CHẨN ĐOÁN TẾ BÀO KHỐI U LÀNH TÍNH/ÁC TÍNH TRONG UNG THƯ VÚ

Lê Tiến Quyết
Khoa Khoa học Máy tính
Trường Đại học Công nghệ Thông tin
Thủ Đức, Việt Nam
21520428@gm.uit.edu.vn

Hồ Nguyễn Thiên Vũ
Khoa Khoa học và Kỹ thuật thông tin
Trường Đại học Công nghệ Thông tin
Thủ Đức, Việt Nam
22521689@gm.uit.edu.vn

Nguyễn Phi Long
Khoa Khoa học và Kỹ thuật Thông tin
Trường Đại học Công nghệ Thông tin
Thủ Đức, Việt Nam
22520818@gm.uit.edu.vn

***Tóm tắt nội dung*—Trước tình hình số lượng các ca mắc bệnh ung thư vú ngày một gia tăng như hiện nay, việc áp dụng những mô hình học máy và trí tuệ nhân tạo nhằm chẩn đoán tình trạng của căn bệnh này càng trở nên cấp thiết. Nắm bắt được xu thế đó, trong nghiên cứu này, chúng tôi đã đề xuất các phương pháp chẩn đoán mức độ lành tính và ác tính của tế bào ung thư vú dựa trên những số liệu của tế bào được thu thập từ hình ảnh số hóa của một vết chích hút kim nhỏ của khối u vú. Nghiên cứu được thực hiện nhằm cải thiện độ chính xác của việc chẩn đoán, từ đó góp phần nâng cao hiệu quả cũng như tối ưu hóa quá trình điều trị ung thư vú. Với việc áp dụng các phương pháp tiền xử lý, phân chia dữ liệu, các mô hình SVM, Random Forest, Logistic Regression cùng phương pháp tinh chỉnh tham số GridSearchCV, nghiên cứu đã đạt được kết quả rất khả quan với độ chính xác tổng thể lần lượt là 97.37, 96.49 và 97.37. Điều đó khẳng định mức độ đáng tin cậy cũng như hiệu suất cao trong việc giải quyết bài toán chẩn đoán ung thư vú.**

I. GIỚI THIỆU ĐỀ TÀI

Vào 19/10/2023, theo Bộ trưởng Bộ Y tế Đào Hồng Lan thì ung thư đang trở thành gánh nặng lớn của các quốc gia, trong đó có Việt Nam. Ung thư vú là bệnh thường gặp và theo thống kê mỗi năm cả nước có 21.555 ca mắc mới, chiếm 25,8% tổng số các loại ung thư thường gặp ở nữ giới. Đáng lo ngại hơn, hiện nay ung thư vú đang có xu hướng ngày càng trẻ hóa. Do vậy, việc đề xuất một ứng dụng kết hợp giữa khoa học công nghệ và y tế giúp phát hiện sớm được bệnh khi thăm khám nhằm cải thiện được tình hình sức khỏe của người dân là vô cùng cần thiết.

Với sự phát triển của lĩnh vực học máy như hiện nay, điều này là hoàn toàn khả thi. Dù khó có thể đạt được độ chính xác hoàn toàn tuyệt đối, ứng dụng này vẫn sẽ giúp bác sĩ có thể rút ngắn thời gian chẩn đoán và điều trị cho bệnh nhân. Vì thế, trong nghiên cứu này, chúng tôi đề xuất áp dụng các mô hình học máy nhằm giải quyết bài toán chẩn đoán mức độ lành tính và ác tính của tế bào ung thư vú.

II. GIỚI THIỆU BỘ DỮ LIỆU

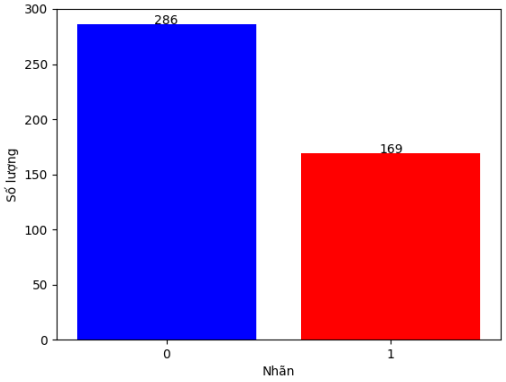
Trong nghiên cứu này, chúng tôi sử dụng bộ dữ liệu tên "Breast Cancer Wisconsin (Diagnostic) Data Set" có sẵn trên https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data. Các đặc trưng của bộ dữ liệu này được xây dựng dựa trên hình ảnh số hóa của một vết chích hút kim nhỏ của khối u vú.

