TRƯỜNG ĐẠI HỌC MỞ HÀ NỘI

**KHOA CÔNG NGHỆ THÔNG TIN**

----------------------



**BÁO CÁO BÀI TẬP LỚN**

**MÔN: NHẬP MÔN KHAI PHÁ DỮ LIỆU VÀ MÁY HỌC**

**Nhóm 15**

**Đề tài: Dự đoán nguy cơ mắc bệnh tiểu đường bằng phương**

**pháp học máy có giám sát**

**GVHD: ThS.Vũ Xuân Hạnh**

**Sinh viên thực hiện:**

**Vũ Thị Linh – 2110A05**

**Nguyễn Hồng Long – 2110A05**

**Phạm Văn Long – 2110A05**

**Nguyễn Đức Lượng – 2110A05**

**Hà Nội – 2024**

**PHÂN CÔNG CÔNG VIỆC VÀ KẾ HOẠCH THỰC HIỆN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TV**  **CV** | **Vũ Thị Linh** | **Nguyễn Hồng Long** | **Phạm Văn Long** | **Nguyễn Đức Lượng** |
| **TỔNG QUAN VỀ ĐỀ TÀI** | **x** | **x** | **x** | **x** |
| **THU THẬP DỮ LIỆU** | **x** | **x** | **x** | **x** |
| **TIỀN XỬ LÝ DỮ LIỆU** |  | **x** |  | **x** |
| **XÂY DỰNG MÔ HÌNH HỌC MÁY CÓ GIÁM SÁT** | **x** |  | **x** |  |
| **ĐÁNH GIÁ MÔ HÌNH HỌC MÁY** | **x** | **x** |  |  |
| **TÍCH HỢP MÔ HÌNH HỌC MÁY VÀO ỨNG DỤNG** |  |  | **x** | **x** |
| **KẾT LUẬN** | **x** | **x** | **x** | **x** |

Mục Lục

[**I.** **TỔNG QUAN VỀ ĐỀ TÀI** 3](#_Toc164005132)

[**1.** **Lý do chọn đề tài** 3](#_Toc164005133)

[**II.** **THU THẬP DỮ LIỆU** 3](#_Toc164005134)

[**1.** **Dữ liệu được thu thập ở đâu ?** 3](#_Toc164005135)

[**2.** **Đặc điểm của bộ dữ liệu thu thập** 3](#_Toc164005136)

[**III.** **TIỀN XỬ LÝ DỮ LIỆU** 5](#_Toc164005137)

[**IV.** **XÂY DỰNG MÔ HÌNH HỌC MÁY CÓ GIÁM SÁT** 6](#_Toc164005138)

[**1.** **Naive Bayes** 6](#_Toc164005139)

[**1.1.** **Cơ sở lý thuyết** 6](#_Toc164005140)

[**1.2.** **Áp dụng cơ sở lý thuyết vào xây dựng mô hình cho bài toán** 7](#_Toc164005141)

[**2.** **Decision Tree (Cây quyết định)** 7](#_Toc164005142)

[**2.1.** **Cơ sở lý thuyết** 7](#_Toc164005143)

[**2.2.** **Áp dụng cơ sở lý thuyết vào xây dựng mô hình cho bài toán** 7](#_Toc164005144)

[**3.** **Random forests** 7](#_Toc164005145)

[**3.1.** **Cơ sở lý thuyết** 7](#_Toc164005146)

[**3.2.** **Áp dụng cơ sở lý thuyết vào xây dựng mô hình cho bài toán** 8](#_Toc164005147)

[**V.** **ĐÁNH GIÁ MÔ HÌNH HỌC MÁY** 8](#_Toc164005148)

[**VI.** **TÍCH HỢP MÔ HÌNH HỌC MÁY VÀO ỨNG DỤNG** 11](#_Toc164005149)

[**VII.** **KẾT LUẬN** 14](#_Toc164005150)

[**TÀI LIỆU THAM KHẢO** 15](#_Toc164005151)

1. **TỔNG QUAN VỀ ĐỀ TÀI**
   * + 1. **Lý do chọn đề tài**

Bệnh tiểu đường là một vấn đề sức khỏe công cộng nghiêm trọng trên toàn thế giới. chính vì thế việc dưh đoán nguy cơ bị bệnh tiểu đường có thể giúp phát hiện sớm và can thiệp kịp thời, từ đó giảm nguy cơ phát triển các biến chứng và tăng khả năng quản lý bệnh của cá nhân.

