

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA ĐA PHƯƠNG TIỆN



BÁO CÁO
KHAI PHÁ DỮ LIỆU ĐA PHƯƠNG TIỆN

ĐỀ TÀI: XÂY DỰNG WEBSITE DỰ ĐOÁN BỆNH TIỂU ĐƯỜNG

Giảng viên hướng dẫn: TS. Đỗ Thị Liên

Sinh viên thực hiện

Mã sinh viên

1. Đỗ Văn Tuấn

B19DCPT209

2. Nguyễn Hữu Quang

B19DCPT184

3. Mai Quốc Cường

B19DCPT024

4. Trần Quang Huy

B19DCPT109

5. Bùi Thị Mai

B19DCPT154

Hà Nội – 2023

Mục lục

| | |
|--|----|
| 1. Phát biểu bài toán | 3 |
| 2. Kiến trúc tổng quát của hệ thống (Front-end, Back-end) | 4 |
| 2.1. Back-end. | 4 |
| 2.1.1. Miêu tả dữ liệu về dataset. | 4 |
| 2.1.2. Quá trình xử lý Back-end | 5 |
| 2.1.3. Mô hình khai phá dữ liệu | 9 |
| 2.1.4. Thực nghiệm và đánh giá mô hình | 11 |
| 2.1.5. Cài đặt và triển khai Back-end. | 12 |
| 2.2. Front-end | 13 |
| 2.2.1. Phương pháp tiếp cận và giải quyết vấn đề cho ứng dụng Front-end. | 13 |
| 2.2.2. Công nghệ giải quyết phía Front-end. | 14 |
| 2.2.3. Phân tích thiết kế. | 14 |
| 2.2.4. Cài đặt và triển khai Front-end. | 17 |

PHÂN CHIA CÔNG VIỆC

| Sinh viên | Công việc |
|------------------|-------------------------------------|
| Nguyễn Hữu Quang | Train data, model, dataset |
| Trần Quang Huy | Train data, model, dataset |
| Mai Quốc Cường | Frontend, Backend, hệ thống website |
| Đỗ Văn Tuấn | Frontend, Backend, hệ thống website |
| Bùi Thị Mai | Frontend, Backend, hệ thống website |

1. Phát biểu bài toán

Bài toán phát hiện tiểu đường là một bài toán quan trọng trong lĩnh vực y tế, nhằm dự đoán khả năng mắc bệnh tiểu đường của một cá nhân dựa trên các đặc trưng y tế và yếu tố rủi ro. Mục tiêu của bài toán là xây dựng một mô hình hoặc hệ thống có khả năng phân loại các cá nhân thành hai nhóm: bệnh nhân mắc tiểu đường và bệnh nhân không mắc tiểu đường.

Các dấu hiệu của bệnh tiểu đường bao gồm:

- Cảm thấy khát và uống nước nhiều hơn thường lệ.
- Thường xuyên đi tiểu, đặc biệt vào ban đêm.
- Mất cân nặng một cách không rõ ràng.
- Mệt mỏi, mất năng lượng.
- Da khô, ngứa, nổi mẩn hoặc nhiễm trùng da thường xuyên. Thèm ăn nhiều, đặc biệt là thèm đồ ngọt.
- Thường xuyên bị nhiễm khuẩn, viêm nhiễm và thương lành chậm.

Bệnh tiểu đường có thể gây ra các biến chứng nguy hiểm nếu không được kiểm soát tốt bao gồm:

- Biến chứng tim mạch: Tiểu đường có thể gây ra tăng nguy cơ mắc các bệnh tim mạch như đau tim, nhồi máu cơ tim và đột quỵ.
- Suy thận: Bệnh tiểu đường có thể làm suy giảm chức năng thận và dẫn đến suy thận.
- Mất thị lực: Tiểu đường có thể gây ra các vấn đề mắt như đục thủy tinh thể, viêm võng mạc và đục thể thực quản, có thể dẫn đến mất thị lực.
- Biến chứng dây thần kinh: Tiểu đường có thể gây ra tổn thương dây thần kinh, dẫn đến nhức đầu, đau mỏi, tê liệt và khó khăn trong việc đi lại.
- Nhiễm trùng: Người mắc tiểu đường có nguy cơ cao hơn mắc các nhiễm trùng da, nhiễm trùng đường tiểu và nhiễm trùng khác.