Bộ dữ liệu có 31 thuộc tính và 569 mẫu, trong đó thuộc tính đích "diagnosis" thể hiện nhãn của dữ liệu mang một trong hai giá trị 0 (lành tính) hoặc 1 (ác tính) và 30 thuộc tính còn lại thuộc loại dữ liệu số thực thể hiện các đặc trưng của dữ liệu. Dựa trên sự phân chia về nhãn, bộ dữ liệu gồm 357 mẫu lành tính và 212 mẫu ác tính. Không có mẫu nào mang giá trị NULL. Từ bộ dữ liệu gốc, nhóm chia thành hai bộ dữ liệu gồm: 80% để huấn luyện mô hình và 20% để kiểm tra mô hình.

A. Tổng quan về bộ dữ liệu huấn luyện

Bộ dữ liệu huấn luyện gồm có 455 mẫu. Mỗi mẫu dữ liệu gồm 31 cột, trong đó cột "diagnosis" thể hiện nhãn của dữ liệu mang một trong hai giá trị 0 (lành tính) hoặc 1 (ác tính) và 30 cột còn lại là 30 số thực thể hiện đặc trưng của dữ liệu.

Có 286 mẫu mang giá trị 0 và 169 mẫu mang giá trị 1 trong bộ dữ liệu huấn luyện.



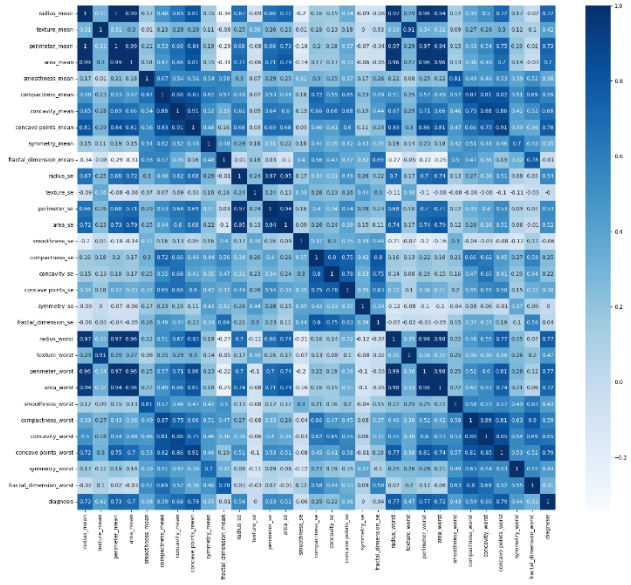
Hình 1. Số mẫu dữ liệu của mỗi nhãn

B. Phân tích dữ liệu huấn luyện

Phân tích mức độ tương quan giữa các cột với nhau bằng độ tương quan Pearson:

$$\rho(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

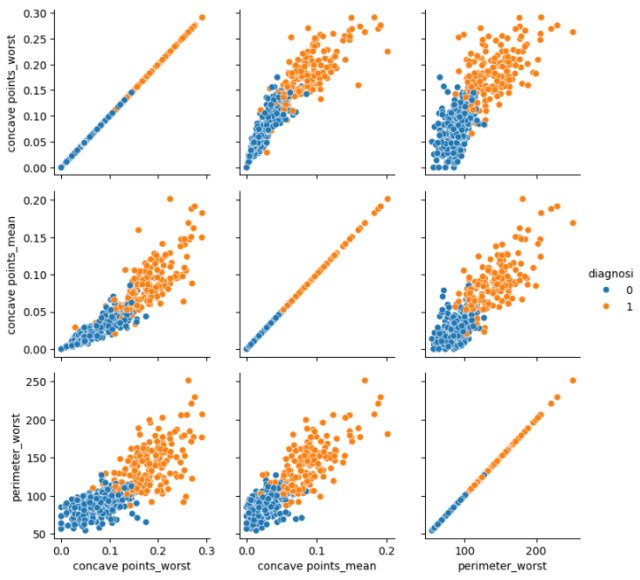
với $\rho(X,Y)$ là độ tương quan giữa X và Y, $cov(X,Y)$ là hiệp phương sai giữa X và Y, σ_X và σ_Y lần lượt là giá trị độ lệch chuẩn của X và Y.



