IE212.M12.CNCL

Công nghệ Dữ liệu lớn

Nhóm 11

Xây dựng hệ thống dự đoán đột quỵ

18521051 - Phặm Thăng Long
18520888 - Lê Nhị Khang
18520757 - Võ Đoàn Minh Hiếu

Mục lục

I.GIỚI THIỆU

Giới thiệu

Đột quỵ là một tổn thương đến não xảy ra khi dòng máu cung cấp cho não bị gián đoạn hoặc giảm đáng kể. Não bị thiếu oxy và dinh dưỡng và các tế bào não bắt đầu chết trong vòng vài phút.

800.000

Người Mỹ bị đột quỵ hàng năm

137.000

Người trong số đó tử vong

Giới thiệu

Để hiện thực được mục đích của để tài, nhóm xử dụng 3 công cụ

PYSPARK

là một giao diện cho phép truy cập Spark bằng cách sử dụng Python

SPARK STREAMING

là một phần bổ sung cho Spark để xử lý lượng dữ liệu lớn tức thì và đảm bảo chống chịu lỗi

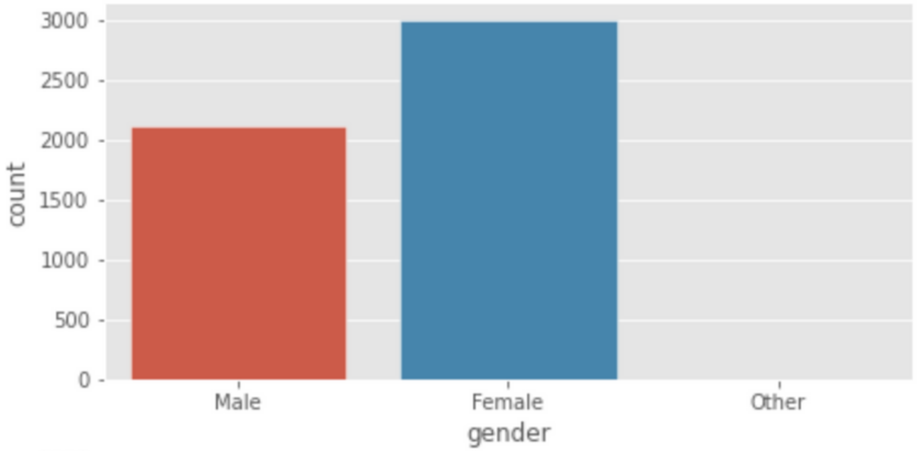
FLUTTER

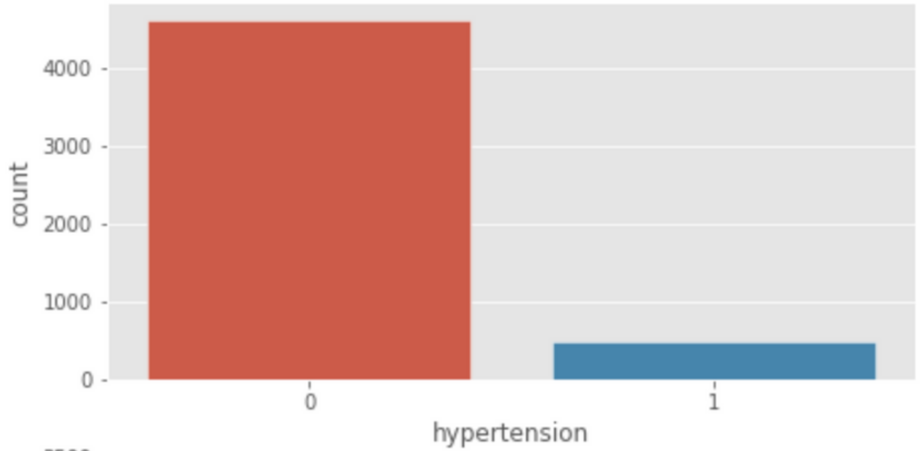
Là framework giúp xây dựng ứng dụng mobile đa nền tảng