* + - 1. Mục đích và xác định vấn đề:

Mục đích: xây dựng một mô hình học máy để dự đoán nguy cơ bị bệnh tiểu đường của một người dựa trên các yếu tố như tuổi, giới tính, BMI, di truyền, lịch sử gia đình, mức độ hoạt động thể chất và các chỉ số sinh học khác. Điều này đòi hỏi việc thu thập và xử lý dữ liệu lâm sàng từ các nghiên cứu và cơ sở dữ liệu y tế.

Xác định vấn đề:

* Thu thập dữ liệu.
* Tiền xử lý dữ liệu và mã hóa dữ liệu.
* Xây dựng mô hình học máy có giám sát.
* Đánh giá hiệu suất.
* Triển khai ứng dụng.

1. **THU THẬP DỮ LIỆU**
   * + 1. **Dữ liệu được thu thập ở đâu ?**

* Dữ liệu được thu thập trên Kaggle
* Gồm 1 bộ dữ liệu về bệnh tiểu đường gồm 100.000 bản ghi và 9 trường trong đó có 8 thuộc tính và 1 nhãn phân loại.
  + - 1. **Đặc điểm của bộ dữ liệu thu thập**
* Định nghĩa: bệnh tiểu đường là 1 tình trạng mãn tính, xác định bởi lượng đường huyết tăng cao trong máu gây tổn thương cho thận, tim, mắt theo thời gian.
* Đặc điểm của bộ dữ liệu thu thập:
  + Gender (Giới tính):
    - Gồm: Male (Nam) và Female (Nữ)
    - Giải thích: Giới tính khác nhau ảnh hướng có thể khác nhau. VD, nam giới có ngay cơ mắc bệnh tiểu đường cao hơn 1 chút sao với nữ giới hay phụ nữ có tiền sử mắc tiểu đường thai kỳ (tiểu đường khi mang thai) về sau có nguy cơ mắc tiểu đường tuýp 2 cao hơn.
  + Age (Tuổi):
    - Phạm vi trong tập dữ liệu từ 1 – 80 tuổi.
    - Giải thích: khi con người già đi, nguy cơ mắc bệnh tiểu đường tăng lên do giảm hoạt động thể chất, nồng độ   
      hóc – môn thay đổi, tình trạng sức khỏe kém đi.
  + Hypertension (Tăng huyết áp):
    - Gồm 2 giá trị 0 và 1.
    - Giải thích: Tăng huyết áp là một tình trạng bệnh lý trong đó huyết áp trong động mạch tăng cao liên tục. Nó có giá trị 0 hoặc 1 trong đó 0 cho biết họ không bị tăng huyết áp và với 1 có nghĩa là họ bị tăng huyết áp.
  + Heart\_disease (Bệnh tim):
    - Gồm 2 giá trị 0 và 1.
    - Giải thích: Bệnh tim là một tình trạng y tế khác có liên quan đến việc tăng nguy cơ phát triển bệnh tiểu đường. Nó có giá trị 0 hoặc 1 trong đó 0 cho biết họ không mắc bệnh tim và với 1 có nghĩa là họ mắc bệnh tim.
  + Smoking\_history (Lịch sử hút thuốc):
    - Gồm 6 loại: not current, former, No Info, current, never và ever.
    - Giải thích: Tiền sử hút thuốc cũng được coi là yếu tố nguy cơ của bệnh tiểu đường và có thể làm trầm trọng thêm các biến chứng liên quan đến bệnh tiểu đường.
  + BMI (Chỉ số cơ thể BMI):
    - Phạm vi trong tập dữ liệu từ 10.16 đến 71.55 (kg/m2).
    - Giải thích: là thước đo lượng mỡ trong cơ thể dựa trên cân nặng và chiều cao. Giá trị BMI cao hơn có liên quan đến nguy cơ mắc bệnh tiểu đường cao hơn. Phạm vi chỉ số BMI trong tập dữ liệu là từ 10,16 đến 71,55. BMI dưới 18,5 là thiếu cân, 18,5-24,9 là bình thường, 25-29,9 là thừa cân và từ 30 trở lên là béo phì.
  + HbA1c\_Level:
    - Phạm vi trong tập dữ liệu từ 3.5 đến 9 (%).
    - Giải thích: là thước đo lượng đường trong máu trung bình của một người trong 2-3 tháng qua. Mức độ cao hơn cho thấy nguy cơ phát triển bệnh tiểu đường cao hơn. Hầu hết mức HbA1c trên 6,5% cho thấy bệnh tiểu đường.
  + Blood\_glucose\_level:
    - Phạm vi trong tập dữ liệu từ 80 đến 300 (mg/dL).
    - Giải thích: Mức đường huyết đề cập đến lượng glucose trong máu tại một thời điểm nhất định. Mức đường huyết cao là dấu hiệu chính của bệnh tiểu đường.
  + Diabetes (Tiểu đường):
    - Gồm 2 giá trị 0 và 1.
    - Giải thích: Bệnh tiểu đường là biến mục tiêu được dự đoán, với giá trị 1 biểu thị sự hiện diện của bệnh tiểu đường và 0 biểu thị sự vắng mặt của bệnh tiểu đường.