Hình 2. Mức độ tương quan giữa các cột dữ liệu

Dựa vào ma trận tương quan ở Hình 2, với giá trị tương quan lần lượt là 0.788885, 0.778115 và 0.774998, ta thấy các thuộc tính "concave points_worst", "concave points_mean" và "perimeter_worst" tương quan rất mạnh với thuộc tính đích. Bên cạnh đó, ta cũng nhận thấy mức độ tương quan giữa các thuộc tính "fractal_dimension_mean", "texture_se" và "symmetry_se" so với thuộc tính đích là rất thấp với giá trị tương quan lần lượt là -0.01, 0 và 0.

Tiếp đến, nhằm mang lại cái nhìn tổng quát hơn về cách các biến số tương quan với nhau, chúng tôi tạo Pairplot cho 3 thuộc tính có giá trị tương quan mạnh nhất với thuộc tính đích là "concave points_worst", "concave points_mean" và "perimeter_worst".



Hình 3. Pairplot cho 3 thuộc tính có tương quan mạnh nhất với thuộc tính đích

Dựa vào Hình 3, ta thấy các thuộc tính từng đôi một có quan hệ tuyến tính với nhau. Khi một biến số tăng thì biến số kia cũng tăng theo, được biểu hiện qua xu hướng dữ liệu đi lên từ trái sang phải. Đồng thời, khi giá trị của các thuộc tính càng tăng thì nhãn 1 của thuộc tính đích xuất hiện càng nhiều và ngược lại. Bên cạnh đó, các biểu đồ trên đường chéo cho thấy "perimeter_worst" và "concave points_worst" có phân phối lệch với đuôi dài về phía giá trị cao, trong khi "concave points_mean" có phân phối đối xứng hơn.

Các bước phân tích trên đã giúp đánh giá nhanh chóng các mối quan hệ tiềm năng trong tập huấn luyện, từ đó mang lại những giá trị hữu ích cho việc lựa chọn đặc trưng cũng như hỗ trợ cho việc phân tích thống kê và xây dựng mô hình dự đoán.

III. PHƯƠNG PHÁP THỰC HIỆN

A. Phương pháp phân lớp nhị phân

Phân lớp nhị phân trong Machine Learning là quá trình gán nhãn dữ liệu cho đối tượng vào một trong hai lớp khác nhau dựa vào việc dữ liệu đó có hay không có các đặc trưng (feature) của bộ phân lớp. Đây là một bài toán quan trọng và rộng rãi ứng dụng trong thực tế. Ví dụ, nó được sử dụng để nhận dạng khuôn mặt, phát hiện email spam, và nhiều ứng dụng khác. Trong bài toán của chúng tôi là phân loại khối u "lành tính" hoặc "ác tính".

B. Các công cụ sử dụng

Chúng tôi thực hiện phân tích dữ liệu cùng với xây dựng mô hình Machine Learning hoàn toàn bằng ngôn ngữ lập trình Python đi kèm là các thư viện Sklearn hỗ trợ phát triển và triển khai mô hình học máy. Sklearn còn cung cấp các công cụ phân tích, kiểm tra mô hình như confusion matrix và các phép đo đánh giá hiệu suất, giúp chia tập dữ liệu hay làm giảm chiều dữ liệu như PCA.

Chúng tôi sử dụng các công cụ xử lý dữ liệu như Pandas và Numpy để dễ dàng thao tác như xử lý dữ liệu, chuẩn hóa, thống kê và khám phá dữ liệu… Các công cụ trực quan hóa mà chúng tôi sử dụng là Matplotlib, Seaborn để hiển thị dữ liệu, vẽ biểu đồ, tạo đồ thị để trực quan hóa kết quả và phân tích dữ liệu.

Bên cạnh đó, chúng tôi còn sử dụng công cụ để tinh chỉnh mô hình và đánh giá hiệu suất: GridSearchCV - một trong những công cụ phổ biến để tìm kiếm bộ siêu tham số tốt nhất cho mô hình, sử dụng Pipeline để mã hóa và tự động hóa các công đoạn, quy trình làm cần thiết để tạo mô hình học máy.

