**TRƯỜNG ĐẠI HỌC SƯ PHẠM HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN**

**Học phần: COMP261– Trí tuệ nhân tạo**

**DỰ ĐOÁN BỆNH NHÂN BỊ BỆNH TIỂU ĐƯỜNG THÔNG QUA MÔ HÌNH HỌC MÁY**

**Giảng viên: Thầy Lê Xuân Hiền**

**Sinh viên thực hiện : Nguyễn Hoàng Tùng - 735105112**

**MỤC LỤC**

[1. GIỚI THIỆU DỰ ÁN 3](#_Toc202129054)

[2. MÔ TẢ DỮ LIỆU 3](#_Toc202129055)

[3. TIỀN XỬ LÝ DỮ LIỆU 4](#_Toc202129056)

[4. KHÁM PHÁ VÀ TRỰC QUAN HOÁ DỮ LIỆU 5](#_Toc202129057)

[5. XÂY DỰNG VÀ ĐÁNH GIÁ MÔ HÌNH 7](#_Toc202129058)

[6. TUNING VÀ LỰA CHỌN MÔ HÌNH 11](#_Toc202129059)

[7. TRIỂN KHAI VÀ ỨNG DỤNG THỰC TẾ 12](#_Toc202129060)

[8. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 13](#_Toc202129061)

[9. TÀI LIỆU THAM KHẢO 13](#_Toc202129062)

# GIỚI THIỆU DỰ ÁN

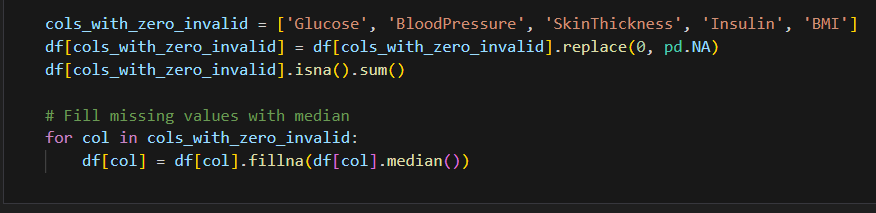
* Mục tiêu:
  + Ứng dụng các thuật toán học máy (Machine Learning) để xây dựng mô hình có khả năng **dự đoán khả năng mắc bệnh tiểu đường** ở bệnh nhân dựa trên các thông số y tế (chỉ số BMI, huyết áp, số lần mang thai, mức insulin, …).
* Lý do chọn đề tài:
  + Bệnh tiểu đường là một trong những bệnh mãn tính nguy hiểm, có thể gây ra nhiều biến chứng nghiêm trọng nếu không được phát hiện và điều trị kịp thời.
  + Việc ứng dụng AI vào lĩnh vực y tế giúp hỗ trợ bác sĩ phát hiện sớm bệnh lý, giảm gánh nặng cho hệ thống y tế và nâng cao chất lượng cuộc sống cho bệnh nhân.