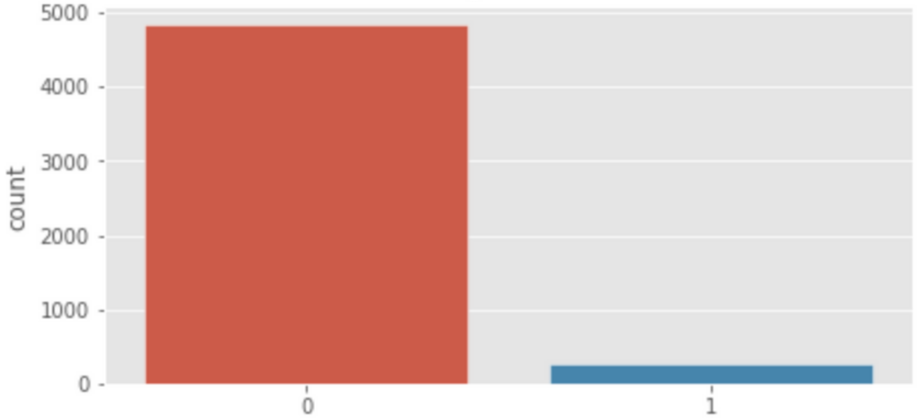
II. CÁCH TIẾP CẬN

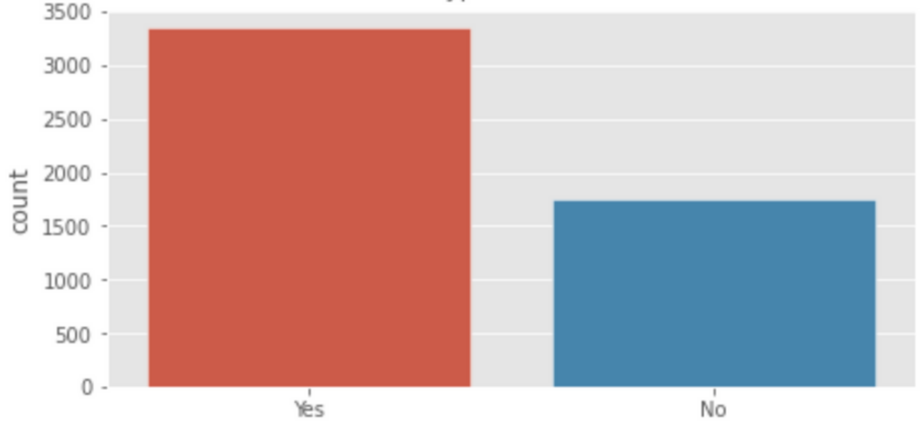
PHÂN TÍCH DỮ LIỆU

BẢNG PHÂN TÍCH DỮ LIỆU DATA TỪ KAGGLE

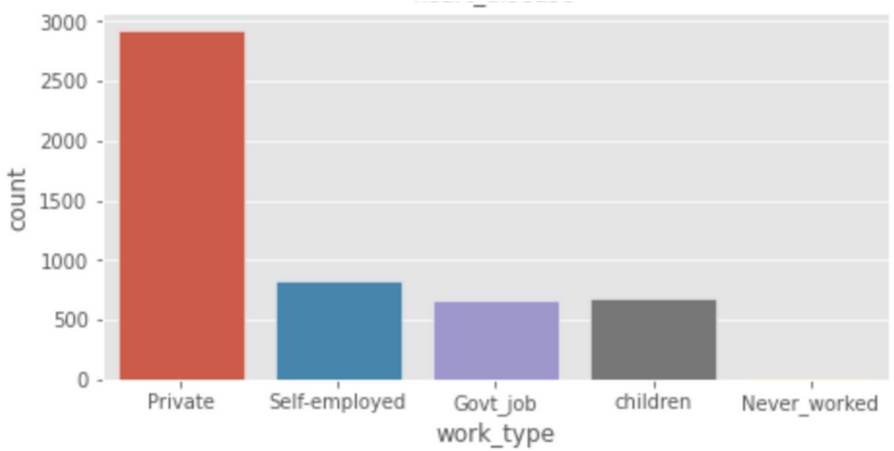


