BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM

BÁO CÁO CUỐI KÌ

ĐỀ TÀI: DỰ ĐOÁN BỆNH ĐỘT QUỴ BẰNG TRÍ TUỆ NHÂN TẠO

GVHD: GV. Trần Tiến Đức

Môn Học: Trí Tuệ Nhân Tạo

Sinh viên thực hiện

20110457 Trần Tiến Đạt

20110450 Nguyễn Thành Danh

Tp. Hồ Chí Minh, ngày 30 tháng 11 năm 2022

MỤC LỤC

I. Lý Do Chọn Đề Tài

Công nghệ AI (Artificial Intelligence) là công nghệ được quan tâm phát triển và ứng dụng bậc nhất hiện nay, được ứng dụng trong nhiều lĩnh vực như nhận dạng khuôn mặt, xử lý giọng nói, xữ lý dữ liệu lớn…

AI là công nghệ sử dụng đến kỹ thuật số có khả năng thực hiện những nhiệm vụ mà bình thường phải cần tới trí thông minh của con người, được xem là phổ biến nhất. Đặc trưng của công nghệ AI là năng lực "tự học" của máy tính (Machine Learning, do đó có thể tự phán đoán, phân tích trước các dữ liệu mới mà không cần sự hỗ trợ của con người, đồng thời có khả năng xử lý dữ liệu với số lượng rất lớn và tốc độ cao.

Cùng với xu thế phát triển mạnh mẽ của AI, hiện nay việc áp dụng AI vào các lĩnh vực trong đời sống đã không còn xa lạ, đặc biệt là lĩnh vực y tế chăm sóc sức khỏe.

Đột quỵ (stroke) còn gọi là tai biến mạch máu não thường xảy ra đột ngột khi nguồn máu cung cấp cho não bị tắc nghẽn, gián đoạn hoặc suy giảm. Khi đó, não bị thiếu oxy, dinh dưỡng và các tế bào não bắt đầu chết trong vòng vài phút. Người bị đột quỵ có nguy cơ tử vong cao nếu không được phát hiện và cấp cứu kịp thời. Đây là một trong những bệnh lý thần kinh nguy hiểm và phổ biến nhất. Đột quỵ rất khó phát hiện do không có triệu chứng cụ thể, diễn ra rất đột ngột cho nên việc dự đoán sớm để phòng ngừa là 1 vấn đề mà công nghệ trí tuệ nhân tạo có thể giải quyết để giúp y học phát triển trong việc phòng ngừa và chữa trị.

Từ thực tiển trên nhóm em muốn áp dụng AI để giải quyết vấn đề phát hiện sớm bệnh đột quỵ thông qua dữ liệu mà người dùng cung cấp. Bằng các kiến thức đã học ở môn Trí Tuệ Nhân Tạo nhóm em đã xây dựng 1 trang web sử dụng python và thư viện Scikit-learn thông qua quá trình học bộ dữ liệu bệnh nhân đã được cung cấp bởi kaggle.com tạo thành ứng dụng hoàn chỉnh.

Mục tiêu của đề tài:

+ Áp dụng các thuật toán AI/ML.

+Xử lý chuẩn hóa bộ dataset (kaggle.com cung cấp).

+Sử dụng ngôn ngữ python và các thư viện liên quan để thực hiện.

+ Dùng streamlit tạo giao diện trực quan thân thiện.

II. Tổng Quan Bài Toán

1. Tìm Hiểu Về Bài Toán

Trước khi có thể được triển khai trong các ứng dụng chăm sóc sức khỏe, các hệ thống AI cần được "huấn luyện" thông qua dữ liệu được tạo ra từ các hoạt động lâm sàng như sàng lọc, chẩn đoán, chỉ định điều trị, để có thể tìm hiểu các nhóm đối tượng, liên kết tương tự giữa chúng, tính năng chủ đề và kết quả quan tâm.

Ở đây nhóm đã tìm được bộ dữ liệu cung cấp bởi Kaggle (Stroke Prediction Dataset).Bộ dữ liệu này được sử dụng để dự đoán liệu một bệnh nhân có khả năng bị đột quỵ hay không dựa trên các thông số đầu vào như giới tính, tuổi tác, các bệnh khác nhau và tình trạng hút thuốc. Mỗi hàng trong dữ liệu cung cấp thông tin liên quan về bệnh nhân.

Sau khi đã có bộ dataset nhóm tiến hành phân tích chuẩn hóa tiếp theo sẽ chọn thuật toán phân loại (Classification Algorithm) để cho máy học từ đó xây dựng ra các model để sử dụng hiệu quả.

