TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

PHAN VĂN HIẾU – 51800868
NGUYỄN ĐỨC TÍN – 51800248

DỰ ĐOÁN ĐỘT QUỴ VỚI DỮ LIỆU KHÔNG CÂN BẰNG

DỰ ÁN CÔNG NGHỆ THÔNG TIN 2

KHOA HỌC MÁY TÍNH

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

PHAN VĂN HIẾU – 51800868
NGUYỄN ĐỨC TÍN – 51800248

DỰ ĐOÁN ĐỘT QUỴ VỚI DỮ LIỆU KHÔNG CÂN BẰNG

DỰ ÁN CÔNG NGHỆ THÔNG TIN 2

KHOA HỌC MÁY TÍNH

Người hướng dẫn
TS. Trịnh Hùng Cường

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024

LỜI CẢM ƠN

Nhóm chúng em xin bày tỏ lòng tri ân sâu sắc đến thầy Trịnh Hùng Cường vì sự tận tâm và hướng dẫn nhiệt tình trong suốt quá trình thực hiện Dự án Công nghệ thông tin 2. Sự quan tâm, thời gian mà thầy đã dành cho chúng em thật đáng quý và chúng em vô cùng biết ơn về điều đó.

Nhóm chúng em cũng muốn gửi lời cảm ơn chân thành đến toàn thể giảng viên khoa Công nghệ thông tin của trường Đại học Tôn Đức Thắng. Những kiến thức quý báu mà quý Thầy/Cô đã truyền đạt trong suốt quãng thời gian đại học không chỉ giúp chúng em hoàn thành tốt các bài báo cáo và dự án tại trường, mà còn là nền tảng vững chắc để chúng em tự tin bước vào công việc và cuộc sống sau này.

Do điều kiện hiện tại và kinh nghiệm còn hạn chế của sinh viên, bài báo cáo Dự án Công nghệ thông tin 2 của chúng em chắc chắn không tránh khỏi những thiếu sót. Chúng em rất mong nhận được những góp ý và nhận xét từ quý Thầy/Cô để có cơ hội học hỏi và cải thiện trong tương lai.

Một lần nữa, chúng em xin chân thành cảm ơn thầy Trịnh Hùng Cường. Sự tận tâm và hướng dẫn nhiệt tình của Thầy trong suốt quá trình học tập môn Dự án Công nghệ thông tin 2 đã để lại ấn tượng sâu đậm và mang lại giá trị to lớn cho chúng em.

TP.Hồ Chí Minh, ngày ... tháng ... năm 2024

Tác giả

Nguyễn Đức Tín

Phan Văn Hiếu

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Chúng tôi xin cam đoan đây là công trình nghiên cứu của riêng chúng tôi và được sự hướng dẫn khoa học của TS. Trịnh Hùng Cường. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Dự án Công nghệ thông tin 2 còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung Dự án của mình. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do chúng tôi gây ra trong quá trình thực hiện (nếu có).

TP.Hồ Chí Minh, ngày ... tháng ... năm 2024

Tác giả

Nguyễn Đức Tín

Phan Văn Hiếu

DỰ ĐOÁN ĐỘT QUỴ
VỚI DỮ LIỆU KHÔNG CÂN BẰNG
TÓM TẮT

Ngày nay, tác hại nghiêm trọng của căn bệnh đột quỵ trong đời sống hiện đại đã thúc đẩy việc cải tiến phương pháp chẩn đoán và điều trị bệnh cho căn bệnh này. Sự kết hợp giữa y học và công nghệ đã tạo điều kiện cho các nhà khoa học quản lý bệnh nhân hiệu quả hơn thông qua việc tận dụng và lưu trữ hồ sơ y tế điện tử. Vì vậy, việc nghiên cứu mối tương quan giữa các yếu tố, nguy cơ trong hồ sơ bệnh án và đánh giá mức độ ảnh hưởng của chúng đến khả năng dự đoán đột quỵ là vô cùng cần thiết cho tổ chức y tế.

Nghiên cứu này tập trung vào việc phân tích có hệ thống các yếu tố trong hồ sơ sức khỏe điện tử nhằm dự đoán đột quỵ một cách chính xác. Bằng cách áp dụng các kỹ thuật khai thác dữ liệu, chúng em đã xác định được những yếu tố chính góp phần vào việc dự đoán tai biến. Kết quả cho thấy tuổi tác, bệnh tim mạch, chỉ số đường huyết trung bình và tình trạng cao huyết áp là những yếu tố quan trọng nhất trong việc phát hiện nguy cơ đột quỵ.

Sau khi phân tích dữ liệu, chúng em tiến hành thử nghiệm với các mô hình học máy, bao gồm: K-Nearest Neighbors, Random Forest, Support Vector Classifier và CNN. Do tập dữ liệu ban đầu không cân bằng nên nhóm đã sử dụng kỹ thuật Undersampling, Oversampling, SMOTE, ... để tạo ra một tập dữ liệu cân bằng và báo cáo kết quả dựa trên tập dữ liệu này.

Để đánh giá và so sánh hiệu suất của các mô hình, chúng em sử dụng các chỉ số: độ chính xác (accuracy), độ chính xác dương tính (precision), độ nhớ lại/nhạy cảm (recall/sensitivity), điểm F1 (F1 score) và đường cong AUC (AUC curve). Việc sử dụng đa dạng các chỉ số này giúp đánh giá toàn diện hiệu suất của các mô hình, đặc biệt trong bối cảnh dữ liệu không cân bằng.

STROKE PREDICTION WITH IMBALANCED DATA

ABSTRACT

Today, the severe consequences of stroke in modern life have driven improvements in diagnostic and treatment methods for this disease. The combination of medicine and technology has enabled scientists to manage patients more effectively through the utilization and storage of electronic medical records. Therefore, researching the correlation between factors and risks in medical records and evaluating their impact on stroke prediction capability is extremely necessary for healthcare organizations.

This research focuses on systematically analyzing factors in electronic health records to accurately predict stroke. By applying data mining techniques, we have identified the main factors contributing to stroke prediction. Results show that age, heart disease, average blood glucose levels, and hypertension are the most important factors in detecting stroke risk.

After data analysis, we conducted experiments with machine learning models, including: K-Nearest Neighbors, Random Forest, Support Vector Classifier and CNN. Due to the initial dataset being imbalanced, we used the SMOTE technique to create a balanced dataset and reported results based on this dataset.

To evaluate and compare the performance of the models, we used the following metrics: accuracy, precision, recall/sensitivity, F1 score and AUC curve. The use of these diverse metrics helps comprehensively evaluate the performance of models, especially in the context of imbalanced data.

MỤC LỤC

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

DANH MỤC CÁC CHỮ VIẾT TẮT

AI	Artificial Intelligence
CNN	Convolutional Neural Network
DT	Decision Tree
EEG	Electroencephalogram
KNN	K-Nearest Neighbors
LGBM	Light Gradient Boosting Machine
LR	Linear Regression
LSTM	Long Short Term Memory
ML	Machine Learning
RF	Random Forest
SVC	Support Vector Classifier
SVM	Support Vector Machine

CHƯƠNG 1. MỞ ĐẦU

1.1 **Lý do chọn đề tài**

Trong quá trình học tập, nhóm sinh viên chúng em đã được tiếp xúc với nhiều môn học liên quan đến trí tuệ nhân tạo (AI) như Nhập môn học máy, Nhập môn Xử lý đa phương tiện và Xử lý ngôn ngữ lớn. Những môn học này, kết hợp với niềm đam mê sẵn có về AI, đã thôi thúc nhóm tìm hiểu sâu hơn về các đề tài nghiên cứu mà Khoa đề xuất.

Với mong muốn được mở rộng kiến thức và khả năng nghiên cứu, nhóm đã tham gia quá trình đăng ký đề tài theo quy định của trường. Sau khi cân nhắc kỹ lưỡng, chúng em quyết định chọn hướng nghiên cứu do thầy Trịnh Hùng Cường hướng dẫn, tập trung vào lĩnh vực mô hình học máy.

Lý do lựa chọn hướng nghiên cứu này không chỉ vì nó phù hợp với sở trường cá nhân, mà còn bởi tính ứng dụng thực tiễn cao của nó. Nhóm tin rằng việc tham gia nghiên cứu này sẽ giúp không chỉ tiếp thu được nhiều kiến thức mới mà còn hiểu sâu sắc hơn về các mô hình học máy. Những kinh nghiệm và kỹ năng thu được từ quá trình nghiên cứu này chắc chắn sẽ là nền tảng quý giá, hỗ trợ đắc lực cho công việc của chúng em trong tương lai.

1.2 Mục tiêu thực hiện đề tài

Mục tiêu 1: Nghiên cứu tổng quan về bệnh đột quỵ và đánh giá các phương pháp phát hiện bệnh hiện nay.

Mục tiêu 2: Khảo sát và phân tích các mô hình học máy được sử dụng phổ biến trong các nghiên cứu khoa học gần đây về dự đoán bệnh đột quỵ.

Mục tiêu 3: Nắm vững cơ sở dữ liệu và các mô hình học máy được áp dụng trong nghiên cứu này.

Mục tiêu 4: Tổng hợp và rút ra bài học kinh nghiệm về quy trình nghiên cứu, cũng như phương pháp xây dựng và tối ưu hóa mô hình phù hợp với bài toán đặt ra.

1.3 Đối tương và phạm vi nghiên cứu

Đối tượng nghiên cứu:

- Bệnh đột quỵ.
- Các mô hình học máy được sử dụng phổ biến trong các nghiên cứu khoa học gần đây.

Phạm vi nghiên cứu:

- Về phần tài liệu: Các bài báo khoa học gần đây liên quan đến việc sử dụng mô hình học máy để dự đoán bệnh đột quỵ. Bộ dữ liệu Stroke Prediction Dataset từ Kaggle.
- Thời gian thực hiện đề tài: Từ ngày 15/05/2024 đến ngày 01/08/2024.

1.4 Ý nghĩa khoa học và thực tiễn cửa đề tài

1.4.1 *Ý nghĩa khoa học của đề tài*

Trong lĩnh vực nghiên cứu khoa học, việc khai thác và phân tích dữ liệu đóng vai trò then chốt. Sự kết hợp giữa thông tin thu thập được và các phương pháp thống kê tạo nền tảng vững chắc cho các bước tiến trong khoa học và công nghệ hiện đại.

Nghiên cứu về ứng dụng mô hình học máy trong dự đoán bệnh đột quỵ mang lại nhiều giá trị khoa học đáng kể, đồng thời có tiềm năng ứng dụng cao trong y tế. Một số điểm nổi bật của hướng nghiên cứu này bao gồm:

- Tăng cường khả năng dự báo bệnh đột quỵ thông qua việc áp dụng các mô hình học máy trên bộ dữ liệu y tế chuyên biệt như Stroke Prediction Dataset từ Kaggle.
- Mở rộng phạm vi ứng dụng: Các phương pháp và kết quả nghiên cứu có thể được áp dụng để phát triển mô hình dự đoán cho các bệnh lý khác, không giới hạn ở đột quỵ.
- Đóng góp vào sự phát triển của lĩnh vực học máy bằng cách đánh giá hiệu suất, độ chính xác và độ tin cậy của các mô hình khác nhau trong bối cảnh y tế, đặc biệt là trong dự đoán bệnh đột quỵ.
- Tăng cường hiểu biết về các mô hình học máy và ứng dụng thực tiễn của chúng trong lĩnh vực y tế, cụ thể là trong dự đoán bệnh đột quỵ.

Thông qua việc áp dụng các mô hình học máy vào bài toán cụ thể này, nghiên cứu không chỉ góp phần nâng cao hiểu biết về kỹ thuật mà còn mở ra khả năng ứng dụng rộng rãi trong thực tiễn y tế.

1.4.2 *Ý nghĩa thực tiễn của đề tài*

Đột quỵ hay còn được gọi là tai biến mạch máu não, một trong những nguyên nhân hàng đầu gây tử vong và tàn tật trên toàn cầu. Hiện tượng này xảy ra khi lưu lượng máu đến não bị gián đoạn hoặc suy giảm đáng kể, dẫn đến tình trạng thiếu oxy và dinh dưỡng, gây tổn thương não trong thời gian ngắn.

Tuy nhiên, nhờ những tiến bộ trong y học hiện đại, quan niệm cho rằng đột quỵ không thể điều trị đã không còn phù hợp. Đột quỵ hiện được xem là một tình trạng cấp cứu y tế, đòi hỏi sự can thiệp nhanh chóng và chính xác.

Việc nghiên cứu ứng dụng các mô hình học máy trong dự đoán đột quỵ mang lại nhiều ý nghĩa thực tiễn quan trọng, góp phần tích cực vào sự phát triển của ngành y tế và xã hội. Như được biết, tầm quan trọng của việc dự đoán đột quỵ ngày càng được nhấn mạnh, do tác động nghiêm trọng của nó đối với sức khỏe cộng đồng ngày nay.

Xu hướng sử dụng các mô hình học máy trong dự đoán đột quỵ đang ngày càng phát triển và thu hút sự quan tâm rộng rãi. Các nghiên cứu hiện tại thì tập trung vào việc áp dụng và đánh giá hiệu quả của các mô hình học máy phổ biến, nhằm xác định độ chính xác và khả năng ứng dụng thực tế của chúng trong lĩnh vực này.

1.5 Cơ sở khoa học của việc chọn đề tài

Đề tài nghiên cứu về việc áp dụng các mô hình học máy để dự đoán bệnh đột quỵ được lựa chọn dựa trên nền tảng khoa học chắc chắn và sự liên kết chặt chẽ giữa y học và học máy. Dưới đây là một số lý do khoa học đằng sau việc chọn đề tài này:

- Các mô hình học máy có khả năng học từ dữ liệu lớn và phức tạp.
- Sự sẵn có của dữ liệu lớn về bệnh nhân. Như vậy, việc lựa chọn nghiên cứu về việc sử dụng các mô hình học máy để dự đoán bệnh đột quỵ dựa trên cơ sở khoa học về khả năng học từ dữ liệu, tích hợp

thông tin đa dạng và tiềm năng dự đoán, mang lại nhiều tiềm năng cho việc cải thiện chăm sóc sức khỏe.

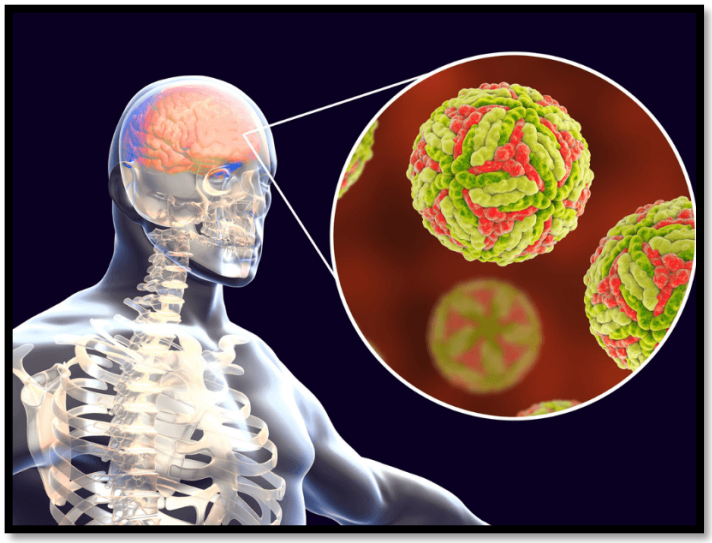
CHƯƠNG 2. TỔNG QUAN VỀ ĐỀ TÀI VÀ CÁC NGHIÊN CỨU CÓ LIÊN QUAN

2.1 Tổng quan về bệnh đột quỵ

2.1.1 *Giới thiệu về bệnh đột quỵ*

Đột quỵ hay còn được biết đến như tai biến mạch máu não là một tình trạng y tế khẩn cấp xảy ra khi lượng máu cung cấp cho một phần não không đủ do một số nguyên nhân cụ thể. Đột quỵ có thể xảy ra khi một động mạch mang máu đến não bị tắc nghẽn bởi mảng bám hoặc khi một động mạch trong não bị vỡ gây ra chảy máu. Trong số các bệnh bệnh tim mạch chính, đột quỵ là một trong những bệnh nguy hiểm nhất, có thể đe dọa đến tính mạng, nhưng nếu được phát hiện sớm, bệnh nhân có thể được cứu sống.

Khi não không nhận được đủ máu và oxy, các tế bào não sẽ bắt đầu chết sau một khoảng thời gian ngắn. Điều này có thể dẫn đến các hậu quả nghiêm trọng như tê liệt, khó nói, mất đi trí nhớ, tổn thương não vĩnh viễn và thậm chí dẫn đến tử vong.



Hình 2.1: Bệnh đột quỵ

(Nguồn: microlife.com.vn)

2.1.2 Các loại đột quỵ thường gặp

Đột quỵ là hậu quả của hai tình trạng chính: thiếu máu não cục bộ và xuất huyết não:

Trường hợp thiếu máu não cục bộ: mạch máu cung cấp máu cho não bị tắc nghẽn.

Trường hợp xuất huyết não: máu thoát khỏi thành mạch và tràn vào bên trong các mô não đột ngột, gây tổn thương đến não bộ. Khi máu từ tổn thương kích thích các mô não thì sẽ gây ra phù não, máu tập trung thành một khối gọi là tụ máu. Tình trạng này làm tăng áp lực lên các mô xung quanh, cuối cùng giết chết các tế bào não và vỡ mạch não.

Dựa trên hai nguyên nhân này, y học đã phân loại đột quỵ thành ba nhóm:

- Đột quỵ nhồi máu não (Đột quỵ thiếu máu não cục bộ)
- Đột quỵ xuất huyết não
- Đột quỵ nhỏ (Cơn thiếu máu não thoáng qua).

Đột quỵ thiếu máu não cục bộ trên thế giới chiếm khoảng đến “87%” tổng số ca đột quỵ. Nguyên nhân gây ra tình trạng này là do máu không thể lưu thông đến não bình thường, do tắc nghẽn gây ra bởi các mảng bám hoặc cục máu đông. Cục máu đông có thể hình thành ở trong hoặc ngoài các mạch máu não. Một số yếu tố làm tăng nguy cơ mắc bệnh bao gồm huyết áp cao, rung nhĩ, bệnh tiểu đường, thừa cân và sử dụng thuốc lá

Đột quỵ xuất huyết não xảy ra khi một động mạch trong não vỡ, dẫn đến máu chảy ra khỏi nhu mô não và gây tổn thương. Hiện tượng này có hai dạng chính: xuất huyết dưới nhện và xuất huyết nội sọ. Tăng huyết áp và vỡ túi phình động mạch não thường là nguyên nhân chính. Các yếu tố rủi ro khác bao gồm việc lạm dụng rượu bia, hút thuốc, sử dụng ma túy, chấn thương sọ não do tai nạn và các bất thường bẩm sinh ở mạch máu não, ...

Cơn thiếu máu não thoáng qua, còn được gọi là cơn đột quỵ nhỏ, có các triệu chứng giống như các loại đột quỵ khác nhưng chỉ kéo dài trong vài phút hoặc biến mất trong 24 giờ. Đây cũng được coi là “đột quỵ cảnh báo” vì nó cho thấy nguy cơ

đột quỵ nghiêm trọng trong tương lai gần. Các cơn đột quỵ nhỏ này cần được can thiệp kịp thời vì rất khó phân biệt với đột quỵ nguy hiểm đe dọa tính mạng.

Các yếu tố tăng nguy cơ mắc đột quỵ bao gồm tuổi cao, tiền sử gia đình có đột quỵ, huyết áp cao, tiểu đường, hút thuốc lá, tăng mỡ máu, bệnh tim mạch và lối sống không lành mạnh.

Việc nhận biết các triệu chứng đột quỵ và đưa người bệnh đến bệnh viện ngay lập tức rất quan trọng để tăng cơ hội phục hồi. Các triệu chứng thông thường của đột quỵ bao gồm tê liệt một bên cơ thể, khó nói, khó thức giấc, chóng mặt và mất cân bằng.

Đột quỵ được coi là một tình trạng khẩn cấp y tế và cần được chăm sóc y tế ngay lập tức.

2.1.3 Phương pháp phát hiện bệnh đột quỵ

Hiện nay, có một số phương pháp phổ biến để phát hiện bệnh đột quỵ, bao gồm:

- **Đánh giá triệu chứng:** Bác sĩ sẽ thu thập thông tin về các triệu chứng mà bệnh nhân gặp phải, chẳng hạn như tê liệt, khó nói, khó cử động, hoặc chóng mặt. Việc này giúp bác sĩ đưa ra đánh giá sơ bộ về khả năng có đột quỵ. Ví dụ, khi một bệnh nhân xuất hiện tình trạng liệt nửa người và rối loạn ngôn ngữ, điều này có thể làm bác sĩ hoài nghi về khả năng xảy ra đột quỵ và từ đó chỉ định thêm các kiểm tra chẩn đoán để làm rõ tình trạng bệnh.
- **Xét nghiệm máu:** Xét nghiệm máu được thực hiện nhằm đánh giá các chỉ số y tế quan trọng và phát hiện các yếu tố nguy cơ như mỡ máu cao, tiểu đường hay tình trạng của viêm nhiễm. Nếu kết quả cho thấy mức mỡ máu cao hoặc tình trạng tiểu đường, những yếu tố này có thể làm tăng khả năng phát triển bệnh đột quỵ.
- **Xét nghiệm huyết động mạch:** Xét nghiệm Doppler là một phương pháp được sử dụng để kiểm tra tình trạng tuần hoàn máu qua các động mạch của não. Xét nghiệm này giúp phát hiện nhanh chóng kịp thời các

vấn đề như sự tắc nghẽn trong động mạch, từ đó góp phần vào việc chẩn đoán đột quỵ.

- **Kiểm tra hình ảnh não:** Các kỹ thuật hình ảnh như cộng hưởng từ (MRI), cắt lớp vi tính (CT scan) và chụp mạch não (angiogram) có thể được sử dụng để xem xét chi tiết về cấu trúc và mô não. Phương pháp này giúp phát hiện tổn thương hoặc sự tắc nghẽn trong não. Ví dụ, khi bệnh nhân được đưa vào máy CT scan, hình ảnh có thể cho thấy một khu vực não bị thiếu máu, chỉ ra sự hiện diện của đột quỵ.
- **Đánh giá chức năng não:** Các phương pháp như điện não đồ (Electroencephalogram – EEG) và các xét nghiệm chức năng thần kinh để đo lường hoạt động điện của não và hệ thần kinh. Ví dụ, EEG có khả năng ghi lại các hoạt động điện trong não và nếu phát hiện ra những bất thường, điều này có thể báo hiệu sự tổn thương ở một khu vực nào đó của não, có thể do đột quỵ gây ra.

Hầu hết các tài liệu nghiên cứu dựa vào hình ảnh chụp MRI và CT. Các phương pháp kiểm tra hình ảnh não đã được đề cập trước đó để phân loại các bệnh tim mạch, bao gồm cả đột quỵ. Đây là những phương pháp tốn kém cho việc chẩn đoán sớm đột quỵ.

Các phương pháp này thường được kết hợp sử dụng để đưa ra chẩn đoán chính xác về đột quỵ. Bác sĩ sẽ dựa vào kết quả từ nhiều loại xét nghiệm để có cái nhìn tổng thể về tình trạng sức khỏe của bệnh nhân và từ đó, đưa ra chẩn đoán cuối cùng.

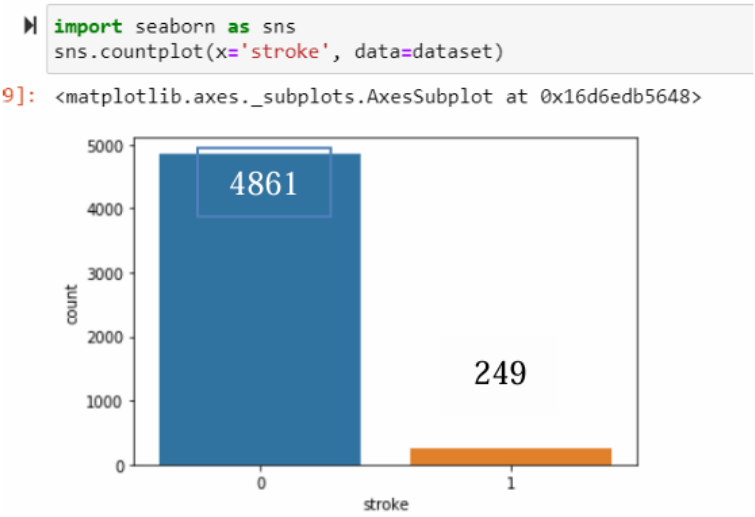
2.2 **Các nghiên cứu liên quan**

Trong nghiên cứu đầu tiên [5], tác giả tập trung vào lý thuyết của các mô hình học máy truyền thống mà không đề cập đến các mô hình học sâu, như chúng em đang nghiên cứu.