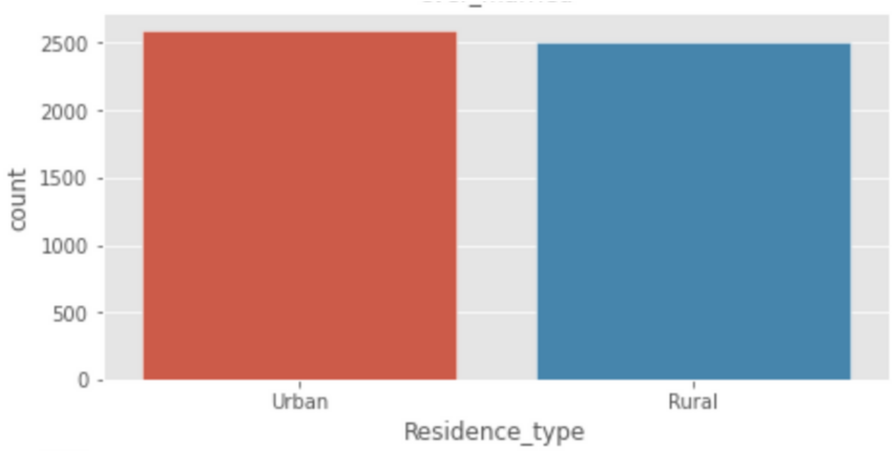


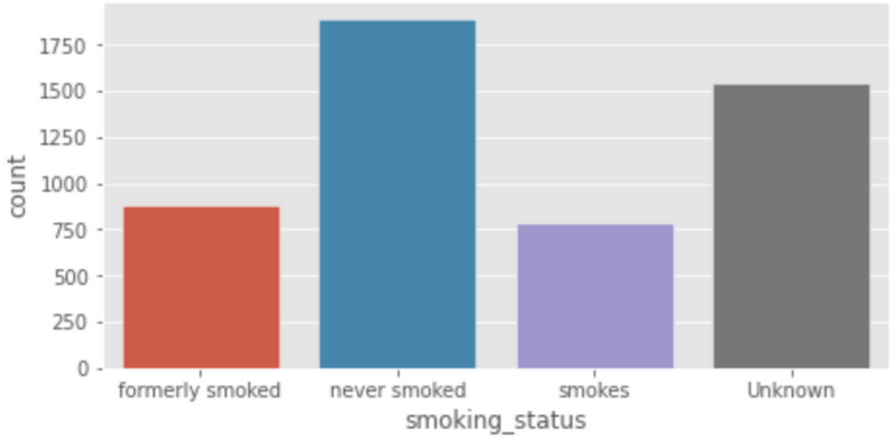


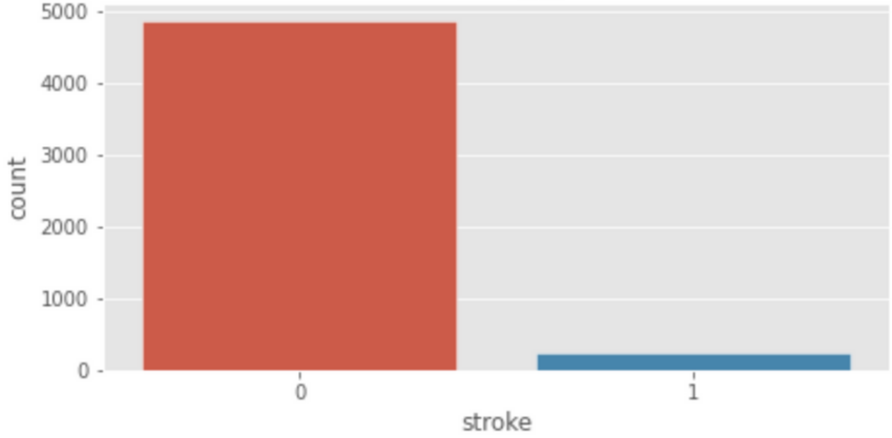


BẢNG PHÂN TÍCH DỮ LIỆU DATA TỪ KAGGLE





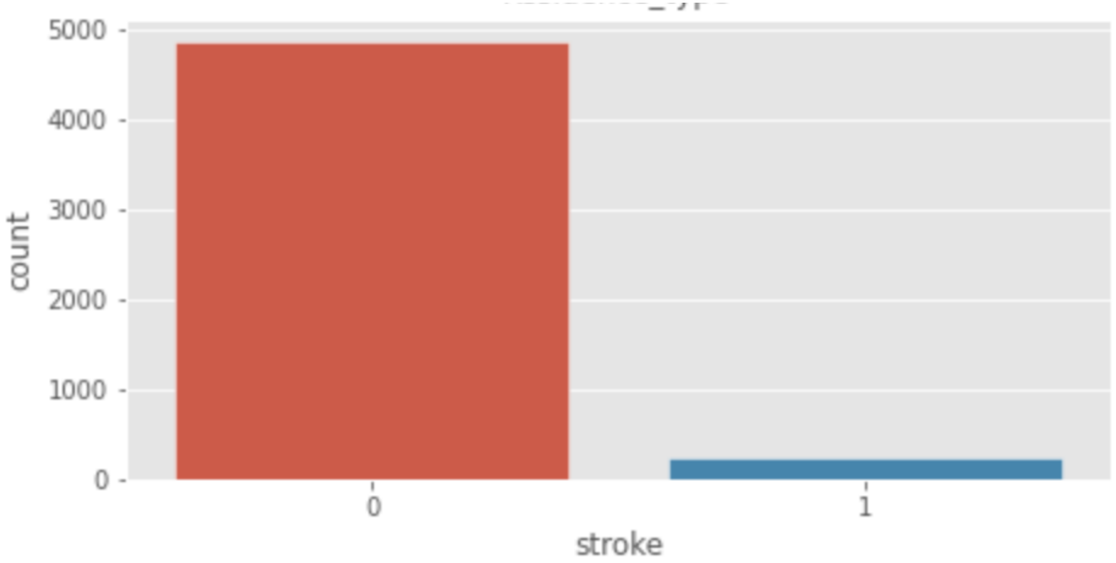