2. Giải Quyết Bài Toán

Sau khi xác định được yêu cầu đầu ra và vào của ứng dụng nhóm đã sử dụng Machine Learning để học bộ dữ liệu ban đầu xây dựng nên các model thông qua Supervised Learning (học có giám sát) cụ thể ở đây nhóm chọn thuật toán KNN (K-Nearest Neighbors) do việc dễ dàng triển khai cũng nhưng mạng lại độ chính xác khá cao.

Để cho máy tính có thể học được bộ dữ liệu một cách tối ưu nhóm sẽ tiến hành khai phá và chuẩn hóa lại bộ dữ liệu (file .csv) loại bỏ các feature không cần thiết cho việc training mang lại hiệu quả cao nhất. Nhóm sử dụng các thư viện imblearn để xử lý dữ liệu và sklearn training model. Sau khi đã xây dựng thành công model nhóm sẽ sử dụng streamlit để phát triển ứng dụng Web dựa trên Python 3.10.6 64bit.

III. Thuật Toán Xử Lý

Supervised Learning (Học có giám sát)

Supervised learning là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (*input, outcome*) đã biết từ trước. Cặp dữ liệu này còn được gọi là (*data, label*), tức (*dữ liệu, nhãn*). Supervised learning là nhóm phổ biến nhất trong các thuật toán Machine Learning.

Một cách toán học, Supervised learning là khi chúng ra có một tập hợp biến đầu vào $X=\{x1,x2,...,xN\}$ và một tập hợp nhãn tương ứng $Y=\{y1,y2,...,yN\}$, trong đó xi,yi là các vector. Các cặp dữ liệu biết trước $(xi,yi)\in X\times Y$ được gọi là tập *training data* (dữ liệu huấn luyện). Từ tập training data này, chúng ta cần tạo ra một hàm số ánh xạ mỗi phần tử từ tập X sang một phần tử (xấp xỉ) tương ứng của tập Y:

$$yi\approx f(xi),\ \forall i=1,2,...,N$$

Mục đích là xấp xỉ hàm số f thật tốt để khi có một dữ liệu x mới, chúng ta có thể tính được nhãn tương ứng của nó $y=f(x)$.

Thuật toán ***supervised learning*** còn được tiếp tục chia nhỏ ra thành hai loại chính:

+ *Classification* (Phân loại)

+ *Regression* (Hồi quy)

Thuật toán KNN (K-Nearest Neighbors)

K-nearest neighbor là một trong những thuật toán supervised-learning đơn giản nhất (mà hiệu quả trong một vài trường hợp) trong Machine Learning. Khi training, thuật toán này không học một điều gì từ dữ liệu training (đây cũng là lý do thuật toán này được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới.

KNN là thuật toán đi tìm đầu ra của một điểm dữ liệu mới bằng cách chỉ dựa trên thông tin của K điểm dữ liệu trong training set gần nó nhất (K-lân cận), không quan tâm đến việc có một vài điểm dữ liệu trong những điểm gần nhất này là nhiễu.

Không có hàm mất mát hay bài toán tối ưu nào cần thực hiện trong quá trình huấn luyện KNN. Mọi tính toán được tiến hành ở bước kiểm tra. Vì KNN ra quyết định dựa trên các

điểm gần nhất nên có hai vấn đề ta cần lưu tâm. Thứ nhất, khoảng cách được định nghĩa như thế nào. Thứ hai, cần phải tính toán khoảng cách như thế nào cho hiệu quả

Với vấn đề thứ nhất, mỗi điểm dữ liệu được thể hiện bằng một vector đặc trưng, khoảng cách giữa hai điểm chính là khoảng cách giữa hai vector đó. Có nhiều loại khoảng cách khác nhau tuỳ vào bài toán, nhưng khoảng cách được sử dụng nhiều nhất là khoảng cách Euclid.

