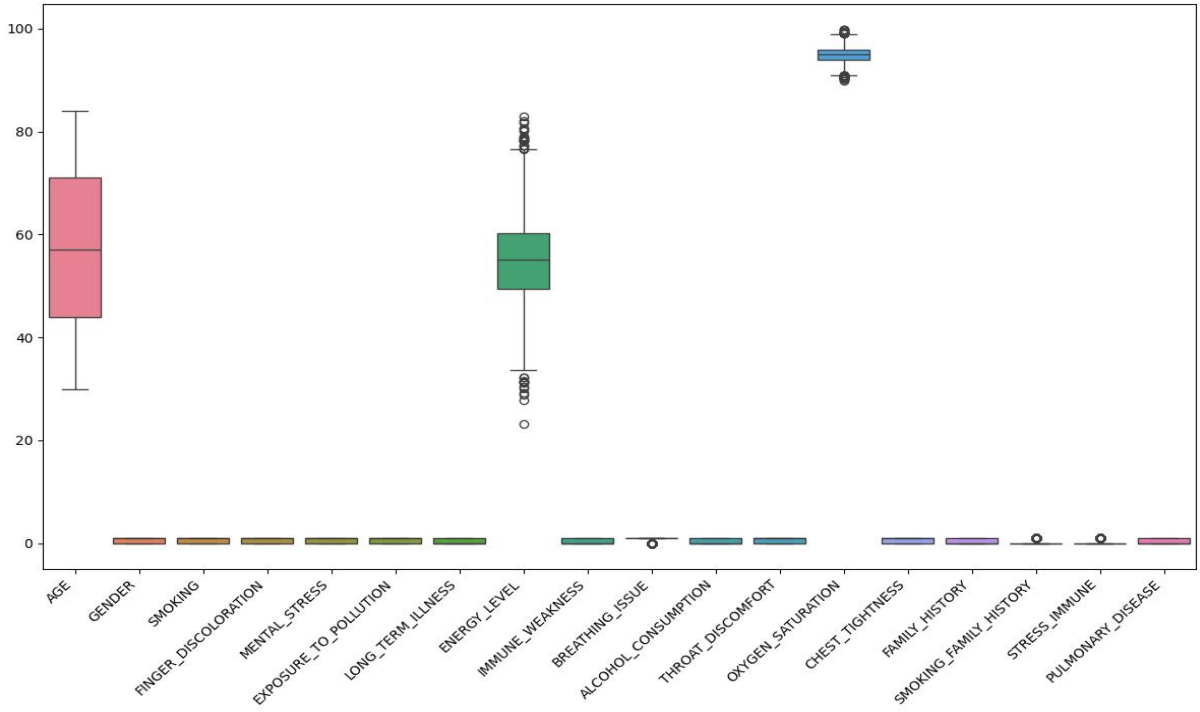


Hình 3.6: Phân phối của nhãn dữ liệu

Ý nghĩa:

Sự phân bố này cho thấy mức độ cân bằng hoặc không cân bằng giữa hai nhóm, cung cấp cơ sở để đánh giá tác động của các yếu tố như hút thuốc (SMOKING), tiếp xúc với ô nhiễm (EXPOSURE_TO_POLLUTION), và các chỉ số sức khỏe như độ bão hòa oxy (OXYGEN_SATURATION) hoặc mức năng lượng (ENERGY_LEVEL). Nếu tỷ lệ "YES" cao bất thường, điều này nhấn mạnh tầm quan trọng của việc phòng ngừa và nhận thức về các yếu tố nguy cơ liên quan đến sức khỏe phổi.

2. Phát hiện ngoại lai



Hình 3.7: Phát hiện ngoại lai

Ý nghĩa:

Nhằm phát hiện ra những giá trị ngoại lai có thể làm sai lệch các chỉ số thống kê như trung bình, độ lệch chuẩn, và ảnh hưởng đến mô hình dự đoán bệnh phổi. Ví dụ, một giá trị OXYGEN_SATURATION bất thường có thể che giấu mối quan hệ thực sự giữa độ bão hòa oxy và tình trạng bệnh.