MẤT CÂN BẰNG DỮ LIỆU

- Hiện tại tập dataset đang mất cần bằng giữa trường Stroke

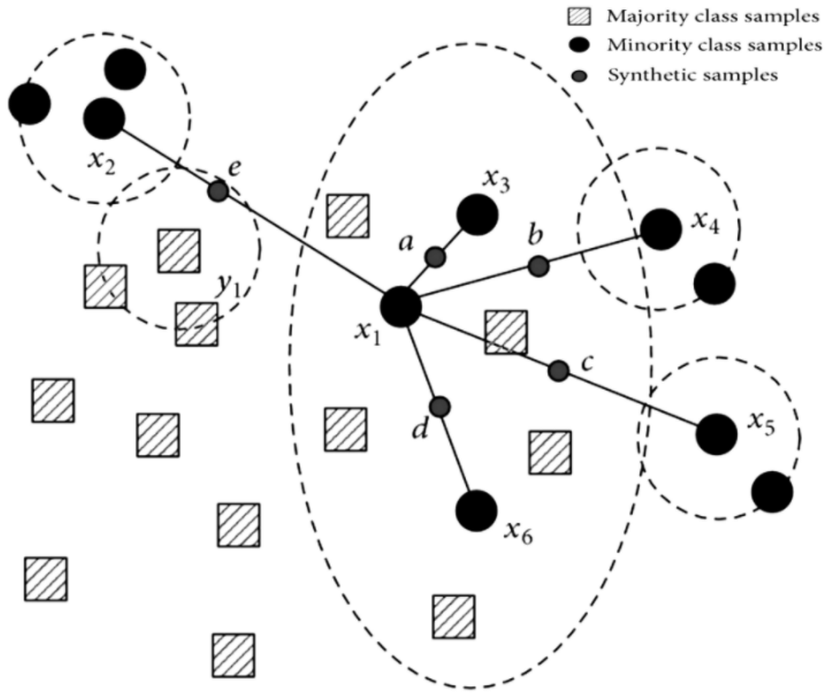
- Số lượng stroke có giá trị 0 nhiều hơn rất nhiều so với giá trị 1