Vấn đề thứ hai cần được lưu tâm hơn, đặc biệt với các bài toán có tập huấn luyện lớn và vector dữ liệu có kích thước lớn. Giả sử các vector huấn luyện là các cột của ma trận X ∈ R d×N với d và N lớn. KNN sẽ phải tính khoảng cách từ một điểm dữ liệu mới z ∈ R d đến toàn bộ N điểm dữ liệu đã cho và chọn ra K khoảng cách nhỏ nhất. Nếu không có cách tính hiệu quả, khối lượng tính toán sẽ rất lớn

Khoảng cách Euclid từ một điểm z tới một điểm xi trong tập huấn luyện được định nghĩa bởi $\|\mathbf{z}-\mathbf{x}_i\|_2$. Người ta thường dùng bình phương khoảng cách Euclid $\|\mathbf{z}-\mathbf{x}_i\|_2^2$ để tránh phép tính căn bậc hai. Việc bình phương này không ảnh hưởng tới thứ tự của các khoảng cách.

$$\|\mathbf{z}-\mathbf{x}_i\|_2^2 = (\mathbf{z}-\mathbf{x}_i)^T(\mathbf{z}-\mathbf{x}_i) = \|\mathbf{z}\|_2^2 + \|\mathbf{x}_i\|_2^2 - 2\mathbf{x}_i^T\mathbf{z}$$

IV. Xây dựng ứng dụng

1. Khai phá xử lý dữ liệu

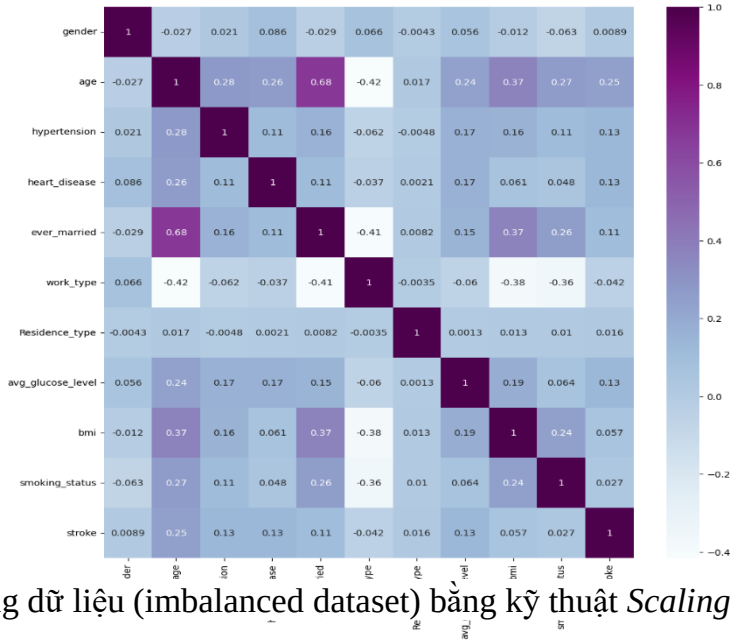
Dữ liệu ban đầu là file CSV với 4982 bộ dữ liệu gồm 11 cột (bỏ đi cột id)

healthcare-dataset-stroke-data.csv (316.97 kB)

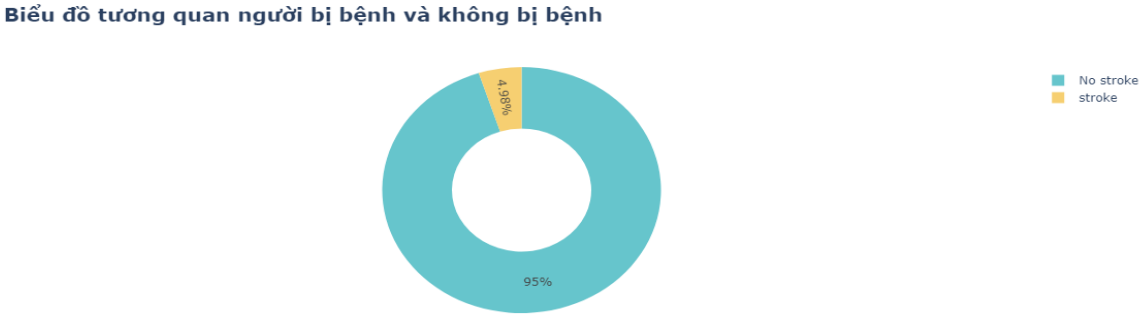
Detail | Compact | Column — 12 of 12 columns

id	gender	age	hypertensi...	heart_dise...	ever_marri...	work_type	Residence...	avg_gluco...	bmi	smoking_s...
9046	Male	67	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked
51676	Female	61	0	0	Yes	Self-employed	Rural	202.21	N/A	never smoked
31112	Male	80	0	1	Yes	Private	Rural	105.92	32.5	never smoked
60182	Female	49	0	0	Yes	Private	Urban	171.23	34.4	smokes
1665	Female	79	1	0	Yes	Self-employed	Rural	174.12	24	never smoked
56669	Male	81	0	0	Yes	Private	Urban	186.21	29	formerly smoked
53882	Male	74	1	1	Yes	Private	Rural	70.09	27.4	never smoked