C. Mô hình sử dụng

1. Support Vector Machine

Support Vector Machine (SVM) là một thuật toán học có giám sát (Supervised Learning) được sử dụng rộng rãi trong lĩnh vực phân loại dữ liệu. Đặc điểm nổi bật của SVM là khả năng tạo ra một siêu phẳng (hyperplane) tối ưu để phân chia các điểm dữ liệu vào các nhóm riêng biệt.

Mục tiêu chính của SVM là tìm một siêu phẳng trong không gian đa chiều sao cho khoảng cách từ các điểm dữ liệu gần nhất đến siêu phẳng đó là lớn nhất. Khoảng cách này được gọi là độ rộng giới hạn (margin). Siêu phẳng cần có độ rộng giới hạn lớn nhất để đảm bảo tính tổng quát và khả năng phân loại tốt trên dữ liệu mới.

Trong quá trình xây dựng siêu phẳng, SVM sử dụng các điểm dữ liệu gần nhất (support vectors) để xác định vị trí và hướng của siêu phẳng. Các điểm dữ liệu này là những điểm

gần nhất với siêu phẳng và quan trọng trong việc xác định biên phân chia.

Phương trình của siêu phẳng trong SVM có dạng:

$$w_1x_1 + w_2x_2 + \cdots + w_nw_n + \ b = 0$$

Trong đó:

- $w_1, w_2, ..., w_n$ là các hệ số của siêu phẳng.
- $x_1, x_2, ..., w_n$ là các thuộc tính của điểm dữ liệu, và b là một hằng số.

Để xử lý các tình huống phức tạp và dữ liệu phi tuyến, SVM sử dụng các hàm kernel như linear, polynomial, Gaussian (RBF), sigmoid, … để ánh xạ dữ liệu từ không gian ban đầu sang một không gian cao chiều hơn. Việc sử dụng kernel giúp SVM phân loại dữ liệu một cách hiệu quả hơn và tạo ra các biên phân lớp linh hoạt hơn.

Một thách thức khi sử dụng SVM là khi số lượng thuộc tính (p) trong dữ liệu lớn hơn rất nhiều so với số lượng điểm dữ liệu (n). Trường hợp này có thể dẫn đến kết quả không tốt do việc tìm siêu phẳng phân chia trở nên phức tạp hơn và có nguy cơ overfitting. Để giải quyết vấn đề này, cần điều chỉnh các tham số như hằng số *C* trong SVM để tìm được giải pháp tối ưu.

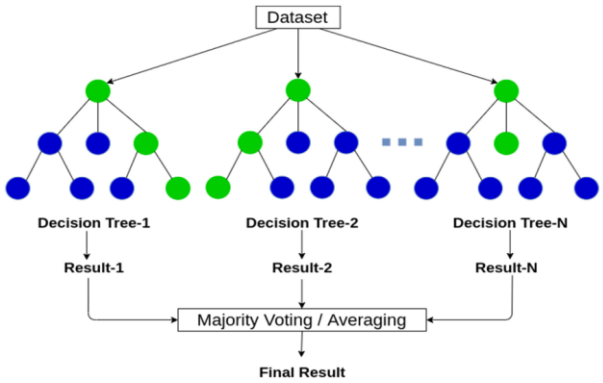
Nhìn chung, SVM là một công cụ mạnh trong việc phân loại dữ liệu và tạo ra các biên phân lớp tối ưu. Tuy nhiên, chúng ta cần tinh chỉnh các tham số và lựa chọn kernel để đạt được hiệu suất tốt nhất.

2. Random Forest

Random Forest (RF) là thuật toán học có giám sát (supervised learning) có thể giải quyết cả hai bài toán hồi quy (regression) và phân lớp (classification).Nó là thuật toán linh hoạt và dễ sử dụng. Nói sâu về thuật toán RF thì ta có thể hiểu Random là ngẫu nhiên, Forest là rừng nên ở đây thuật toán RF sẽ xây dựng nhiều cây quyết định bằng thuật toán Decision Tree, tuy nhiên mỗi cây quyết định sẽ khác nhau (có yếu tố random). Sau đó kết quả dự đoán được tổng hợp từ các cây quyết định.