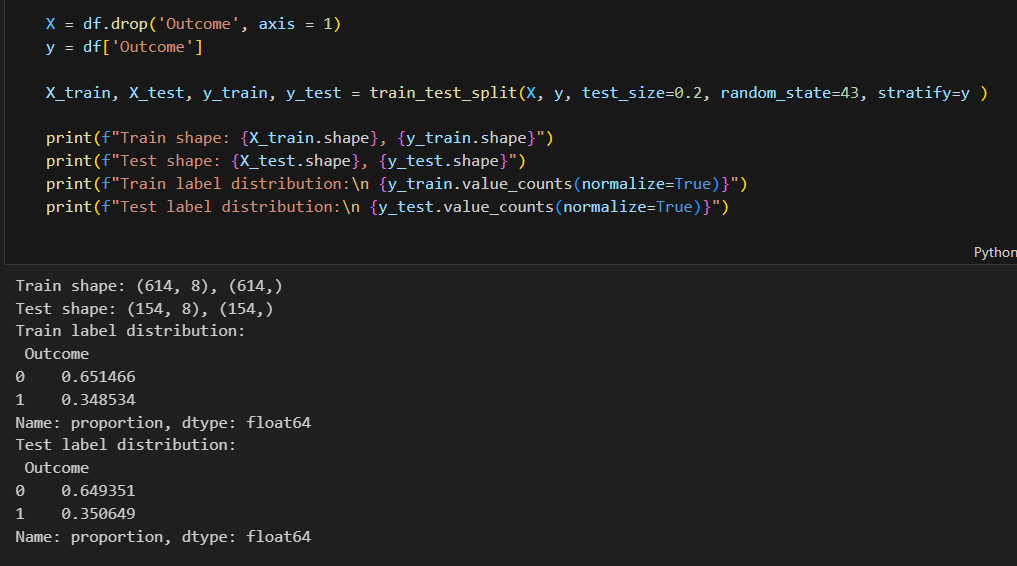
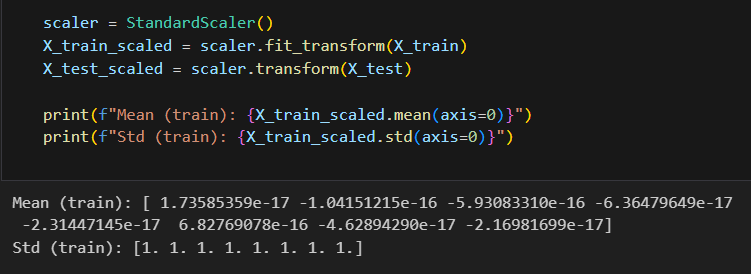
# MÔ TẢ DỮ LIỆU

* Nguồn dữ liệu: [🌟Diabetes Dataset For Beginners](https://www.kaggle.com/code/melikedilekci/diabetes-dataset-for-beginners/input)
* Quy mô: 768 mẫu dữ liệu, 8 thuộc tính lâm sàng, 1 nhãn.
* Các thuộc tính lâm sàng:
  + Pregnancies: Số lần mang thai.
  + Glucose: Nồng độ glucose trong máu.
  + BloodPressure: Huyết áp.
  + SkinThickness: Độ dày da.
  + Insulin: Nồng độ insulin.
  + BMI: Chỉ số khối cơ thể.
  + DiabetesPedigreeFunction: Chỉ số di truyền.
  + Age: Tuổi.
* Nhãn: Outcome: 0 (không bệnh), 1 (có bệnh).
* Nhận xét dữ liệu thô:
  + Dữ liệu ban đầu ổn định, không có giá trị rỗng.
  + Dựa trên mẫu dữ liệu đầu vào ta thấy có sự chênh lệch khá lớn giữa số người bị bệnh và người không bị bệnh.
  + Dữ liệu có sự phân tán lớn ở một số thuộc tính. Như là Glucose, Insulin, BMI có nhiều giá trị ngoại lại – các điểm nằm ngoài “râu” của boxplot.
  + Ma trận tương quan ( correlation matrix ) cho thấy Glucose có mức tương quan mạnh với Outcome.