Tình trạng hiện nay: Tiểu đường đã trở thành một vấn đề sức khỏe đáng lo ngại trên toàn thế giới. Theo Tổ chức Y tế Thế giới (WHO), số người mắc tiểu đường đã tăng gấp đôi từ năm 1980 và đạt khoảng 422 triệu người vào năm 2014. Các yếu tố nguy cơ gồm di truyền, lối sống không lành mạnh, béo phì, thiếu hoạt động thể chất và tuổi tác đều đóng vai trò trong sự gia tăng này. Mặc dù tiểu đường có thể kiểm soát được, nhưng nếu không được phát hiện và quản lý tốt, nó có thể gây ra các biến chứng nghiêm trọng như bệnh tim, đột quỵ, thậm chí mất thị lực và suy thận.

Để giải quyết bài toán này, quá trình khai phá dữ liệu thường bao gồm các bước sau:

- Thu thập dữ liệu: Dữ liệu y tế liên quan đến tiểu đường được thu thập từ bệnh viện, phòng khám hoặc các nguồn dữ liệu y tế khác. Dữ liệu này bao gồm các chỉ số sinh lý, kết quả xét nghiệm, thông tin về thói quen ăn uống và lối sống của bệnh nhân.

- Tiền xử lý dữ liệu: Dữ liệu thu thập được thường không hoàn hảo, chứa nhiều và thiếu sót. Quá trình tiền xử lý dữ liệu bao gồm loại bỏ dữ liệu nhiễu, xử lý các giá trị thiếu, chuẩn hóa dữ liệu và chuyển đổi dữ liệu thành định dạng phù hợp cho việc phân tích.
- Xác định các đặc trưng: Các đặc trưng quan trọng để phát hiện tiểu đường có thể bao gồm tuổi, giới tính, chỉ số khối cơ thể (BMI), mức đường huyết, huyết áp, lịch sử gia đình và các chỉ số xét nghiệm khác. Qua việc phân tích dữ liệu, các đặc trưng quan trọng sẽ được xác định.
- Xây dựng mô hình: Các thuật toán khai phá dữ liệu như học máy và học sâu có thể được áp dụng để xây dựng mô hình dự đoán tiểu đường. Các mô hình này sẽ học từ dữ liệu đã được gán nhãn để tìm ra các quy luật, mẫu và mối quan hệ trong dữ liệu để phân loại các trường hợp tiểu đường và không tiểu đường.
- Đánh giá và tinh chỉnh mô hình: Mô hình được xây dựng sẽ được đánh giá bằng các phương pháp đánh giá hiệu suất như cross-validation hoặc chia dữ liệu thành tập huấn luyện và tập kiểm tra. Nếu mô hình không đạt hiệu suất mong đợi, ta có thể tinh chỉnh các tham số, chọn lại đặc trưng hoặc sử dụng các phương pháp tăng cường dữ liệu để cải thiện kết quả.
- Áp dụng mô hình: Sau khi mô hình đã được đánh giá và tinh chỉnh, nó có thể được áp dụng vào dữ liệu mới để dự đoán xác suất mắc bệnh tiểu đường của một cá nhân dựa trên các đặc trưng y tế của họ.

Tuy bài toán phát hiện tiểu đường có thể trở nên phức tạp và đòi hỏi sự cẩn thận trong quá trình xử lý dữ liệu và xây dựng mô hình, nhưng nó có thể cung cấp thông tin quan trọng và hỗ trợ trong việc chẩn đoán và điều trị bệnh tiểu đường.

2. Kiến trúc tổng quát của hệ thống (Front-end, Back-end)

- Kiến trúc: Client - Server

Client: là những thứ bạn nhìn thấy trên browser (trình duyệt web)

Server: là những gì xử lý ở phía sau mà người dùng bình thường không thấy được

- Cách truyền dữ liệu: sử dụng phương thức HTTP request

Server nhận các request được gửi từ client rồi xử lý dữ liệu và trả lại, hiển thị kết quả lên client đúng theo yêu cầu của người dùng.

2.1. Back-end.

2.1.1. Miêu tả dữ liệu về dataset.

Trong lĩnh vực back-end, dataset thường ám chỉ tập hợp các dữ liệu được lưu trữ và quản lý bởi hệ thống back-end. Đây là nơi mà dữ liệu được xử lý, lưu trữ và truy xuất từ cơ sở dữ liệu.