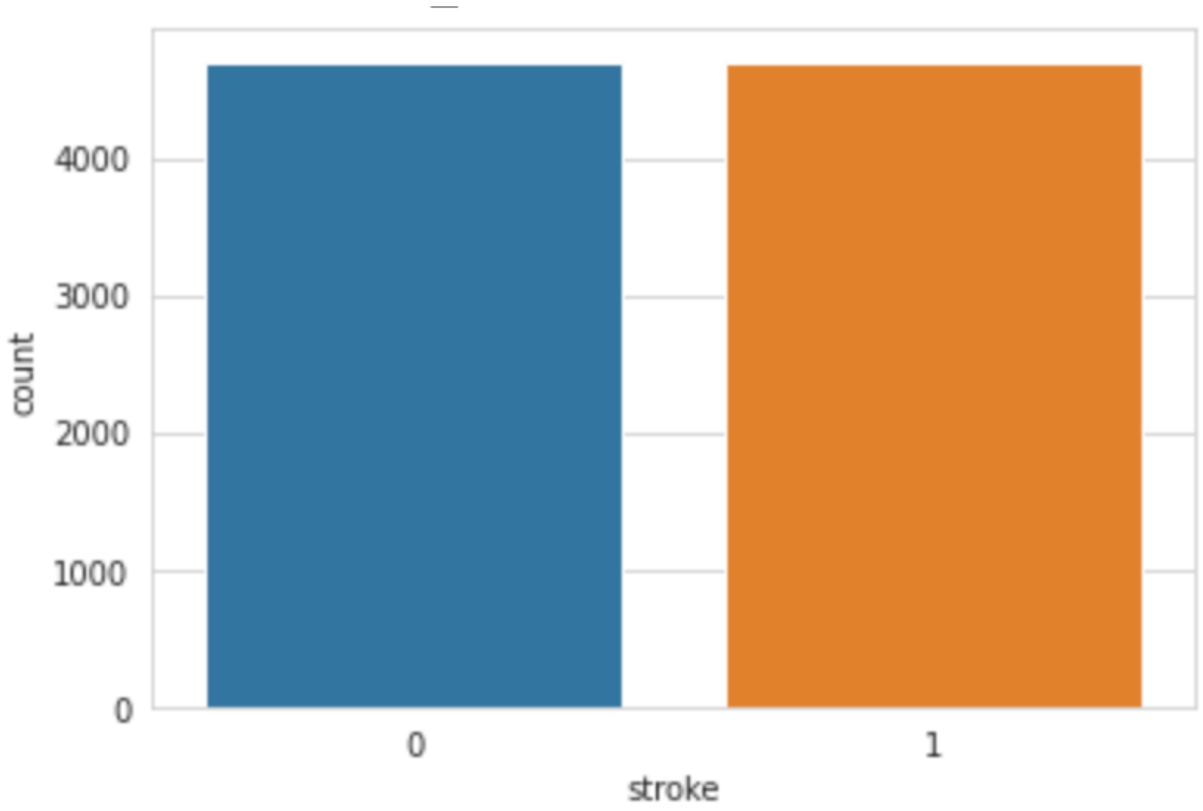
MẤT CÂN BẰNG DỮ LIỆU

- Nhóm quyết định sử dụng thư Smote để cân bằng giữ liệu

- SMOTE sẽ sử dụng thuật toán K-mean để nhân bản data, giúp cần bằng data khi bị mất cần bằng dữ liệu



SAU KHI XÀI SMOTE



STRING-INDEXER

- Sử dụng String indexer trong Pyspark sẽ tương đồng với Label Indexer.

- Việc đưa dạng data String sang dạng số sẽ giúp cho việc training dữ liệu trở nên dễ dàng hơn.

- Gender
- Ever_married
- Work_type

- Residence_type
- Smoking_status

ONEHOT-ENCODER

- One-hot encoding là quá trình biến đổi từng giá trị thành các đặc trưng nhị phân chỉ chứa giá trị 1 hoặc 0.

- Mỗi mẫu trong đặc trưng phân loại sẽ được biến đổi thành một vector có kích thước m chỉ với một trong các giá trị là 1 (biểu thị nó là active)

- Gender
- Ever_married
- Work_type

- Residence_type
- Smoking_status

STANDARD-SCALER

- Ý tưởng đằng sau StandardScaler là sẽ biến đổi dữ liệu của bạn sao cho phân phối của nó sẽ có giá trị trung bình là 0 và độ lệch chuẩn là 1.

- Với phân phối dữ liệu, mỗi giá trị trong tập dữ liệu sẽ bị trừ giá trị trung bình mẫu và sau đó chia cho độ lệch chuẩn của toàn bộ dữ liệu.

Standardization:

$$z = \frac{x - \mu}{\sigma}$$

with mean:

$$\mu = \frac{1}{N}\sum_{i=1}^{N}(x_i)$$

and standard deviation:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

Min-Max scaling:

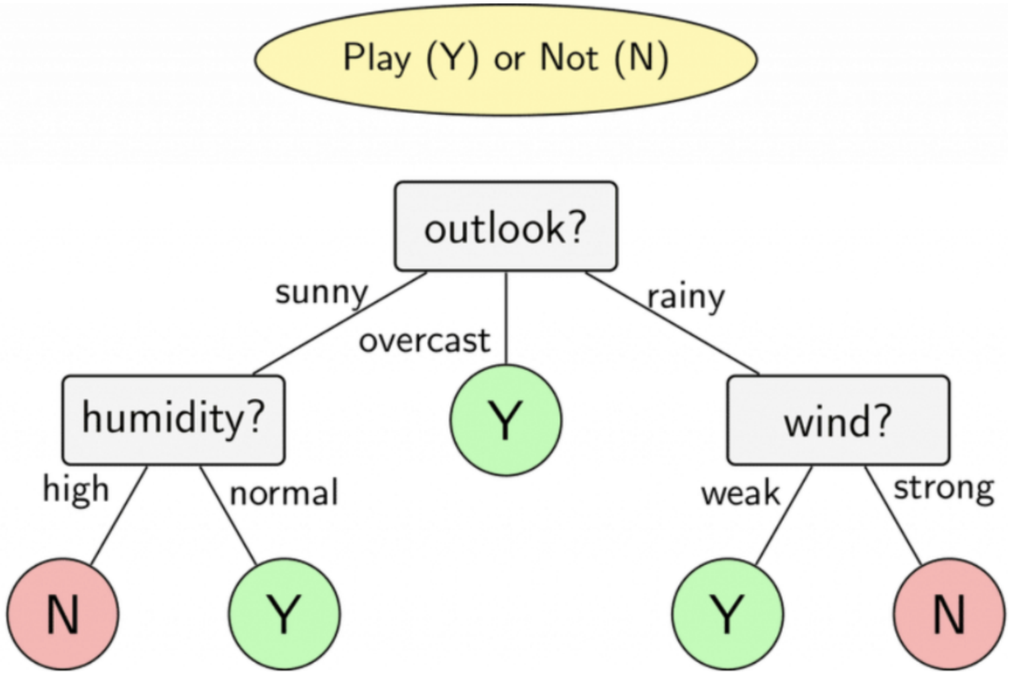
$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