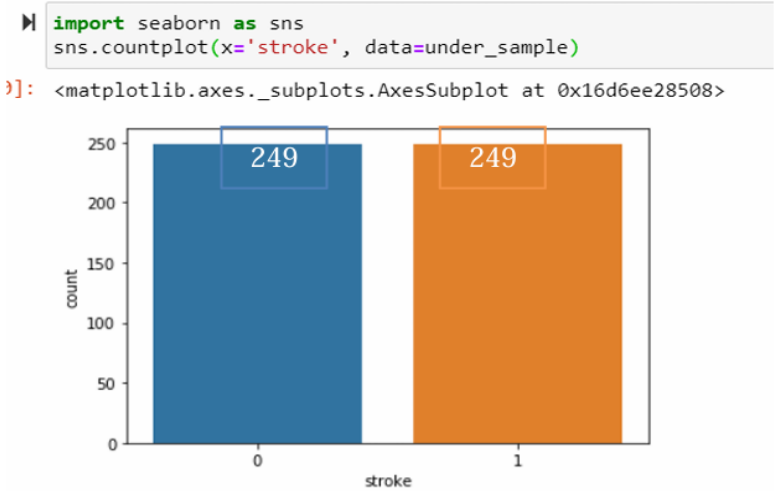
Nghiên cứu này xây dựng 6 mô hình học máy dựa trên các thuật toán được nêu trong bài báo, bao gồm: Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors, Support Vector Machine và Naive Bayes. Bộ dữ liệu được sử

dụng trong nghiên cứu này được chọn từ Kaggle và bao gồm các yếu tố quan trọng như giới tính, tuổi, tình trạng hôn nhân, chỉ số sức khỏe và sử dụng thuốc lá.

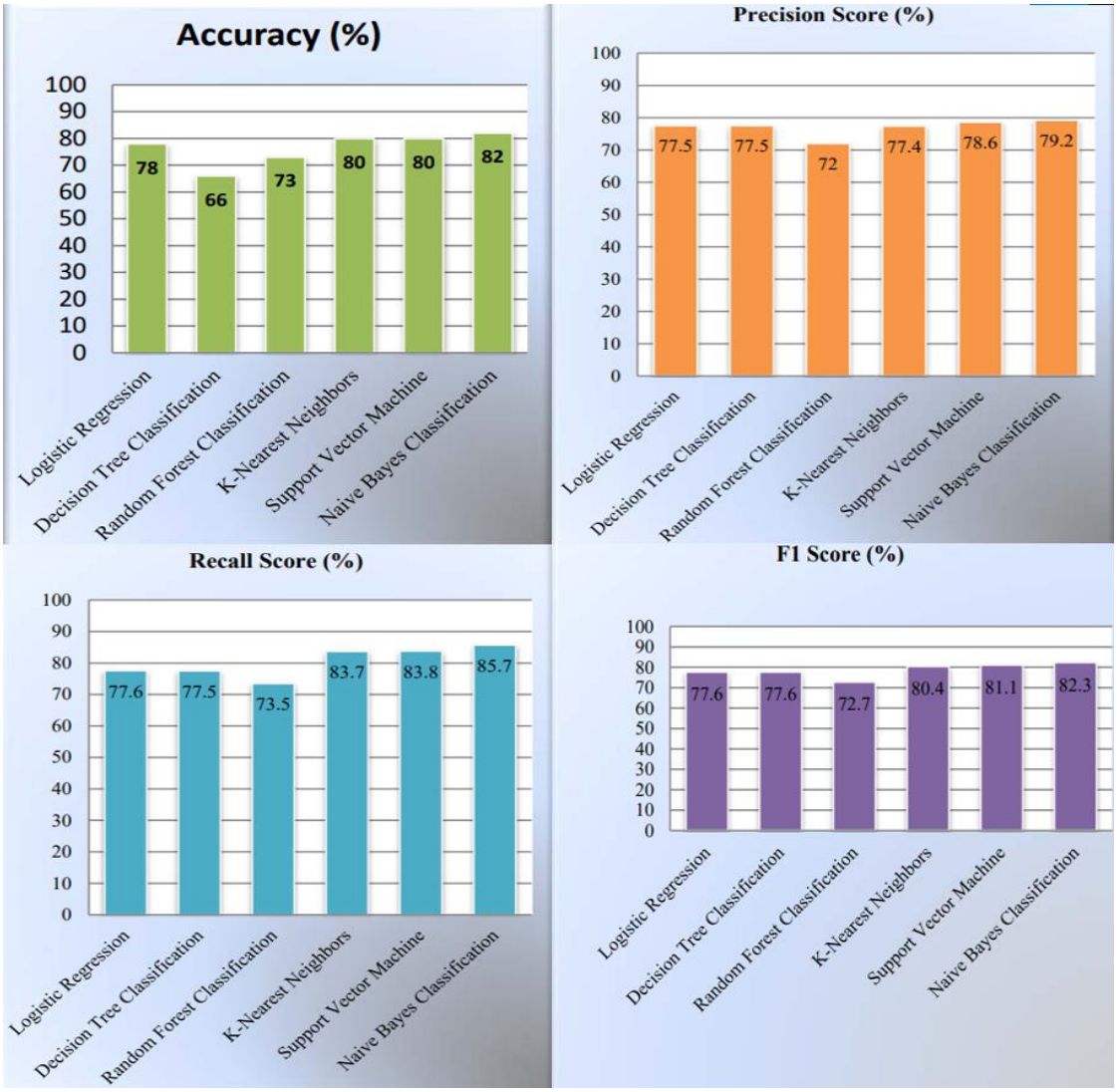
Sau khi thực hiện tiền xử lý dữ liệu, mã hóa nhãn và xử lý dữ liệu mất cân bằng bằng kỹ thuật Undersampling, tác giả tiến hành xây dựng các mô hình học máy. Hiệu suất của các mô hình được so sánh qua nhiều chỉ số, bao gồm độ chính xác, độ chính xác dương tích cực, độ chính xác phân loại sai và đường cong ROC. Sau đây là các hình ảnh mô tả quá trình xử lý dữ liệu và kết quả của các mô hình trong nghiên cứu đầu tiên [5]:



Hình 2.2: Dữ liệu cột stroke.



Hình 2.3: Dữ liệu sau khi áp dụng Undersampling.



Hình 2.4: Kết quả chạy của các mô hình tại nghiên cứu đầu tiên.

Trong nghiên cứu thứ hai này của nhóm chúng em [6], tác giả đã áp dụng các mô hình học máy và Deep Neural Network. Nghiên cứu này sử dụng nhiều mô hình phân loại bao gồm XGBoost, AdaBoost, LightGBM, Random Forest, Decision Tree, Logistic Regression, K Nearest Neighbors, SVM, Naive Bayes và mạng nơ-ron sâu với 3 và 4 lớp. Kết quả cho thấy Random Forest đạt độ chính xác cao nhất (99%) trong các mô hình học máy, trong khi mạng nơ-ron sâu 4 lớp đạt độ chính xác 92,39% trong các phương pháp học sâu.

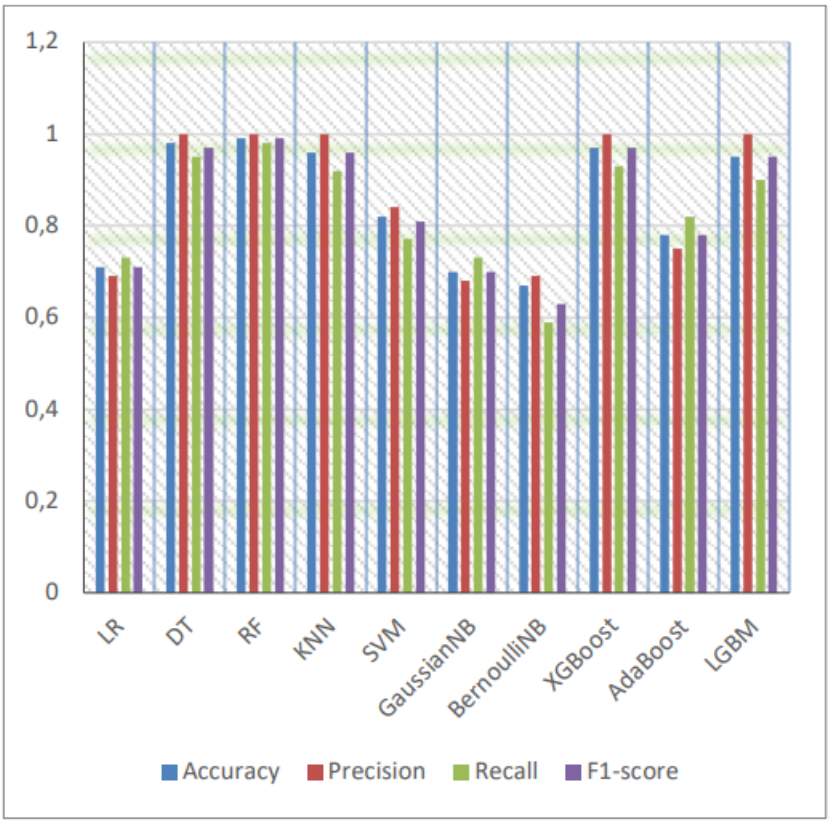
Trong nghiên cứu này, tác giả đã áp dụng phương pháp tiền xử lý dữ liệu khác so với nghiên cứu đầu tiên. Cụ thể, nghiên cứu này sử dụng Oversampling thay vì

Undersampling như trong nghiên cứu trước. Oversampling được chọn vì nó phù hợp hơn với các mô hình phức tạp như Deep Learning hoặc Gradient Boosting, có khả năng học các hình dạng phức tạp của dữ liệu tốt hơn so với các mô hình đơn giản như Logistic Regression.

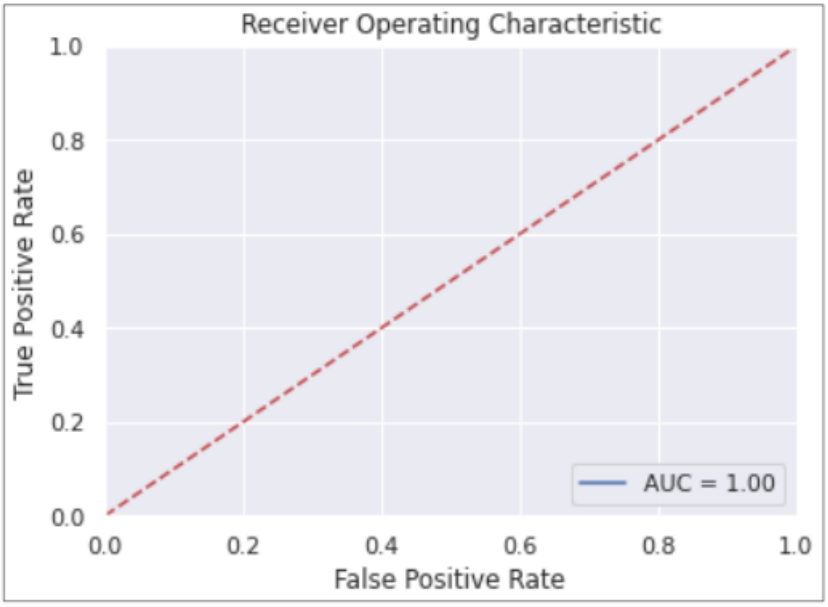
Hiệu suất của các mô hình được so sánh qua nhiều chỉ số, bao gồm độ chính xác, độ chính xác dương tích cực, độ chính xác phân loại sai và đường cong ROC. Dưới đây là các hình ảnh mô tả kết quả chạy của nghiên cứu thứ hai:

Algorithm	Accuracy	Precision	Recall	F1-score	AUC
LR	0.71	0.69	0.73	0.71	0.79
DT	0.98	1.00	0.95	0.97	0.98
RF	0.99	1.00	0.98	0.99	1.00
KNN	0.96	1.00	0.92	0.96	0.98
SVM	0.82	0.84	0.77	0.81	-
GaussianNB	0.70	0.68	0.73	0.70	0.78
BernoulliNB	0.67	0.69	0.59	0.63	0.71
XGBoost	0.97	1.00	0.93	0.97	0.98
AdaBoost	0.78	0.75	0.82	0.78	0.76
LGBM	0.95	1.00	0.90	0.95	0.96

Hình 2.5: Kết quả



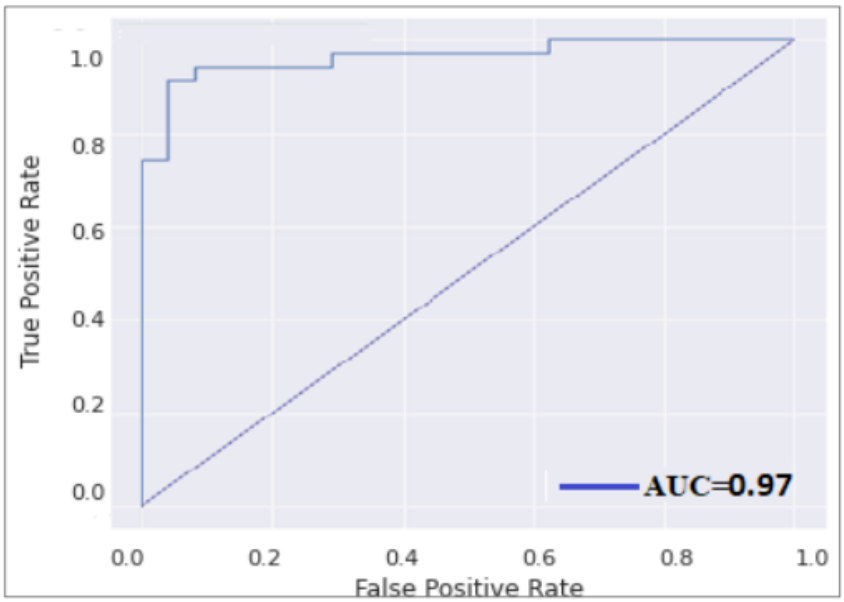
Hình 2.6: Biểu đồ so sánh các chỉ số đánh giá của các thuật toán ML



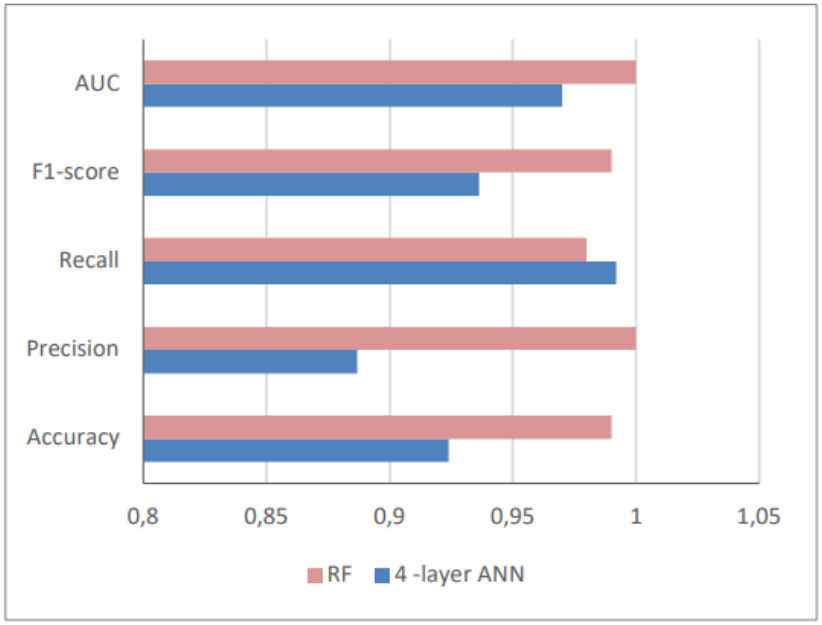
Hình 2.7: Diện tích dưới đường cong ROC cho Random Forest

Algorithm	Accuracy	Precision	Recall	F1-score	AUC
4 -layer ANN	0.9239	0.8867	0.992	0.9364	0.97
3-layer ANN	0.8401	0.7709	0.974	0.8606	0.91

Hình 2.8: So sánh giữa các cách tiếp cận ở học sâu



Hình 2.9: Diện tích dưới đường cong của phương pháp ANN 4 lớp



Hình 2.10: So sánh giữa Random Forest và ANN 4 lớp

2.3 Những thách thức chưa được giải quyết trong nghiên cứu liên quan

Đầu tiên, mặc dù các vấn đề liên quan đến sự mất cân đối trong bộ dữ liệu đã được giải quyết một cách hiệu quả, đạt được kết quả ấn tượng với thuật toán Random Forest có độ chính xác lên tới 99%, nhưng cả hai nghiên cứu đều chưa khai thác hết tiềm năng khi chỉ sử dụng dữ liệu dạng số. Một hướng tiếp cận có thể cải thiện là thu thập dữ liệu hình ảnh, như từ chụp CT não để nâng cao khả năng dự đoán đột quỵ trong tương lai.

Thứ hai, cả hai nghiên cứu đều tập trung vào việc đánh giá hiệu suất của mô hình mà không đi sâu vào phân tích khám phá dữ liệu, điều này cần thiết để hiểu rõ hơn về bản chất của dữ liệu gốc. Phân tích dữ liệu sâu hơn có thể giúp xác định các biến số quan trọng cho việc dự đoán và cân nhắc các phương pháp xử lý dữ liệu mất cân bằng một cách hiệu quả hơn, từ đó tối ưu hóa hiệu suất của mô hình.

Mặc dù vẫn còn những hạn chế, không thể phủ nhận rằng cả hai nghiên cứu đã cung cấp cái nhìn toàn diện về quy trình nghiên cứu và đề xuất các phương pháp tiếp cận có giá trị trong việc dự đoán đột quỵ.

2.4 Những vấn đề cần được khắc phục trong nghiên cứu

Khi tiến hành nghiên cứu và triển khai các mô hình học máy trong việc dự báo đột quỵ, việc giải quyết một loạt các vấn đề chủ chốt là điều cần thiết để cải thiện chất lượng và tính ứng dụng của mô hình. Sau đây là danh sách các vấn đề then chốt cần được quan tâm:

- **Phân tích và Khám phá dữ liệu**: Thực hiện phân tích để hiểu rõ phân phối của các biến số chính như tuổi, huyết áp và lượng đường trong máu, … có trong bộ dữ liệu. Điều này giúp tìm ra mối liên hệ giữa các yếu tố nguy cơ và bệnh đột quỵ. Kết quả từ việc phân tích này có thể hỗ trợ việc xác định các biến số quan trọng cho mô hình dự đoán và xem xét các giải pháp cho vấn đề dữ liệu không cân bằng, nhằm đảm bảo mô hình hoạt động hiệu quả.
- **Dữ liệu không cân bằng**: Dữ liệu đột quỵ thường không cân bằng, với số lượng mẫu âm tính cao hơn đáng kể so với mẫu dương tính. Điều

này có thể dẫn đến việc mô hình phân loại không chính xác đối với lớp thiểu số. Việc áp dụng các phương pháp như undersampling và oversampling là cần thiết để cân bằng lại bộ dữ liệu.

- **Xử lý Dữ liệu Thiếu và Nhiễu**: Trong lĩnh vực y tế, dữ liệu thường không hoàn chỉnh và chứa nhiễu. Việc xử lý cẩn thận các giá trị thiếu và loại bỏ nhiễu là cần thiết để tăng cường độ tin cậy của mô hình.
- **Chọn lọc và Đánh giá Mô hình**: Cần tiến hành nghiên cứu so sánh giữa các mô hình học máy để tìm ra mô hình phù hợp nhất cho việc dự đoán đột quỵ. Điều này bao gồm việc áp dụng các kỹ thuật đánh giá như kiểm định chéo (cross-validation) và phân tích đường cong ROC để đánh giá hiệu suất mô hình một cách toàn diện.
- **Giải thích mô hình:** Việc giải thích cách thức hoạt động và quyết định của mô hình là điều cực kỳ quan trọng. Điều này không chỉ giúp các bác sĩ và chuyên gia y tế hiểu rõ nguyên nhân một bệnh nhân được phân loại theo một cách nhất định mà còn cung cấp cơ sở vững chắc cho các quyết định lâm sàng.
- **Triển khai và Ứng dụng thực tế:** Sau khi mô hình được phát triển, việc triển khai và tích hợp nó vào hệ thống y tế thực tế là bước tiếp theo quan trọng. Điều này đảm bảo rằng mô hình có thể được sử dụng để hỗ trợ việc dự đoán và ngăn chặn đột quỵ, góp phần vào việc cải thiện sức khỏe cộng đồng. Đây là những bước cần thiết để đảm bảo mô hình không chỉ có giá trị lý thuyết mà còn có khả năng ứng dụng thực tiễn cao.

CHƯƠNG 3. CƠ SỞ LÝ THUYẾT

3.1 **Tổng quan về học máy**

3.1.1 ***Học máy là gì?***

Học máy là một lĩnh vực thuộc AI và khoa học máy tính, học máy tập trung vào việc phát triển các phương pháp cho phép máy móc tự động học hỏi từ dữ liệu để giải quyết các bài toán cụ thể. Trọng tâm của học máy là tạo ra và tối ưu các thuật toán phức tạp để thực hiện các phép tính. Lĩnh vực này có nhiều ứng dụng đa dạng, từ việc nhận diện hình ảnh, dự báo tài chính đến xử lý ngôn ngữ tự nhiên và nhiều lĩnh vực khác nữa.

3.1.2 Các loại học máy chính

Học có giám sát (Supervised Learning):

Định nghĩa: Phương pháp học mà trong đó mô hình được huấn luyện trên dữ liệu đã được gán nhãn.

Đặc điểm:

- Dữ liệu huấn luyện gồm cặp đầu vào và đầu ra mong muốn.
- Mục tiêu là học ánh xạ từ đầu vào sang đầu ra.

Ứng dụng*:* Phân loại, hồi quy

Ví dụ: Nhận dạng chữ viết tay, dự đoán giá nhà, ...

Học không giám sát (Unsupervised Learning):

Định nghĩa: Phương pháp học từ dữ liệu không được gán nhãn.

Đặc điểm:

- Tìm kiếm cấu trúc hoặc mẫu ẩn trong dữ liệu.
- Không có đầu ra mục tiêu cụ thể.

Ứng dụng: Phân cụm, giảm chiều dữ liệu, phát hiện bất thường.

Ví dụ: Nén dữ liệu, phân khúc khách hàng, ...

Học tăng cường (Reinforcement Learning):

Định nghĩa: Phương pháp học dựa trên tương tác với môi trường và phản hồi.

Đặc điểm:

- Học cách đưa ra quyết định tối ưu.
- Học thông qua thử nghiệm và sai lầm.

Ứng dụng: Trò chơi, Robot tự hoạt động, tối ưu hóa quy trình.

Ví dụ: Xe tự lái, ...

Các phương pháp bổ sung:

Học bán giám sát (Semi-supervised Learning):

- Kết hợp dữ liệu có nhãn và không nhãn.
- Có lợi khi có ít dữ liệu được gán nhãn.

Học sâu (Deep Learning):

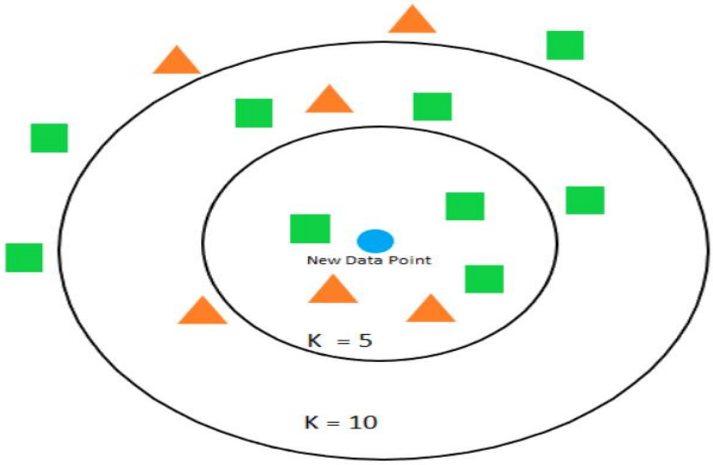
- Sử dụng mạng nơ-ron nhiều lớp.
- Xử lý dữ liệu phức tạp hiệu quả trong văn bản, hình ảnh, âm thanh.

Mỗi phương pháp đều có các ưu điểm và thách thức riêng, phù hợp với các loại bài toán và dữ liệu khác nhau.

3.2 Một số mô hình học máy phổ biến

3.2.1 *K-Nearest Neighbor (KNN)*

KNN là viết tắt của thuật toán K-Nearest Neighbors trong khai phá dữ liệu và học máy. KNN biết đến như là một trong những thuật toán supervised-learning đơn giản nhất trong Machine Learning (ML).



Hình 3.1: Mô tả thuật toán KNN

(Nguồn: www.debuggercafe.com)

Ý tưởng cơ bản của thuật toán KNN là dựa vào việc tính khoảng cách giữa điểm dữ liệu đang cần phân loại hoặc dự đoán và các điểm dữ liệu đã có nhãn hoặc giá trị dự đoán trong bộ dữ liệu train. KNN chọn "k" những điểm dữ liệu gần nhất với điểm đang xét và sử dụng thông tin từ những điểm này để đưa ra quyết định về lớp hoặc giá trị của điểm đó.

Trong quá trình training, thuật toán này không học một điều gì từ tập huấn luyện, mà mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới.

K-Nearest Neighbors thuộc neighbors trong thư viện sklearn.

Kết quả đánh giá sẽ dựa trên RMSE, R2_score. K-NN trong Python: sklearn.neighbors.KneighborsRegressor(), ngoài ra còn có K-NN classifier (KNeighborsClassifier()).

Ưu điểm:

- Đơn giản và dễ hiểu: Vì KNN không đòi hỏi quá nhiều kiến thức chuyên sâu về học máy.
- Không cần được huấn luyện trước.
- Phân loại tốt nếu số lượng mẫu đủ lớn.