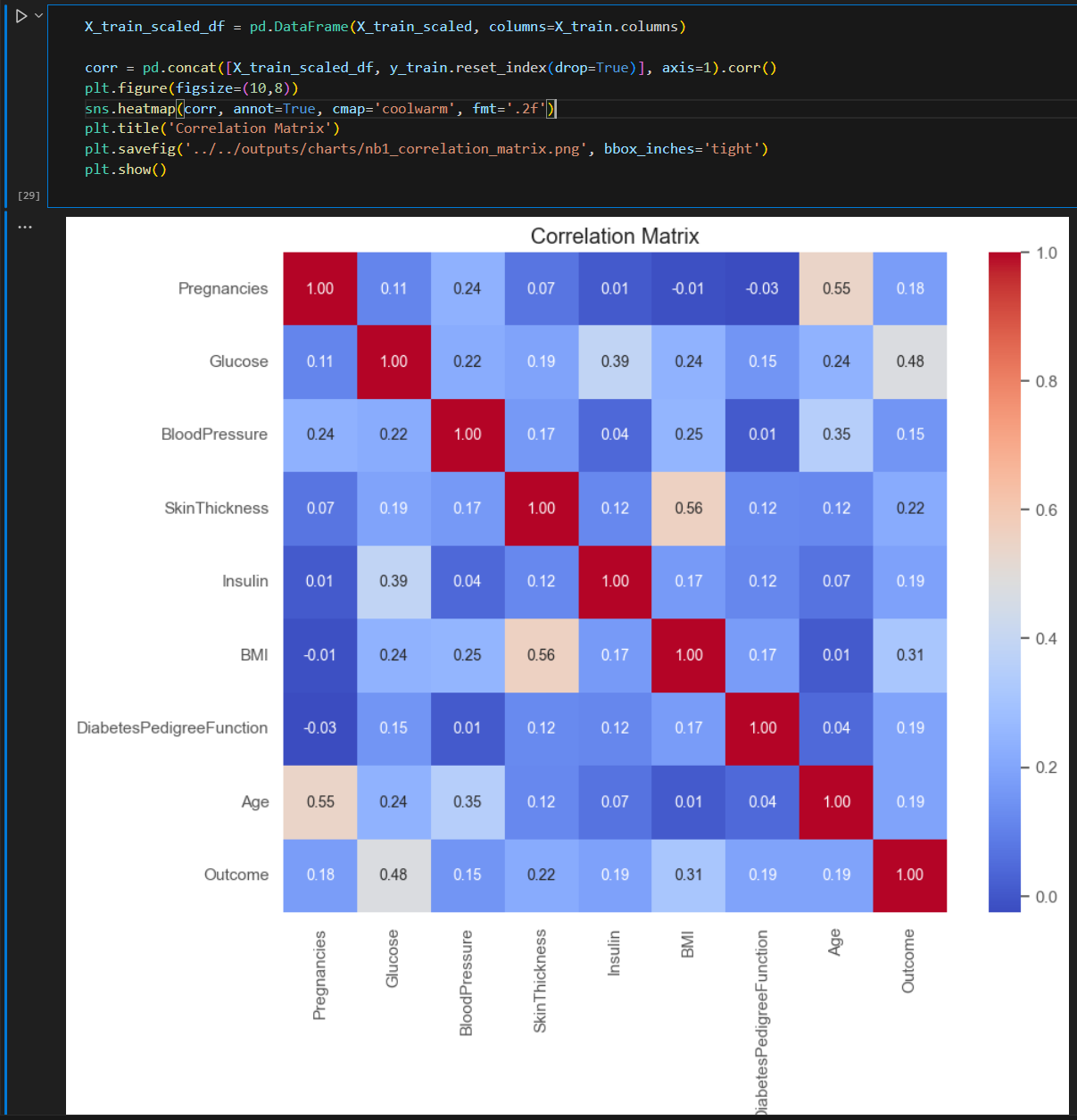
# TIỀN XỬ LÝ DỮ LIỆU

* Kiểm tra và xử lý dữ liệu không phù hợp.
  + Thay giá trị 0 bằng trung vị ở các cột Glucose, BloodPressure, SkinThickness, Insulin, BMI.