- Hearth_disease
- Avg_glucose_level
- Bmi
- Age
- Hypertension

XÂY DỰNG HỆ THỐNG DỰ ĐOÁN

DECISION TREE

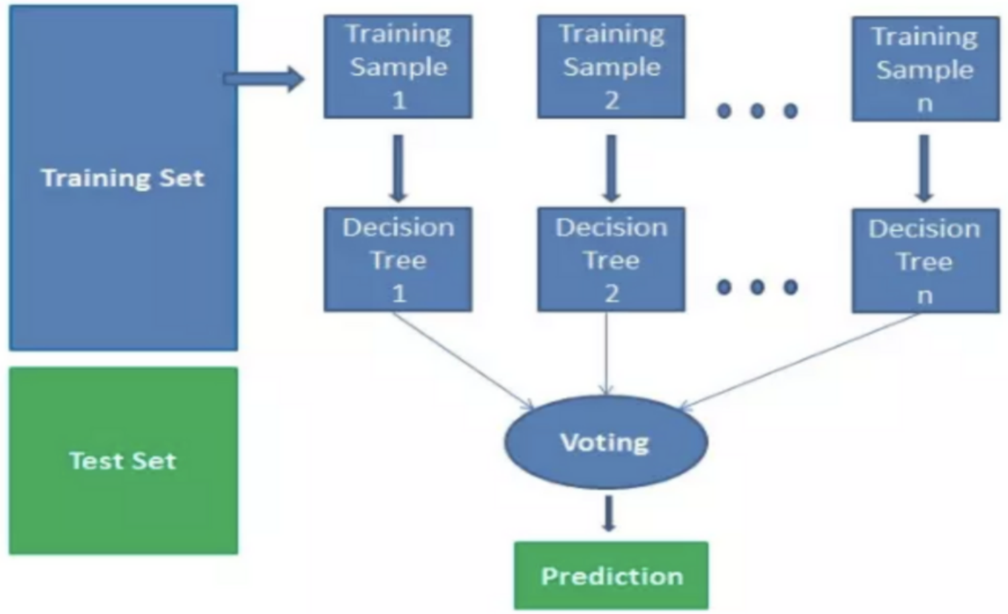
- Cây quyết định (Decision Tree) là một cây phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào dãy các luật



Kết quả độ chính xát: 0.8281907433380085

RANDOM FOREST

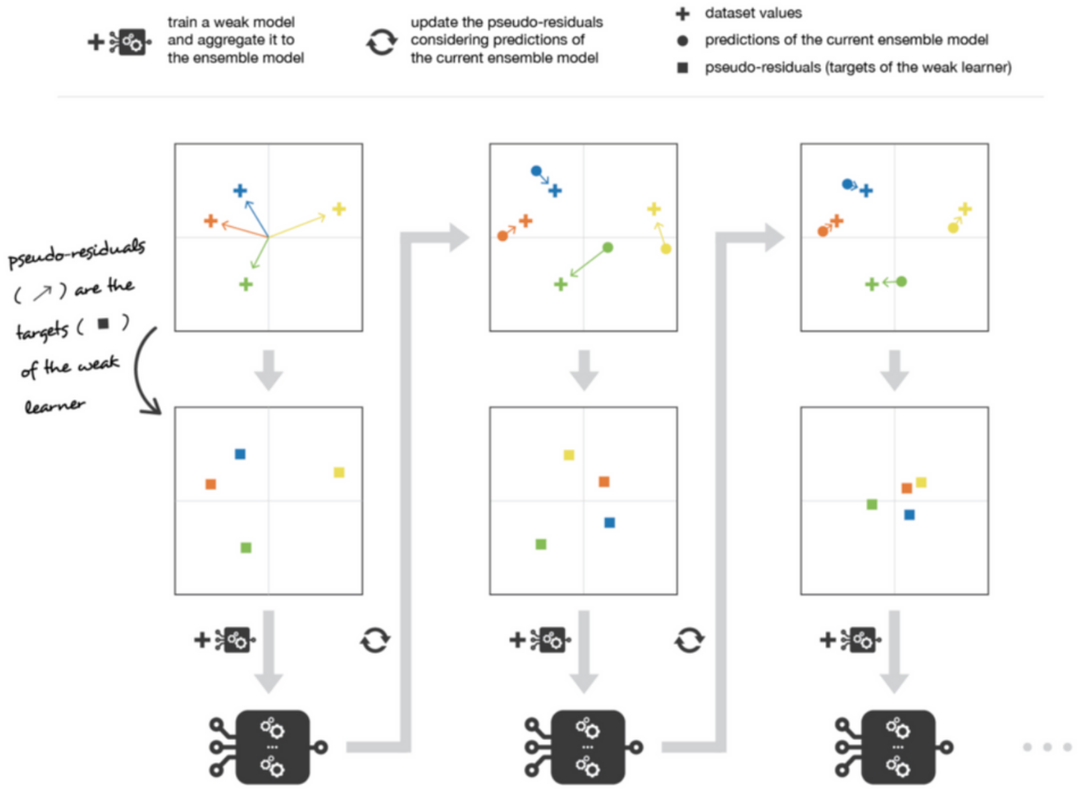
- Random Forests là thuật toán học có giám sát (supervised learning). Nó có thể được sử dụng cho cả phân lớp và hồi quy. Nó cũng là thuật toán linh hoạt và dễ sử dụng nhất. Một khu rừng bao gồm cây cối. Người ta nói rằng càng có nhiều cây thì rừng càng mạnh