3. Phân bố tuổi

Hình 3.8: Phân bố tuổi

Ý nghĩa:

Mối liên hệ với bệnh phổi: Tuổi cao hơn thường liên quan đến nguy cơ mắc bệnh phổi cao hơn do tích lũy các yếu tố nguy cơ như hút thuốc (SMOKING), tiếp xúc lâu dài với ô nhiễm (EXPOSURE_TO_POLLUTION), và suy giảm chức năng phổi tự nhiên. Sự tập trung vào nhóm tuổi trung niên và cao tuổi trong dữ liệu nhấn mạnh tầm quan trọng của việc theo dõi sức khỏe phổi ở các nhóm này.

Tác động đến phân tích: Phân bố tuổi ảnh hưởng đến việc đánh giá các yếu tố nguy cơ khác. Ví dụ, người lớn tuổi có thể có tỷ lệ "YES" cao hơn trong cột PULMONARY_DISEASE, đòi hỏi phân tích phân tầng theo tuổi để xác định tác động riêng lẻ của các biến khác.

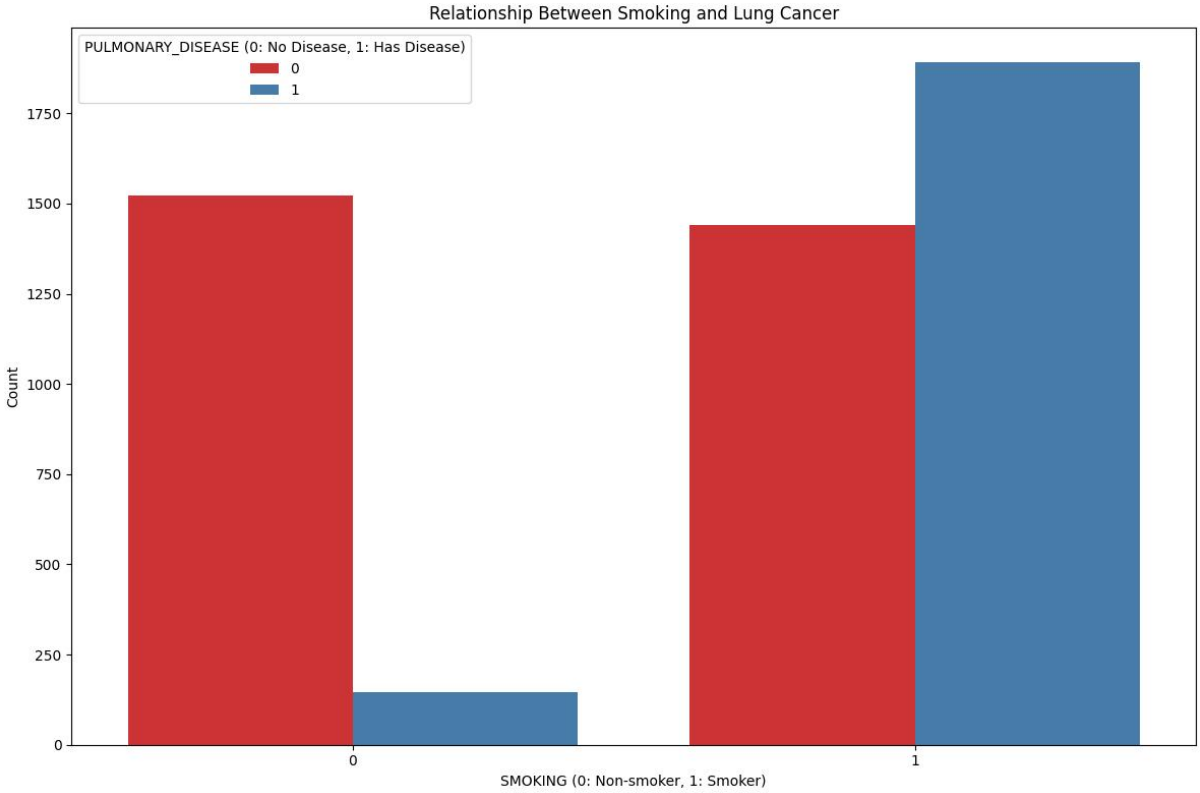
4. Phân bố giới tính

Hình 3.9: Phân bố giới tính

Ý nghĩa:

Mối liên hệ với bệnh phổi: Giới tính có thể ảnh hưởng đến nguy cơ mắc bệnh phổi do sự khác biệt về hành vi (ví dụ, nam giới có thể hút thuốc nhiều hơn) hoặc yếu tố sinh học (ví dụ, sự khác biệt về cấu trúc phổi hoặc phản ứng miễn dịch). Nếu tỷ lệ "YES" trong cột PULMONARY_DISEASE cao hơn ở một giới, điều này gợi ý cần điều tra thêm về các yếu tố liên quan.

5. Quan hệ giữa hút thuốc và ung thư phổi

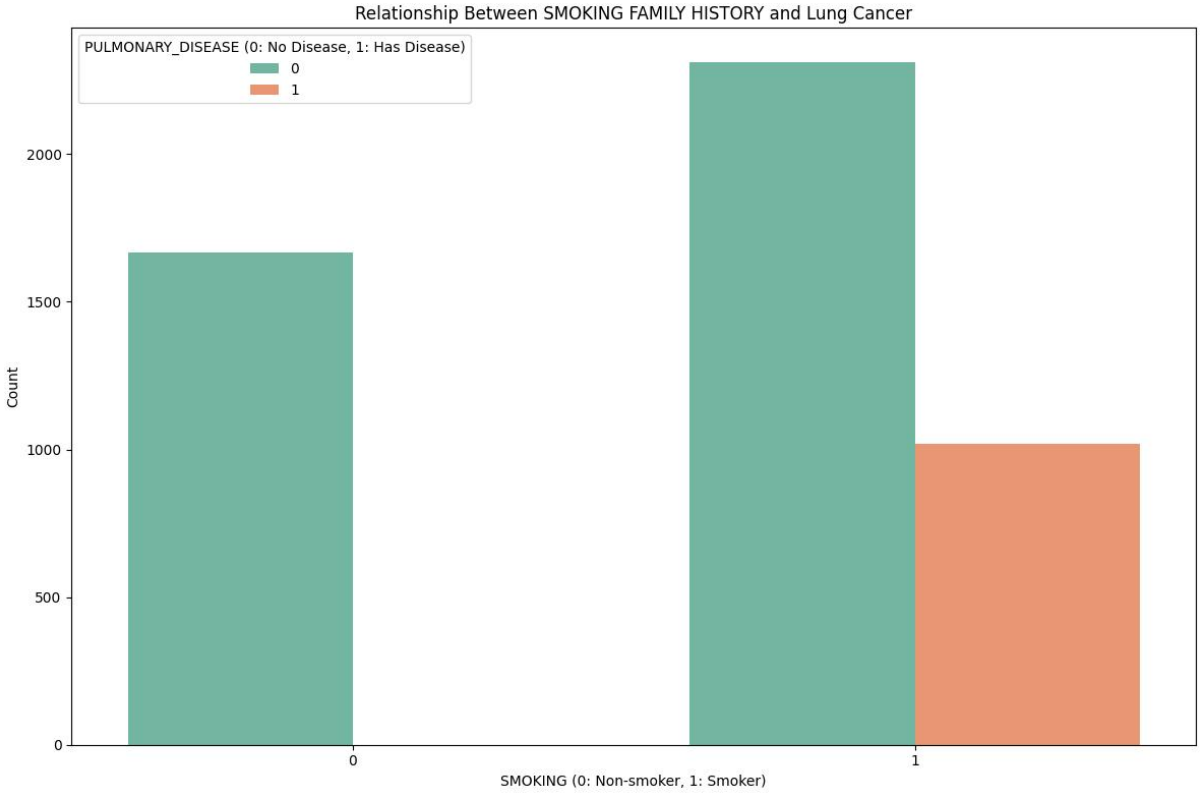


Hình 3.10: Quan hệ giữa hút thuốc và ung thư phổi

Ý nghĩa:

Mối liên hệ tiềm tàng: Hút thuốc là nguyên nhân chính gây ung thư phổi do các chất độc hại trong khói thuốc (như hydrocarbon thơm đa vòng, nitrosamine) làm tổn thương DNA và gây đột biến ở tế bào phổi. Dữ liệu này có thể phản ánh xu hướng tương tự, với nhóm hút thuốc cho thấy nguy cơ cao hơn.

6. Quan hệ giữa gia đình và ung thư phổi



Hình 3.11: Quan hệ giữa gia đình và ung thư phổi

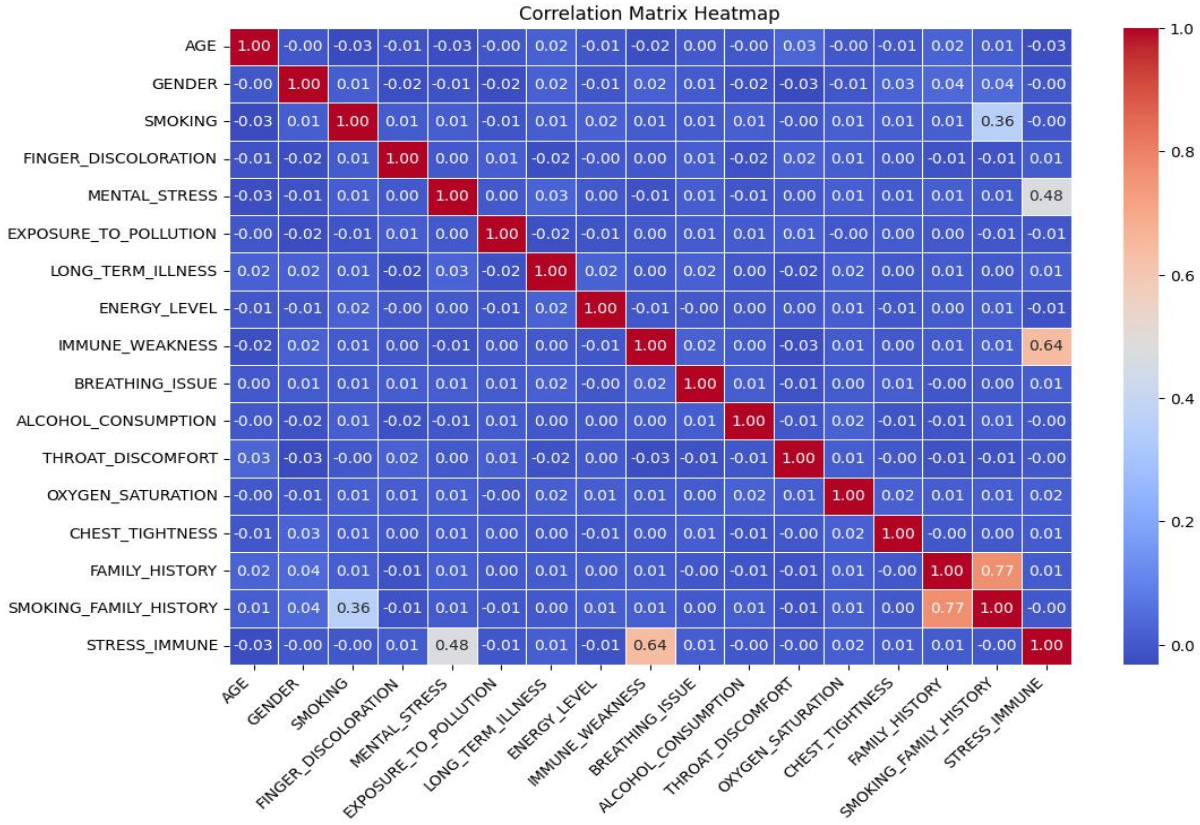
Ý nghĩa:

Mối liên hệ tiềm tàng:

Di truyền: Tiền sử gia đình mắc bệnh phổi có thể chỉ ra khuynh hướng di truyền, như đột biến ở các gene (ví dụ, EGFR, KRAS) liên quan đến ung thư phổi, làm tăng nguy cơ ngay cả ở những người không hút thuốc.

Môi trường gia đình: Tiếp xúc thụ động với khói thuốc từ thành viên gia đình hút thuốc có thể gây tổn thương phổi lâu dài, góp phần vào nguy cơ mắc bệnh.

7. Ma trận tương quan



Hình 3.12: Ma trận tương quan

Ý nghĩa:

Mối liên hệ chính: Hút thuốc (SMOKING) và độ bão hòa oxy (OXYGEN_SATURATION) cho thấy mối quan hệ mạnh nhất với PULMONARY_DISEASE, nhấn mạnh vai trò của yếu tố lối sống và chỉ số sinh lý trong nguy cơ ung thư phổi.

Tác động đến phân tích: Các biến có tương quan cao (ví dụ, > 0.7 hoặc < -0.7) với nhau (như SMOKING và SMOKING_FAMILY_HISTORY) có thể gây đa cộng tuyến, cần thận trọng khi xây dựng mô hình dự đoán.

3.3. Đánh giá mô hình

Tôi sử dụng 5 mô hình đó là:

Models	Decision Tree
	Random Forest
	K-NN
	Naive Bayes
	XGBoost

Bảng 3.2 Danh sách models

Quy Trình Huấn Luyện, Đánh Giá Mô Hình Dự Đoán Bệnh Phổi

Mục tiêu: Xây dựng và đánh giá mô hình dự đoán ung thư phổi (dựa trên cột PULMONARY_DISEASE) từ tập dữ liệu "Lung Cancer Dataset.csv" qua ba lần huấn luyện, so sánh hiệu quả của các phương pháp tiền xử lý và điều chỉnh tham số.

Lần 1: Huấn Luyện Mô Hình Chưa Tiền Xử Lý, Tham Số Mặc Định

Mô tả:

Sử dụng tập dữ liệu gốc ("Lung Cancer Dataset.csv") mà không áp dụng bất kỳ bước tiền xử lý nào, giữ nguyên các giá trị thô của các biến như AGE, GENDER, SMOKING, FAMILY_HISTORY, OXYGEN_SATURATION, v.v.

Tham số:

Áp dụng các tham số mặc định của mô hình (ví dụ, mô hình học máy như Random Forest, KNN, XGBoost) mà không điều chỉnh.

Quy trình:

Chia tập dữ liệu thành tập huấn luyện (80%) và tập kiểm tra (20%).

Huấn luyện mô hình với dữ liệu thô.

Đánh giá hiệu suất bằng các chỉ số: độ chính xác (accuracy), độ nhạy (recall), độ đặc hiệu (precision), F1-score.

Mục đích: Thiết lập đường cơ sở (baseline) để so sánh hiệu quả của mô hình khi chưa xử lý dữ liệu hoặc tối ưu hóa.