Dữ liệu trong tập dataset cho bài toán này bao gồm 768 mẫu và có 9 đặc trưng (features) tương ứng cho mỗi bệnh nhân. Tập dữ liệu này bao gồm các trường thông tin sau:

- Tuổi (Age): Đây là đặc trưng liên tục, biểu thị tuổi của bệnh nhân.
- Tình trạng mang thai (Pregnancies): Đây là đặc trưng rời rạc, biểu thị số lần mang thai của bệnh nhân.
- Mức đường huyết dùng glucose (Glucose): Đây là đặc trưng liên tục, biểu thị mức đường huyết sau khi bệnh nhân uống dung dịch glucose và được đo sau 2 giờ.
- Huyết áp (Blood Pressure): Đây là đặc trưng liên tục, biểu thị huyết áp của bệnh nhân.
- Độ dày da (Skin Thickness): Đây là đặc trưng liên tục, biểu thị độ dày của da gấp sau chỏm triceps của bệnh nhân.
- Insulin: Đây là đặc trưng liên tục, biểu thị mức độ insulin huyết thanh của bệnh nhân.
- Chỉ số khối cơ thể (BMI): Đây là đặc trưng liên tục, biểu thị chỉ số khối cơ thể của bệnh nhân.
- Lịch sử gia đình (Diabetes Pedigree Function): Đây là đặc trưng liên tục, biểu thị một chức năng đo lường xác suất tiểu đường dựa trên lịch sử gia đình của bệnh nhân.
- Nhãn (Outcome): Đây là nhãn nhị phân (0 hoặc 1), biểu thị xác suất bệnh nhân có tiểu đường (1) hoặc không (0).

Mỗi mẫu trong tập dữ liệu sẽ có các giá trị của các đặc trưng trên và nhãn tương ứng để xác định xem một người có tiểu đường hay không.

2.1.2. Quá trình xử lý Back-end

- Tiền xử lý dữ liệu, tách các tập dữ liệu
- Tiền xử lý dữ liệu: thêm các dữ liệu bị trống
Xử lý trong phần mềm Weka

Bước 1: dùng filter NumericCleaner để tìm ra các dữ liệu bị thiếu thành NaN và đánh dấu



weka.gui.GenericObjectEditor



weka.filters.unsupervised.attribute.NumericCleaner

About

A filter that 'cleanses' the numeric data from values that are too small, too big or very close to a certain value, and sets these values to a pre-defined default.

[More](#)[Capabilities](#)

attributeIndices 2-8

closeTo 0.0

closeToDefault 0.0

closeToTolerance 1.0E-6

debug False

decimals -1

doNotCheckCapabilities False

includeClass False

invertSelection False

maxDefault 1.7976931348623157E308

maxThreshold 1.7976931348623157E308

minDefault NaN

minThreshold -1.7976931348623157E308

Open...

Save...

OK

Cancel

Name: Glucose
Missing: 5 (1%)

Distinct: 135

Type: Numeric
Unique: 19 (2%)

Name: BloodPressure
Missing: 35 (5%)

Distinct: 46

Type: Numeric
Unique: 8 (1%)

Name: SkinThickness
Missing: 221 (29%)

Distinct: 52

Type: Numeric
Unique: 5 (1%)

| | | |
|-------------------------------------|---------------|-----------------------------------|
| Name: Insulin Missing: 372 (48%) | Distinct: 186 | Type: Numeric Unique: 93 (12%) |
| Name: BMI Missing: 11 (1%) | Distinct: 63 | Type: Numeric Unique: 28 (4%) |

Bước 2: dùng filter RemoveWithValues để xóa các dữ liệu NaN đã được đánh dấu đó

weka.gui.GenericObjectEditor

weka.filters.unsupervised.instance.RemoveWithValues

About

Filters instances according to the value of an attribute. [More](#) [Capabilities](#)

attributeIndex: 3

debug: False

doNotCheckCapabilities: False

dontFilterAfterFirstBatch: False

invertSelection: False

matchMissingValues: True

modifyHeader: False

nominalIndices: first-last

splitPoint: 0.0

Open... Save... OK Cancel

| | | |
|--|---------------|-----------------------------------|
| Name: Glucose Missing: 0 (0%) | Distinct: 135 | Type: Numeric Unique: 19 (2%) |
| Name: BloodPressure Missing: 0 (0%) | Distinct: 46 | Type: Numeric Unique: 8 (1%) |
| Name: SkinThickness Missing: 0 (0%) | Distinct: 52 | Type: Numeric Unique: 5 (1%) |
| Name: Insulin Missing: 0 (0%) | Distinct: 186 | Type: Numeric Unique: 95 (24%) |

| | | |
|-----------------|--------------|-----------------|
| Name: BMI | Distinct: 51 | Type: Numeric |
| Missing: 0 (0%) | | Unique: 22 (6%) |