1. **TIỀN XỬ LÝ DỮ LIỆU**
   * + 1. Loại bỏ các bản ghi có dữ liệu bị nhiễu, không nhất quán.

* Loại bỏ các bản ghi bị trùng lặp: loại 3854 bản ghi bị trùng lặp.
* Loại bỏ dữ liệu không nhất quán của thuộc tính gender: thuốc tính gender gồm 2 giá trị male và female nên những bản ghi có trường gender là orther sẽ được loại bỏ = > 18 bản ghi.
* Loại bỏ dữ liệu không nhất quán của thuộc tính age: thuộc tính age được tính theo năm nên những bản ghi có trường age là số thập phân sẽ được loại bỏ = > 2013 bản ghi.
  + - 1. Mã hóa bộ dữ liệu.
* Biến đổi các thuộc tính danh nghĩa ở các trường gender, smoking\_history và diabetes.
  + Gender: female = 0; male = 1.
  + Smoking\_history: current = 0; not current = 1; former = 2; never = 3; ever = 4; no info = 5.
  + Diabetes: 0 = no; 1 = yes.

1. **XÂY DỰNG MÔ HÌNH HỌC MÁY CÓ GIÁM SÁT**
   * + 1. **Naive Bayes**
   1. **Cơ sở lý thuyết**

* Một phân loại Naive Bayes dựa trên ý tưởng nó là một lớp được

dự đoán bằng các giá trị của đặc trưng cho các thành viên của lớp đó.

Các đối tượng là một nhóm (group) trong các lớp nếu chúng có cùng

các đặc trưng chung. Có thể có nhiều lớp rời rạc hoặc lớp nhị phân.

* Các luật Bayes dựa trên xác suất để dự đoán chúng về các lớp có sẵn dựa trên các đặc trưng được trích rút. Trong phân loại Bayes, việc học được coi như xây dựng một mô hình xác suất của các đặc trưng và sử dụng mô hình này để dự đoán phân loại cho một ví dụ mới.
* Biến chưa biết hay còn gọi là biến ẩn là một biến xác suất chưa được quan sát trước đó. Phân loại Bayes sử dụng mô hình xác suất trong đó phân loại là một biến ẩn có liên quan tới các biến đã được quan sát. Quá trình phân loại lúc này trở thành suy diễn trên mô hình xác suất
* Lý do chọn Naïve Bayes
  + Hiệu suất và tốc độ: Naive Bayes là một thuật toán đơn giản và nhanh chóng, phù hợp với các tập dữ liệu lớn và có nhiều biến độc lập. Điều này làm cho nó phù hợp cho việc xử lý các tập dữ liệu y tế lớn.
  + Xử lý dữ liệu dạng văn bản: Trong một số trường hợp, dữ liệu y tế có thể chứa nhiều thông tin dạng văn bản như hồ sơ bệnh án. Naive Bayes là lựa chọn phổ biến cho việc phân loại văn bản.
  + Phân loại đa lớp: Naive Bayes có thể xử lý các bài toán phân loại đa lớp, tức là dự đoán một cá nhân thuộc vào một trong nhiều lớp (ví dụ: mắc bệnh tiểu đường, không mắc bệnh, hoặc rủi ro cao, trung bình, thấp).
  + Xử lý dữ liệu thưa thớt: Với các tập dữ liệu mà có nhiều giá trị bị thiếu hoặc dữ liệu không đầy đủ, Naive Bayes có thể xử lý hiệu quả nhờ vào giả định về sự độc lập giữa các biến.
  + Hiệu quả với dữ liệu không cân bằng: Trong các tập dữ liệu y tế, có thể gặp phải tình trạng mẫu dữ liệu không cân bằng, tức là một lớp có số lượng mẫu ít hơn so với lớp khác. Naive Bayes có thể hoạt động tốt trong điều kiện này.
  1. **Áp dụng cơ sở lý thuyết vào xây dựng mô hình cho bài toán**
* Đọc dữ liệu từ file CSV:

data = pd.read\_csv('diabetes\_prediction\_dataset\_final.csv')

* Tách dữ liệu thành tập train và test:

X = data[['gender', 'age', 'hypertension', 'heart\_disease', 'smoking\_history', 'bmi', 'HbA1c\_level', 'blood\_glucose\_level']]

y = data['diabetes']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

* Huấn luyện mô hình Naïve Bayes:

model = NaiveBayesClassifier(n\_estimators=100, random\_state=42)

model.fit(X\_train, y\_train)

* Đánh giá hiệu suất mô hình trên tập test:

y\_pred = model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print("chinh xac:", accuracy)

* Lưu mô hình:

pickle.dump(model, open('model\_RDF.pkl', 'wb'))

1. **Decision Tree (Cây quyết định)**
   1. **Cơ sở lý thuyết**

* 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. Các thuộc tính của đối tượngncó thể thuộc các kiểu dữ liệu khác nhau như Nhị phân (Binary) , Định danh (Nominal), Thứ tự (Ordinal), Số lượng (Quantitative) trong khi đó thuộc tính phân lớp phải có kiểu dữ liệu là Binary hoặc Ordinal.
* Lý do chọn Decision Tree :
  + Mô hình sinh ra các quy tắc dễ hiểu cho người đọc, tạo ra bộ luật với mỗi nhánh lá là một luật của cây.
  + Dữ liệu đầu vào có thể là là dữ liệu missing, không cần chuẩn hóa hoặc tạo biến giả
  + Có thể làm việc với cả dữ liệu số và dữ liệu phân loại
  + Có thể xác thực mô hình bằng cách sử dụng các kiểm tra thống kê
  + Có khả năng là việc với dữ liệu lớn
  1. **Áp dụng cơ sở lý thuyết vào xây dựng mô hình cho bài toán**
* Đọc dữ liệu từ file CSV:

data = pd.read\_csv('diabetes\_prediction\_dataset\_final.csv')

* Tách dữ liệu thành tập train và test:

X = data[['gender', 'age', 'hypertension', 'heart\_disease', 'smoking\_history', 'bmi', 'HbA1c\_level', 'blood\_glucose\_level']]

y = data['diabetes']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

* Huấn luyện mô hình Decision Tree:

model = DecisionTreeClassifier(n\_estimators=100, random\_state=42)

model.fit(X\_train, y\_train)

* Đánh giá hiệu suất mô hình trên tập test:

y\_pred = model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print("chinh xac:", accuracy)

* Lưu mô hình:

pickle.dump(model, open('model\_RDF.pkl', 'wb'))