=> Kết quả:

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

Model	Dataset	Accuracy	Precision	Recall	F1-Score
Decision Tree	Train	100.00%	1.0000	1.0000	1.0000
	Test	84.50%	0.8049	0.7772	0.7908
	Difference	15.50%	0.1951	0.2228	0.2092
Random Forest	Train	100.00%	1.0000	1.0000	1.0000
	Test	91.20%	0.8937	0.8700	0.8817
	Difference	8.80%	0.1063	0.1300	0.1183
K-NN	Train	78.50%	0.7454	0.7319	0.7386
	Test	64.50%	0.5281	0.5491	0.5384
	Difference	14.00%	0.2173	0.1829	0.2002
Naive Bayes	Train	86.17%	0.8118	0.8681	0.8390
	Test	86.90%	0.8030	0.8647	0.8327
	Difference	-0.73%	0.0089	0.0034	0.0063
XGBoost	Train	99.40%	0.9886	0.9970	0.9928
	Test	90.60%	0.8856	0.8621	0.8737
	Difference	8.80%	0.1031	0.1349	0.1191

Bảng 3.3 Lần 1: (chưa tiền xử lý, tham số mặc định)

Lần 2: Huấn Luyện Sau Tiền Xử Lý, Áp Dụng SMOTE và ROS

Mô tả:

Tiền xử lý dữ liệu để cải thiện chất lượng, sau đó áp dụng kỹ thuật cân bằng dữ liệu SMOTE (Synthetic Minority Oversampling Technique) và ROS (Random Over-Sampling) để xử lý mất cân bằng trong nhãn PULMONARY_DISEASE.

Quy trình tiền xử lý dữ liệu:

Xử lý giá trị thiếu: Điền giá trị trung bình/trung vị cho các cột số (AGE, ENERGY_LEVEL, OXYGEN_SATURATION) hoặc loại bỏ nếu cần.

Chuẩn hóa dữ liệu: Áp dụng chuẩn hóa (scaling) như Min-MaxScaler cho các cột số để đồng nhất phạm vi.

Phát hiện và xử lý ngoại lai: Dùng quy tắc IQR để loại bỏ hoặc điều chỉnh các giá trị bất thường.

Cân bằng dữ liệu: Áp dụng SMOTE để tạo mẫu tổng hợp cho lớp thiểu số (ví dụ, PULMONARY_DISEASE = "YES"). Áp dụng ROS để nhân bản ngẫu nhiên các mẫu của lớp thiểu số, đảm bảo tỷ lệ nhãn cân bằng hơn.

Huấn luyện mô hình với tham số mặc định trên tập dữ liệu đã xử lý.

Đánh giá hiệu suất bằng các chỉ số: Accuracy, Precision, Recall, F1-score.

Mục đích: Đánh giá tác động của tiền xử lý và cân bằng dữ liệu đến hiệu suất mô hình, đặc biệt cải thiện khả năng dự đoán cho lớp thiểu số.

=> Kết quả:

Lần 2: (sau tiền xử lý, SMOTE)

Model	Dataset	Accuracy	Precision	Recall	F1-Score
Decision Tree	Train	100.00%	1.0000	1.0000	1.0000
	Test	82.79%	0.7655	0.8439	0.8028
	Difference	17.21%	0.2345	0.1561	0.1972
Random Forest	Train	100.00%	1.0000	1.0000	1.0000
	Test	90.18%	0.8717	0.8951	0.8833
	Difference	9.82%	0.1283	0.1049	0.1167
K-NN	Train	89.04%	0.8917	0.8887	0.8902
	Test	88.97%	0.8444	0.9000	0.8713
	Difference	0.07%	0.0473	-0.0113	0.0189
Naive Bayes	Train	82.02%	0.8133	0.8313	0.8222
	Test	84.31%	0.7766	0.8732	0.8220
	Difference	-2.29%	0.0367	-0.0419	0.0002
XGBoost	Train	97.45%	0.9674	0.9822	0.9747
	Test	87.85%	0.8420	0.8707	0.8561
	Difference	9.60%	0.1254	0.1114	0.1186

Bảng 3.4 Lần 2: (sau tiền xử lý, SMOTE)

Lần 2: (sau tiền xử lý, RandomOverSampler)