Về mặt kỹ thuật, nó là một phương pháp tổng hợp (dựa trên cách tiếp cận phân chia và chinh phục) của các cây quyết định được tạo ra trên một tập dữ liệu được chia ngẫu nhiên. Bộ sưu tập phân loại cây quyết định này còn được gọi là rừng. Cây quyết định riêng lẻ được tạo ra bằng cách sử dụng chỉ báo chọn thuộc tính như tăng thông tin, tỷ lệ tăng và chỉ số Gini cho từng thuộc tính. Mỗi cây phụ thuộc vào một mẫu ngẫu nhiên độc lập. Trong bài toán phân loại, mỗi phiếu bầu chọn và lớp phổ biến nhất được chọn là kết quả cuối cùng. Trong trường hợp hồi quy, mức trung bình của tất cả các kết quả đầu ra của cây được coi là kết quả cuối cùng. Nó đơn giản và mạnh mẽ hơn so với các thuật toán phân loại phi tuyến tính khác. RF hoạt động theo 4 bước:

- Chọn các mẫu ngẫu nhiên từ tập dữ liệu đã cho.
- Thiết lập cây quyết định cho từng mẫu và nhận kết quả dự đoán từ mỗi cây quyết định cây.
- Hãy bỏ phiếu cho mỗi kết quả dự đoán.
- Chọn kết quả được dự đoán nhiều nhất là dự đoán cuối cùng



Hình 4. Giải thuật Random Forest

3. Logistic Regression

Logistic Regression (LR) là một thuật toán học có giám sát (supervised learning) được sử dụng cho bài toán phân loại nhị phân. LR làm việc dựa trên nguyên tắc của một hàm phi tuyến (sigmoid function) chuyển đầu vào của nó thành xác suất thuộc về một trong hai lớp nhị phân. Sau đó phân lớp vào một trong hai lớp {0,1}.

Để xác suất hóa đầu ra thì LR đã sử dụng hàm sigmoid có dạng:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

và giá trị của đầu ra nằm trong khoảng từ 0 đến 1.

Với đầu vào là ma trận dữ liệu X và trọng số w thì đầu ra là:

$$\sigma(f(X)) = \frac{1}{1 + e^{-f(X)}}$$

trong đó $f(X) = w^T X$. Lúc này ta so sánh giá trị đầu ra này với một ngưỡng (threshold thường là 0.5) để quyết định nó thuộc về lớp nào. Nghĩa là $\sigma\big(f(X)\big) \geq$ threshold thì thuộc về lớp 1 hoặc $\sigma\big(f(X)\big) <$ threshold thì thuộc về lớp 0.

Việc huấn luyện mô hình là tìm ra bộ trọng số w sao cho giá trị hàm mất mát của mô hình nhỏ nhất. Hàm mất mát sử dụng cho Logistic Regression là Binary Cross Entropy có dạng:

$$L(w) = -\frac{1}{n}\sum_{i=1}^{n}[y_i log p_i + (1 - y_i)\log(1 - p_i)]$$

Trong đó:

- n là số lượng mẫu dữ liệu trong tập huấn luyện
- y_i là giá trị nhãn thực tế của mẫu thứ i
- p_i là giá trị xác suất dự đoán của mô hình cho mẫu thứ i.

Với mỗi mẫu, giá trị xác suất dự đoán p_i càng xa giá trị nhãn thực tế y_i thì giá trị của hàm mất mát càng lớn.

Mô hình LR ngoài phân loại ra nó còn cung cấp xác suất dự đoán cho từng lớp giúp ta có thể đánh giá mức độ chắc chắn của dự đoán. Tuy vậy, mô hình sẽ hoạt động kém hơn khi gặp dữ liệu phi tuyến.

D. Các giá trị tham số

Chúng tôi sử dụng GridSearchCV để tìm bộ các tham số tốt nhất trên Pipeline của từng mô hình với tập các giá trị cho trước.