1. **Random forests**
   1. **Cơ sở lý thuyết**

* Random forest là một phương pháp thống kê mô hình hóa bằng máy (machine learning statistic) dùng để phục vụ các mục đích phân loại, tính hồi quy và các nhiệm vụ khác bằng cách xây dựng nhiều cây quyết định (Decision tree)
* Thuật toán Random Forest gồm nhiều cây quyết định, mỗi cây quyết định đều có những yếu tố ngẫu nhiên:
  + Lấy ngẫu nhiên dữ liệu để xây dựng cây quyết định.
  + Lấy ngẫu nhiên các thuộc tính để xây dựng cây quyết định.
* Do mỗi cây quyết định trong thuật toán Random Forest không dùng tất cả dữ liệu training, cũng như không dùng tất cả các thuộc tính của dữ liệu để xây dựng cây nên mỗi cây có thể sẽ dự đoán không tốt, khi đó mỗi mô hình cây quyết định không bị overfitting mà có thế bị underfitting, hay nói cách khác là mô hình có độ lệch chuẩn cao. Tuy nhiên, kết quả cuối cùng của thuật toán Random Forest lại tổng hợp từ nhiều cây quyết định, thế nên thông tin từ các cây sẽ bổ sung thông tin cho nhau, dẫn đến mô hình có độ lệch chuẩn thấp và phương sai thấp, hay mô hình có kết quả dự đoán tốt.
* Lý do chọn thuật toàn Random forests
  + Trong thuật toán Decision Tree, khi xây dựng cây quyết định nếu để độ sâu tùy ý thì cây sẽ phân loại đúng hết các dữ liệu trong tập training dẫn đến mô hình có thể dự đoán tệ trên tập validation/test, khi đó mô hình bị overfitting, Bằng cách kết hợp nhiều cây quyết định khác nhau và sử dụng kỹ thuật lấy mẫu ngẫu nhiên, Random Forests giúp giảm thiểu hiện tượng overfitting và tăng tính ổn định của mô hình.
  1. **Áp dụng cơ sở lý thuyết vào xây dựng mô hình cho bài toán**
* Đọc dữ liệu từ file CSV:

data = pd.read\_csv('diabetes\_prediction\_dataset\_final.csv')

* Tách dữ liệu thành tập train và test:

X = data[['gender', 'age', 'hypertension', 'heart\_disease', 'smoking\_history', 'bmi', 'HbA1c\_level', 'blood\_glucose\_level']]

y = data['diabetes']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

* Huấn luyện mô hình Random Forest:

model = RandomForestClassifier(n\_estimators=100, random\_state=42)

model.fit(X\_train, y\_train)

* Đánh giá hiệu suất mô hình trên tập test:

y\_pred = model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print("chinh xac:", accuracy)

* Lưu mô hình:

pickle.dump(model, open('model\_RDF.pkl', 'wb'))

1. **ĐÁNH GIÁ MÔ HÌNH HỌC MÁY**

Đánh giá mô hình thông qua ma trận nhầm lần (Confusing Matrix).

* Ma trận nhầm lẫn (confusion matrix) là công cụ trực quan và hiệu quả để đánh giá hiệu suất của mô hình phân loại. Nó thể hiện rõ ràng mức độ chính xác của mô hình trong việc dự đoán các lớp (classes) khác nhau.
* Phương pháp đánh giá:
  + - Tính toán các thước đo
      * Accuracy: tỉ lệ giữa số điểm dữ liệu được dự đoán đúng và tổng số điểm dữ liệu.
      * Precision: tỉ lệ số điểm true positive trong số những điểm được phân loại là positive
      * Recall: tỉ lệ số điểm true positive trong số những điểm thực sự là positive
      * F1 Score: Điểm cân bằng giữa precision và recall.
    - Phân tích các thước đo trong ma trận: Một mô hình tốt khi cả Precision và Recall đều cao có nghĩa là ít phân loại nhầm và tỷ lệ bỏ sót thấp
* Công thức tính:
  + - Accuracy= (TP+TN)/(TP+TN+FP+FN)
    - Precision= TP/(TP+ FP)
    - Recall= TP/(TP+ FN)
    - Accuracy=2\* (Precision\*Recall )/(Precision+Recall)

Trong đó:

TP( True Positive): Số lượng các trường hợp mà mô hình dự đoán đúng là positive (dương tính) và thực tế cũng là positive.

TN( True Negative): Số lượng các trường hợp mà mô hình dự đoán đúng là negative và thực tế cũng là negative.

FP( False Positive): Số lượng các trường hợp mà mô hình dự đoán là positive nhưng thực tế lại là negative (âm tính).

FN( False Negative): Số lượng các trường hợp mà mô hình dự đoán là negative nhưng thực tế lại là positive.