Kết quả độ chính xát: 0.8269921033740 13

GBT

- Gradient Boosting là một dạng tổng quát hóa của AdaBoost. Thuật toán để phân loại. Nó hỗ trợ các nhãn nhị phân, cũng như các tính năng liên tục và phân loại



Kết quả độ chính xát: 0.8612959719789842

TỔNG KẾT THUẬT TOÁN

- Sau khi nhóm sử dụng 3 thuật toán thì GBT có độ chính xác cao nhất.

- Nhóm sẽ sử GBT là model chính

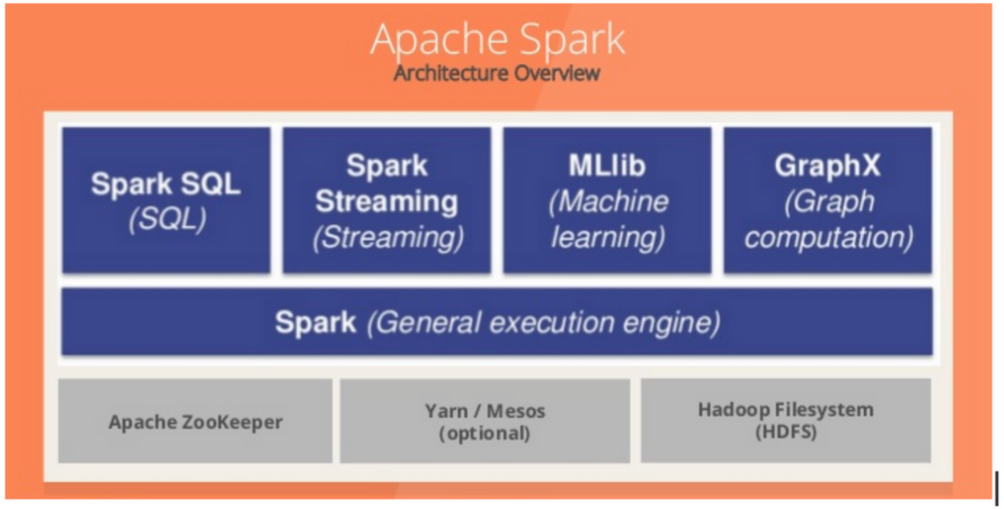
Decision Tree	0.8281907433380085
Random Forest	0.826992103374013
GBT	0.8612959719789842

XÂY DỰNG HỆ THỐNG STREAMING

Spark-Streaming

- Spark Streaming dựa trên Spark Core, là một phần bổ sung cho Spark để xử lý lượng dữ liệu lớn tức thì và đảm bảo chống chịu lỗi.

- Spark Streaming đóng vai trò cung cấp nền tảng để đẩy dữ liệu vào các mô hình phân tích tức thời, tăng hiệu năng của mô hình



Flutter

- Flutter là framework được xử dụng để xây ứng dụng mobile đa nền tảng

- Nhóm kết hợp việc xử dụng Flutter kết với Spark-Streaming core để xây dựng một ứng dụng Mobile