Đối với hai mô hình SVM và LR, chúng tôi xác định các tham số sau:

- "*max_iter*": xác định số lần lặp tối đa mà thuật toán tối ưu hóa sẽ thực hiện để tìm kiếm điểm hội tụ (convergence). Nếu giá trị "*max_iter*" quá nhỏ, thuật toán có thể không hội tụ, nghĩa là không tìm được giá trị tối ưu cho các hệ số (coefficients) của mô hình. Nếu giá trị quá lớn, thời gian tính toán có thể kéo dài mà không cải thiện đáng kể kết quả.
- "*tol*" (tolerance): là ngưỡng dung sai để ngừng thuật toán. Khi sự thay đổi của hàm mục tiêu (objective function) giữa hai lần lặp liên tiếp nhỏ hơn giá trị này, thuật toán sẽ dừng lại. Giá trị "*tol*" nhỏ có nghĩa là thuật toán sẽ chạy cho đến khi đạt được mức độ chính xác rất cao, nhưng có thể tốn nhiều thời gian hơn. Giá trị "*tol*" lớn hơn có thể giúp thuật toán dừng lại sớm hơn, nhưng có thể dẫn đến việc không đạt được giá trị tối ưu tốt nhất.
- "*C*": là tham số điều chỉnh cho regularization trong LR. Nó là nghịch đảo của sức mạnh regularization ($C = \frac{1}{\lambda}$, với λ là hệ số regularization). Regularization giúp ngăn chặn mô hình bị overfitting bằng cách thêm một hình phạt vào hàm mục tiêu. Giá trị *C* lớn (tức là Regularization yếu) cho phép mô hình linh hoạt hơn, có thể phù hợp hơn với dữ liệu huấn luyện nhưng dễ bị overfitting (quá khớp). Giá trị *C* nhỏ (tức là regularization mạnh) làm cho mô hình ít linh hoạt hơn, có thể ngăn chặn overfitting nhưng dễ bị underfitting (thiếu khớp).

Đối với RF, chúng tôi tìm kiếm giá trị cho:

- "*n_estimators*": xác định số lượng cây quyết định (Decision Trees) sẽ được xây dựng trong rừng ngẫu nhiên (Random Forest). Việc số lượng cây lớn hay nhỏ có tác động rất nhiều đến hiệu suất của mô hình.
- "*max_depth*": xác định độ sâu tối đa của mỗi cây quyết định trong rừng ngẫu nhiên. Giống như số lượng cây, việc xác định độ sâu cũng đóng vai trò quan trọng trong hiệu suất của mô hình.

Khi chúng tôi thực hiện việc giảm chiều và áp dụng với mỗi mô hình tương ứng, chúng tôi còn phải xác định thêm một tham số đó là:

- "*n_components*": xác định số lượng thành phần chính (principal components) cần giữ lại sau khi áp dụng PCA. Thành phần chính là các hướng trong không gian đặc trưng mà phương sai của dữ liệu là lớn nhất. Bằng cách chọn một số lượng ít hơn các thành phần chính, chúng ta có thể giảm số chiều của dữ liệu trong khi vẫn giữ lại phần lớn thông tin của dữ liệu gốc.

E. Các độ đo sử dụng

Chúng tôi sử dụng **Accuracy** và **F1-score** để đánh giá mô hình. Thông tin cụ thể của các độ đo này như sau:

- **Accuracy**: là tỷ lệ giữa số lượng dự đoán đúng và tổng số lượng dự đoán. Nói cách khác, độ đo này cho biết mô hình dự đoán đúng bao nhiêu phần trăm trên toàn bộ tập dữ liệu.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- **F1-score**: là trung bình điều hòa của *precision* và *recall*. Nó cung cấp một cách để đánh giá sự cân bằng giữa *precision* và *recall*. *F1-score* sẽ cao khi cả *precision* và *recall* đều cao. Ngoài ra, nó có xu hướng gần với giá trị nhỏ nhất giữa *precision* và *recall*.

$$Precision = \frac{TP}{TP + FP} \mid Recall = \frac{TP}{TP + FN}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

- **Precision**: là tỷ lệ giữa số lượng true positive (TP - các trường hợp mà mô hình dự đoán đúng là positive) và tổng số lượng dự đoán là positive (là tổng của TP và false positive - FP, các trường hợp mà mô hình dự đoán sai là positive). Precision cho biết khi mô hình dự đoán một trường hợp là positive, nó chính xác bao nhiêu phần trăm.