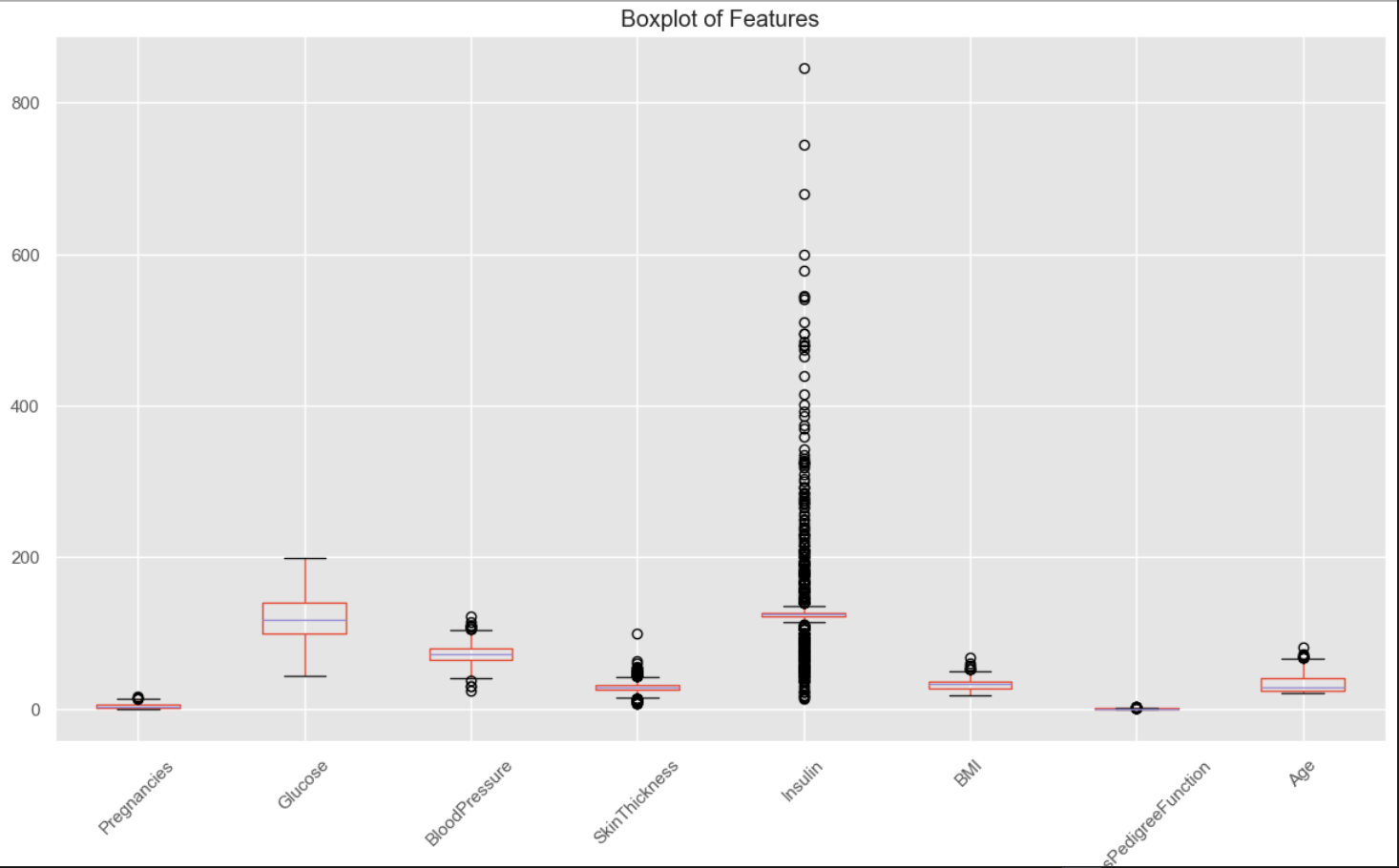


* Chia tập huấn luyện/kiểm tra: 80/20.
  + 
* Chuẩn hoá dữ liệu.
  + 

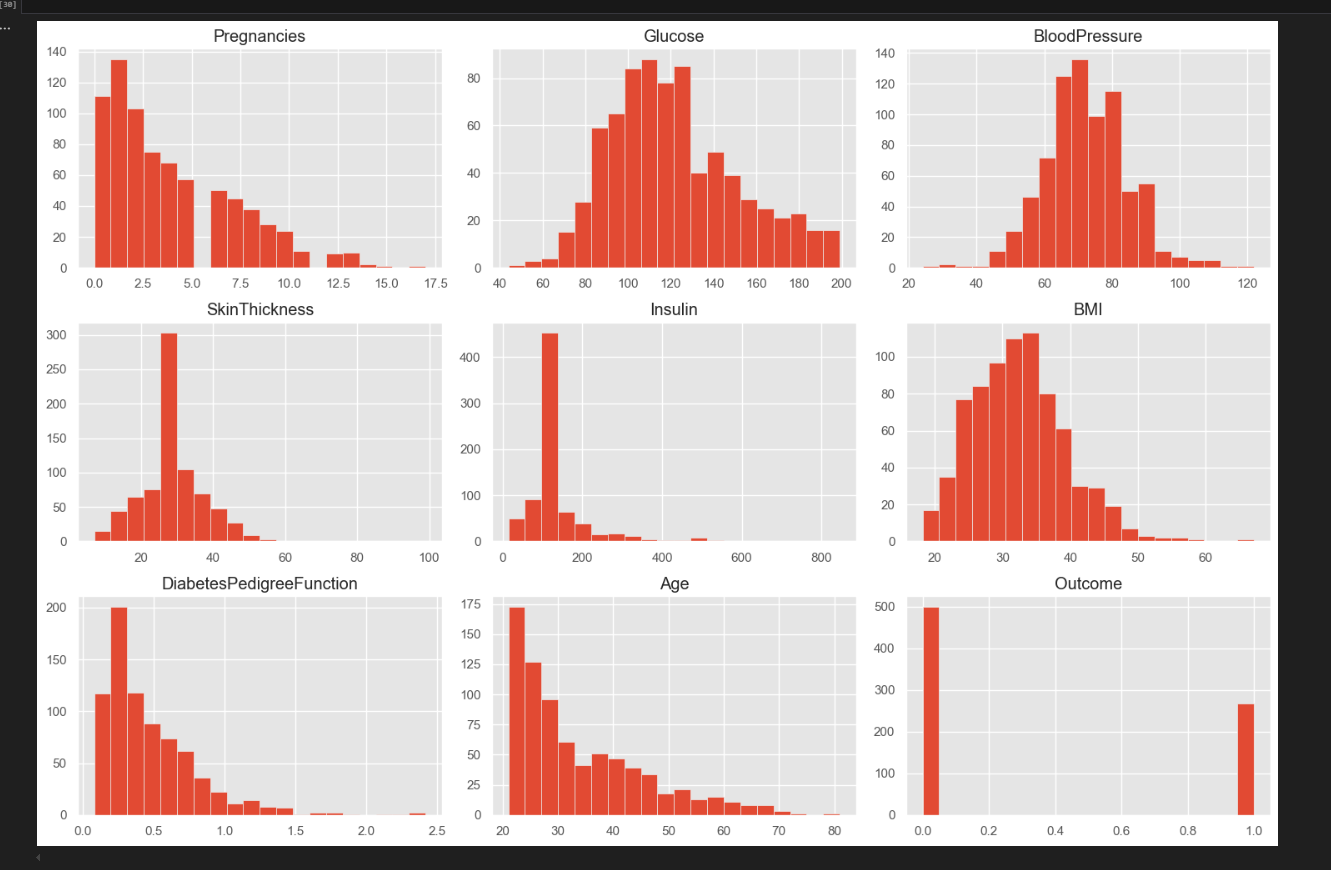
# KHÁM PHÁ VÀ TRỰC QUAN HOÁ DỮ LIỆU



* Thông qua Correlation Matrix ta thấy được Glucose có tương quan mạnh với Outcome.



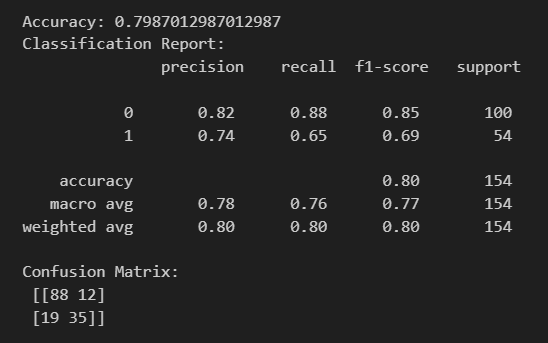
* Hình này cho thấy Insulin có rất nhiều outlier với giá trị lớn vượt xa các đặc trưng khác. Một số các chỉ số khác như SkinThickness và BloodPressure cũng có một số outlier nhưng không quá nghiêm trọng.



* Phân phối các đặc trưng như Pregnancies, Insulin, DiabetesPedigreeFunction và Age bị lệch phải.
* Cần cân nhắc xử lý outlier ( đặc biệt là với Insulin ).
* Việc chuẩn hoá dữ liệu là quan trọng do phân phối không đồng đều.