Nhược điểm:

- Khó khăn trong việc lựa chọn tham số k.
- Không có giai đoạn học tập, mọi việc đều thực hiện trong quá trình kiểm tra.

Ứng dụng:

- Phân loại tài liệu.
- Nén ảnh.
- Phân tích dữ liệu địa lý.
- Phân khúc khách hàng trong marketing.

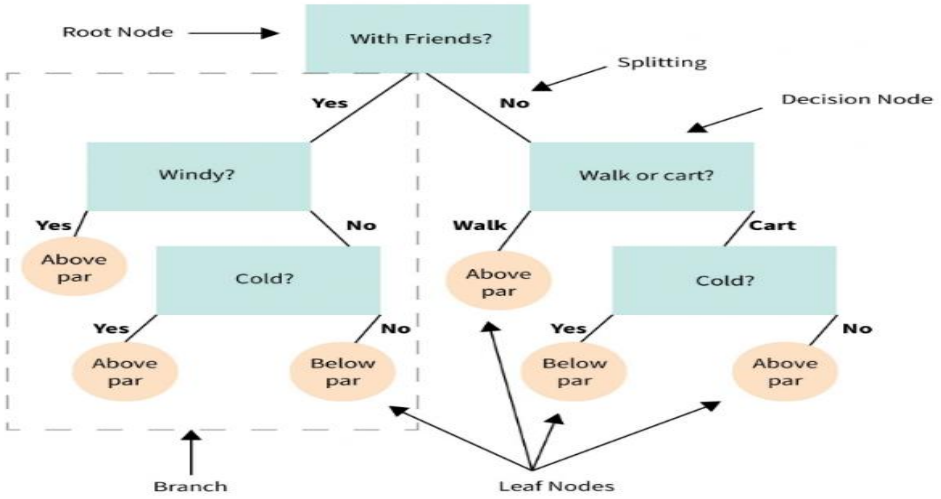
Một số tham số phổ biến của KNN:

- *n_neighbors (k):* Đây là số lượng láng giềng gần nhất mà thuật toán KNN sẽ xem xét khi thực hiện dự đoán hoặc phân loại.

- *weights*: Tham số này quyết định cách mà các láng giềng ảnh hưởng đến dự đoán. Mặc định là “uniform”, hàm trọng lượng được sử dụng trong dự đoán. Ngoài ra còn có tùy chọn là “distance”.
- *algorithm*: Tham số này xác định cách tính khoảng cách và tìm láng giềng. Mặc định là “auto”, “auto” sẽ tự động chọn thuật toán phù hợp nhất dựa trên tập dữ liệu. Ngoài ra còn có các tùy chọn khác cho tham số này “ball_tree”, “kd_tree”, “brute”.
- Ngoài ra còn có: *p, leaf_size, n_jobs, ...*

3.2.2 Decision Tree (DT)

Decision Tree (DT) là một mô hình học có giám sát, DT có thể được áp dụng cho phân loại (Classification) và hồi quy (Regression). Thuật toán DT hoạt động bằng cách tạo ra một biểu đồ có cấu trúc giống cây để mô hình hóa quá trình ra quyết định và kết quả có thể có.



Hình 3.2: Ví dụ thuật toán Decision Tree

(Nguồn: www.masterdatascience.org)

Cách thức hoạt động của Decision Tree:

1. Chọn thuộc tính.
2. Phân chia dữ liệu.
3. Lặp lại bước 1 và bước 2.

4. Xác định nhãn.

Ưu điểm:

- Decision Tree khá dễ hiểu và diễn giải.
- Cần ít dữ liệu để train.
- Dữ liệu dạng số bao gồm rời rạc và liên tục, dữ liệu hạng mục có thể được DT xử lý tốt.

Nhược điểm:

- Không chắc chắn xây dựng được cây tối ưu.
- Dễ bị quá khớp (Overfitting): Cây quyết định có thể trở nên phức tạp quá mức và phân chia dữ liệu một cách quá chính xác dẫn đến hiện tượng quá khớp.
- Thuộc tính có nhiều giá trị thường sẽ được ưu tiên. Tuy nhiên có thể khắc phục bằng cách sử dụng Gain Ratio.

Ứng dụng:

- Chuẩn đoán y tế.
- Dự đoán rủi ro tín dụng.
- Phân tích hành vi khách hàng.

Decision Tree là một mô hình học máy linh hoạt và trực quan, tạo nên nền tảng cho nhiều thuật toán phức tạp hơn trong học máy.

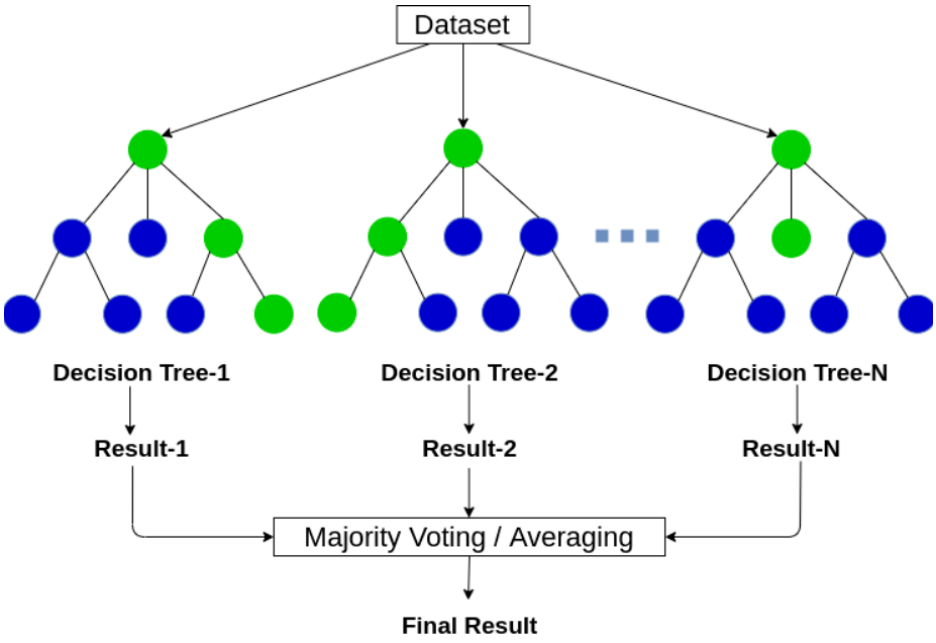
3.2.3 Random Forest (RF)

Random Forest (RF) là một phương pháp học máy tiên tiến dựa trên việc kết hợp nhiều cây quyết định để tạo ra một mô hình mạnh mẽ hơn và ổn định hơn. Là một mô hình được tối ưu hóa từ Decision Trees, với nhiều cải tiến đáng kể. Có nhiều ưu điểm đối với loại mô hình này bao gồm tốc độ huấn luyện nhanh chóng, khả năng mở rộng và đặc tính hội tụ (tính chất không lặp lại của mô hình).

Ý tưởng của thuật toán này chính là tạo ra *một tập hợp cây quyết định (Decision Trees)* mà từng cây được huấn luyện dựa vào *nhiều mẫu con* khác nhau và cho ra kết quả dự báo là voting từ toàn bộ Decision Trees.

Random Forest là một phương pháp “ensemble learning”. Trong sklearn, sử dụng RandomForestClassifier cho bài toán phân loại và RandomForestRegressor cho bài toán hồi quy.

Random Forest còn sử dụng kỹ thuật "feature randomness": Là mỗi cây chỉ xem xét một tập con ngẫu nhiên của các đặc trưng khi tìm kiếm Split tốt nhất tại mỗi Node. Việc này giúp tăng tính đa dạng giữa các cây và làm cho mô hình tổng thể mạnh mẽ hơn, ít bị Overfitting hơn.



Hình 3.3: Random Forest

(Nguồn: medium.com)

Ưu điểm:

- Không gặp phải vấn đề về Overfitting.
- Có thể sử dụng cho cả hai bài toán phân loại và hồi quy.
- Hoạt động tốt trên dữ liệu lớn mà không cần quá nhiều tiền xử lý dữ liệu.

Nhược điểm:

- Đào tạo chậm so với các thuật toán khác.
- Mô hình có thể trở nên quá phức tạp khi số lượng cây tăng lên và vì thế nên tốn nhiều tài nguyên tính toán.

- Thuật toán hoạt động kém hơn các phương pháp tuyến tính khi một tập hợp có nhiều đặc điểm thưa thớt.

Ứng dụng:

- Phân loại hình ảnh và nhận dạng đối tượng.
- Dự đoán giá cả trong tài chính.
- Dự đoán về bệnh và phân tích các dữ liệu y tế.
- Phân tích lựa chọn của người tiêu dùng trong Marketing.
- Random Forest là một trong những thuật toán mạnh mẽ và phổ biến trong học máy, đặc biệt là hiệu quả trong nhiều bài toán thực tế.

Một số tham số cho mô hình RF:

- n_estimators = 100: mặc định là 100 (Số lượng cây).
- max_depth = None: mặc định là None (Chiều sâu tối đa của cây).
- random_state = None: mặc định là None (Kiểm soát tính ngẫu nhiên trong thuật toán).

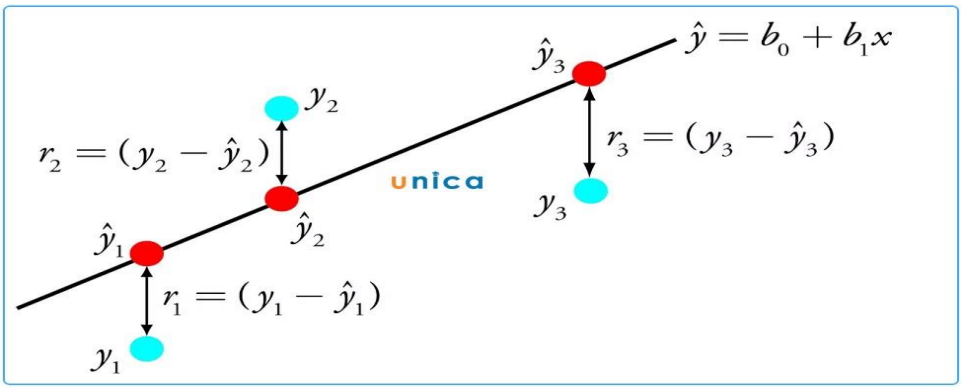
3.2.4 Linear Regression (LR)

Hồi quy Tuyến tính (Linear Regression - LR) là một thuật toán đơn giản nhưng mạnh mẽ và được sử dụng rộng rãi trong khoa học dữ liệu.

Linear Regression là một loại phân tích thống kê được sử dụng để dự đoán mối quan hệ giữa hai biến. Nó giả định mối quan hệ tuyến tính giữa biến độc lập và biến phụ thuộc và nhằm mục đích tìm ra đường phù hợp nhất để mô tả mối quan hệ. Đường này được xác định bằng cách giảm thiểu tổng bình phương chênh lệch giữa giá trị dự đoán và giá trị thực tế.

Phương pháp này giả định rằng mối quan hệ giữa các biến có thể được xấp xỉ bằng một phương trình tuyến tính.

Linear Regression (LR) là một trong những thuật toán cơ bản và phổ biến nhất của Supervised Learning trong đó đầu ra dự đoán là liên tục.



Hình 3.4: Linear Regression

(Nguồn: Unica)

Trong Linear Regression sẽ có 2 bài toán đó là *Hồi quy đơn biến* và *Hồi quy đa biến*.

Phương trình hồi quy đơn biến có dạng như phương trình sau đây:

$$y = ax + b$$

với *x*: biến độc lập, *y*: biến phụ thuộc *x*.

Còn Hồi quy tuyến tính đa biến là sẽ có nhiều biến độc lập *x1, x2, ..., xn* và nhiều hệ số *a1, a2, ..., an* thay vì một biến *x* duy nhất.

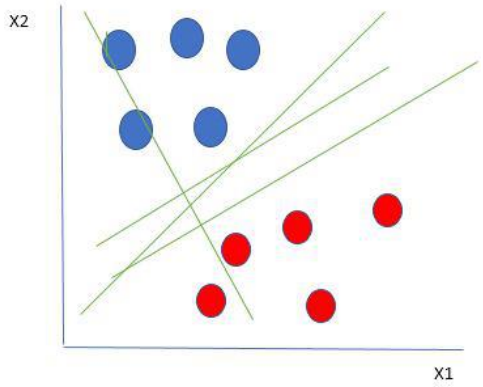
3.2.5 Support Vector Machines – SVM

Support Vector Machines (SVM) là một thuật toán học có giám sát và được sử dụng cho cả bài toán phân loại nhưng cũng có thể áp dụng cho hồi quy. Support Vector Machines thường được dùng cho bài toàn phân loại.

Khi sử dụng cho phân loại, gọi là Support Vector Classifier.

Khi sử dụng cho hồi quy, gọi là Support Vector Regression.

SVM là một trong những phương pháp đa năng và phổ biến. Sự linh hoạt và hiệu quả của SVM trong việc giải quyết các vấn đề đa dạng hay phức tạp đã giúp mô hình được ứng dụng rộng rãi trong nhiều ngành khoa học và công nghiệp khác nhau.



Hình 3.5: Support Vector Machine

(Nguồn cit.ctu.edu.vn)

Là quá trình tìm kiếm một mặt phẳng phân chia tối ưu nhất trong không gian nhiều chiều, gọi là siêu phẳng (Hyperplane). Mục tiêu là tạo ra một ranh giới phân loại rõ ràng nhất giữa các nhóm dữ liệu.

SVM đặc biệt chú trọng vào việc tối đa hóa khoảng cách, hay còn gọi là Margin, giữa siêu phẳng và các mẫu dữ liệu gần nó nhất của mỗi lớp. Những mẫu dữ liệu nằm sát ranh giới này là các (Support Vectors), đóng vai trò quyết định trong việc xác định vị trí của siêu phẳng.

Để xử lý các trường hợp dữ liệu không thể phân tách tuyến tính thì SVM sẽ sử dụng kỹ thuật ánh xạ dữ liệu vào không gian có số chiều cao hơn thông qua các hàm nhân. Trong không gian mới này thì việc tìm kiếm một siêu phẳng phân tách trở nên khả thi hơn và cho phép SVM giải quyết các bài toán phân loại phức tạp một cách hiệu quả.

Ưu điểm:

- Mạnh mẽ trong việc phòng chống Overfitting.
- Hiệu quả trong không gian nhiều chiều.
- Linh hoạt thông qua việc sử dụng các Kernel khác nhau.
- Chỉ sử dụng một tập con của các điểm huấn luyện (Support Vectors) trong quyết định cuối cùng.

Nhược điểm:

- Không thể xử lý tốt với dữ liệu lớn do thời gian huấn luyện cao.
- Kém hiệu quả khi dữ liệu có nhiều nhiễu.
- Cần điều chỉnh cẩn thận các tham số để đạt hiệu suất tốt.
- Chưa thể hiện rõ tính xác suất.

Một số tham số quan trọng cho mô hình SVM:

- C (Regularization Parameter):
- Điều chỉnh sự cân bằng giữa việc tối đa hóa biên và tránh phân loại sai.
- Giá trị C nhỏ: Margin rộng, nhiều lỗi phân loại hơn.
- Giá trị C lớn: Margin hẹp, ít lỗi phân loại.
- Kernel:
- Linear: Cho dữ liệu tuyến tính phân tách.
- RBF (Radial Basis Function): Phổ biến cho dữ liệu phi tuyến.
- Polynomial: Cho dữ liệu phi tuyến, có thể điều chỉnh độ.
- Sigmoid: Ít được sử dụng.
- Gamma (cho Kernel RBF):
- Định nghĩa mức độ ảnh hưởng của một mẫu đơn lẻ.
- Giá trị lớn: ảnh hưởng "gần", có thể dẫn đến overfitting.
- Giá trị nhỏ: ảnh hưởng "xa", có thể underfitting.
- Degree (cho Kernel đa thức):
- Xác định độ của hàm đa thức trong kernel

Lựa chọn tham số:

Thường sử dụng Grid Search với Cross-validation để tìm bộ tham số tốt nhất.

SVM là một thuật toán mạnh mẽ, linh hoạt và đặc biệt là hiệu quả trong nhiều bài toán phân loại và hồi quy phức tạp.

Ứng dụng:

- Phân tích dữ liệu Marketing.
- Nhận dạng: chữ viết tay, ảnh, tiếng nói.
- Phân loại văn bản ...

3.2.6 Gradient Boosting

Gradient Boosting là một kỹ thuật trong học máy thuộc họ Boosting, mục tiêu của nó là tạo ra một mô hình dự đoán mạnh mẽ bằng cách kết hợp nhiều mô hình đơn giản lại với nhau. Ý tưởng chính là xây dựng các mô hình liên tiếp sao cho mỗi mô hình mới tối ưu hóa khắc phục những lỗi mà các mô hình trước đó đã dự đoán sai.

Một số tham số cho Gradient Boosting:

- n_estimators: Đây là số lượng cây quyết định (mô hình yếu) trong chuỗi Gradient Boosting. Một giá trị lớn có thể cải thiện hiệu suất của mô hình, nhưng cũng làm tăng thời gian huấn luyện.
- learning_rate: Tham số này điều chỉnh mức độ đóng góp của mỗi mô hình yếu vào mô hình tổng. Giá trị nhỏ có thể giúp tránh overfitting nhưng cũng làm giảm tốc độ hội tụ của mô hình.
- random_state.

3.2.7 Light Gradient Boosting Machine (LGBM)

LGBM (Light Gradient Boosting Machine) là một thuật toán học máy thuộc lớp Gradient Boosting, được phát triển bởi Microsoft và sau đó được phổ biến rộng rãi trong cộng đồng dữ liệu lớn và học máy. Nó là một thuật toán cải tiến của Gradient Boosting và là một trong những thuật toán thường được sử dụng trong các cuộc thi và ứng dụng thực tiễn.

LGBM là thuật toán ensemble learning, trong đó nhiều cây quyết định nhỏ (cây con) được xây dựng và tổng hợp lại để hình thành một mô hình mạnh hơn.

Ưu điểm:

- Tốc độ huấn luyện nhanh.
- Hỗ trợ parallel learning.
- LGBM thường có hiệu suất tốt với dữ liệu lớn.

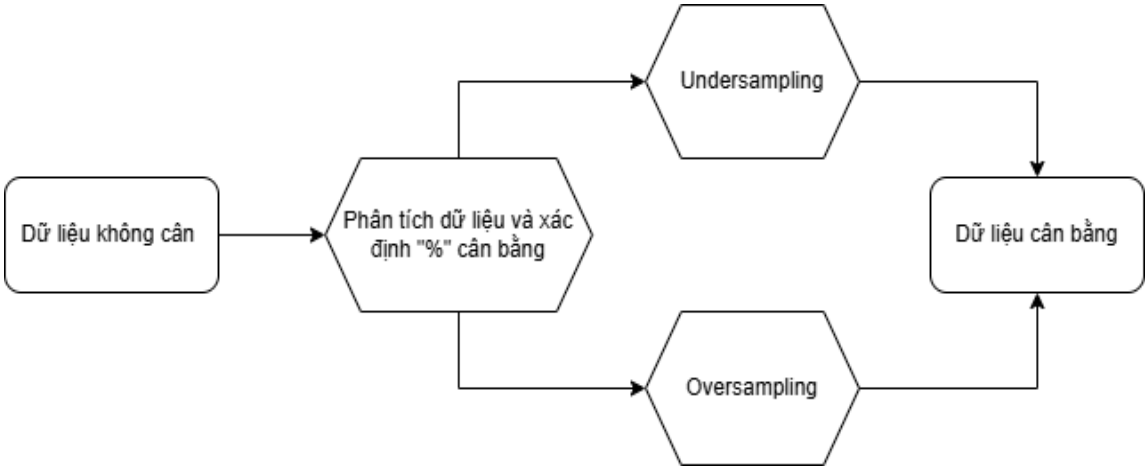
Nhược điểm:

- Khó hiểu đối với người mới bới khi sử dụng cần phải thay đổi tham số để có thể có hiệu suất tốt hơn.
- Dễ bị overfitting.

3.3 **Xử lý dữ liệu không cân bằng**

Việc không cân bằng dữ liệu là điều mà đối với bất cứ ai mà thường xuyên làm việc với dữ liệu ắt hẳn sẽ gặp phải không ít thì nhiều. Đặc biệt là đối với dữ liệu trong đề tài chúng em đang nghiên cứu chính là dữ liệu của bệnh nhân để dự đoán bệnh đột quỵ thì việc phải cân bằng dữ liệu là điều tất yếu.

Cân bằng dữ liệu chính là yếu tố đặc biệt quan trọng trong quá trình xử lý và đào tạo mô hình học máy. Đặc biệt chính là quá chênh lệch ở cột dự đoán trong tập dữ liệu giữa các lớp hay giá trị.



Hình 3.6: Mô hình xử lý dữ liệu không cân bằng

Trong quá trình xử lý dữ liệu, hệ thống sẽ phân tích từng bộ dữ liệu để xác định các số lượng mẫu trong mỗi nhóm (có lỗi và không có lỗi) cũng như mức độ chênh lệch giữa chúng. Từ kết quả phân tích, mỗi phương pháp sẽ có những cách tiếp cận riêng để cân bằng dữ liệu. Ví dụ như: Oversampling sẽ bổ sung thêm mẫu, Undersampling sẽ loại bớt mẫu, còn SMOTE sẽ tạo ra các mẫu mới. Sau đó, các chiến lược này sẽ được triển khai cho từng tập dữ liệu cụ thể.

Sự mất cân bằng trong dữ liệu có thể gây ảnh hưởng tiêu cực đến hiệu suất của các mô hình học máy. Khi được huấn luyện trên những tập dữ liệu không cân bằng, các mô hình này thường cho kết quả dự đoán kém chính xác. Hậu quả là những mô hình này có thể không đạt được chất lượng mong muốn, nên dẫn đến việc các nhà phát triển phải bỏ ra thêm thời gian và nguồn lực để giải quyết những vấn đề phát sinh từ dự đoán sai lệch của mô hình.

Những biện pháp xử lý sự mất cân bằng trong phân phối dữ liệu giữa các lớp có thể được phân thành hai nhóm chính: kỹ thuật tăng mẫu và kỹ thuật giảm mẫu.

Kỹ thuật tăng mẫu còn gọi là Oversampling, nó tập trung vào việc làm phong phú thêm dữ liệu của những lớp có ít mẫu. Ngược lại, kỹ thuật giảm mẫu Undersampling thì hướng đến việc giảm bớt số lượng mẫu từ những lớp chiếm ưu thế về số lượng.

Mục tiêu chung của cả hai nhóm kỹ thuật này là tạo ra một tập dữ liệu có sự phân bố cân đối hơn giữa các lớp, từ đó cải thiện hiệu quả của quá trình học máy và phân loại.

3.4 Kỹ thuật Undersampling

Undersampling là một tập hợp các phương pháp nhằm giải quyết vấn đề mất cân bằng trong phân phối dữ liệu giữa các lớp. Kỹ thuật này tập trung vào việc giảm bớt số lượng mẫu từ lớp chiếm ưu thế vượt trội (lớp đa số) để tạo ra sự cân bằng hơn với lớp ít mẫu hơn (lớp thiểu số). Mục tiêu của Undersampling là điều chỉnh tỷ lệ giữa các lớp, giúp cải thiện đáng kể sự chênh lệch trong dữ liệu.

Ví dụ, phương pháp này có thể biến đổi một tập dữ liệu có tỷ lệ mất cân bằng nghiêm trọng như 1:100 thành các tỷ lệ thuận lợi hơn như là 1:10, 1:2 hoặc thậm chí là 1:1 trong một số trường hợp. Bằng cách loại bỏ có chọn lọc các mẫu từ lớp đa số.

Undersampling sẽ giúp tạo ra một tập dữ liệu gọn nhẹ hơn và cân đối hơn. Điều này giúp chúng ta có thể nâng cao hiệu suất của các mô hình học máy, đặc biệt là trong việc nhận diện và xử lý các mẫu thuộc lớp thiểu số.

Bảng 3.1: Các phương pháp Undersampling

Random Undersampling	- Là một phương pháp đơn giản và trực tiếp trong nhóm kỹ thuật Undersampling. - Tập trung vào lớp có số lượng mẫu vượt trội (lớp đa số). - Tiến hành lựa chọn ngẫu nhiên một số mẫu từ lớp đa số này. - Loại bỏ các mẫu đã được chọn ra khỏi tập dữ liệu dùng để đào tạo mô hình. - Đặc điểm nổi bật của kỹ thuật này là tính ngẫu nhiên trong việc chọn lựa và loại bỏ mẫu. Không có tiêu chí cụ thể nào được áp dụng để quyết định mẫu nào sẽ bị loại bỏ ngoài yếu tố ngẫu nhiên. - Mục tiêu chính của Random Undersampling là giảm kích thước của lớp đa số, từ đó tạo ra sự cân bằng hơn giữa các lớp trong tập dữ liệu. - Tuy nhiên, cần lưu ý rằng phương pháp này có thể dẫn đến mất mát thông tin do việc loại bỏ mẫu một cách không có chọn lọc.
Cluster	- Là một kỹ thuật Undersampling nâng cao, nhằm mục đích tạo ra các nhóm đại diện cho toàn bộ tập dữ liệu. - Phân chia dữ liệu của lớp đa số thành nhiều cụm (clusters). - Mỗi cụm sẽ được thiết kế để phản ánh một khía cạnh hoặc là đặc tính cụ thể của dữ liệu mà ta quan tâm. - Từ các cụm này, có thể chọn lọc một số lượng nhất định để đưa vào quá trình nghiên cứu hoặc huấn luyện mô hình.