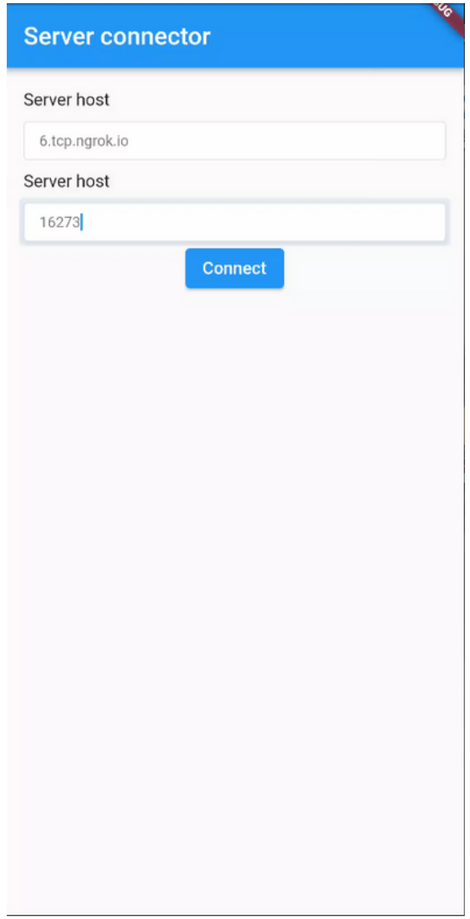
Xây dựng server

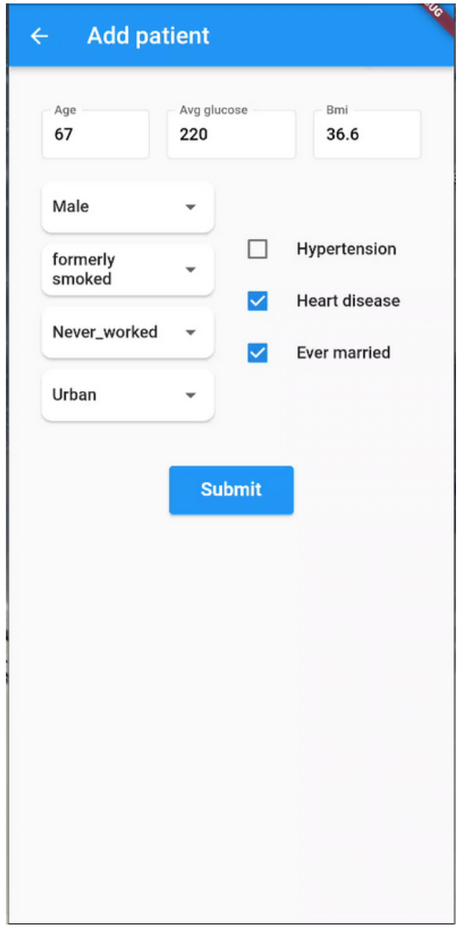
Xử dụng Dart xây dựng 2 server TCP

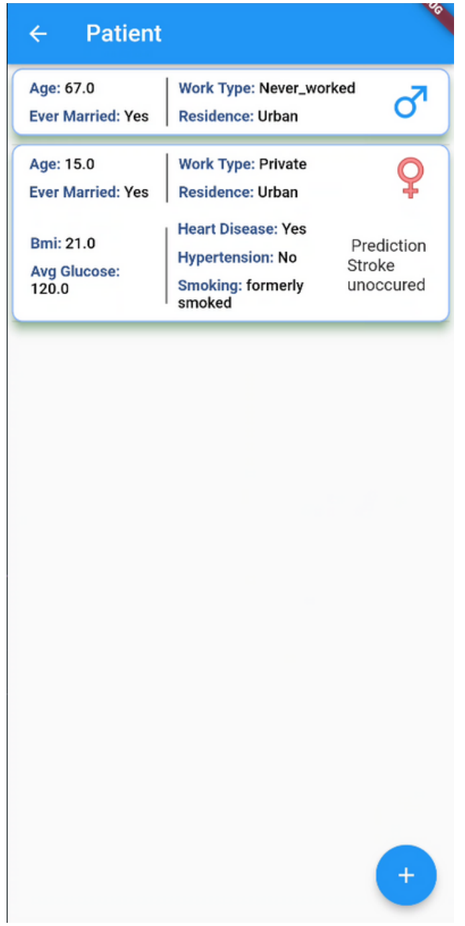
- 1 Server để kết nối với Py-spark dùng trong data structured streaming ,query trong streaming data và trả về các dự đoán

- 1 Server để kết nối với các thiết bị liên quan nhằm thu thập dữ liệu các bệnh nhân cần dự đoán

Một số hình ảnh của ứng dụng







III. KẾT LUẬN

Kết luận

- Theo những kết quả đã thể hiện trong báo cáo này việc sử dụng Spark Streaming một phần mở rộng từ lõi của Spark Api đã giúp xử lý tốt được luồng dữ liệu liên tục và giúp kết nối giữa các thiết bị với Spark trở nên dễ dàng.

- Tuy nhiên kết quả đạt được với phương pháp tiếp cận này vẫn chưa đưa đến độ chính xác, cần cải thiện các hướng đi về phân tích dữ liệu và thuật toán để tăng độ chính xác cho hệ thống.

IV. DEMO

THANK YOU

SOURCE-CODE:

HTTPS://GITHUB.COM/SKIZDUKION/HEALTHCARE_PREDICTION

HTTPS://COLAB.RESEARCH.GOOGLE.COM/DRIVE/1UBJRPK1R7XGN6TJVMDGQF-WQGKM9JDGH

HTTPS://COLAB.RESEARCH.GOOGLE.COM/DRIVE/1TTQPUUD9FTBXMSSRBPYXHNNGD7FJMGNF

DEMO:

HTTPS://DRIVE.GOOGLE.COM/FILE/D/1OLX7OY2TYDIGTP0JVK36SLWXGX340GGA/VIEW