* Đánh giá mô hình NaïveBayes:

A screenshot of a computer

Description automatically generated

* + Nhận xét:
    - Mô hình có Accuracy cao (94,18%)
    - Mô hình có Precision cao (94,1%)
    - Mô hình có Recall rất cao (94,2%)
    - Mô hình có: F1 Score cao (94,1%)
      * Mô hình có hiệu suất tốt
* Đánh giá mô hình Decision Trees:

A screenshot of a computer

Description automatically generated

* + Nhận xét:
    - Mô hình có Accuracy cao (94,89%)
    - Mô hình có Precision cao (95%)
    - Mô hình có Recall cao (94,9%)
    - Mô hình có: F1 Score cao (94,9%)
      * Mô hình có hiệu suất rất tốt
* Đánh giá mô hình Random forests

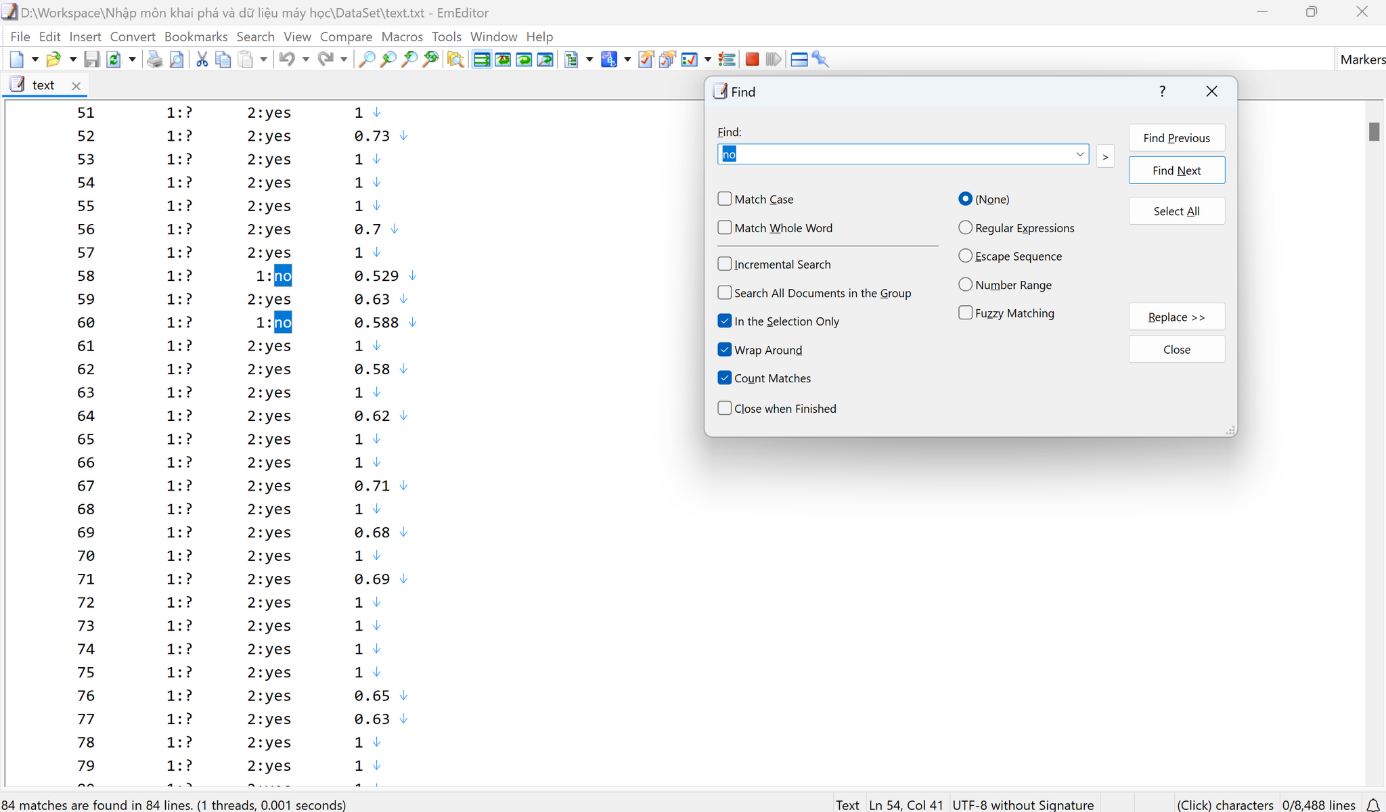
A white screen with black text

Description automatically generated

* Nhận xét:
  + - Mô hình có Accuracy cao (96,86%)
    - Mô hình có Precision cao (96,8%)
    - Mô hình có Recall cao (96,9%)
    - Mô hình có: F1 Score cao (96,6%)
* Mô hình có hiệu suất rất tốt
* So sánh 3 mô hình

|  |  |  |  |
| --- | --- | --- | --- |
|  | Decision Tree | Random forests | Naïve Bayes |
| Accuracy | 94,89% | 96,86% | 94,18% |
| Precision | 95% | 96,8% | 94,1% |
| Recall | 94,9% | 96,9% | 94,2% |
| F1 – Score | 94,9% | 96,6% | 94,1% |

* + - Nhận xét:
* Cả 3 mô hình đều hiệu suất tốt, tuy nhiên mô hình của thuật toán Random Forests có bốn chỉ số cao hơn so với mô hình của 2 thuật toàn Decision Tree và Naïve Bayes