	- Ưu điểm chính của phương pháp này là khả năng bảo toàn cấu trúc tổng thể của dữ liệu, ngay cả khi giảm đáng kể số lượng mẫu. Bằng cách tập trung vào các cụm đại diện, có thể duy trì được những đặc trưng quan trọng của tập dữ liệu gốc, đồng thời sẽ giảm thiểu nguy cơ mất mát thông tin quan trọng. - Phương pháp này đặc biệt hữu ích khi cần cân bằng giữa việc giảm kích thước dữ liệu và duy trì tính đại diện của mẫu.
Tomek links	- Là một kỹ thuật Undersampling tiên tiến, tập trung vào việc làm sáng tỏ ranh giới giữa các lớp dữ liệu. - Xác định và xử lý các vùng giao thoa không mong muốn giữa lớp đa số và lớp thiểu số. - Quy trình loại bỏ có chọn lọc: Các mẫu thuộc lớp đa số sẽ được xóa bỏ một cách có kế hoạch. - Tiêu chí dừng: Quá trình này tiếp tục cho đến khi các cặp mẫu gần nhau nhất đều thuộc cùng một lớp. - Mục tiêu chính của Tomek links là tăng cường sự phân biệt giữa các lớp. - Bằng cách này, kỹ thuật giúp: - Làm nổi bật đặc trưng của lớp thiểu số. - Tạo ra một ranh giới rõ ràng hơn giữa các lớp. - Giảm thiểu khả năng nhầm lẫn trong quá trình phân loại. - Kết quả là các vùng dữ liệu của lớp thiểu số trở nên dễ nhận diện và phân biệt hơn, nâng cao hiệu suất của mô hình học máy và đặc biệt trong việc nhận dạng chính xác các mẫu thuộc lớp thiểu số.

3.5 Kỹ thuật Oversampling

Oversampling là phương pháp đối lập với Undersampling, mục đích nhằm cân bằng tập dữ liệu bằng cách tăng cường số lượng mẫu trong lớp thiểu số. Đặc điểm chính của Oversampling bao gồm:

Mục tiêu: Tăng số lượng mẫu của lớp ít dữ liệu hơn.

Phương pháp cơ bản: Nhân bản các mẫu hiện có của lớp thiểu số.

Hạn chế: Không tạo ra thông tin mới, chỉ làm tăng số lượng mẫu hiện có.

Tuy nhiên, Oversampling tiên tiến có khả năng tạo ra các biến thể mới của mẫu thiểu số. Dẫn đến: Tăng đa dạng trong tập dữ liệu mới. Khả năng tạo ra tập dữ liệu quá cụ thể (Overly Specific).

Ưu điểm: Cải thiện độ chính xác của mô hình học máy trên tập dữ liệu huấn luyện.

Nhược điểm:

Có thể dẫn đến hiệu suất kém hơn khi áp dụng mô hình cho dữ liệu mới.

Nguy cơ tạo ra các trường hợp không thực tế trong tập dữ liệu mới.

Khi áp dụng Oversampling, cần cân nhắc kỹ lưỡng để đảm bảo cân bằng giữa việc cải thiện hiệu suất mô hình và tránh tạo ra dữ liệu không phản ánh đúng thực tế.

Bảng 3.2 Các phương pháp Oversampling

Random Oversampling	- Random Oversampling là một kỹ thuật đơn giản trong nhóm phương pháp Oversampling, với các đặc điểm chính:
	- Nguyên lý hoạt động: Phương pháp này làm phong phú dữ liệu bằng cách tạo thêm bản sao từ lớp thiểu số.
	- **Quy trình lựa chọn:**
	  - Không sao chép toàn bộ mẫu của lớp thiểu số.
	  - Thay vào đó, chọn ngẫu nhiên một số mẫu từ lớp này.
	  - Việc chọn mẫu được thực hiện có hoàn lại, nghĩa là một mẫu có thể được chọn nhiều lần.
	- **Tính ngẫu nhiên:** Đặc trưng quan trọng của phương pháp này là tính không định trước trong việc chọn mẫu để nhân bản.
	- **Mục tiêu:** Tăng số lượng mẫu trong lớp thiểu số để cân bằng với lớp đa số.
	- Phương pháp này có ưu điểm là dễ thực hiện và hiệu quả trong việc cân bằng nhanh chóng tỷ lệ giữa các lớp. Tuy nhiên, cần lưu ý rằng nó không tạo ra thông tin mới và có thể dẫn đến nguy cơ Overfitting nếu sử dụng không cẩn thận.

SMOTE	- SMOTE: Được thiết kế để khắc phục vấn đề khi dữ liệu của lớp thiểu số có số lượng mẫu ít hơn nhiều so với lớp đa số sẽ gây ra hiện tượng mất cân đối. - Phương pháp này thường đem lại kết quả tốt trong việc nâng cao hiệu suất của mô hình học máy trong các bài toán phân loại.
ADASYN	- ADASYN (Adaptive Synthetic) là một phương pháp cải tiến từ SMOTE. - **Trọng tâm cải tiến:** • Tập trung vào các mẫu "khó học" trong lớp thiểu số. • Phân bổ nguồn lực tạo mẫu mới không đồng đều, ưu tiên cho các trường hợp phức tạp. - **Cơ chế hoạt động:** • Xác định mức độ khó khăn trong việc học của từng mẫu thuộc lớp thiểu số. • Tạo nhiều mẫu tổng hợp hơn cho những trường hợp được đánh giá là khó học. - **Ưu điểm:** • Giảm thiểu xu hướng thiên lệch trong học máy gây ra bởi sự mất cân bằng dữ liệu. • Có khả năng điều chỉnh ranh giới quyết định một cách linh hoạt. • Tập trung nguồn lực vào việc cải thiện khả năng phân loại cho các mẫu khó.

	Đặc điểm nổi bật: • Phương pháp có tính thích ứng cao, tự động điều chỉnh dựa trên đặc tính của dữ liệu. • Cải thiện khả năng tổng quát hóa của mô hình bằng cách tăng cường học tập từ các trường hợp khó. - ADASYN không chỉ đơn thuần cân bằng số lượng mẫu giữa các lớp mà kỹ thuật này còn chú trọng vào việc tăng cường chất lượng học, đặc biệt đối với những mẫu dữ liệu khó khăn. - Phương pháp này giúp mô hình có khả năng thích nghi tốt hơn với các tình huống phức tạp trong dữ liệu thực tế.

3.6 **Kỹ thuật Synthetic Minority Over-sampling Technique (SMOTE)**

SMOTE (Synthetic Minority Over-sampling Technique) là một phương pháp cải tiến từ Random Oversampling, được đề xuất bởi Chawla và cộng sự của mình [7]. Các đặc điểm chính của SMOTE bao gồm:

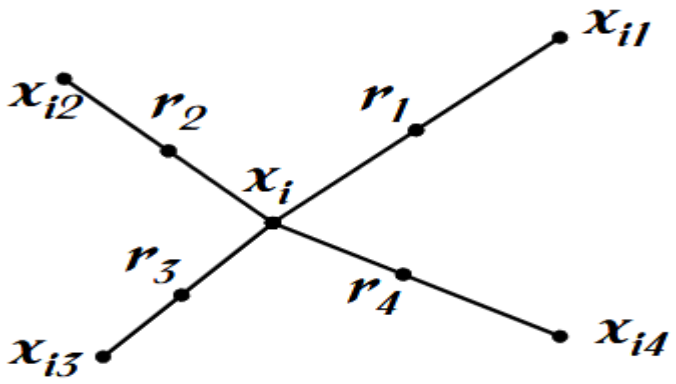
Mục tiêu: Nhằm khắc phục hạn chế của kỹ thuật Oversampling và Undersampling truyền thống. Để cải thiện khả năng tổng quát hóa của mô hình phân loại trên dữ liệu thực tế.

Nguyên lý hoạt động: Tạo ra các mẫu mới cho lớp thiểu số thay vì đơn thuần sao chép. Phân tích và sử dụng các thuộc tính của mẫu dữ liệu trong lớp thiểu số. Quy trình: Kiểm tra tất cả thuộc tính của từng mẫu trong lớp thiểu số. Tạo mẫu mới bằng cách nội suy giữa các mẫu lân cận trong lớp thiểu số.

Đặc điểm nổi bật: Tập trung vào đặc trưng của dữ liệu thay vì chỉ quan tâm đến số lượng mẫu. Tập trung vào mối quan hệ giữa các thuộc tính và giá trị của chúng.

Cơ chế tạo mẫu:

1. Chọn một mẫu x(i) từ lớp thiểu số làm cơ sở. Xác định vùng lân cận của mẫu này. Tạo mẫu mới bằng cách nội suy giữa x(i) và các mẫu lân cận.



Hình 3.7: Minh họa cách tạo các điểm dữ liệu tổng hợp trong thuật toán SMOTE (Nguồn [8])

2. Thiết lập tham số:

- Xác định hệ số N (số nguyên) để điều chỉnh mức độ tăng mẫu.
- Mục tiêu là đạt được sự phân phối gần như cân bằng giữa các lớp (tỷ lệ 1:1).

3. Lựa chọn mẫu gốc: Chọn ngẫu nhiên một mẫu từ lớp thiểu số trong tập huấn luyện.
4. Xác định lân cận:

- Tìm K mẫu gần nhất (lân cận) của mẫu đã chọn.
- Sử dụng một thước đo khoảng cách cụ thể để xác định độ gần.

5. Tạo mẫu mới:

- Thực hiện phép nội suy ngẫu nhiên giữa mẫu gốc và các mẫu lân cận.
- Tạo ra các mẫu tổng hợp mới (ví dụ: r1 đến r4).

6. Lặp lại quy trình: Quá trình này được lặp đi lặp lại cho đến khi đạt được số lượng mẫu mong muốn.

Cuối cùng, chọn ngẫu nhiên N mẫu từ K láng giềng gần nhất.

Với mỗi mẫu được chọn:

- Tính vector chênh lệch giữa mẫu gốc và mẫu láng giềng.
- Tạo số ngẫu nhiên trong khoảng [0, 1].
- Nhân vector chênh lệch với số ngẫu nhiên. Cộng kết quả vào mẫu gốc để tạo mẫu mới.

Công thức: Mẫu_mới = Mẫu_gốc + rand(0,1) * (Mẫu_láng_giềng - Mẫu_gốc)

Đặc điểm quan trọng:

- SMOTE không đơn thuần sao chép mẫu hiện có, mà tạo ra các mẫu mới có ý nghĩa.
- Phương pháp này dựa trên mối quan hệ không gian giữa các mẫu trong lớp thiểu số.

Việc sử dụng nội suy ngẫu nhiên giúp tạo ra sự đa dạng trong các mẫu mới.

Quá trình này tạo ra các mẫu tổng hợp đa dạng, nằm giữa mẫu gốc và các láng giềng. Giúp cân bằng dữ liệu và cải thiện khả năng học của mô hình trên lớp thiểu số.

Ý nghĩa: Mở rộng nghiên cứu về mối quan hệ giữa mẫu gốc và mẫu tổng hợp. Đòi hỏi phân tích sâu hơn về cấu trúc và chiều sâu của dữ liệu.

SMOTE không chỉ cân bằng số lượng mẫu giữa các lớp mà còn tạo ra dữ liệu mới có ý nghĩa, đóng góp vào việc tăng cường chất lượng của mô hình học máy trong việc xử lý dữ liệu không cân bằng.

3.7 **Kết luận**

Quá trình xử lý các tập dữ liệu không cân bằng. Áp dụng tăng cường và giảm bớt mẫu. Tăng cường mẫu thì nhằm mục đích bổ sung thêm dữ liệu cho những lớp có ít mẫu hơn. Còn giảm bớt mẫu thì tập trung vào việc giảm số lượng mẫu có trong các lớp có dữ liệu dư thừa, nhằm tạo ra một bộ dữ liệu cân bằng hơn.

CHƯƠNG 4. DỰ ĐOÁN BỆNH ĐỘT QUỴ BẰNG CÁC MÔ HÌNH HỌC MÁY

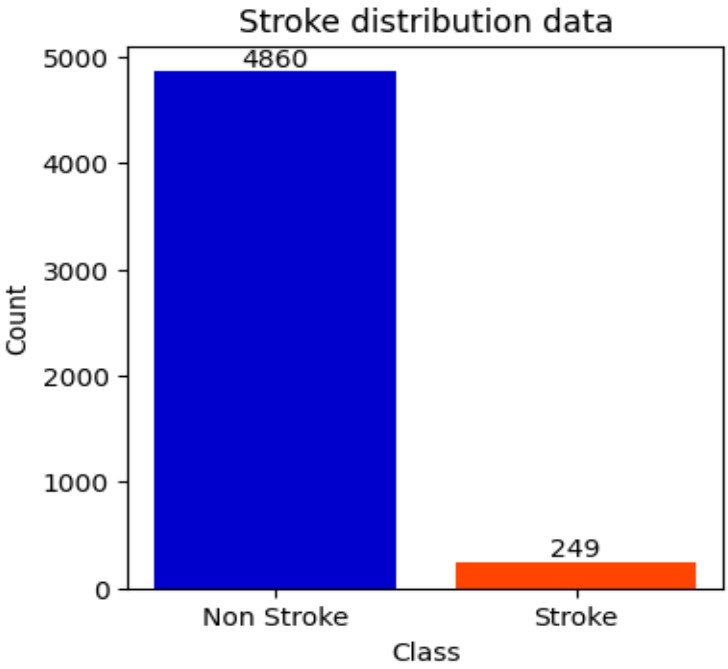
4.1 **Dữ liệu thí nghiệm**

Dự án dự đoán bệnh đột quỵ này nhóm em tập trung vào bộ dữ liệu về Healthcare Dataset Stroke Data thuộc lĩnh vực chăm sóc sức khỏe, nhằm xác định các mẫu và yếu tố chính liên quan đến về bệnh đột quỵ.

Với việc sử dụng các phương pháp như phân tích đơn biến, phân tích lưỡng biến và phân tích đa biến, chúng em nỗ lực để phát hiện các mối quan hệ phức tạp và các yếu tố quan trọng trong bộ dữ liệu.

Bằng cách áp dụng những phương pháp này, chúng em hy vọng sẽ có khả năng khám phá thêm những thông tin quan trọng về những yếu tố nào có thể gây ra đột quỵ để giúp chúng em hiểu rõ hơn về loại bệnh này và hỗ trợ trong việc phòng ngừa và điều trị.

Đầu tiên, chúng em sẽ phân tích đơn biến cột **stroke** trong dữ liệu. Vì cột **stroke** có kiểu dữ liệu là nhị phân nên nhóm sẽ sử dụng biểu đồ cột để biểu diễn sự phân bố dữ liệu:

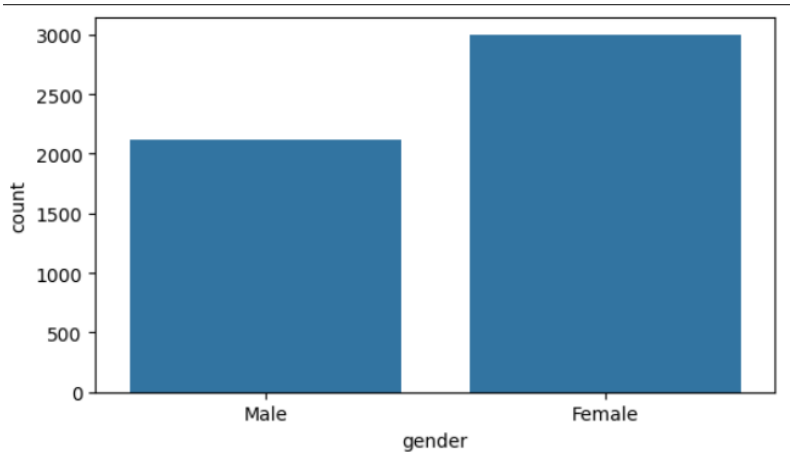


Hình 4.1: Sự phân bố dữ liệu cột Stroke

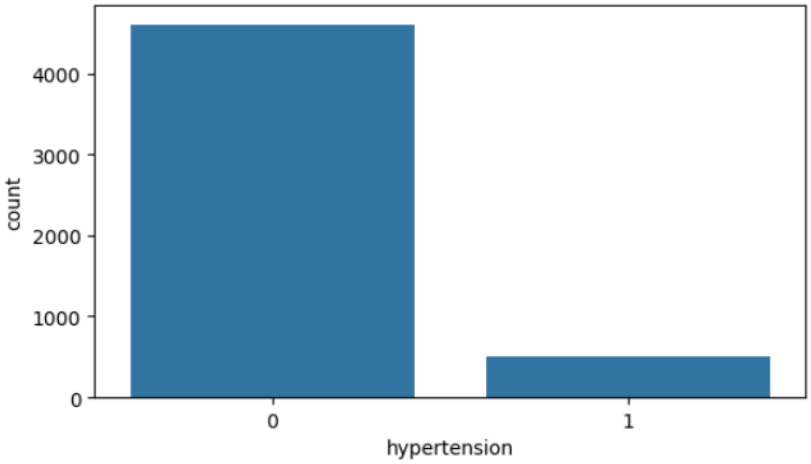
Có thể thấy dữ liệu của cột **Stroke** bị mất cân bằng khá lớn khi mà dữ liệu **Non Stroke** có số lượng vượt trội hơn nhiều so với dữ liệu **Stroke**. Điều này là một nhận định rất quan trọng vì có thể tác động đến việc lựa chọn các mô hình cũng như các thước đo đánh giá mô hình.

Sau khi đã phân tích xong cột Stroke, nhóm sẽ tiến hành phân tích các biến phân loại (Category Vairables) có trong dữ liệu, gồm các cột sau đây: **gender**, **hypertension**, **heart_disease**, **ever_married**, **work_type**, **Residence_type**, **smoking_status.**

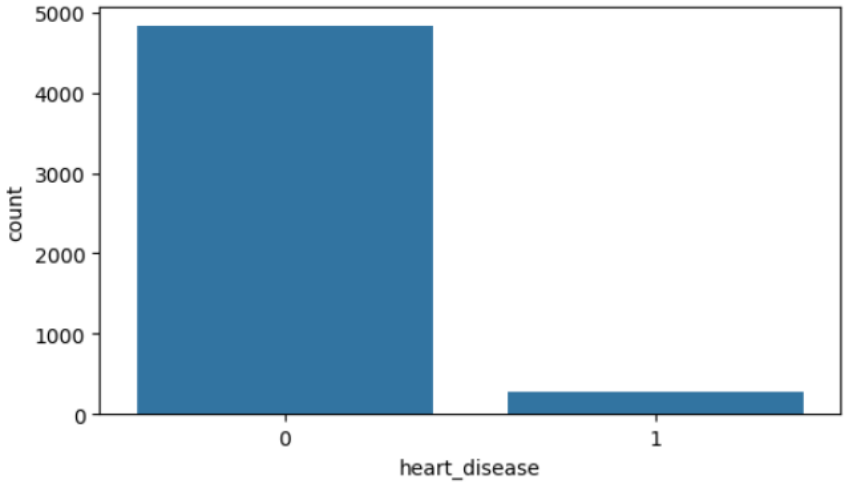
Chúng em sẽ áp dụng biểu đồ cột để biểu diễn:



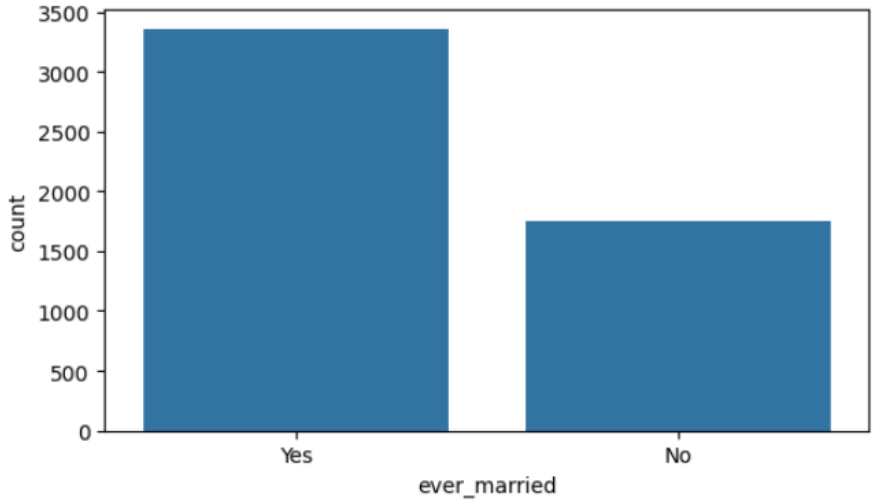
Hình 4.2: Dữ liệu cột gender



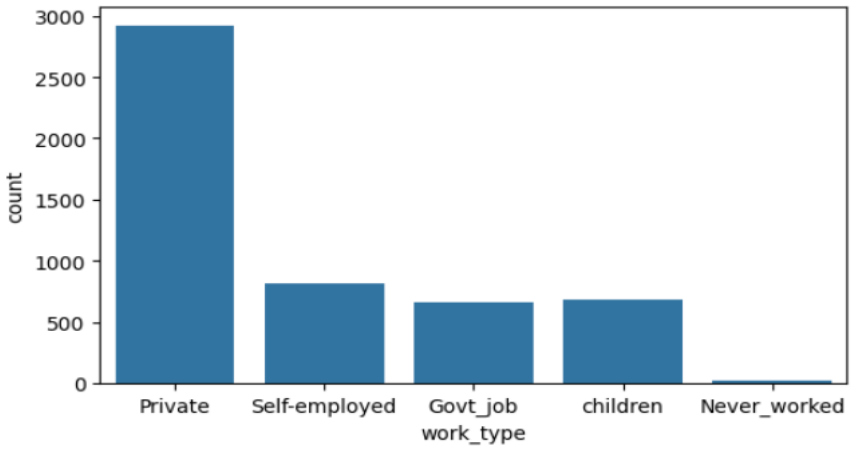
Hình 4.3: Dữ liệu cột hypertension



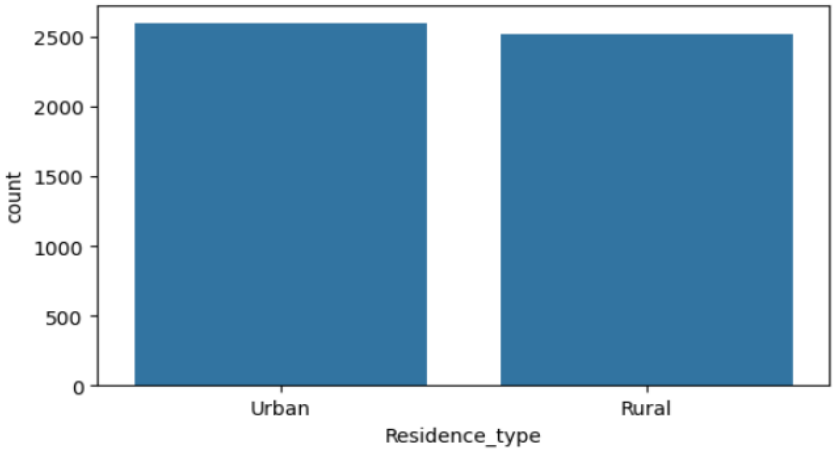
Hình 4.4: Dữ liệu cột heart_disease



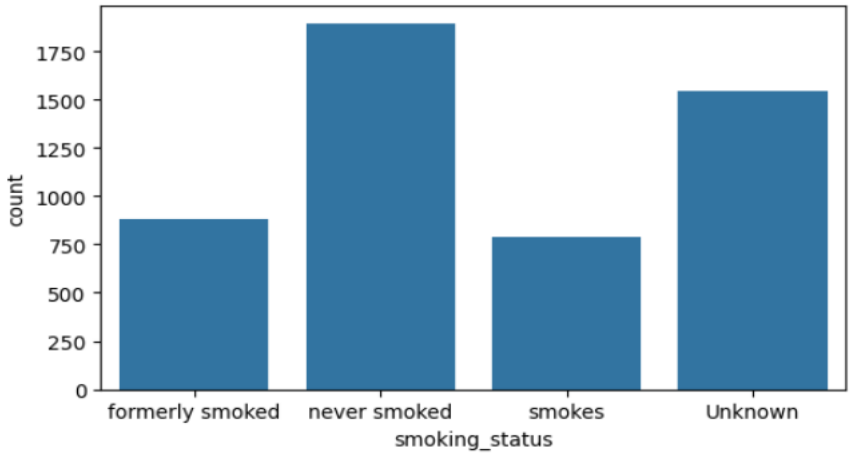
Hình 4.5: Dữ liệu cột ever_married



Hình 4.6: Dữ liệu cột work_type



Hình 4.7: Dữ liệu cột Residence_type

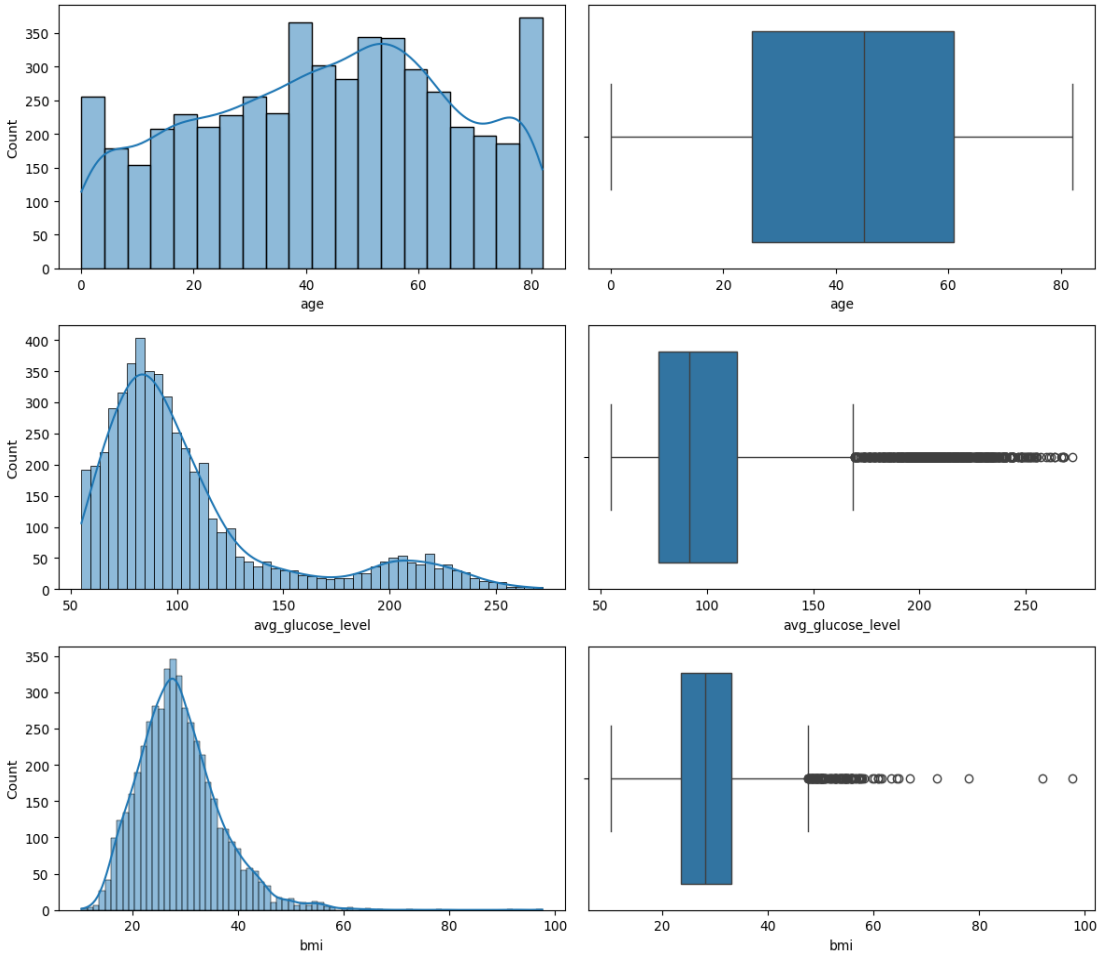


Hình 4.8: Dữ liệu cột smoking_status

Sau khi biểu diễn dữ liệu của các cột trên. Chúng em đã quan sát và có một vài kết luận sau đây:

- **gender**: Cho thấy bệnh nhân nữ nhiều hơn bệnh nhân nam.
- **hypertension**: Phần lớn cho thấy các bệnh nhân không bị cao huyết áp.
- **heart_disease**: Phần lớn cho thấy các bệnh nhân không bị bệnh tim.
- **ever_married**: Phần lớn cho thấy các bệnh nhân đã từng kết hôn.
- **work_type**: Hầu hết các bệnh nhân đều làm công việc tư nhân (*private*) và phần còn lại là phân bố đều ra ở các mục: đang tự kinh doanh (*self-employed*), làm việc nhà nước (*gotv_job*) và trẻ em (*children*). Bệnh nhân không đi làm có ít dữ liệu nhất.
- **residence_type**: Số lượng bệnh nhân sống ở nông thôn và thành phố có số lượng gần như bằng nhau.
- **smoking_status**: Hầu hết phần lớn các bệnh nhân chưa bao giờ hút thuốc. Dữ liệu về bệnh nhân hút thuốc và ít hút thuốc có số lượng ít hơn. Ngoài ra, dữ liệu tồn tại một số lượng lớn bệnh nhân không rõ là có hút thuốc hay là không.

Việc tiếp theo, nhóm sẽ tiến hành phân tích các biến liên tục (Continous Vairables) trong dữ liệu, gồm các cột: **age**, **avg_glucose_level**, **bmi**. Với biến liên tục, chúng em sử dụng biểu đồ histograms, boxplots để biểu diễn:



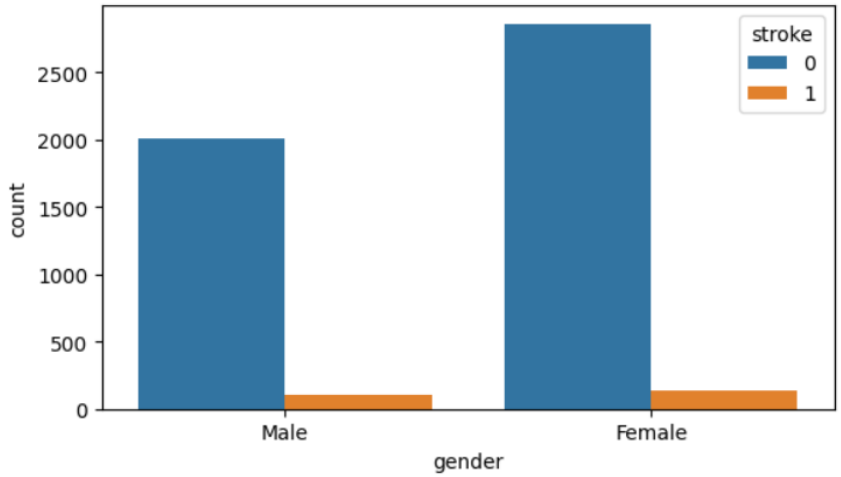
Hình 4.9: Dữ liệu của các cột age, avg_glucose_level và bmi

Sau khi biểu diễn dữ liệu, chúng em đã quan sát và đạt một số kết luận sau đây:

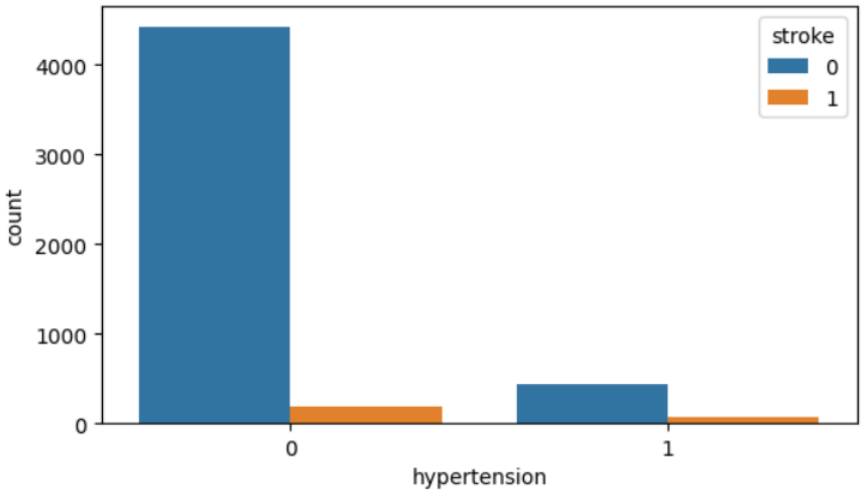
- **age**: Đa dạng độ tuổi từ trẻ đến già, phần lớn các bệnh nhân nằm trong độ tuổi từ 40 đến 80 tuổi.
- **avg_glucose_level**: Phần lớn hầu hết các bệnh nhân có mức đường huyết trung bình trong khoảng 50 đến 125, nhưng cũng có nhiều bệnh nhân có mức đường huyết cao hơn. Sự phân phối dữ liệu của cột này bị lệch phải.

- **bmi**: Bệnh nhân phần lớn có chỉ số BMI trong khoảng từ 20 đến 40, khoảng được coi là bình thường đến thừa cân. Có một số ngoại lệ có giá trị **bmi** cực cao.

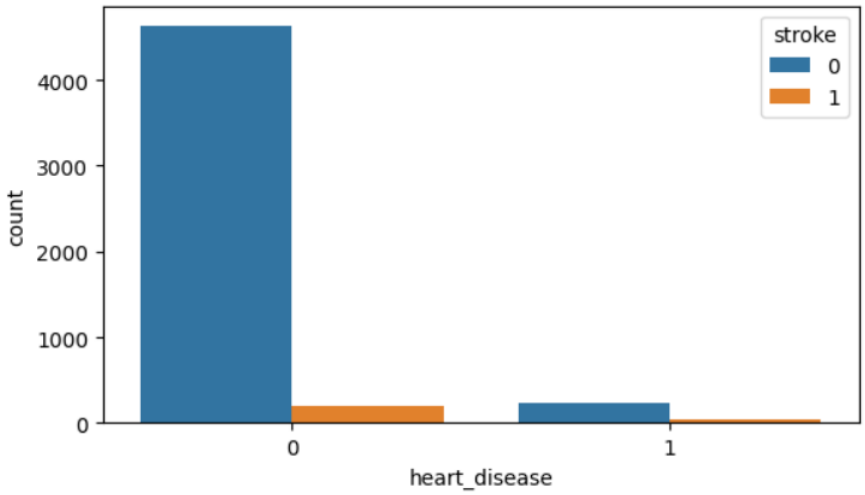
Sau việc hoàn thành phân tích đơn biến, chúng em tiến hành đến phân tích lưỡng biến. Phân tích lưỡng biến sẽ xem xét đến các mối quan hệ giữa biến mục tiêu (cột stroke) các biến khác. Đối với các biến phân loại, nhóm sẽ sử dụng biểu đồ cột đã được phân chia các mục theo biến mục tiêu. Đối với các biến liên tục, chúng em sẽ sử dụng biểu đồ vilolin đã được phân chia các mục theo biến mục tiêu.



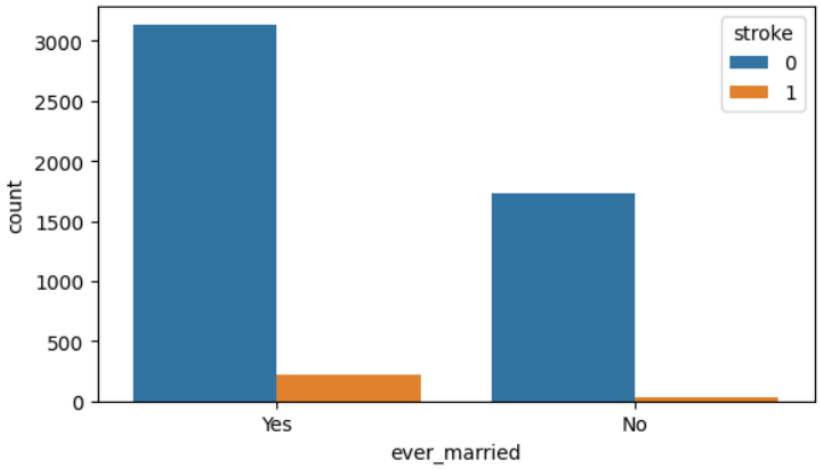
Hình 4.10: Dữ liệu cột gender khi được phân chia các mục theo biến mục tiêu



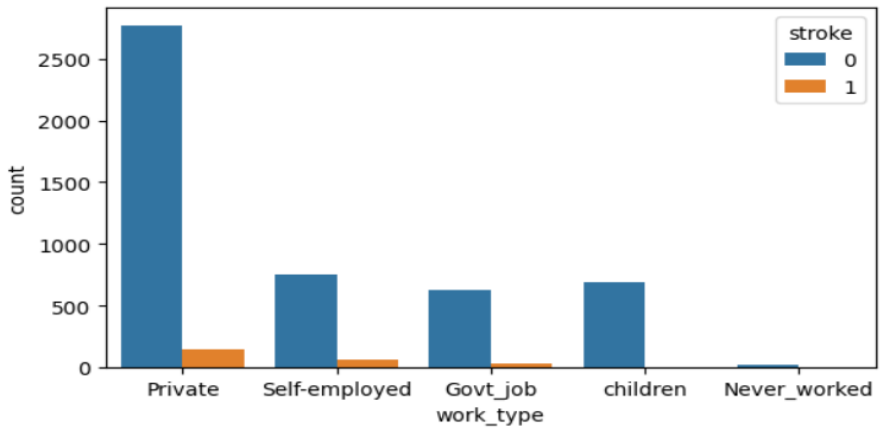
Hình 4.11: Dữ liệu cột hypertension khi được phân chia các mục theo biến mục tiêu



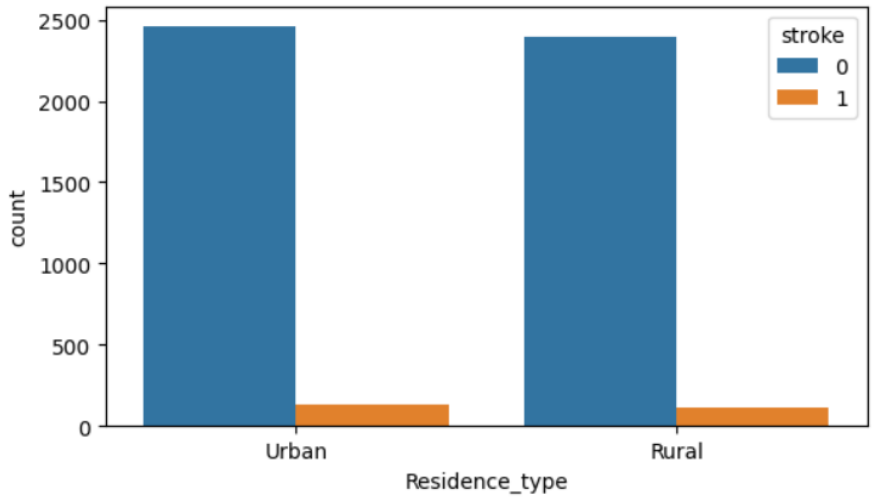
Hình 4.12: Dữ liệu cột heart_disease khi được phân chia các mục theo biến mục tiêu



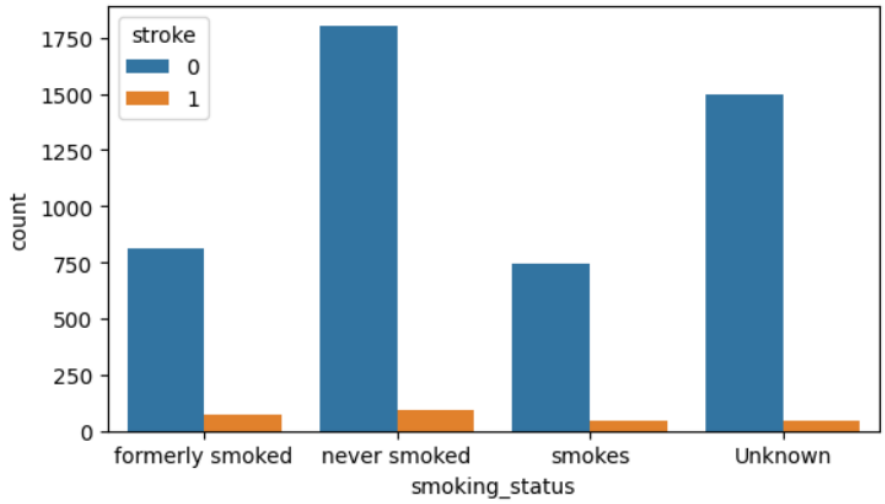
Hình 4.13: Dữ liệu cột ever_married khi được phân chia các mục theo biến mục tiêu



Hình 4.14: Dữ liệu cột work_type khi được phân chia các mục theo biến mục tiêu



Hình 4.15: Dữ liệu cột Residence_type khi được phân chia các mục theo biến mục tiêu



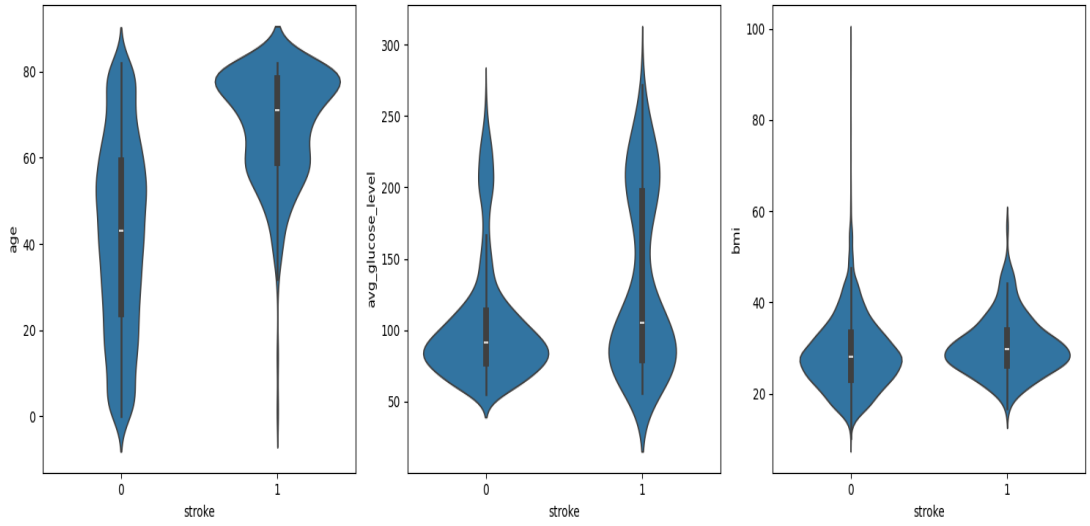
Hình 4.16: Dữ liệu smoking_status khi được phân chia các mục theo biến mục tiêu

Sau khi phân tích lưỡng biến với các biến phân loại trong dữ liệu, nhóm có những kết luận sau đây:

- **gender**: Nam bệnh nhân và nữ bệnh nhân đều có tỷ lệ mắc bệnh đột quỵ tương tự nhau, tuy nhiên số bệnh nhân nam có nhiều hơn một chút.
- **hypertension**: Khi bệnh nhân bị tăng huyết áp sẽ mắc tỷ lệ đột quỵ cao hơn khi so với những người không bị tăng huyết áp.

- **heart_disease**: Khi người bị mắc bệnh tim thì có tỷ lệ đột quỵ cao hơn người không mắc bệnh tim.
- **ever_married**: Người đã kết hôn có tỷ lệ bị đột quỵ cao hơn là những người chưa kết hôn.
- **work_type**: Những bệnh nhân làm việc tư nhân (*private*) thì có tỷ lệ bị đột quỵ cao hơn so với các loại công việc khác.
- **Residence_type**: Các trường hợp đột quỵ tỷ lệ ở thành phố và nông thôn gần như là bằng nhau.
- **smoking_status**: Bệnh nhân từng hút thuốc hoặc hiện đang hút thuốc có tỷ lệ bị đột quỵ cao hơn so với những người chưa bao giờ hút thuốc.

Tiếp tục, nhóm sẽ tiến hành phân tích lưỡng biến với các biến liên tục trong dữ liệu:



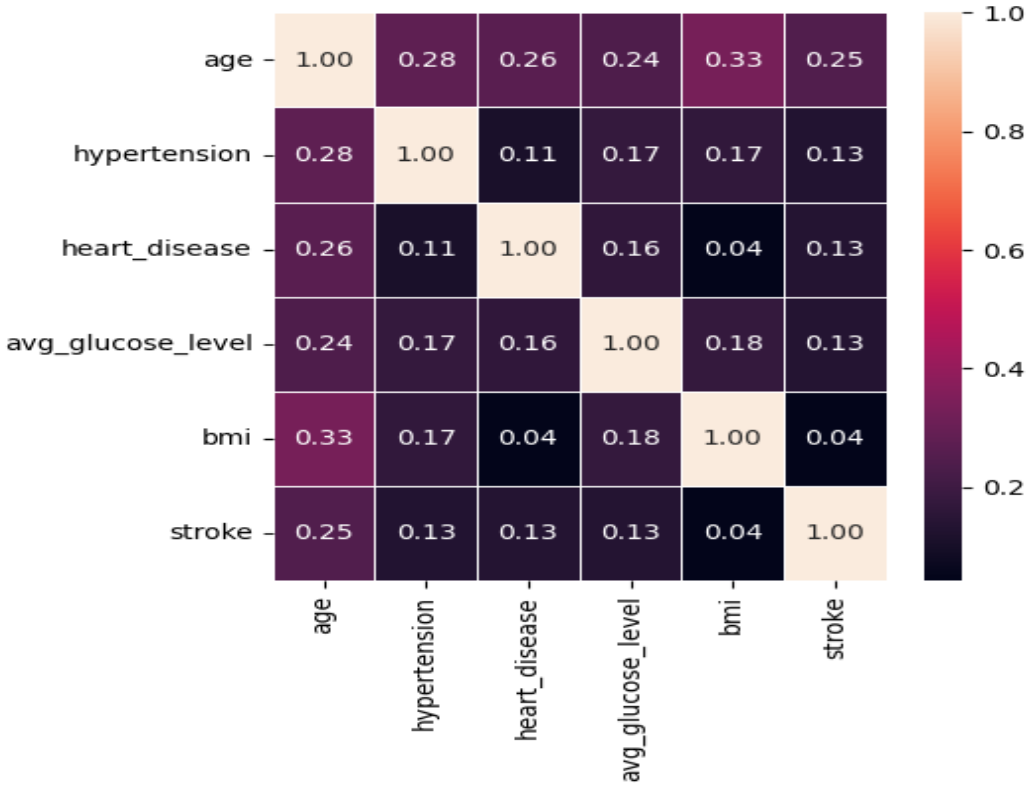
Hình 4.17: Dữ liệu của các cột age, avg_glucose_level và bmi khi được phân chia các mục theo biến mục tiêu

Sau đây là những kết luận sau khi phân tích lưỡng biến với biến liên tục:

- **age**: Những bệnh nhân lớn tuổi thường gặp đột quỵ hơn. Dường như độ tuổi trung bình của bệnh nhân đột quỵ cao hơn so với bệnh nhân không bị đột quỵ.

- **avg_glucose_level**: Người bị đột quỵ dường như có mức độ đường huyết trung bình cao hơn bệnh nhân không bị đột quỵ. Nồng độ glucose ở những bệnh nhân bị bệnh đột quỵ có sự phân bố rộng hơn.
- **bmi**: Bệnh nhân đột quỵ và không đột quỵ thì sự phân bố **bmi** khá giống nhau, cho thấy chỉ số **bmi** có thể không phải là yếu tố dự báo mạnh mẽ trong bài toán dự đoán về bệnh đột quỵ.

Sau phân tích đơn biến và lưỡng biến, chúng em sẽ tiến đến đa biến. Việc phân tích đa biến sẽ đánh giá mối quan hệ giữa nhiều biến cùng một lúc. Nhóm sẽ sử dụng ma trận tương quan để xem xét cách các biến liên tục liên quan đến nhau cũng như đến biến mục tiêu. Đối với các biến phân loại, sẽ sử dụng biểu đồ cặp (Pairplot) hoặc ma trận biểu đồ phân tán (Scatterplot Matrix).