Model	Dataset	Accuracy	Precision	Recall	F1-Score
Decision Tree	Train	100.00%	1.0000	1.0000	1.0000
	Test	85.53%	0.8414	0.8024	0.8215
	Difference	14.47%	0.1586	0.1976	0.1785
Random Forest	Train	100.00%	1.0000	1.0000	1.0000
	Test	91.50%	0.9055	0.8878	0.8966
	Difference	8.50%	0.0945	0.1122	0.1034
K-NN	Train	92.01%	0.9152	0.9261	0.9206
	Test	90.18%	0.8735	0.8927	0.8830
	Difference	1.83%	0.0417	0.0334	0.0376
Naive Bayes	Train	85.74%	0.8405	0.8823	0.8609
	Test	85.53%	0.8000	0.8683	0.8327
	Difference	0.22%	0.0405	0.0140	0.0281
XGBoost	Train	98.53%	0.9774	0.9936	0.9855
	Test	90.28%	0.8811	0.8854	0.8832
	Difference	8.25%	0.0964	0.1083	0.1022

Bảng 3.5 Lần 2: (sau tiền xử lý, RandomOverSampler)

Lần 3: Huấn Luyện Với Tùy Chỉnh Tham Số, SMOTE và ROS

Mô tả:

Dựa trên dữ liệu đã tiền xử lý và cân bằng từ lần 2, điều chỉnh tham số của mô hình để tối ưu hóa hiệu suất.

Quy trình:

Sử dụng tập dữ liệu đã tiền xử lý và cân bằng (SMOTE + ROS) từ lần 2.

Tùy chỉnh tham số: Áp dụng kỹ thuật Grid Search để tìm tổ hợp tham số tối ưu (ví dụ, số cây trong Random Forest, số k trong KNN …). Tập trung vào các tham số ảnh hưởng đến độ chính xác và khả năng khái quát hóa của mô hình. Huấn luyện mô hình với các tham số tối ưu.

Đánh giá hiệu suất bằng các chỉ số: Accuracy, Precision, Recall, F1-score

Mục đích: Tối ưu hóa mô hình để đạt hiệu suất cao nhất, xác định sự cải thiện từ việc điều chỉnh tham số so với hai lần trước.

Best Parameters, Smote

Model	Original Parameters	Best Parameters
Decision Tree	- max_depth = [3, 5, 10, 15, 20, None] - min_samples_split = [2, 5, 10, 20] - min_samples_leaf = [1, 2, 4, 10] - criterion = ['gini', 'entropy']	- max_depth = 10 - min_samples_leaf = 10
Random Forest	- n_estimators = [5, 10, 15] - max_depth = [5, 10, 15, 20, None] - min_samples_split = [2, 5, 10] - min_samples_leaf = [1, 2, 4] - bootstrap = [True, False]	- bootstrap = False - max_depth = 15 - min_samples_leaf = 2 - n_estimators = 15

K-Nearest Neighbor	- n_neighbors = [3, 5, 7, 9, 11] - weights = ['uniform', 'distance'] - p = [1, 2]	- n_neighbors = 9 - p = 1 - weights = 'distance'
Bayes (GNB)	- var_smoothing = [1e-9, 1e-8, 1e-7, 1e-6, 1e-5]	- GaussianNB()
XGBoost	- n_estimators = [5, 10, 15] - max_depth = [3, 6, 10, 15] - learning_rate = [0.01, 0.05, 0.1, 0.2, 0.3] - subsample = [0.6, 0.8, 1.0] - colsample_bytree = [0.6, 0.8, 1.0]	- max_depth = 15 - learning_rate = 0.3 - n_estimators = 15 - colsample_bytree = 1.0

Bảng 3.6 Lần 3: (best parameters, Smote)

Lần 3: (tùy chỉnh tham số, SMOTE)