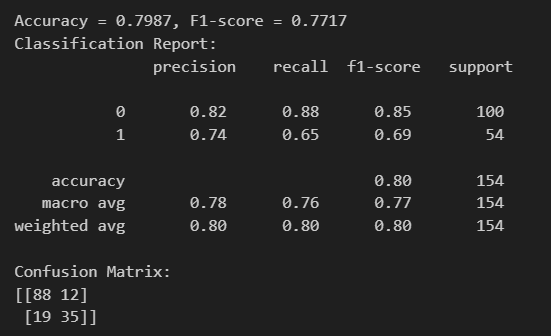
# XÂY DỰNG VÀ ĐÁNH GIÁ MÔ HÌNH

* Mô hình: Logistic Regression, K-Nearest Neighbors, Random Forest, Support Vector Machine.
* Chỉ số đánh giá: Accuracy, Precision, Recall, F1-score, Confusion Matrix.
  1. **Logistic Regression**



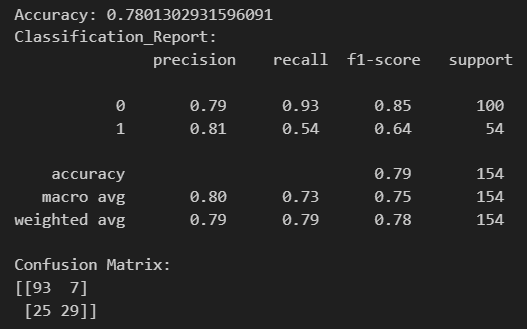
**Nhận xét:**

* Mô hình có độ chính xác khá tốt (~80.5%), hiệu suất tổng thể ổn định.
* Hiệu suất giữa hai lớp không đồng đều, đặc biệt là với lớp 1.
  + Lớp 0 (không bị bệnh):
    - Precision: 82% -> Trường hợp dự đoán "không bị bệnh" đúng.
    - Recall: 89% -> Mô hình rất tốt trong việc dự đoán người không bị bệnh.
    - F1-score: 86% -> Cân bằng tốt giữa Precision và Recall.
    - Kết luận: Mô hình hoạt động tốt với lớp 0.
  + Lớp 1 (bị bệnh):
    - Precision: 76% -> Trường hợp dự đoán "bị bệnh" đúng.
    - Recall: 65% -> Mô hình bỏ sót khá nhiều người thực sự bị bệnh.
    - F1-score: 70% -> Cần phải cải thiện.
    - Kết luận: Mô hình có thiếu sót lớn trong việc tìm ra người bị bệnh.
* Tổng kết: Mô hình chưa đủ tốt trong việc nhận diện bệnh nhân bị bệnh.
  1. **K-Nearest Neighbors**



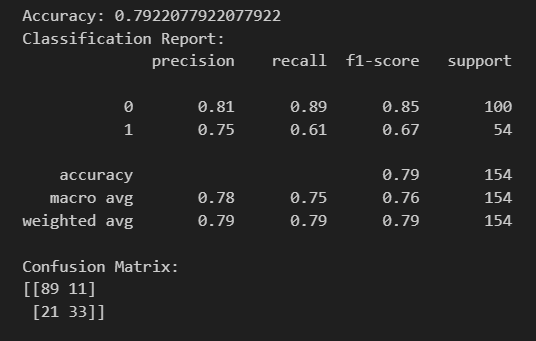
**Nhận xét:**

* Mô hình có độ chính xác khá tốt (~81.2%), hiệu suất tổng thể ổn định.
* Hiệu suất giữa hai lớp không đồng đều, đặc biệt là với lớp 1.
  + Lớp 0 (không bị bệnh):
    - Precision: 81% → Dự đoán "không bị bệnh" đúng với độ chính xác cao.
    - Recall: 93% → Mô hình rất tốt trong việc phát hiện người không bị bệnh.
    - F1-score: 87% → Cân bằng rất tốt giữa độ chính xác và khả năng bao phủ.
    - Kết luận: Mô hình hoạt động rất tốt với lớp 0.
  + Lớp 1 (bị bệnh):
    - Precision: 82% → Dự đoán "bị bệnh" đúng với độ chính xác khá cao.
    - Recall: 59% → Mô hình bỏ sót nhiều người thực sự bị bệnh.
    - F1-score: 69% → Hiệu suất chưa cao, cần cải thiện.
    - Kết luận: Mô hình còn thiếu sót lớn trong việc phát hiện người bị bệnh.
* Tổng kết: Mô hình vẫn chưa đủ mạnh để đảm bảo phát hiện đầy đủ bệnh nhân vì tỷ lệ bỏ sót bệnh (False Negative) còn cao.
  1. **Random Forest**