Sau khi đã có bộ dữ liệu tiến hành phân tích để lượt bỏ các cột (feature) không cần thiết



Xử lý mất căng bằng dữ liệu (imbalanced dataset) bằng kỹ thuật *Scaling* và *Oversampling*



Dữ liệu trước khi xử lý

	age	hypertension	heart_disease	ever_married	avg_glucose_level	bmi
0	5.670554	0	1	1	29.484279	1.193238
1	7.585196	0	1	1	11.402697	0.589390
2	3.019512	0	0	1	21.021563	0.869222
3	7.437916	1	0	1	21.447202	-0.662492
4	7.732476	0	0	1	23.227819	0.073909
...	...	...	...	...	...	...
9461	5.516197	0	0	1	6.319999	0.064135
9462	3.938426	1	0	1	7.213486	-0.021234
9463	6.060548	0	0	1	31.298742	2.423669
9464	7.696631	0	1	1	26.834668	0.326911
9465	4.544848	0	0	0	9.600652	1.966870

9466 rows × 6 columns

Dữ liệu sau khi xử lý

2. Training Model

Từ bộ dữ liệu đã xử lý ở trên nhóm xử dụng thư viện scikit-learn để training. Scikit-learn là một thư viện Python mã nguồn mở dành cho học máy. Thư viện này hỗ trợ các thuật toán hiện đại như KNN, XGBoost, random forest, SVM và một số thuật toán khác. Scikit-learn được sử dụng rộng rãi trong các cuộc thi kaggle cũng như trong các công ty

công nghệ nổi tiếng. Nó giúp tiền xử lý, giảm chiều dữ liệu (lựa chọn tham số), phân loại, hồi quy, phân cụm và model selection.

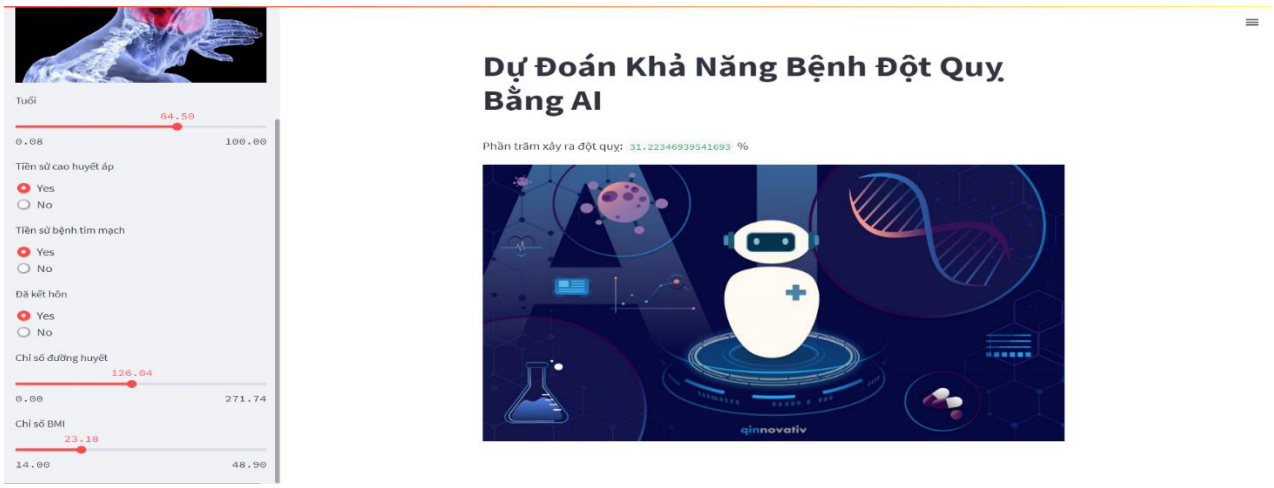
```
from sklearn.neighbors import KNeighborsClassifier
knn_model=KNeighborsClassifier(n_neighbors=20,weights="distance",algorithm="auto",metric="minkowski")
knn_model.fit(X_train_os,y_train_os)
evalate_model(knn_model,X_test_os,y_test_os)
```

```
[[708 236]
 [ 25 925]]
              precision    recall  f1-score   support

           0       0.97      0.75      0.84       944
           1       0.80      0.97      0.88       950

    accuracy                           0.86      1894
   macro avg       0.88      0.86      0.86      1894
weighted avg       0.88      0.86      0.86      1894
```

Sau khi thực hiện training tiến hành kiểm tra các giá trị của model. Ở hình bên trên các chỉ số *recall, f1-score, precision* ở mức ổn và ***accuracy*** ở mức chấp nhận được 0.86. Cho nên nhóm sẽ lưu và sử dụng model này để phát triển ứng dụng.