- **Recall**: là tỷ lệ giữa số lượng true positive (TP - các trường hợp mà mô hình dự đoán đúng) và tổng số lượng thực sự positive (là tổng của TP và false negative - FN, các trường hợp mà mô hình dự đoán sai). Recall cho biết mô hình tìm ra được bao nhiêu phần trăm các trường hợp positive.

Trong đó:

- TP (True Positive): số lượng dự đoán đúng là positive.
- TN (True Negative): số lượng dự đoán đúng là negative.
- FP (False Positive): số lượng dự đoán sai là positive.
- FN (False Negative): số lượng dự đoán sai là negative.

IV. KẾT QUẢ THỰC NGHIỆM

A. Kết quả trên tập đánh giá

Pipeline	***Accuracy***	***F1-score***
MinMaxScaler + SVM	**97.37**	**96.47**
MinMaxScaler + PCA + SVM	**97.37**	**96.47**
RF	96.49	95.24
MinMaxScaler + RF	96.49	95.24
MinMaxScaler + LR	**97.37**	**96.47**
MinMaxScaler + PCA + LR	**97.37**	**96.47**

Bảng 1. Kết quả huấn luyện và tinh chỉnh mô hình (%)

B. Phân tích và nhận định kết quả

Qua kết quả huấn luyện và tinh chỉnh mô hình được trình bày ở Bảng 1, ta có thể rút ra một số nhận xét rằng các pipeline ở cùng mô hình cho ra kết quả như nhau ở các độ đo. Điều này cho thấy việc thay đổi phần tiền xử lý (preprocessing) không ảnh hưởng đáng kể đến hiệu suất của mô hình.

Ở mô hình RF và SVM, việc thêm phương pháp giảm chiều dữ liệu PCA vào pipeline không làm tăng hiệu suất mô hình. Có thể những thành phần chính được chọn không mang nhiều thông tin quan trọng cũng như gây mất mát thông tin, từ đó làm giảm khả năng dự đoán của mô hình.

Các mô hình với pipeline mang lại hiệu suất tối ưu nhất: SVM với MinMaxScaler + SVM, Logistic Regression với MinMaxScaler + LogisticRegression. Điều này có thể do SVM là mô hình phân loại tuyến tính và việc chuẩn hóa dữ liệu giúp tối ưu hóa việc tìm ra siêu mặt phẳng. Cùng với đó, Logistic Regression cũng là mô hình tuyến tính, nên việc chuẩn hóa dữ liệu cũng có tác động tích cực.

Mô hình Random Forest cho mang lại hiệu suất thấp nhất trong 3 mô hình. Đồng thời, phương pháp MinMaxScaler cũng tỏ ra không hiệu quả trong việc làm tăng hiệu suất mô hình. Điều này có thể do Random Forest không nhạy với việc chuẩn hóa dữ liệu theo phương pháp này.

Nhìn chung, kết quả của cả 3 mô hình với những Pipeline trên đều trên 95%, thể hiện hiệu suất rất tốt trong việc giải quyết bài toán.

V. PHÂN TÍCH LỖI, HƯỚNG PHÁT TRIỂN

A. Phân tích lỗi

Dựa trên kết quả ở Bảng 1, việc sử dụng giảm chiều dữ liệu PCA không làm tăng hiệu suất ở một số mô hình. Điều này có thể do PCA có thể loại bỏ một số thông tin quan trọng nào đó hoặc các thành phần được chọn không mang nhiều thông tin quan trọng hoặc mô hình lúc này không phù hợp với dữ liệu đã được giảm chiều

Số lượng mẫu dữ liệu không đủ lớn gây hạn chế cho khả năng học của mô hình do không thể tổng quát hóa các trường hợp khác nhau có thể xảy ra.

B. Hướng phát triển

Với kích thước dữ liệu như đã nói ở mục II, bộ dữ liệu của nhóm vẫn còn hạn chế về số lượng. Vậy nên cần tăng số lượng mẫu cũng như chất lượng của bộ dữ liệu. Ngoài ra, chúng ta có thể thay đổi dữ liệu từ tập các thông tin của khối u thành hình ảnh chứa khối u đó.

Về mô hình, có thể xem xét việc kết hợp nhiều mô hình với nhau để tận dụng các điểm mạnh của từng cái. Tiến hành thử nghiệm thêm các phương pháp khác nhau và tinh chỉnh nhiều tham số hơn để cải thiện kết quả. Áp dụng các mô hình mạng học sâu như VGG, ResNet, InceptionNet … để học được các đặc trưng phức tạp nhằm cải thiện khả năng tổng quát hóa của mô hình.