**Nhận xét:**

* Mô hình có độ chính xác ở mức khá (~78.3%), tuy nhiên chưa thực sự cao để đảm bảo độ tin cậy trong các bài toán nhạy cảm như y tế.
* Hiệu suất giữa hai lớp không đồng đều, đặc biệt là với lớp 1.
  + Lớp 0 (không bị bệnh):
    - Precision: 81% → Dự đoán "không bị bệnh" đúng với độ chính xác cao.
    - Recall: 93% → Mô hình rất tốt trong việc phát hiện người không bị bệnh.
    - F1-score: 87% → Cân bằng tốt giữa độ chính xác và khả năng bao phủ.
    - Kết luận: Mô hình hoạt động hiệu quả với lớp 0.
  + Lớp 1 (bị bệnh):
    - Precision: 82% → Dự đoán "bị bệnh" đúng với độ chính xác khá cao.
    - Recall: 59% → Mô hình bỏ sót đáng kể người thực sự bị bệnh.
    - F1-score: 69% → Hiệu suất chưa cao, cần được cải thiện.
    - Kết luận: Mô hình còn thiếu sót lớn trong việc phát hiện người bị bệnh — đây là điểm yếu nghiêm trọng nếu ứng dụng trong y tế.
* Tổng kết: Dù precision cao cho cả hai lớp, nhưng recall của lớp 1 còn thấp → mô hình chưa phù hợp nếu mục tiêu là giảm thiểu rủi ro bỏ sót bệnh nhân. Cần cải thiện để tăng độ bao phủ với người thực sự bị bệnh.
  1. **Support Vector Machine**

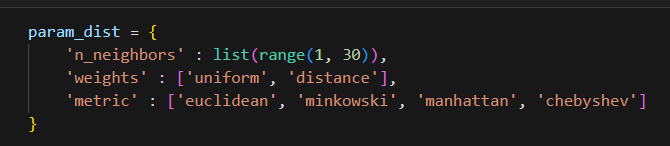


**Nhận** **xét**:

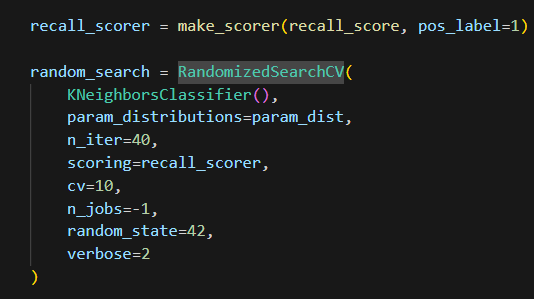
* Mô hình có độ chính xác tương đối tốt (~79.2%), tuy nhiên vẫn còn hạn chế khi xét đến hiệu suất giữa hai lớp.
* Hiệu suất không đồng đều giữa hai lớp, đặc biệt là với lớp 1.
  + Lớp 0 (không bị bệnh):
    - Precision: 81% → Dự đoán "không bị bệnh" đúng với độ chính xác cao.
    - Recall: 89% → Mô hình khá tốt trong việc phát hiện người không bị bệnh.
    - F1-score: 85% → Hiệu suất cao, cân bằng giữa precision và recall.
    - Kết luận: Mô hình hoạt động tốt với lớp 0.
  + Lớp 1 (bị bệnh):
    - Precision: 75% → Dự đoán "bị bệnh" đúng với độ chính xác trung bình.
    - Recall: 61% → Mô hình bỏ sót một tỷ lệ lớn người thực sự bị bệnh.
    - F1-score: 67% → Hiệu suất chưa cao, cần được cải thiện.
    - Kết luận: Mô hình còn hạn chế trong việc phát hiện đúng các trường hợp bị bệnh – điều này đáng lo ngại nếu áp dụng vào các lĩnh vực nhạy cảm như y tế.
* Tổng kết: Mô hình có xu hướng thiên lệch về lớp 0, hoạt động chưa hiệu quả với lớp 1. Cần cải thiện khả năng nhận diện người bị bệnh để giảm rủi ro bỏ sót bệnh nhân.