  + - * Mô hình Random forests có hiệu suất tốt hơn cho bài toán đặt ra
* Đánh giá mô hình Random Forest sau khi kiểm thử với tập dữ liệu testing gồm 8488 bản ghi chỉ toàn positive:



* + - * Có 84 bản ghi bị dự đoán sai thành negative trên tổng 8488 bản ghi suy ra độ chính xác của mô hình sau kiểm thử là: 99,01%

1. **TÍCH HỢP MÔ HÌNH HỌC MÁY VÀO ỨNG DỤNG**

* Thuật toán

# Import thư viện cần thiết

from flask import Flask, render\_template, request

import pickle

import numpy as np

from sklearn.ensemble import RandomForestClassifier

# Load model Random Forest

filename = 'model\_RDF.pkl'

classifier = pickle.load(open(filename, 'rb'))

# Chạy ứng dụng web

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

return render\_template('index.html')

@app.route('/predict', methods=['POST'])

def predict():

if request.method == 'POST':

gender = int(request.form['gender'])

age = int(request.form['age'])

hypt = int(request.form['hypertension'])

hd = int(request.form['heart\_disease'])

smk = int(request.form['smoking\_history'])

bmi = float(request.form['bmi'])

hba1c = float(request.form['hba1c\_level'])

bgl = int(request.form['blood\_glucose\_level'])

data = np.array([[gender, age, hypt, hd, smk, bmi, hba1c, bgl]])

my\_prediction = classifier.predict(data)