3. Tạo ứng dụng với Streamlit

Code xử lý của ứng dụng sẽ nằm trong file app.py chứa thư viện *streamlit, panda, encoder_data, dill.* Để đọc dữ liệu của người dùng sau đó gọi ra model để xử lý và đưa ra kết quả.



Giao diện của ứng dụng

V. Cài đặt kiểm thử

Link code: Github

Dùng lệnh py -m streamlit run app.py để chạy hoặc truy cập link để sử dụng Streamlit App

Kiểm thử

+ Bộ ***dữ liệu 1***:

{Tuổi : **55.3**, Tiền sử huyết áp: **yes**, Tiền sử tim mạch: **yes**, Đã kết hôn: **yes**, Chỉ số đường huyết: **175.20**, Chỉ số BMI: **21.00**}

Kết quả: `60.25084453604844` %

+ Bộ ***dữ liệu 2***:

{Tuổi : **24.07**, Tiền sử huyết áp: **no**, Tiền sử tim mạch: **no**, Đã kết hôn: **no**, Chỉ số đường huyết: **125.00**, Chỉ số BMI: **20.00**}

Kết quả: `0.00`%

+ Bộ ***dữ liệu 3***:

{Tuổi : **75.02**, Tiền sử huyết áp: **yes**, Tiền sử tim mạch:**yes**, Đã kết hôn: **yes**, Chỉ số đường huyết:**170.73**, Chỉ số BMI: **25.60**}

Kết quả: `89.1176679899934`%

+ Bộ ***dữ liệu 4***:

{Tuổi : **40.51**, Tiền sử huyết áp: **yes**, Tiền sử tim mạch:**no**, Đã kết hôn: **yes**, Chỉ số đường huyết:**166.26**, Chỉ số BMI: **22.73**}

Kết quả: `13.436940156106337`%

Như vậy ứng dụng đã cơ bản hoàn thiện được mục tiêu đề ra ban đầu là xây dựng được ứng dụng dự đoán được phần trăm đột quỵ Dựa trên thuật toán KNN và thư viện sklearn

VI. Đánh giá

Sau khi thực hiện đề tài nhóm đã nhận thấy 1 số ưu và khuyết điểm sau

Về ưu điểm:

+ Hiểu được quy trình tạo ra 1 ứng dụng AI

+ Khai phá xử lý bộ dữ liệu cung cấp

+ Tìm hiểu được các thuật toán học máy (Machine Learning)

+ Triển khai ứng dụng của AI vào lĩnh vực y tế.

+ Sử dụng các thư viện của ngôn ngữ python hỗ trợ để phát triển 1 ứng dụng hoàn thiện.

Bên cạnh đó tồn tại 1 số *khuyết điểm* sau:

+ Bộ dữ liệu của Kaggle có độ mất cân bằng khá lớn ảnh hưởng đến chất lượng của model.

+ Thuật toán KNN mang lại độ chính xác ở mức vừa phải không quá cao.

+ Ứng dụng đôi khi hoạt động không chính xác với 1 số bộ dữ liệu.

Hướng *khắc phục* và *phát triển*

+ Tìm thêm các dataset ở các trung tâm báo cáo sức khỏe có uy tính trong và ngoài nước để train.

+ Có thể thay thế các thuật toán giúp tăng độ chính xác như XGBoost, Random Forest, SVM…

+ Phát triển ứng dụng theo dạng API để người dùng có thể sử dụng song song quá trình đó hệ thống sẽ học hỏi nâng cao độ chính xác.

TÀI LIỆU THAM KHẢO

1. Trương Thanh , Trần Châu Mỹ Thanh, Nguyễn Thị Như Ly (2020), "Trí tuệ nhân tạo trong chăm sóc sức khỏe" , Tạp Chí Khoa Học & Công Nghệ Đại Học Duy Tân (tr.25).
2. Vũ Hữu Tiệp (2020), Machine Learning cơ bản.
3. Stroke Prediction Dataset, Kaggle, https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset truy cập 30/11/2022