Model	Dataset	Accuracy	Precision	Recall	F1-Score
Decision Tree	Train	88.50%	0.9030	0.8627	0.8824
	Test	87.55%	0.8284	0.8829	0.8548
	Difference	0.95%	0.0746	-0.0202	0.0276
Random Forest	Train	98.90%	0.9873	0.9907	0.9890
	Test	88.97%	0.8508	0.8902	0.8701
	Difference	9.93%	0.1365	0.1004	0.1189
K-NN	Train	100.00%	1.0000	1.0000	1.0000
	Test	89.98%	0.8642	0.9000	0.8817
	Difference	10.02%	0.1358	0.1000	0.1183

Naive Bayes	Train	82.02%	0.8133	0.8313	0.8222
	Test	84.31%	0.7766	0.8732	0.8220
	Difference	-2.29%	0.0367	-0.0419	0.0002
XGBoost	Train	97.60%	0.9706	0.9817	0.9761
	Test	89.37%	0.8605	0.8878	0.8739
	Difference	8.23%	0.1101	0.0939	0.1022

Bảng 3.7 Lần 3: (tùy chỉnh tham số, SMOTE)

Best Parameters, RandomOverSampler

Model	Original Parameters	Best Parameters
Decision Tree	- max_depth = [3, 5, 10, 15, 20, None] - min_samples_split = [2, 5, 10, 20] - min_samples_leaf = [1, 2, 4, 10] - criterion = ['gini', 'entropy']	- criterion = 'entropy' - max_depth = 10
Random Forest	- n_estimators = [5, 10, 15] - max_depth = [5, 10, 15, 20, None] - min_samples_split = [2, 5, 10] - min_samples_leaf = [1, 2, 4] - bootstrap = [True, False]	- bootstrap = False - min_samples_split = 5 - n_estimators = 15
K-Nearest Neighbor	- n_neighbors = [3, 5, 7, 9, 11] - weights = ['uniform', 'distance'] - p = [1, 2]	- n_neighbors = 11 - p = 1 - weights = 'distance'

Bayes (GNB)	- var_smoothing = [1e-9, 1e-8, 1e-7, 1e-6, 1e-5]	- GaussianNB()
XGBoost	- n_estimators = [5, 10, 15] - max_depth = [3, 6, 10, 15] - learning_rate = [0.01, 0.05, 0.1, 0.2, 0.3] - subsample = [0.6, 0.8, 1.0] - colsample_bytree = [0.6, 0.8, 1.0]	- max_depth = 15 - learning_rate = 0.3 - n_estimators = 15 - colsample_bytree = 1.0

Bảng 3.8 Lần 3: (best parameters, RandomOverSampler)

Lần 3: (tùy chỉnh tham số, RandomOverSampler)

Model	Dataset	Accuracy	Precision	Recall	F1-Score
Decision Tree	Train	94.65%	0.9476	0.9452	0.9464
	Test	88.87%	0.8713	0.8585	0.8649
	Difference	5.78%	0.0763	0.0866	0.0815
Random Forest	Train	100.00%	1.0000	1.0000	1.0000
	Test	90.79%	0.8958	0.8805	0.8881
	Difference	9.21%	0.1042	0.1195	0.1119
K-NN	Train	100.00%	1.0000	1.0000	1.0000
	Test	91.50%	0.8937	0.9024	0.8981
	Difference	8.50%	0.1063	0.0976	0.1019
Naive Bayes	Train	85.74%	0.8405	0.8823	0.8609
	Test	85.53%	0.8000	0.8683	0.8327
	Difference	0.22%	0.0405	0.0140	0.0281
XGBoost	Train	97.54%	0.9706	0.9805	0.9755
	Test	91.19%	0.9007	0.8854	0.8930
	Difference	6.34%	0.0698	0.0951	0.0825

Bảng 3.9 Lần 3: (tùy chỉnh tham số, RandomOverSampler)

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.
2. **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.
3. **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] https://www.coursera.org/learn/machine-learning

[3] Giáo trình Học máy cơ bản (Nhà xuất bản khoa học và kỹ thuật)

[4] Các slide bài thuyết trình Học máy cơ bản, khoa học dữ liệu.