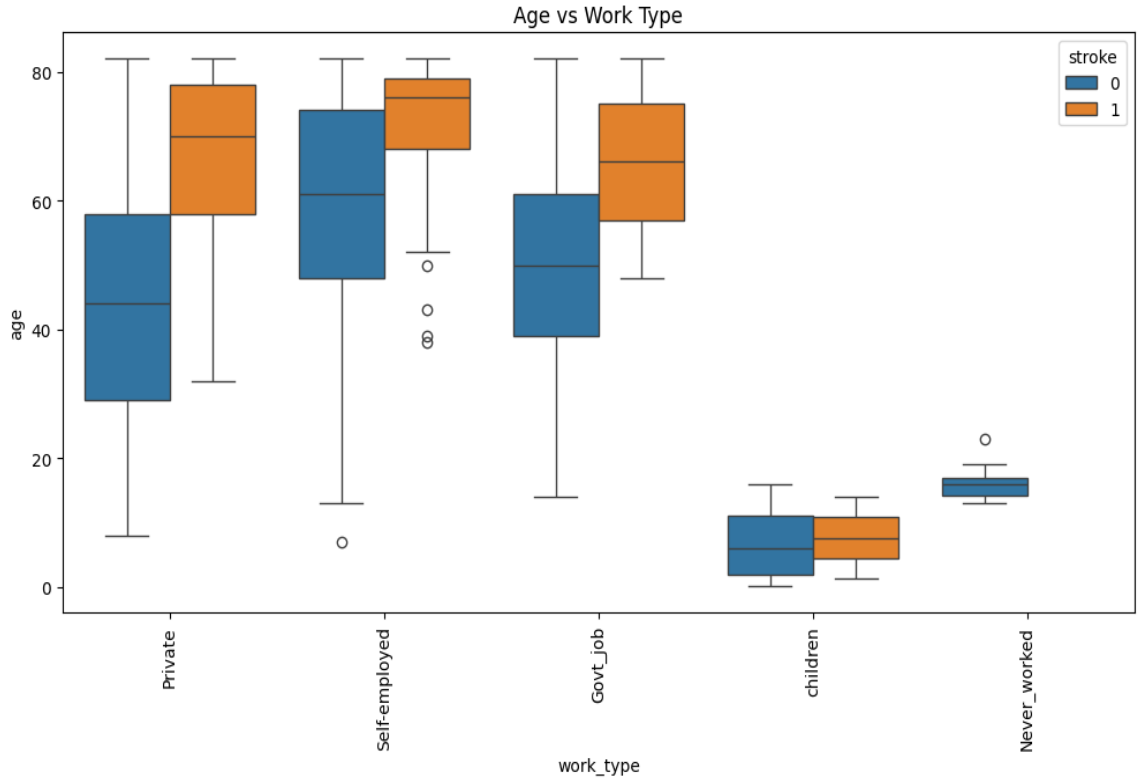


Hình 4.18: Ma trận tương quan của các biến liên tục

Sau đây là một vài kết luận khi dựa vào ma trận tương quan của các biến liên tục:

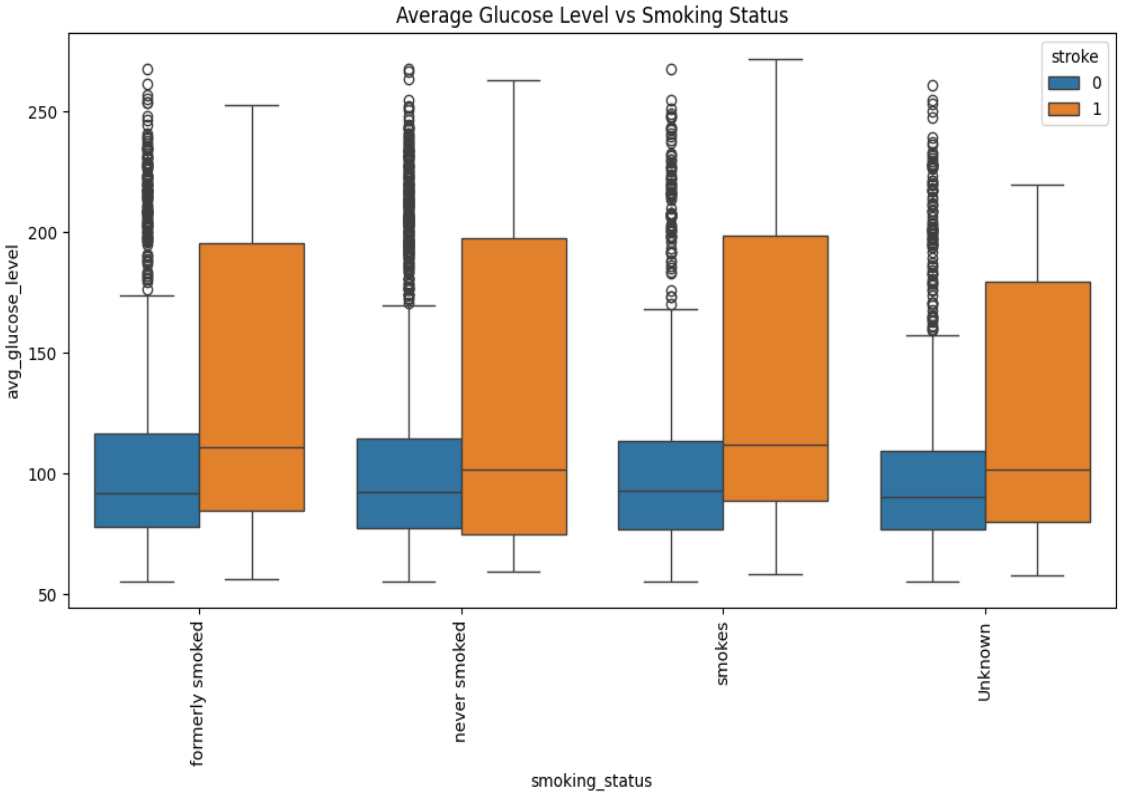
- **age**: Cho thấy mối tương quan tích cực cao nhất cột **stroke**. Cho thấy người lớn tuổi có thể có nguy cơ bị đột quỵ cao hơn, điều này hợp lý với những nghiên cứu ý tế thực tế.
- **hypertension** và **heart_disease**: Có mối tương quan tích cực với cột **stroke**, cho thấy những người bị tăng huyết áp hoặc bệnh tim có nhiều khả năng bị đột quỵ hơn.
- **avg_glucose_level**: Có mối tương quan tích cực với cột **stroke**, cho thấy khả năng dẫn đến bị đột quỵ cao hơn khi mức độ đường huyết trung bình cao hơn.

Tiếp theo, nhóm chúng em sẽ áp dụng biểu đồ hộp (Box Plots) để có thể hiểu rõ hơn mối liên hệ giữa các biến liên tục và biến phân loại có trong dữ liệu.



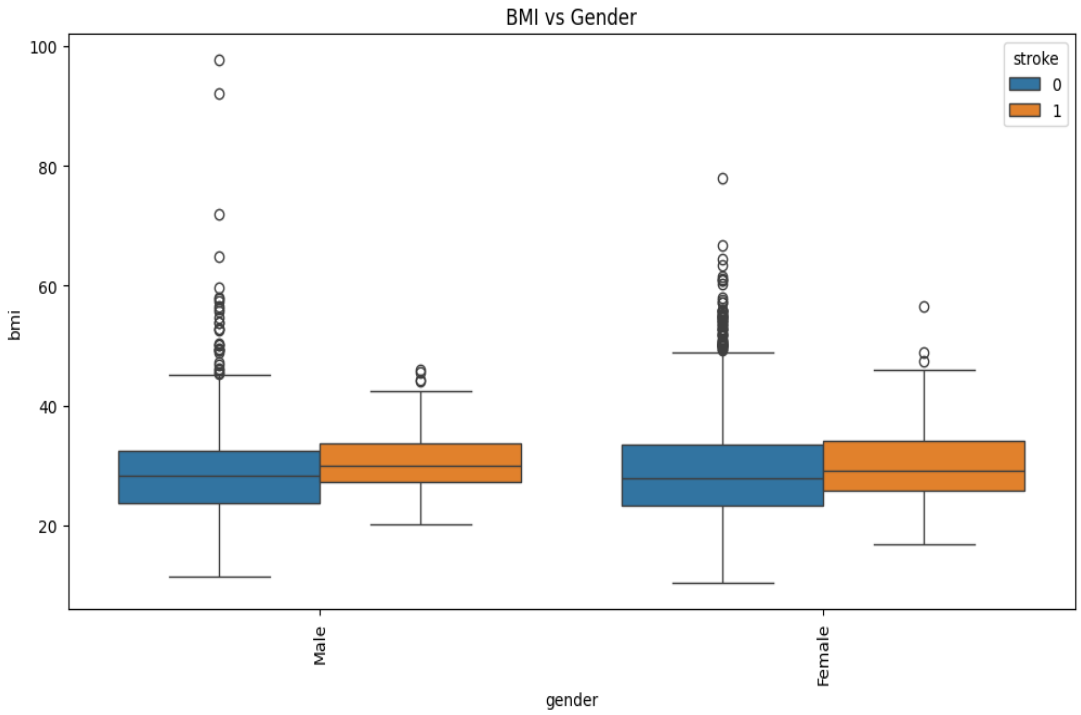
Hình 4.19: Biểu đồ hộp giữa cột age và cột work_type khi được phân chia các mục theo biến mục tiêu

Dựa vào hình 4.19, ta có thể thấy những người lớn tuổi đã đi làm, đặc biệt là những người *làm việc tư nhân* (*private*) hoặc *tự kinh doanh* (*self-employed*) sẽ có tỷ lệ **mắc đột quỵ cao hơn**. Đặc biệt với độ tuổi trung bình của nhóm người *tự kinh doanh* (*self-employed*) bị bệnh nhân đột quỵ cao nhất. Còn với *trẻ em* (*children*) thì trường hợp đột quỵ rất hiếm gặp.



Hình 4.20: Biểu đồ hộp giữa cột avg_glucose_level và smoking_status khi được phân chia các mục theo biến mục tiêu

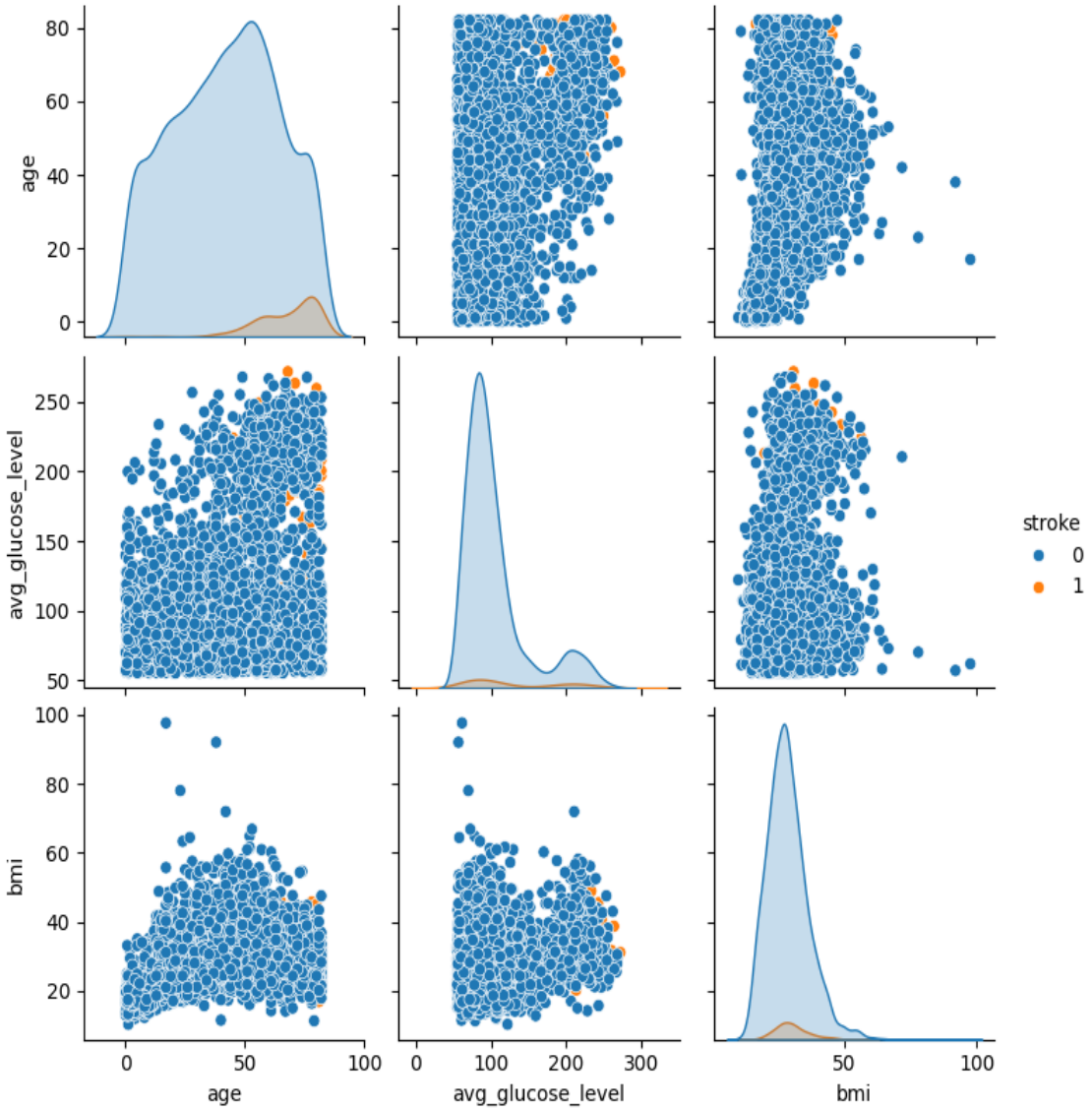
Dựa vào hình 4.20, có thể thấy mức độ glucose trung bình ở bệnh nhân đột quỵ cao hơn ở tất cả các loại tình trạng hút thuốc. Những bệnh nhân từng hút thuốc hoặc hiện đang hút thuốc có sự phân bố nồng độ glucose rộng hơn → cho thấy sự thay đổi nhiều hơn với các bệnh nhân thuộc hai loại này.



Hình 4.21: Biểu đồ hộp giữa cột bmi và gender khi được phân chia các mục theo biến mục tiêu

Dựa vào hình 4.21, thì có thể thấy sự phân bổ **bmi** của nam và nữ khá là giống nhau. Từ đó chúng em có thể đưa ra kết luận tỷ lệ đột quỵ dường như không biến đổi nhiều theo chỉ số **bmi** ở mỗi giới tính. *Nhưng lại có một số phụ nữ có chỉ số bmi cực cao đã bị đột quỵ.*

Cuối cùng, nhóm em sẽ kiểm tra sự phân bố của cột **age**, **avg_glucose_level** và **bmi** ở bệnh nhân bị đột quỵ và không bị đột quỵ bằng cách sử dụng biểu đồ cặp. Biểu đồ cặp sẽ cho phép hình dung ra mối quan hệ theo cặp giữa ba cột này, được phân tách bằng biến mục tiêu.



Hình 4.22: Biểu đồ cặp của các cột age, avg_glucose_level và bmi

Hình 4.22, đem lại một cái nhìn sâu sắc hơn về mối quan hệ giữa các cột **age**, **avg_glucose_level** và **bmi**:

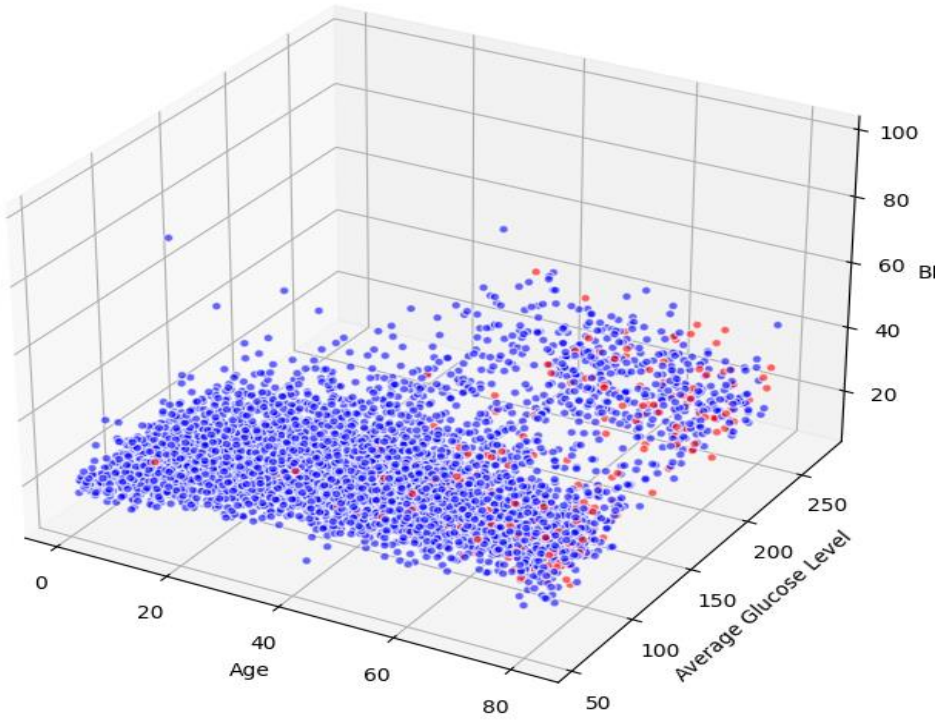
- **age** và **avg_glucose_level**: Không có mối liên hệ rõ ràng nào giữa tuổi và mức đường huyết trung bình. Tuy nhiên, bệnh nhân đột quỵ (được biểu thị bằng màu cam) có xu hướng lớn tuổi hơn và có lượng đường huyết cao hơn.
- **age** và **bmi**: Không có mối quan hệ rõ ràng nào giữa tuổi tác và chỉ số **bmi**. Bệnh nhân đột quỵ có xu hướng lớn tuổi hơn nhưng chỉ số **bmi**

của họ không có sự thay đổi đáng chú ý khi so với bệnh nhân không bị đột quỵ.

- **avg_glucose_level** và **bmi**: Thể hiện không có mối quan hệ rõ rệt về mức đường huyết trung bình và **bmi**. Tuy nhiên, bệnh nhân đột quỵ thường có xu hướng có mức đường huyết cao hơn, bất kể chỉ số **bmi** của họ.

Từ những kết luận trên, nhóm có thể nhận thấy rằng bệnh nhân đột quỵ có xu hướng là *người lớn tuổi* và *có mức đường huyết cao hơn*, trong khi sự phân bố chỉ số *bmi* của họ tương tự như bệnh nhân không bị đột quỵ.

Để có cái nhìn sâu sắc hơn nữa về mối liên hệ giữa các cột **age**, **avg_glucose_level** và **bmi** , chúng em sẽ sử dụng biểu đồ phân tán ba chiều để phân tích:



Hình 4.23: Biểu đồ phân tán ba chiều của các cột age, avg_glucose_level và bmi

Dựa vào hình 4.23, có thể thấy được *bệnh nhân bị đột quỵ thường có xu hướng già hơn và có mức đường huyết cao hơn* → điều này phù hợp với những kết luận trước đó của nhóm. Chỉ số **bmi** dường như không phân biệt được bệnh nhân đột quỵ với bệnh nhân không bị đột quỵ vì có sự trùng lặp đáng kể về giá trị **bmi** của cả hai nhóm.

4.2 **Cách tiếp cận các mô hình học máy**

4.2.1 ***Mô hình K-Nearest Neighbors (KNN)***

KNN là một trong những thuật toán phân loại đơn giản nhưng mạnh mẽ, dựa trên nguyên tắc rằng các điểm dữ liệu có thuộc tính tương tự sẽ thuộc cùng một lớp. Trong bài toán này, KNN sẽ được sử dụng để xác định xác suất một người có nguy cơ bị đột quỵ dựa trên thông tin y tế cá nhân như tuổi, giới tính, có mắc bệnh cao huyết áp hay không, có bệnh tim mạch hay không, chỉ số BMI, thông tin về hút thuốc và nhiều yếu tố khác. Thuật toán sẽ tìm kiếm trong tập dữ liệu các điểm tương tự về thông tin y tế và xác định xem một người có rơi vào lớp đột quỵ hay không.

KNN có những ưu điểm như đơn giản, không cần giả định về sự phân bố của dữ liệu và khả năng tiếp nhận dữ liệu phi tuyến tính. Tuy nhiên, để áp dụng KNN hiệu quả, cần xác định số lượng láng giềng (k) phù hợp và chọn một phương pháp đo khoảng cách phù hợp. Việc sử dụng KNN trong bài toán chuẩn đoán bệnh đột quỵ đòi hỏi cẩn thận trong việc chuẩn bị dữ liệu, lựa chọn tham số và đánh giá hiệu năng của mô hình.

4.2.2 ***Mô hình Random Forest (RF)***

Random Forest là một thuật toán mạnh mẽ được xây dựng dựa trên nguyên tắc Ensemble Learning, tức là nó kết hợp nhiều cây quyết định (Decision Trees) để tạo ra một mô hình phân loại cuối cùng. Trong bài toán này, Random Forest sẽ sử dụng các thông tin y tế cá nhân như tuổi, giới tính, cao huyết áp, bệnh tim mạch, chỉ số BMI, thông tin về hút thuốc và các yếu tố khác để xây dựng nhiều cây quyết định. Sau đó, kết hợp kết quả từ các cây quyết định để đưa ra dự đoán cuối cùng về việc một người có nguy cơ bị đột quỵ hay không.

Random Forest có nhiều ưu điểm như khả năng xử lý cả dữ liệu số và dữ liệu phân loại, khả năng ứng phó với dữ liệu mất cân bằng và khả năng giảm Overfitting. Điều này giúp cải thiện khả năng dự đoán và phòng ngừa bệnh đột quỵ một cách hiệu quả.

4.2.3 Mô hình Support Vector Machines (SVM)

Support Vector Classifier là một thuật toán học máy hiệu quả được sử dụng rộng rãi trong các bài toán phân loại, bao gồm cả dự đoán bệnh đột quỵ. SVM dựa trên nền tảng toán học vững chắc từ lý thuyết học thống kê, đóng góp vào việc duy trì sự ổn định và hiệu suất của mô hình.

Trong bài toán này, SVM sẽ sử dụng các thông tin y tế cá nhân như tuổi, giới tính, cao huyết áp, bệnh tim mạch, chỉ số BMI, thông tin về hút thuốc và các yếu tố khác để xử lý. Trong trường hợp dự đoán đột quỵ, nó sẽ tìm cách phân tách giữa bệnh nhân có nguy cơ đột quỵ và không có nguy cơ.

Mặc dù cần một số điều chỉnh, SVM có thể xử lý hiệu quả dữ liệu không cân bằng thông qua việc điều chỉnh trọng số lớp hoặc kết hợp với các kỹ thuật lấy mẫu. SVM có xu hướng ít bị Overfitting tốt hơn so với một số phương pháp khác, đặc biệt khi được điều chỉnh đúng cách.

Những ưu điểm trên làm cho SVM trở thành một lựa chọn mạnh mẽ cho bài toán dự đoán bệnh đột quỵ, đặc biệt khi được kết hợp với các phương pháp xử lý dữ liệu không cân bằng và được điều chỉnh phù hợp với dữ liệu y tế.

4.3 **Cách tiếp cận mô hình học sâu**

Mô hình CNN có khả năng học tự động và trích xuất những đặc trưng ẩn trong dữ liệu này, từ đó cải thiện khả năng chẩn đoán. CNN sử dụng các lớp tích chập để nhận biết các mẫu, biểu hiện và cấu trúc quan trọng trong dữ liệu. Điều này cho phép nó phát hiện các dấu vết tiềm ẩn của bệnh đột quỵ và xây dựng một mô hình dự đoán dựa trên các thông tin quan trọng này.

Ưu điểm của CNN trong việc chuẩn đoán bệnh đột quỵ bao gồm khả năng xử lý dữ liệu số phức tạp, khả năng tìm ra các mẫu không rõ ràng và khả năng dự đoán

dựa trên dữ liệu chính xác. Điều này giúp nâng cao khả năng phát hiện bệnh đột quỵ sớm và đưa ra dự đoán chính xác hơn.

4.4 **Tiền xử lý dữ liệu**

Sau khi import dữ liệu thành công, chúng em sẽ bắt đầu quá trình tiền xử lý dữ liệu. Đầu tiên, kiểm tra các giá trị null trong dữ liệu. Sau khi kiểm tra, nhóm phát hiện cột **bmi** có 201 giá trị null.

```
# Check null values
data.isnull().sum()
```

```
id                   0
gender               0
age                  0
hypertension         0
heart_disease        0
ever_married         0
work_type            0
Residence_type       0
avg_glucose_level    0
bmi                201
smoking_status       0
stroke               0
dtype: int64
```

Hình 4.24: Kiểm tra các giá trị null trong dữ liệu

Để xử lý 201 dữ liệu null trên, giải pháp của nhóm sẽ là sử dụng cột age để quy ra thành các nhóm tuổi, rồi sau sử dụng nhóm tuổi đó kết với cột gender để điền vào những cột bmi còn trống trong dữ liệu. Trước tiên, sẽ sử dụng cột age để tạo ra cột mới là *age_group*.

```
def age_categorize(age):
    if age < 3.0:
        return 'Toddler'
    elif age >= 3.0 and age <= 12.0:
        return 'Children'
    elif age > 12 and age <= 18.0:
        return 'Teen'
    elif age > 18.0 and age <= 25.0:
        return 'Young Adult'
    elif age > 25.0 and age < 60.0:
        return 'Adult'
    else:
        return 'Senior'

data['age_group'] = data['age'].apply(age_categorize)

data.head()
```

Hình 4.25: Sử dụng cột age để tạo ra cột age_group

```
def impute_bmi(cols):
    bmi = cols[0]
    age_group = cols[1]
    gender = cols[2]
    if pd.isnull(bmi):
        if age_group == 'Senior':
            if gender == 'Male':
                return mean_bmi.loc['Senior', 'Male']
            else:
                return mean_bmi.loc['Senior', 'Female']
        elif age_group == 'Adult':
            if gender == 'Male':
                return mean_bmi.loc['Adult', 'Male']
            else:
                return mean_bmi.loc['Adult', 'Female']
        elif age_group == 'Young Adult':
            if gender == 'Male':
                return mean_bmi.loc['Young Adult', 'Male']
            else:
                return mean_bmi.loc['Young Adult', 'Female']
        elif age_group == 'Teen':
            if gender == 'Male':
                return mean_bmi.loc['Teen', 'Male']
            else:
                return mean_bmi.loc['Teen', 'Female']
        elif age_group == 'Children':
            if gender == 'Male':
                return mean_bmi.loc['Children', 'Male']
            else:
                return mean_bmi.loc['Children', 'Female']
        else:
            if gender == 'Male':
                return mean_bmi.loc['Toddler', 'Male']
            else:
                return mean_bmi.loc['Toddler', 'Female']
    else:
        return bmi

data['bmi'] = data[['bmi','age_group','gender']].apply(impute_bmi,axis=1)
data.head()
```

Hình 4.26: Viết hàm để điền dữ liệu vào những cột bmi bị null

Sau khi xử lý dữ liệu null xong, chúng em tiến hành chia dữ liệu thành tập train – test, kết hợp với hàm StandardScaler để giúp cho mô hình đạt được kết quả chính xác và nhanh chóng hơn. Ở đây nhóm sẽ chia tập train – test theo tỉ lệ 8:2. Sau khi chia dữ liệu thành tập train – test xong, sẽ tiến hành xây dựng những mô hình học máy để tiến hành thực nghiệm.