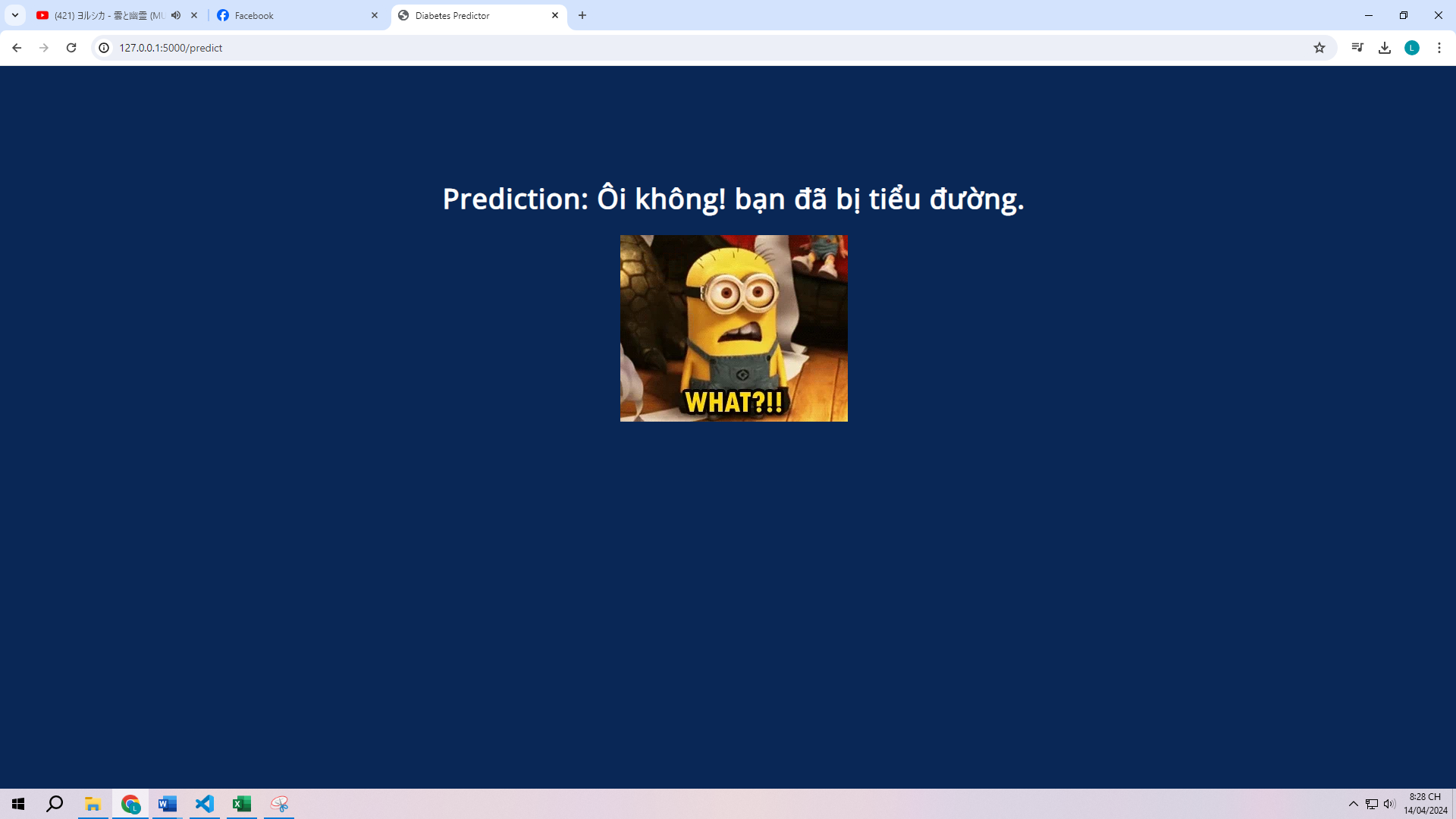
return render\_template('result.html', prediction=my\_prediction)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

* Kết quả





1. **KẾT LUẬN**

Trong quá trình làm bài tập lớn, nhóm đã thực hiện các công việc sau:

* Nghiên cứu về các phương pháp học máy có giám sát.
* Áp dụng các giai đoạn của chu trình học máy vào bài toán thực tế
* Thực hiện đánh giá, so sánh 3 mô hình học để lựa chọn mô hình học máy tốt nhất
* Tích hợp được mô hình học máy vào ứng dụng để giải quyết các vấn đề trong lĩnh vực nghiên cứu y tế.

**TÀI LIỆU THAM KHẢO**

[1] Bài giảng môn Nhập môn Khai phá dữ liệu và Máy học, TS Vũ Xuân Hạnh, Khoa Công nghệ thông tin, Trường Đại Học Mở Hà Nội. (Tham khảo lý thuyết cơ sở)

[2] Vũ Hữu Tiệp – Machine learning cơ bản. (Tham khảo lý thuyết cơ sở)

[3] Phạm Minh Hải, Nguyễn Ngọc Quang – Khái niệm về phương pháp Random Forests trong cuộc cách mạng machine learning. (Tham khảo lý thuyết cơ sở)

[4] Hoàng Anh Công – Nghiên cứu lý thuyết Naïve Bayes. (Tham khảo lý thuyết cơ sở)

[5] Tuấn Nguyễn - Decision Tree algorithm. (Tham khảo lý thuyết cơ sở)

[6] End to end Prediction application using machine learning - https://github.com/ditikrushna/End-to-End-Diabetes-Prediction-Application-Using-Machine-Learning. (VI. Tích hợp mô hình vào ứng dụng)