VI. KẾT LUẬN

Trong đồ án này, nhóm đã áp dụng những kiến thức được học trong môn để giải quyết bài toán. Trước khi huấn luyện mô hình thì nhóm đã trực quan hóa dữ liệu để hiểu hơn về dữ liệu. Sau đó, nhóm đã thực hiện thử nghiệm tinh chỉnh trên nhiều mô hình với những sự kết hợp khác nhau để chọn ra các phương án cho kết quả tốt nhất, thay vì chỉ tập trung vào một phương án cụ thể. Các phương pháp, mô hình được chọn đạt được kết quả tốt nhất trên tập test.

Những phương pháp được đề xuất tuy không thể đưa ra kết quả chính xác tuyệt đối cho các trường hợp bệnh nhân, vì vậy cần được cải tiến thêm ở trong tương lai bằng các kỹ thuật, phương pháp cao hơn để có thể đạt kết quả tốt hơn. Dù vậy, với kết quả đạt được ở hiện tại cũng có thể giúp giảm khối lượng công việc cho các bác sĩ trong việc chẩn đoán lâm sàng, chứ chưa đủ khả năng để thay thế hoàn toàn các bác sĩ.

VII. TÀI LIỆU THAM KHẢO

[1] “Ung thư vú chiếm 1/4 tổng số ca mắc ung thư ở nữ giới, ngày càng trẻ hóa”, Tuổi Trẻ Online, 19/10/2023
[2] “10 Bộ dữ liệu máy học miễn phí năm 2024”, VINBIGDATA, 06/03/2024
[3] Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow, 2nd Edition, Aurélien [2] Géron.
[4] The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd Edition) (Trevor Hastie, Robert Tibshirani, Jerome Friedman)
[5] Pattern Recognition and Machine Learning (Christopher M. Bishop, 2006)
[6] An Introduction to Statistical Learning, with applications in R (ISLR) (James, Witten, Hastie, Tibshirani)

BẢNG PHÂN CÔNG

MỤC	NỘI DUNG	PHỤ TRÁCH	NGÀY BẮT ĐẦU	DEADLINE
A	**TIỀN XỬ LÝ DỮ LIỆU**	**Thiên Vũ Tiến Quyết**	**25/03**	**04/04**
B	**KHÁM PHÁ DỮ LIỆU**	**Thiên Vũ Phi Long**	**06/04**	**12/04**
C	**HUẤN LUYỆN & TINH CHỈNH MÔ HÌNH**	**Chung**	**13/04**	**01/05**
D	**BÁO CÁO**	**Chung**	**02/05**	**28/06**
*	Abstract	Phi Long	02/05	04/05
I	Giới thiệu đề tài			
II	Giới thiệu bộ dữ liệu		05/05	07/05
III	Phương pháp thực hiện	Chung	08/05	30/05
1	Tổng quan phương pháp	Thiên Vũ	08/05	
2	Giới thiệu mô hình	Chung	09/05	23/05
2.1	SVM	Phi Long	09/05	23/05
2.2	Random Forest	Thiên Vũ		
2.3	Logistic Regression	Tiến Quyết		
3	Các độ đo sử dụng	Phi Long	24/05	30/05
IV	Kết quả thực nghiệm	Phi Long Tiến Quyết	01/06	18/06
1	Kết quả trên tập kiểm thử	Tiến Quyết	01/06	18/06
2	Phân tích và nhận định kết quả	Phi Long		
V	Phân tích lỗi & hướng phát triển	Tiến Quyết	19/06	23/06
VI	Kết luận			
*	Tổng hợp file và trình bày báo cáo	Chung	24/06	28/06
E	**SLIDE**	**Chung**	**29/06**	
F	**NỘP ĐỒ ÁN**	**Phi Long**	**30/06**	
G	**THUYẾT TRÌNH**	**Chung**	**30/06**	**02/07**
1	I - II - IV	Phi Long	30/06	
2	III	Thiên Vũ		
3	V - VI	Tiến Quyết		
4	Tập dượt	Chung	01/07	
5	Thuyết trình chính thức	Chung	02/07	