4.5 **Xây dựng mô hình học máy**

Để phát triển các mô hình học máy cho phần thực nghiệm, nhóm chúng em sẽ sử dụng thư viện Scikit-learn, một thư viện học máy phổ biến trong Python. Thư viện

Keras, một thư viện mã nguồn mở được dùng rộng rãi trong lĩnh vực học máy và học sâu. Nhóm sẽ sử dụng thư viện Scikit-learn để thiết lập mô hình học máy, các công cụ tiền xử lý dữ liệu và các phương pháp đánh giá hiệu suất. Đối với thư viện Keras, chúng em sẽ sử dụng thư viện này để thiết lập các mô hình học sâu. Dưới đây là cấu hình chi tiết của các mô hình mà nhóm sẽ áp dụng.

Mô hình K-Nearest Neighbors:

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import RandomizedSearchCV
param_grid = {'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]}
knn = RandomizedSearchCV(KNeighborsClassifier(), param_grid, cv=5)
```

Mô hình Radom Forest:

```
      from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import RandomizedSearchCV
param_grid = {
            'n_estimators': [50, 75, 100, 150, 200, 300],
            'max_depth': [None, 3, 5, 7],
            'min_samples_split': [2, 5, 10],
            'min_samples_leaf': [1, 2, 4],
            'criterion': ['gini', 'entropy']
}

random_forest_classifier =
RandomizedSearchCV(RandomForestClassifier(random_state=42),
param_grid, cv=5)
```

Mô hình Support Vector Classifier

```
from sklearn.svm import SVC
from sklearn.model_selection import RandomizedSearchCV
support_vector_classifier = SVC(random_state=42, probability=True)
```

Mô hình 4 Layers CNN

Đối với các mô hình học sâu, nhóm em sẽ sử dụng sử dụng hàm tối ưu Adam với learning_rate là 0.01, hàm mất mát sẽ là *binary_crossentropy*.

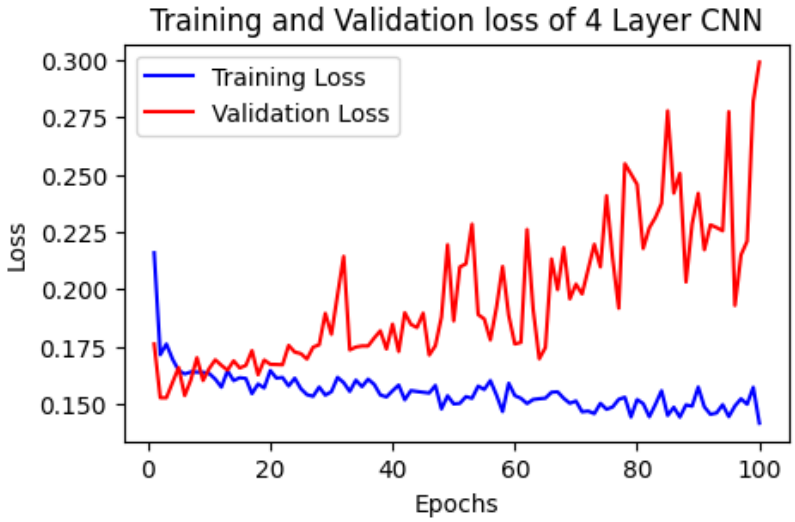
```
from keras.models import Sequential
from keras.layers import Dense, Dropout
from keras.optimizers import Adam
model_4_layer = Sequential()
model_4_layer.add(Dense(60, activation='relu',
input_shape=(X_train_scaled.shape[1],)))

model_4_layer.add(Dropout(0.2))
model_4_layer.add(Dense(30, activation='relu'))
model_4_layer.add(Dropout(0.4))
model_4_layer.add(Dense(30, activation='relu'))
model_4_layer.add(Dropout(0.4))
model_4_layer.add(Dense(1, activation='sigmoid'))

model_4_layer.compile(optimizer=Adam(learning_rate=0.01),
loss='binary_crossentropy', metrics=['accuracy'])
```

CHƯƠNG 5. THỰC NGHIỆM VÀ KẾT QUẢ

Sau khi xây dựng xong các mô hình, nhóm chúng em tiếp tục thực hiện và đánh giá các kết quả thu được. Kết quả sẽ đánh giá qua độ chính xác (accuracy), độ chính xác dương tính (precision), độ nhớ lại/nhạy cảm (recall/sensitivity), điểm F1 (F1 score) và đường cong AUC (AUC curve). Các chỉ số này được tính toán cho cả Class 0 (không mắc/không có nguy cơ đột quỵ) và Class 1 (mắc/có nguy cơ đột quỵ). Dưới đây là bảng kết quả:

Bảng 5.1: Kết quả chạy các mô hình

Algorithm	Accuracy	Precision		Recall		F1-score		AUC
		Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
KNN	0.95	0.95	0.00	1.00	0.00	0.97	1.00	0.68
RF	0.95	0.95	1.00	1.00	0.00	0.97	0.00	0.82
SVC	0.95	0.95	1.00	1.00	0.00	0.97	0.00	0.67
4 Layers CNN	0.95	0.95	1.00	1.00	0.00	0.97	0.00	0.72



Hình 5.1: Training Loss và Validation Loss của mô hình 4 Layers CNN

Sau khi chạy xong các mô hình, dựa trên kết quả thu được chúng em nhận ra rằng các mô hình hiện tại dù có kết quả ROC rất tốt, phần lớn là trên 0.70 (cụ thể là RF: 0.82 và 4 Layers CNN: 0.72), thì tất cả các mô hình chỉ có khả năng nhận diện được những bệnh nhân không bị bệnh đột quỵ (lớp 0) với số điểm (Precision) trên 95%, tỉ lệ nhớ (Recall) 100% và điểm F1-score là 0.97.

Trong khi đó, không mô hình nào có thể nhận biết được những bệnh nhân bị bệnh đột quỵ (lớp 1), (Precision), tỉ lệ nhớ (Recall) và điểm F1-score đều là 0%.

Đối với mô hình 4 Layers CNN, có thể thấy quá trình học của mô hình không tốt khi Training Loss và Validation Loss có kết quả tăng giảm đối lập nhau.

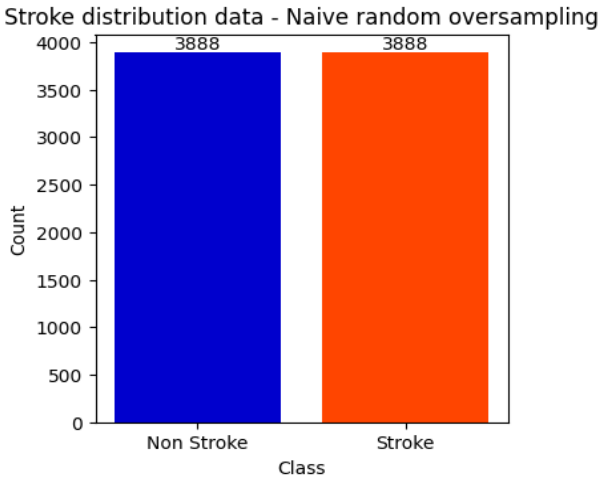
Cùng với việc phân tích dữ liệu trước đó, chúng em có thể thấy được nguyên nhân ở đây là do dữ liệu của cột **stroke** bị mất cân bằng khá lớn khi dữ liệu **Non Stroke** có số lượng lớn hơn rất nhiều so với dữ liệu **stroke** đã khiến cho mô hình bị chệch và có những dự đoán sai lệch. Điều này cũng giải thích vì sao hai bài báo nhóm em nghiên cứu có sử dụng biện pháp cân bằng dữ liệu trước khi đưa vào chạy cùng với mô hình để cho ra kết quả chuẩn xác và thuyết phục hơn.

Để giúp cho các mô hình của nhóm có thể nhận biết được bệnh nhân đột quỵ, chúng em sẽ tiến hành cân bằng dữ liệu. Nhóm chúng em nghiên cứu sử dụng kỹ thuật Undersampling, Oversampling và quyết định sẽ sử dụng kỹ thuật cân bằng **SMOTE** kết hợp giữa Undersampling và Oversampling để có những so sánh trực quan hơn.

5.1 **Oversampling**

5.1.1 ***Naive random Oversampling***

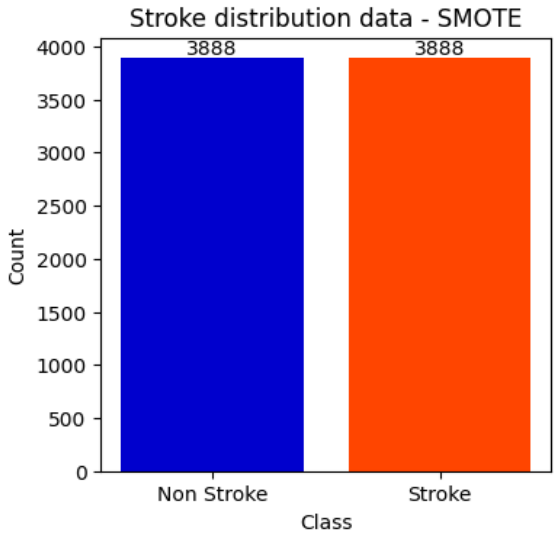
```
ros = RandomOverSampler(random_state=0)
X_ros, y_ros = ros.fit_resample(X_train, Y_train)
plot_stroke_distribution("Stroke distribution data - Naive random oversampling", y_ros.value_counts())
```



Hình 5.2: Biểu đồ cân bằng dữ liệu đột quỵ - Naive Random Oversampling

5.1.2 ***Synthetic Minority Oversampling Technique (SMOTE)***

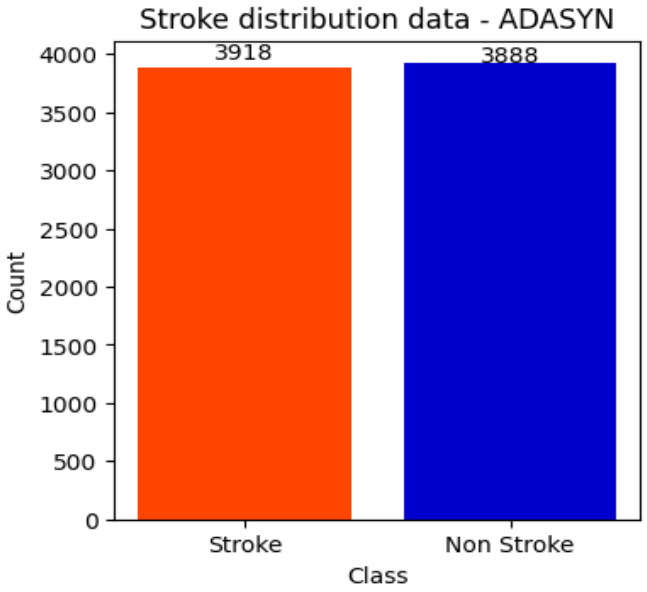
```
X_smote, y_smote = SMOTE().fit_resample(X_train, Y_train)
plot_stroke_distribution("Stroke distribution data - SMOTE", y_smote.value_counts())
```



Hình 5.3: Biểu đồ cân bằng dữ liệu đột quỵ - SMOTE

5.1.3 *Adaptive Synthetic (ADASYN)*

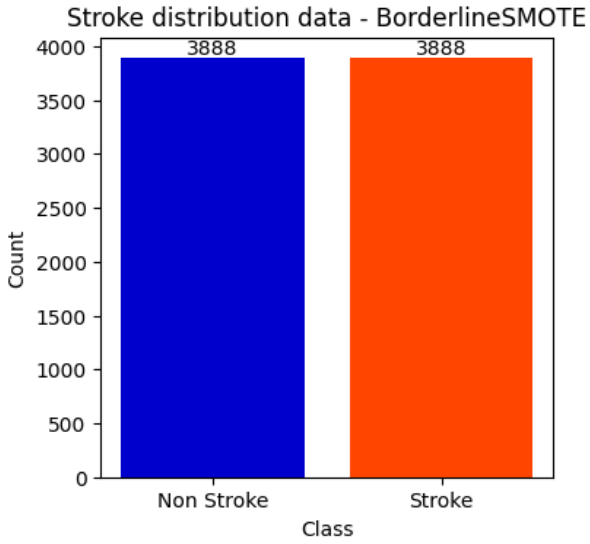
```
X_adasyn, y_adasyn = ADASYN().fit_resample(X_train, Y_train)
plot_stroke_distribution("Stroke distribution data - ADASYN", y_adasyn.value_counts())
```



Hình 5.4: Biểu đồ cân bằng dữ liệu đột quỵ - ADASYN

5.1.4 BorderlineSMOTE

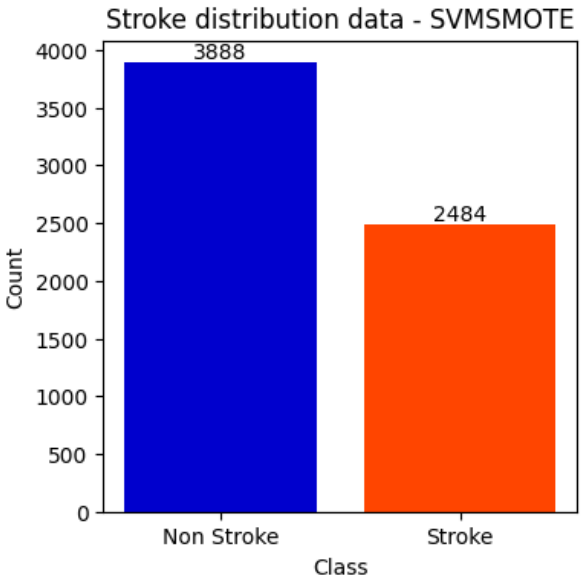
```
X_borderline_smote, y_borderline_smote = BorderlineSMOTE().fit_resample(X_train, Y_train)
plot_stroke_distribution("Stroke distribution data - BorderlineSMOTE", y_borderline_smote.value_counts())
```



Hình 5.5: Biểu đồ cân bằng dữ liệu đột quỵ - BorderlineSMOTE

5.1.5 *SVMSMOTE*

```
X_svmsmote, y_svmsmote = SVMSMOTE().fit_resample(X_train, Y_train)
plot_stroke_distribution("Stroke distribution data - SVMSMOTE", y_svmsmote.value_counts())
```

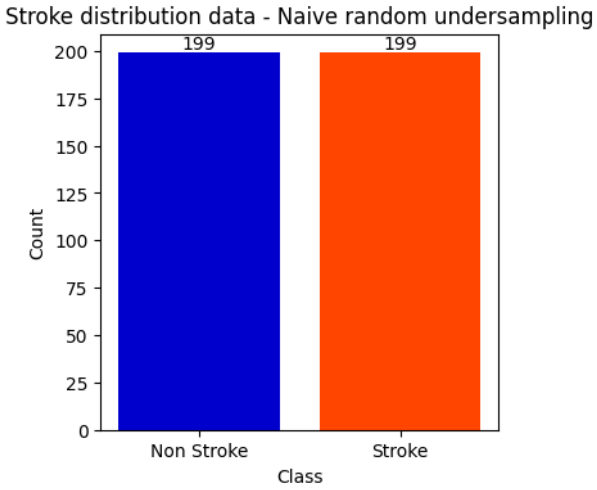


Hình 5.6: Biểu đồ cân bằng dữ liệu đột quỵ - SVMSMOTE

5.2 **Undersampling**

5.2.1 ***Naive random Undersampling***

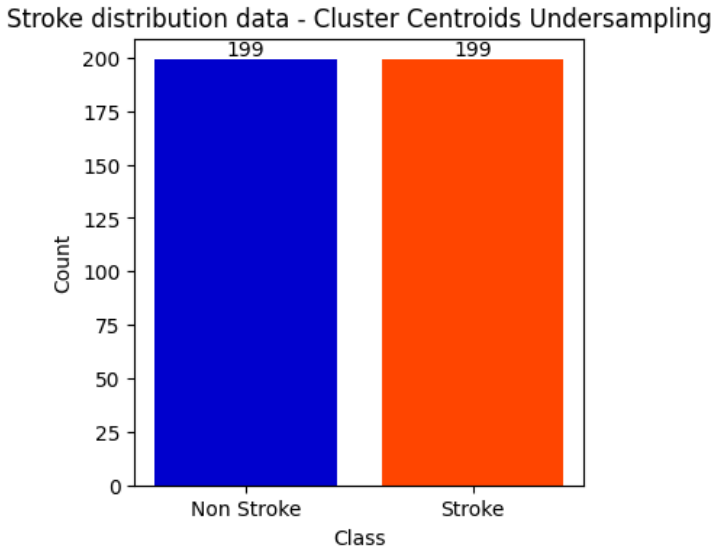
```
rus = RandomUnderSampler(random_state=0)
X_rus, y_rus = rus.fit_resample(X_train, Y_train)
plot_stroke_distribution("Stroke distribution data - Naive random undersampling", y_rus.value_counts())
```



Hình 5.7: Biểu đồ cân bằng dữ liệu đột quỵ - Naive random Undersampling

5.2.2 *Cluster Centroids Undersampling*

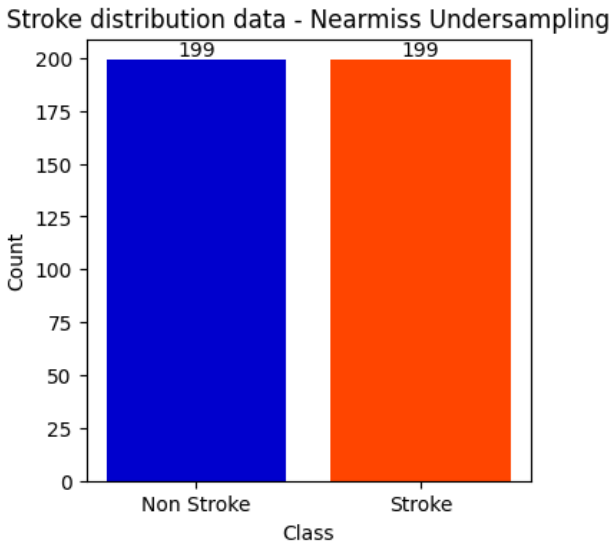
```
cc = ClusterCentroids(random_state=0)
X_cluster_centroids, y_cluster_centroids = cc.fit_resample(X_train, Y_train)
plot_stroke_distribution("Stroke distribution data - Cluster Centroids Undersampling", y_cluster_centroids.value_counts())
```



Hình 5.8: Biểu đồ cân bằng dữ liệu đột quỵ - Cluster Centroids Undersampling

5.2.3 *NearMiss*

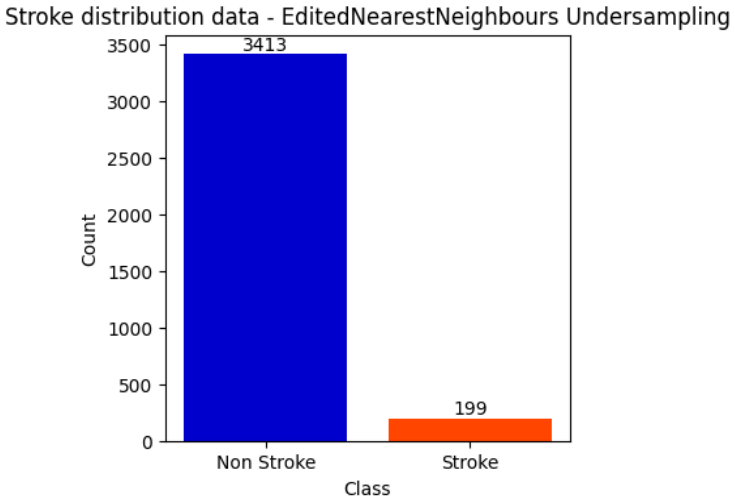
```
nm1 = NearMiss(version=1)
X_near_miss, y_near_miss = nm1.fit_resample(X_train, Y_train)
plot_stroke_distribution("Stroke distribution data - Nearmiss Undersampling", y_near_miss.value_counts())
```



Hình 5.9: Biểu đồ cân bằng dữ liệu đột quỵ - NearMiss

5.2.4 *Edited Nearest Neighbours*

```
enn = EditedNearestNeighbours()
X_enn, y_enn = enn.fit_resample(X_train, Y_train)
plot_stroke_distribution("Stroke distribution data - EditedNearestNeighbours Undersampling", y_enn.value_counts())
y_enn.value_counts()
```

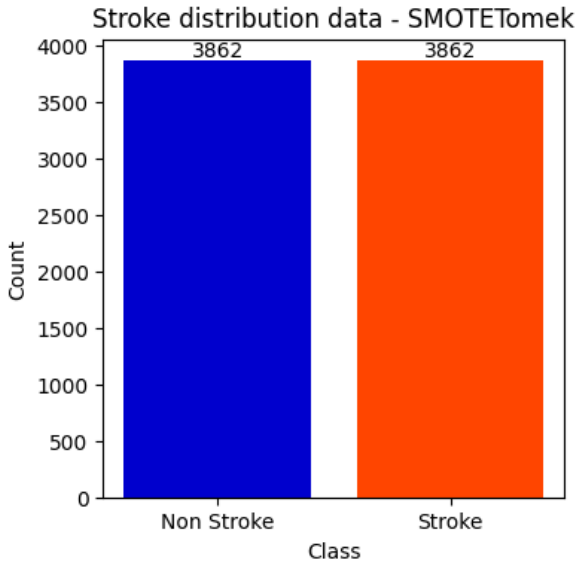


Hình 5.10: Biểu đồ cân bằng dữ liệu đột quỵ - Edited Nearest Neighbours

5.3 Kết hợp Oversampling and Undersampling

5.3.1 *SMOTETomek*

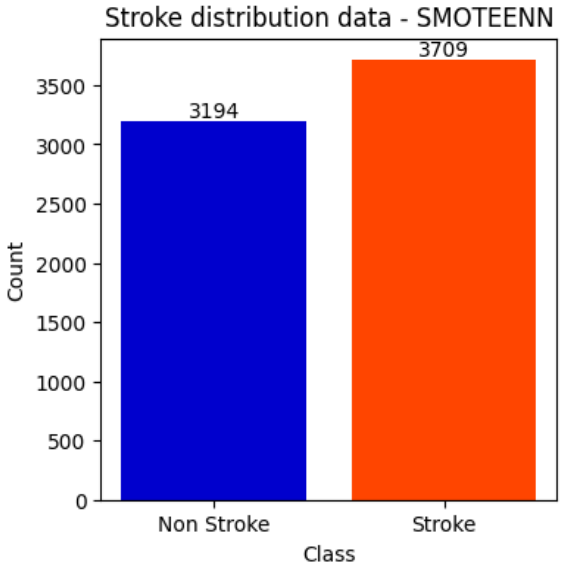
```
smote_tomek = SMOTETomek(random_state=0)
X_smote_tomek, y_smote_tomek = smote_tomek.fit_resample(X_train, Y_train)
plot_stroke_distribution("Stroke distribution data - SMOTETomek", y_smote_tomek.value_counts())
```



Hình 5.11: Biểu đồ cân bằng dữ liệu đột quỵ - SMOTETomek

5.3.2 *SMOTEENN*