# TUNING VÀ LỰA CHỌN MÔ HÌNH

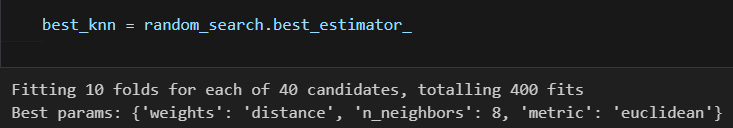
* Thông qua đánh giá, mô hình KNN có các chỉ số tổng thể tốt hơn ba mô hình còn lại.
* Ta tiếp tục phát triển mô hình thông qua tuning hyperparameter.
* Ta có các hyperparameter tiêu biểu cho KNN:



* n\_neigbors : Số lượng "hàng xóm gần nhất".
* weights : Cách tính trọng số.
* metric : Khoảng cách dùng để đo độ gần.
* Sử dụng RandomizedSearchCV tối ưu Recall của lớp 1 (dự đoán đúng bệnh nhân có bệnh).



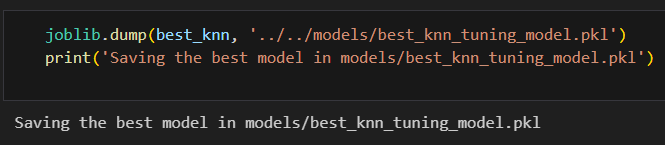
* Mô hình tốt nhất trước tuning: KNN với n\_neighbors = 19.
* Tìm ra tổ hợp hyperparameter tốt nhất



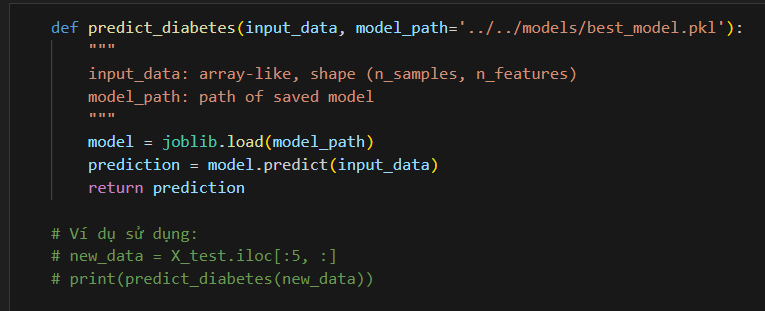
* Đánh giá lại mô hình sau Tuning
  + Sau khi điều chỉnh hyperparameter các chỉ số tổng thể (accuracy, f1-score) và recall của lớp 1 đều giảm so với mô hình mặc định.
  + Việc giữ lại và sử dụng mô hình mặc định sẽ cho ra kết quả tốt hơn.

# TRIỂN KHAI VÀ ỨNG DỤNG THỰC TẾ

* Lưu mô hình hiệu quả nhất bằng joblib



* Xây dựng hàm predict\_diabetes



* Quy trình dự đoán:
  + Load model
  + Tiền xử lý dữ liệu mới
  + Gọi dự đoán và trả về kết quả

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

* KNN là mô hình tốt nhất với dữ liệu hiện tại.
* Ưu điểm là mô hình cho ra kết quả cân bằng giữa các chỉ số đánh giá.
* Hạn chế là Recall lớp 1 còn thấp, cần cân nhắc về các kỹ thuật cân bằng dữ liệu (SMOTE, oversampling).
* Hướng phát triển:
  + Tìm thêm dữ liệu để huấn luyện mô hình, hoặc thử nghiệm các mô hình phức tạp hơn.
  + Có thể tích hợp vào hệ thống thực, kiểm thử với dữ liệu thực tế.

# TÀI LIỆU THAM KHẢO

* [🌟Diabetes Dataset For Beginners](https://www.kaggle.com/code/melikedilekci/diabetes-dataset-for-beginners/input)
* [Diabetes Dataset EDA & Prediction with 7+ Models](https://www.kaggle.com/code/chanchal24/diabetes-dataset-eda-prediction-with-7-models)
* [🍬Diabetes : EDA |🌲Random Forest🌲 + HP⚙️](https://www.kaggle.com/code/tumpanjawat/diabetes-eda-random-forest-hp)
* [Machine Learning cơ bản](https://machinelearningcoban.com/)
* ChatGPT