```
smote_enn = SMOTEENN(random_state=0)
X_smoteenn, y_smoteenn = smote_enn.fit_resample(X_train, Y_train)
plot_stroke_distribution("Stroke distribution data - SMOTEENN", y_smoteenn.value_counts(sort=False))
```



Hình 5.12: Biểu đồ cân bằng dữ liệu đột quỵ - SMOTEENN

5.4 Áp dụng mô hình sau khi cân bằng dữ liệu

5.4.1 *K-Nearest Neighbor (KNN)*

Bảng 5.2: Kết quả mô hình KNN sau khi chạy lại với dữ liệu đã được cân bằng

Algorithm	**Accuracy**	**Precision**		**Recall**		**F1-score**		**AUC**
		Class 0	**Class 1**	**Class 0**	**Class 1**	**Class 0**	**Class 1**	
Trước khi cân bằng								
KNN	*0.95*	*0.95*	*0.00*	*1.00*	*0.00*	*0.97*	*1.00*	*0.68*
Sau khi cân bằng								
Naive random Oversampling	0.92	0.95	0.07	0.96	0.06	0.96	0.07	0.51
SMOTE	0.86	0.96	0.10	0.89	0.24	0.92	0.14	0.56
Adaptive Synthetic	0.86	0.96	0.10	0.89	0.24	0.92	0.14	0.57
Borderline SMOTE	0.88	0.96	0.10	0.91	0.18	0.93	0.12	0.55
SVMSMOTE	0.90	0.96	0.11	0.94	0.16	0.95	0.13	0.59
Naive random Undersampling	0.68	0.98	0.11	0.68	0.78	0.80	0.19	0.77
Cluster Centroids Undersampling	0.17	1.00	0.06	0.13	1.00	0.22	0.11	0.62
NearMiss	0.64	0.96	0.06	0.64	0.48	0.77	0.11	0.59
Edited Nearest Neighbours	0.87	0.96	0.12	0.91	0.24	0.93	0.16	0.57
SMOTETomek	0.86	0.96	0.11	0.90	0.24	0.93	0.15	0.57
SMOTEENN	0.81	0.96	0.10	0.83	0.36	0.89	0.16	0.60

5.4.2 ***Random Forest***

Bảng 5.3: Kết quả mô hình Random Forest sau khi chạy lại với dữ liệu đã được cân bằng

Algorithm	**Accuracy**	**Precision**		**Recall**		**F1-score**		**AUC**
		Class 0	**Class 1**	**Class 0**	**Class 1**	**Class 0**	**Class 1**	
Trước khi cân bằng								
RF	*0.95*	*0.95*	*0.00*	*1.00*	*0.00*	*0.97*	*1.00*	*0.68*
Sau khi cân bằng								
Naive random Oversampling	0.94	0.95	0.10	0.99	0.02	0.97	0.03	0.77
SMOTE	0.88	0.96	0.12	0.91	0.24	0.93	0.16	0.76
Adaptive Synthetic	0.88	0.96	0.14	0.91	0.30	0.93	0.19	0.77
Borderline SMOTE	0.90	0.96	0.14	0.93	0.22	0.95	0.17	0.78
SVMSMOTE	0.85	0.98	0.18	0.86	0.62	0.91	0.28	0.81
Naive random Undersampling	0.60	0.99	0.09	0.59	0.84	0.74	0.17	0.79
Cluster Centroids Undersampling	0.05	1.00	0.05	0.01	1.00	0.01	0.09	0.74
NearMiss	0.25	0.95	0.05	0.24	0.76	0.37	0.09	0.48
Edited Nearest Neighbours	0.94	0.95	0.22	0.99	0.08	0.97	0.12	0.79
SMOTETomek	0.75	0.98	0.12	0.75	0.66	0.85	0.20	0.79
SMOTEENN	0.84	0.97	0.13	0.86	0.42	0.91	0.20	0.77

5.4.3 *Support Vector Classifier*

Bảng 5.4: Kết quả mô hình Support Vector Classifier sau khi chạy lại với dữ liệu đã được cân bằng

Algorithm	**Accuracy**	**Precision**		**Recall**		**F1-score**		**AUC**
		Class 0	**Class 1**	**Class 0**	**Class 1**	**Class 0**	**Class 1**	
Trước khi cân bằng								
SVC	*0.95*	*0.95*	*1.00*	*1.00*	*0.00*	*0.97*	*0.00*	*0.67*
Sau khi cân bằng								
Naive random Oversampling	0.73	0.98	0.11	0.73	0.66	0.84	0.19	0.77
SMOTE	0.77	0.97	0.12	0.78	0.60	0.86	0.20	0.78
Adaptive Synthetic	0.77	0.97	0.12	0.78	0.60	0.86	0.20	0.77
Borderline SMOTE	0.81	0.98	0.15	0.82	0.62	0.89	0.24	0.79
SVMSMOTE	0.84	0.97	0.16	0.86	0.52	0.91	0.24	0.79
Naive random Undersampling	0.68	0.99	0.11	0.67	0.82	0.80	0.20	0.81
Cluster Centroids Undersampling	0.06	1.00	0.05	0.01	1.00	0.02	0.09	0.67
NearMiss	0.27	0.95	0.05	0.24	0.76	0.38	0.09	0.48
Edited Nearest Neighbours	0.95	0.95	1.00	1.00	0.00	0.97	0.00	0.72
SMOTETomek	0.78	0.97	0.12	0.79	0.58	0.87	0.20	0.76
SMOTEENN	0.83	0.97	0.13	0.85	0.44	0.91	0.21	0.77

5.4.4 *4 Layers CNN*

Bảng 5.5: Kết quả mô hình 4 Layers CNN sau khi chạy lại với dữ liệu đã được cân bằng

Algorithm	**Accuracy**	**Precision**		**Recall**		**F1-score**		**AUC**
		Class 0	**Class 1**	**Class 0**	**Class 1**	**Class 0**	**Class 1**	
Trước khi cân bằng								
4 Layers CNN	*0.95*	*0.95*	*1.00*	*1.00*	*0.00*	*0.97*	*0.00*	*0.72*
Sau khi cân bằng								
Naive random Oversampling	0.66	0.97	0.09	0.66	0.64	0.79	0.16	0.71
SMOTE	0.65	0.97	0.08	0.66	0.56	0.78	0.14	0.66
Adaptive Synthetic	0.69	0.97	0.08	0.70	0.52	0.81	0.14	0.68
Borderline SMOTE	0.67	0.96	0.08	0.67	0.52	0.79	0.13	0.65
SVMSMOTE	0.76	0.97	0.10	0.78	0.48	0.86	0.16	0.68
Naive random Undersampling	0.67	0.97	0.08	0.67	0.56	0.80	0.14	0.67
Cluster Centroids Undersampling	0.77	0.96	0.10	0.79	0.44	0.87	0.16	0.63
NearMiss	0.78	0.97	0.10	0.80	0.44	0.88	0.17	0.61
Edited Nearest Neighbours	0.95	0.95	1.00	1.00	0.00	0.97	0.00	0.70
SMOTETomek	0.71	0.96	0.08	0.73	0.48	0.83	0.14	0.66
SMOTEENN	0.69	0.96	0.07	0.71	0.44	0.81	0.12	0.65

Sau khi cân bằng, dữ liệu đã được cải thiện hiệu suất các thuật toán trên các mô hình K-Nearest Neighbor, Random Forest, Support Vector Classifier và 4 Layers CNN.

Phương pháp hiệu quả nhất: SVMSMOTE vì:

- Có F1-score cao và cân bằng nhất cho cả hai lớp trong hầu hết các trường hợp.
- Đạt được AUC cao nhất (lên tới 0.81), cho thấy khả năng phân biệt tốt giữa các lớp.
- Duy trì sự cân bằng tốt giữa Precision và Recall cho cả hai lớp.
- Thể hiện hiệu suất ổn định qua các mô hình khác nhau.

Ngoài ra, SMOTEENN và Borderline SMOTE cũng cho kết quả tốt, nhưng không nhất quán bằng SVMSMOTE.

Ta có thể đưa ra một vài kết luận thông qua các bảng trên:

Thứ nhất, về Accuracy:

- Trước khi cân bằng: Các thuật toán có độ chính xác cao (0.95) nhưng không phản ánh đúng hiệu suất thực tế do dữ liệu không cân bằng.
- Sau khi cân bằng: Độ chính xác giảm đi, nhưng phản ánh chính xác hơn khả năng phân loại của mô hình.

Thứ hai, về Precision và Recall:

- Trước khi cân bằng: Precision cao cho Class 0 (0.95-1.00) nhưng rất thấp cho Class 1 (0.00-1.00). Recall cao cho Class 1 (0.97-1.00) nhưng rất thấp cho Class 0 (0.00-1.00).
- Sau khi cân bằng: Precision và Recall cân bằng hơn giữa hai lớp, cho thấy mô hình công bằng hơn trong việc phân loại.

Thứ ba, về F1-score:

- Trước khi cân bằng: F1-score cao cho Class 1 (0.97-1.00) nhưng rất thấp cho Class 0 (0.00).
- Sau khi cân bằng: F1-score cải thiện đáng kể cho Class 0 và giảm nhẹ cho Class 1, cho thấy sự cân bằng tốt hơn.

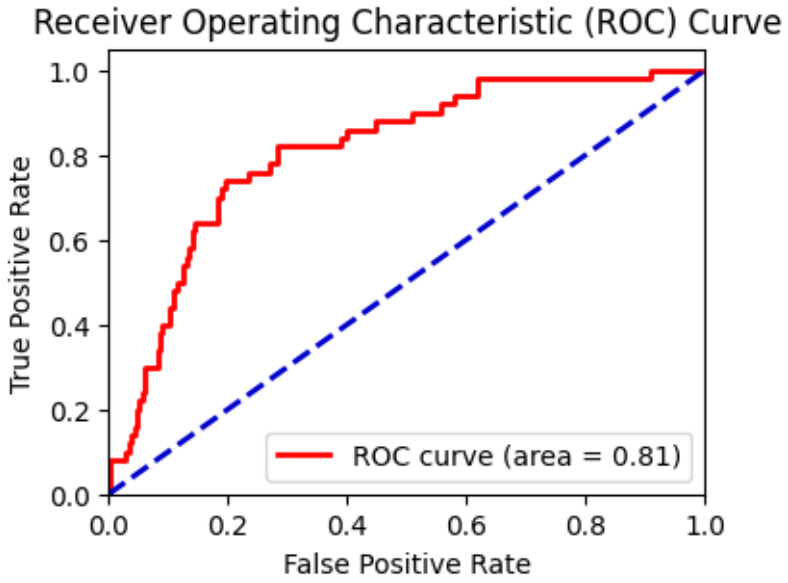
Cuối cùng, về AUC:

- Sau khi cân bằng: AUC cải thiện cho hầu hết các thuật toán, chỉ ra khả năng phân biệt tốt hơn giữa các lớp.

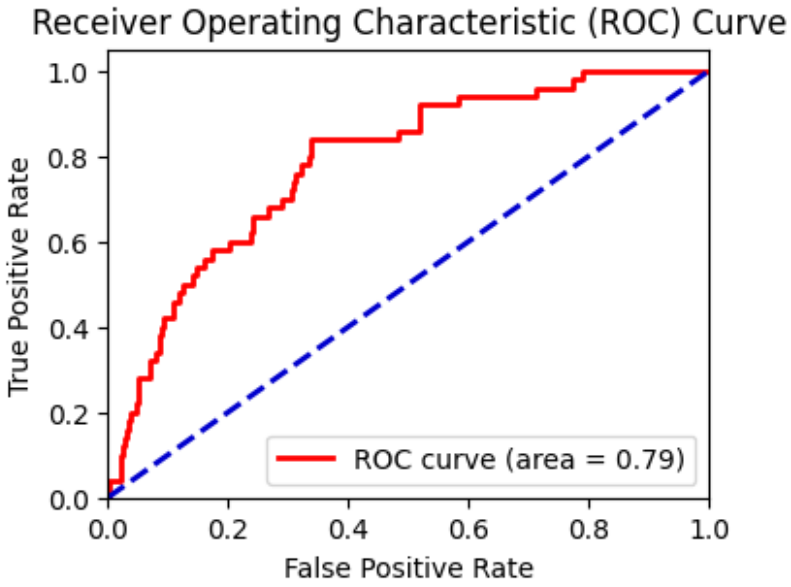
Kết luận:

- Việc cân bằng dữ liệu đã cải thiện đáng kể hiệu suất tổng thể của các mô hình trong việc dự đoán đột quỵ. Giảm thiểu sự chênh lệch giữa các chỉ số của hai lớp. Cải thiện khả năng phát hiện các trường hợp đột quỵ (Class 1) mà không làm giảm đáng kể hiệu suất trên lớp không đột quỵ (Class 0)
- Các thuật toán như SVMSMOTE, SMOTEENN và Borderline SMOTE thường cho kết quả tốt nhất sau khi cân bằng dữ liệu, với sự cải thiện đáng kể ở các chỉ số Precision, Recall và F1-score cho cả hai lớp.
- Việc cân bằng dữ liệu giúp giảm thiểu vấn đề Overfitting đối với lớp đa số (Class 0) và đã cải thiện khả năng tổng quát hóa của mô hình.
- Mặc dù Accuracy giảm sau khi cân bằng, nhưng các chỉ số khác như F1-score và AUC cho thấy mô hình thực sự hiệu quả hơn trong việc phân loại cả hai lớp.
- Việc lựa chọn thuật toán cân bằng dữ liệu phù hợp là quan trọng, vì hiệu suất có thể khác nhau tùy thuộc vào đặc điểm cụ thể của bộ dữ liệu và thuật toán phân loại được sử dụng.

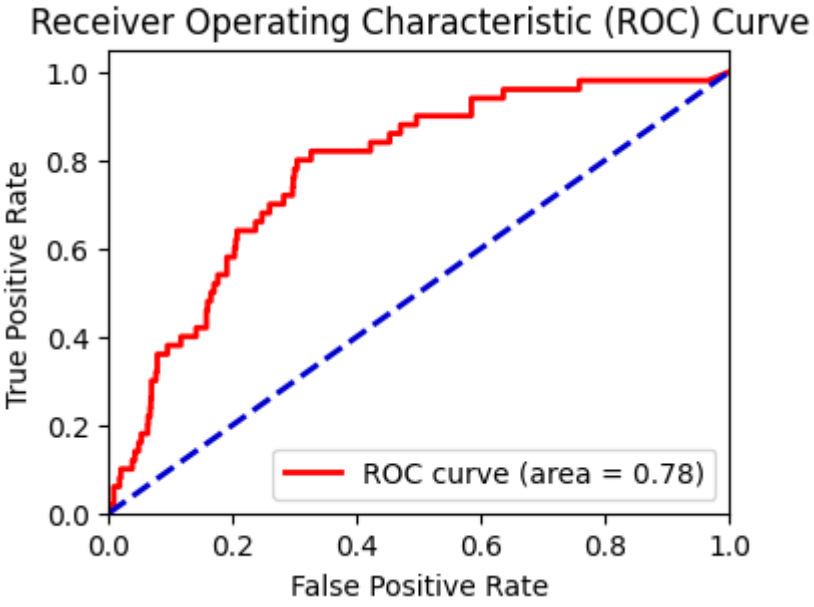
Một vài hình ảnh kết quả của các mô hình hiệu quả sau khi cân bằng dữ diệu:



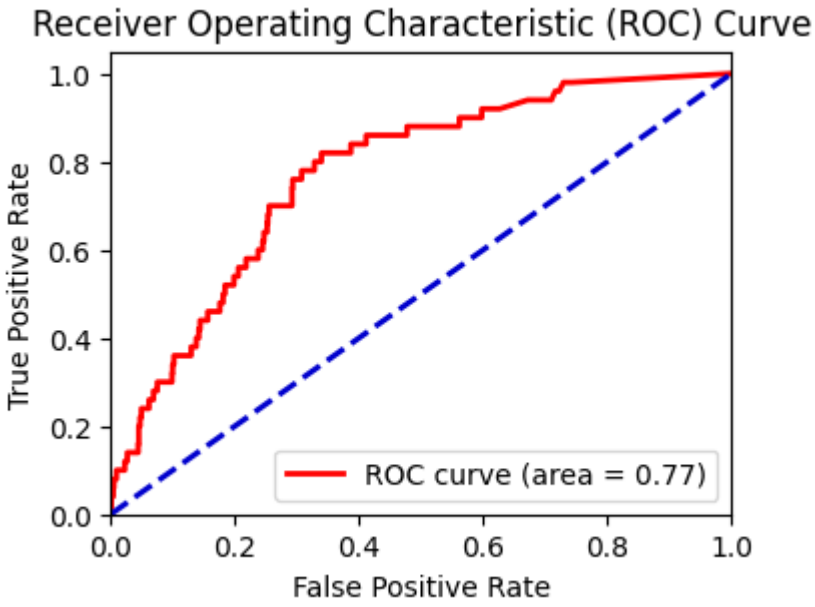
Hình 5.13: AUC của RF – SVMSMOTE



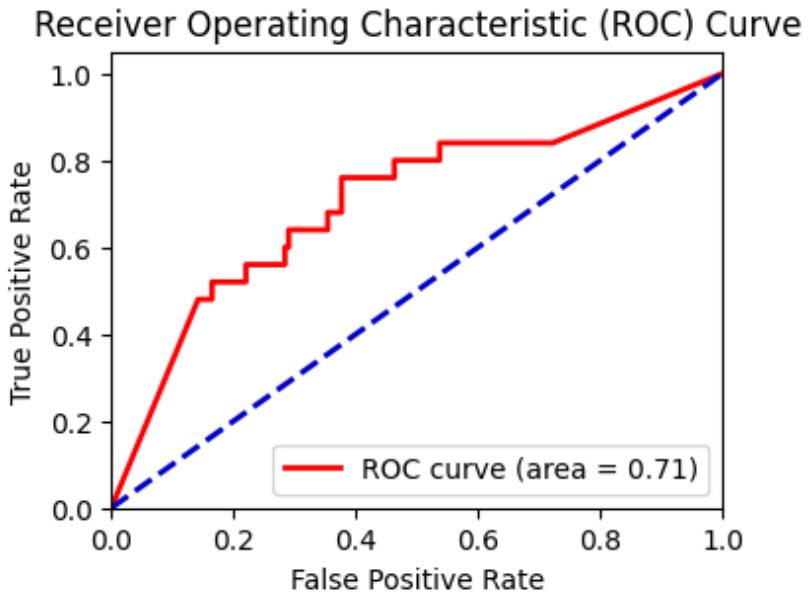
Hình 5.14: AUC của RF – Naive random Undersampling



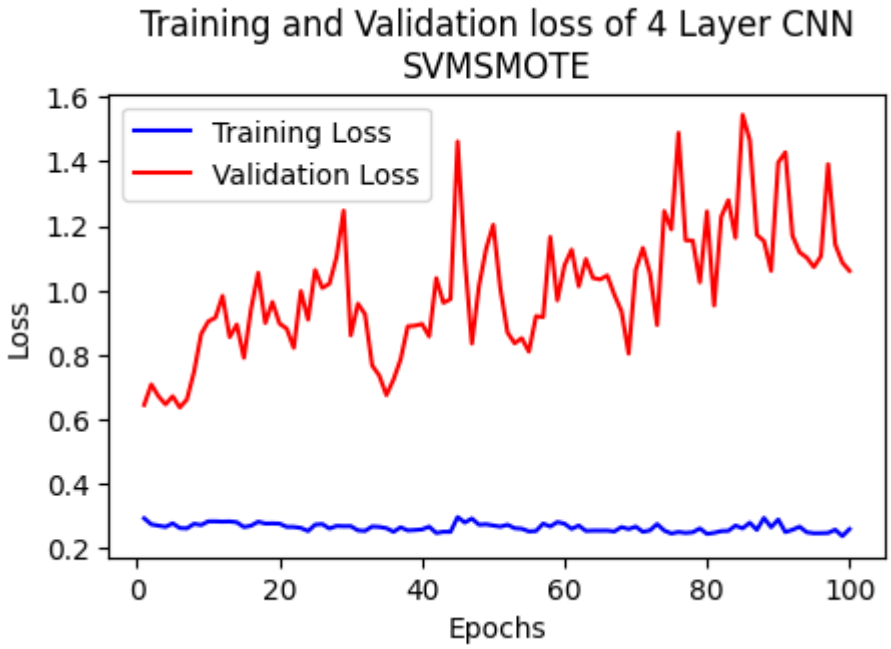
Hình 5.15: AUC của RF – BorderlineSMOTE



Hình 5.16: AUC của SVC – SMOTEENN



Hình 5.17: AUC của 4 Layers CNN – Cluster Centroids Undersampling



Hình 5.18: Training Loss và Validation Loss của 4 Layers CNN sau khi chạy lại với dữ liệu đã được cân bằng

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

So sánh kết quả chạy của nhóm chúng em với hai bài báo nghiên cứu, thì kết quả chạy của nhóm tốt hơn khi so sánh kết quả chạy của bài báo đầu tiên. Lý do ở đây là vì bài báo đầu tiên sử dụng biện pháp cân bằng dữ liệu là Undersampling, việc này đã khiến dữ liệu đầu vào đã bị giảm mạnh khi dữ liệu chỉ còn 249 dòng sau khi cân bằng. Và vì dữ liệu đầu vào quá ít nên các mô hình đã không phát huy hết hiệu suất và cho ra các kết quả không cao. Nhóm chúng em đã áp dụng các kỹ thuật cân bằng dữ liệu tiên tiến hơn như SMOTE, BorderlineSMOTE và SMOTEENN. Kết quả là các mô hình của nhóm chúng em đạt được độ chính xác và các chỉ số đánh giá khác như AUC, F1-score cao hơn đáng kể so với bài báo đầu tiên. Điều này, chứng minh tầm quan trọng của việc lựa chọn phương pháp xử lý dữ liệu không cân bằng phù hợp trong các bài toán dự đoán y tế như dự đoán đột quỵ.

Khi so sánh với bài báo thứ hai, kết quả chạy của nhóm em tương đối tương đồng với kết quả của họ, mặc dù có một số khác biệt nhỏ. Cả hai nghiên cứu đều cho thấy Random Forest và XGBoost là những mô hình hiệu quả nhất trong việc dự đoán đột quỵ. Tuy nhiên, có một số khác biệt nhỏ về độ chính xác và các chỉ số khác giữa hai nghiên cứu. Điều này có thể được giải thích bởi sự khác biệt trong phương pháp cân bằng dữ liệu. Trong bài báo, họ sử dụng biện pháp cân bằng Oversampling, giúp giữ nguyên kích thước của tập dữ liệu trong khi cân bằng nhãn ở cột mục tiêu. Trong khi đó, nhóm em áp dụng sự kết hợp giữa Undersampling và Oversampling, có thể đã làm giảm một phần dữ liệu đầu vào. Mặc dù vậy, kết quả của nhóm em vẫn rất tốt và khá gần với kết quả từ bài báo. Điều này cho thấy cả hai phương pháp đều hiệu quả trong việc cải thiện hiệu suất của các mô hình học máy trong bài toán dự đoán đột quỵ, dù có sự khác biệt nhỏ về lượng dữ liệu đầu vào.

Thông qua quá trình phân tích, tìm hiểu, thực nghiệm và dựa trên nhưng yếu tố khách quan để đánh giá, chúng em kết luận rằng những bệnh nhân có độ tuổi cao, bị các bệnh lý về tim mạch, có mức đường huyết trong máu cao và bị tăng huyết áp là những bệnh nhân có tỉ lệ mắc bệnh đột quỵ cao. Vì dữ liệu ban đầu bị mất cân

bằng, nên việc áp dụng các kỹ thuật cân bằng dữ liệu sẽ giúp cho các mô hình sẽ có một kết quả chạy tốt hơn.

Dựa vào những kết luận trên, để giúp cho các mô hình đạt được kết quả tốt hơn, dữ liệu đầu vào cần phải được mở rộng ra hơn nữa, dữ liệu cũng phải được cân bằng ngay từ đầu để có một kết quả so sánh trực quan hơn giữa các mô hình với nhau. Nên mở rộng hướng tiếp cận đến dữ liệu ảnh thay vì dữ liệu số như hiện tại vì dữ liệu ảnh sẽ cho ra kết quả dự đoán chính xác hơn trong tương lai.

TÀI LIỆU THAM KHẢO

Tiếng Việt:

[1] Hospital, T. A. (2024) Đột quỵ là gì? Nguyên nhân và cách chẩn đoán bệnh, Bệnh viện Đa khoa Tâm Anh | Tâm Anh Hospital. Available at: https://tamanhhospital.vn/dot-quy/

[2] Unica (2025) Machine Learning Là Gì? Thuật Toán Và Ứng Dụng Của Machine Learning, Unica.vn. Unica. Available at: https://unica.vn/blog/machine-learning-la-gi/

[3] Family Caregiver Alliance and reviewed by Thelma Edwards, R. N. (no date) Đột Quỵ (Stroke), Đột Quỵ (Stroke) - Family Caregiver Alliance. Available at: https://www.caregiver.org/vi/resource/dot-quy-stroke/

[4] Nhà thuốc Long Châu. (n.d.). Tìm hiểu về các loại đột quỵ: Nguyên nhân, triệu chứng và cách điều trị bệnh. Available at: https://nhathuoclongchau.com.vn/bai-viet/tim-hieu-ve-cac-loai-dot-quy-nguyen-nhan-trieu-chung-va-cach-dieu-tri-benh.html

Tiếng Anh:

[5] Sailasya, Gangavarapu, and Gorli L. Aruna Kumari. "Analyzing the performance of stroke prediction using ML classification algorithms." *International Journal of Advanced Computer Science and Applications* 12.6 (2021). Available at: https://www.semanticscholar.org/paper/Analyzing-the-Performance-of-Stroke-Prediction-ML-Sailasya-Kumari/df5c7d1bd7a59009dc51b9db903aa7f144241879?p2df

[6] Rahman, S., Hasan, M., & Sarkar, A. K. (2023). Prediction of brain stroke using machine learning algorithms and deep neural network techniques. *European Journal of Electrical Engineering and Computer Science, 7*(1), 23-30. Available at: https://ejece.org/index.php/ejece/article/view/483

[7] Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, *61*, 863-905. Available at: https://www.jair.org/index.php/jair/article/view/11192

[8] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357. Available at: https://www.jair.org/index.php/jair/article/view/10302

[9] Bevans, R. (2023) Simple Linear Regression: An Easy Introduction & Examples, Scribbr. Available at: https://www.scribbr.com/statistics/simple-linear-regression/

[10] Support Vector Machine (SVM) Algorithm (2024) GeeksforGeeks. Available at: https://www.geeksforgeeks.org/support-vector-machine-algorithm/

[11] Contributor, T. (2018) What is over sampling and under sampling?: Definition from TechTarget, WhatIs. TechTarget. Available at: https://www.techtarget.com/whatis/definition/over-sampling-and-under-sampling

[12] Rastogi, R. (2020). Random Forest Classification and it's Mathematical Implementation. Available at: https://medium.com/analytics-vidhya/random-forest-classification-and-its-mathematical-implementation-1895a7bb743e