Bước 3: dùng filter ReplaceMissingValues để thêm các dữ liệu trung bình vào các đoạn dữ liệu bị thiếu đã được đánh dấu

| | | | | |
|------------------|--|--------------------------------------|--|---------------------------------|
| Filter | Choose | ReplaceMissingValues | Apply | Stop |
| Current relation | Relation: diabetes-weka.filters.unsupervised.instance.RemoveWithValues-S0.0-C2-Lfirst-last-M-weka.filters.unsupervised | Attributes: 9 Sum of weights: 392 | Selected attribute Name: Pregnancies Missing: 0 (0%) | Type: Numeric Unique: 3 (1%) |
| Instances: 392 | | | Distinct: 17 | |

Bước 4: lưu lại dataset đã tiền xử lý

- Tách các tập dữ liệu: xử lý bằng python

```
data = pd.read_csv(r"C:\Users\ADMIN\Desktop\KPDL\BTL\diabetes.csv")
train_data, test_data = train_test_split(data, test_size = 0.2)
train_data.to_csv('train.csv', index=False)
test_data.to_csv('test.csv', index=False)
unlabel = test_data.drop("Outcome", axis = 1)
unlabel_data_train, unlabel_data_test = train_test_split(unlabel, test_size = 0.15)
unlabel_data_test.to_csv('unlabel.csv', index=False)
X = data.drop("Outcome", axis = 1)
Y = data['Outcome']
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2)
```
- Dữ liệu sau khi được đọc sẽ được tách thành 2 phần để thực hiện train và test. Tập train data chiếm 80% kích thước so với tập dataset ban đầu, tập test data chiếm 20% kích thước tập dataset ban đầu. Tập unlabel được lấy từ tập test data và bỏ đi nhãn "Out come"
- Train Dataset:

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age | Outcome |
|-----|-------------|---------|---------------|---------------|---------|------|--------------------------|-----|---------|
| 186 | 8 | 181 | 68 | 36 | 495 | 30.1 | 0.615 | 60 | 1 |
| 338 | 9 | 152 | 78 | 34 | 171 | 34.2 | 0.893 | 33 | 1 |
| 438 | 1 | 97 | 70 | 15 | 0 | 18.2 | 0.147 | 21 | 0 |
| 500 | 2 | 117 | 90 | 19 | 71 | 25.2 | 0.313 | 21 | 0 |
| 183 | 5 | 73 | 60 | 0 | 0 | 26.8 | 0.268 | 27 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 341 | 1 | 95 | 74 | 21 | 73 | 25.9 | 0.673 | 36 | 0 |
| 638 | 7 | 97 | 76 | 32 | 91 | 40.9 | 0.871 | 32 | 1 |
| 472 | 0 | 119 | 66 | 27 | 0 | 38.8 | 0.259 | 22 | 0 |
| 641 | 4 | 128 | 70 | 0 | 0 | 34.3 | 0.303 | 24 | 0 |
| 102 | 0 | 125 | 96 | 0 | 0 | 22.5 | 0.262 | 21 | 0 |

614 rows × 9 columns

- Test Dataset:

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age | Outcome |
|-----|-------------|---------|---------------|---------------|---------|------|--------------------------|-----|---------|
| 695 | 7 | 142 | 90 | 24 | 480 | 30.4 | 0.128 | 43 | 1 |
| 568 | 4 | 154 | 72 | 29 | 126 | 31.3 | 0.338 | 37 | 0 |
| 218 | 5 | 85 | 74 | 22 | 0 | 29.0 | 1.224 | 32 | 1 |
| 401 | 6 | 137 | 61 | 0 | 0 | 24.2 | 0.151 | 55 | 0 |
| 608 | 0 | 152 | 82 | 39 | 272 | 41.5 | 0.270 | 27 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 531 | 0 | 107 | 76 | 0 | 0 | 45.3 | 0.686 | 24 | 0 |
| 617 | 2 | 68 | 62 | 13 | 15 | 20.1 | 0.257 | 23 | 0 |
| 96 | 2 | 92 | 62 | 28 | 0 | 31.6 | 0.130 | 24 | 0 |
| 68 | 1 | 95 | 66 | 13 | 38 | 19.6 | 0.334 | 25 | 0 |
| 706 | 10 | 115 | 0 | 0 | 0 | 0.0 | 0.261 | 30 | 1 |

154 rows × 9 columns

- Unlabel Dataset:

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age |
|-----|-------------|---------|---------------|---------------|---------|------|--------------------------|-----|
| 449 | 0 | 120 | 74 | 18 | 63 | 30.5 | 0.285 | 26 |
| 305 | 2 | 120 | 76 | 37 | 105 | 39.7 | 0.215 | 29 |
| 386 | 5 | 116 | 74 | 29 | 0 | 32.3 | 0.660 | 35 |
| 94 | 2 | 142 | 82 | 18 | 64 | 24.7 | 0.761 | 21 |
| 196 | 1 | 105 | 58 | 0 | 0 | 24.3 | 0.187 | 21 |
| 374 | 2 | 122 | 52 | 43 | 158 | 36.2 | 0.816 | 28 |
| 228 | 4 | 197 | 70 | 39 | 744 | 36.7 | 2.329 | 31 |
| 376 | 0 | 98 | 82 | 15 | 84 | 25.2 | 0.299 | 22 |
| 474 | 4 | 114 | 64 | 0 | 0 | 28.9 | 0.126 | 24 |
| 755 | 1 | 128 | 88 | 39 | 110 | 36.5 | 1.057 | 37 |
| 115 | 4 | 146 | 92 | 0 | 0 | 31.2 | 0.539 | 61 |
| 304 | 3 | 150 | 76 | 0 | 0 | 21.0 | 0.207 | 37 |

2.1.3. Mô hình khai phá dữ liệu

a. Mô hình Logistic Regression

Hồi quy logistic là một trong những thuật toán Machine Learning phổ biến nhất, thuộc kỹ thuật Supervised Learning. Nó được sử dụng để dự đoán biến phụ thuộc phân loại bằng cách sử dụng một tập hợp các biến độc lập nhất định.

Hồi quy logistic dự đoán đầu ra của một biến phụ thuộc phân loại. Do đó, kết quả phải là một giá trị phân loại hoặc rời rạc. Nó có thể là Có hoặc Không, 0 hoặc 1, đúng hoặc Sai, v.v. nhưng thay vì đưa ra giá trị chính xác là 0 và 1, nó đưa ra các giá trị xác suất nằm trong khoảng từ 0 đến 1.

- Hàm Logistic (Sigmoid function): Hồi quy logistic sử dụng hàm sigmoid để ánh xạ đầu ra dự đoán vào khoảng [0, 1]. Hàm sigmoid có công thức:

$$\sigma(z) = 1 / (1 + \exp(-z)),$$

Trong đó z là tổ hợp tuyến tính của các biến đầu vào và các tham số của mô hình.

- Phương trình hồi quy Logistic: Phương trình hồi quy Logistic có thể được lấy từ phương trình hồi quy tuyến tính. Các bước toán học để có được phương trình hồi quy logistic được đưa ra dưới đây:

- Phương trình đường thẳng:

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \dots + b_nx_n$$

- Trong hồi quy logistic, y chỉ có thể nằm trong khoảng từ 0 đến 1, vì vậy, hãy chia phương trình trên cho $(1-y)$:

$$f(x) = y/(1-y)$$

Trong đó: $f(x)=0$ nếu $y=0$ và không xác định nếu $y=1$

- Phương trình cuối cùng của hồi quy Logistic:

$$\log\left(\frac{y}{1-y}\right) = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \dots + b_nx_n$$

- Loại hồi quy logistic: Trên cơ sở các loại, Hồi quy logistic có thể được phân thành ba loại:

- + Nhị thức: Trong hồi quy Logistic nhị thức, chỉ có thể có hai loại biến phụ thuộc, chẳng hạn như 0 hoặc 1, Đạt hoặc Không đạt, v.v.
- + Đa thức: Trong hồi quy Logistic đa thức, có thể có 3 hoặc nhiều loại biến phụ thuộc không có thứ tự, chẳng hạn như "mèo", "chó" hoặc "cừu"
- + Thứ tự: Trong hồi quy Logistic thứ tự, có thể có 3 hoặc nhiều loại biến phụ thuộc có thể được sắp xếp theo thứ tự, chẳng hạn như "thấp", "Trung bình" hoặc "Cao"

b. Mô hình KNN

KNN (K-Nearest Neighbors) là một trong những thuật toán học có giám sát đơn giản nhất được sử dụng nhiều trong khai phá dữ liệu và học máy. Ý tưởng của thuật toán này là nó không học một điều gì từ tập dữ liệu học (nên KNN được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán nhãn của dữ liệu mới.

- Nguyên tắc hoạt động:

- KNN xác định lớp của một mẫu dữ liệu mới dựa trên K láng giềng gần nhất trong tập dữ liệu huấn luyện. K là một số nguyên dương được xác định trước (thường là số lẻ để tránh trường hợp số phiếu bầu bằng nhau).
- Để xác định K láng giềng gần nhất, KNN tính toán khoảng cách giữa mẫu dữ liệu mới và các mẫu dữ liệu trong tập huấn luyện, sau đó chọn K láng giềng gần nhất dựa trên khoảng cách này.

- Khoảng cách:

- KNN sử dụng một hàm khoảng cách để đo lường sự tương đồng hoặc sự khác biệt giữa các mẫu dữ liệu. Hàm khoảng cách phổ biến là khoảng cách Euclidean, nhưng cũng có thể sử dụng các hàm khác như khoảng cách Manhattan, khoảng cách cosine, v.v.
- Quyết định và đa số láng giềng:
 - Khi đã xác định K láng giềng gần nhất, KNN sử dụng phương pháp đa số đơn giản để quyết định lớp của mẫu dữ liệu mới. Điều này có nghĩa là lớp được đưa ra là lớp mà có số láng giềng nhiều nhất trong K láng giềng gần nhất. Nếu K là số lẻ, việc đa số sẽ luôn có kết quả rõ ràng.
- Điều chỉnh tham số:
 - Trong KNN, số láng giềng K là một tham số quan trọng. Lựa chọn K phù hợp có thể ảnh hưởng đến hiệu suất của mô hình. Một K quá nhỏ có thể dẫn đến hiện tượng overfitting, trong khi một K quá lớn có thể dẫn đến hiện tượng overfitting. Thông thường, việc chọn K được thực hiện thông qua kỹ thuật cross-validation.
- Đặc điểm:
 - KNN không yêu cầu giả định về phân phối của dữ liệu và có khả năng xử lý các bài toán phân loại phi tuyến.
 - Tuy nhiên, KNN có độ phức tạp tính toán cao khi kích thước của tập dữ liệu huấn luyện lớn, do phải tính toán khoảng cách với tất cả các mẫu dữ liệu.
 - Việc chuẩn bị dữ liệu trước cho KNN cũng là một yếu tố quan trọng, bao gồm việc chuẩn hóa dữ liệu và xử lý các giá trị thiếu.

2.1.4. Thực nghiệm và đánh giá mô hình

So sánh, đánh giá độ chính xác khi thực hiện huấn luyện 2 mô hình KNN và hồi quy Logistic

- Huấn luyện mô hình KNN:

```
knnClassifier = KNeighborsClassifier(n_neighbors = 18)
knnClassifier.fit(X_train, Y_train)
```

- Huấn luyện mô hình Logistic Regression:

```
model = LogisticRegression(solver='lbfgs', max_iter=1000)
model.fit(X_train, Y_train)
```

- Đánh giá mô hình: Sử dụng tập dữ liệu kiểm tra, đánh giá hiệu suất của mô hình KPDL bằng các chỉ số đánh giá phù hợp với bài toán như độ chính xác (accuracy), độ tương đồng (similarity), hoặc độ đo F1 (F1 score).

KNN:

```
print('Accuracy score of the training data by KNN : ',round(roc_auc_score(Y_test,y_pred),9))
```

```
Accuracy score of the training data by KNN : 0.711970571
```

Logistic Regression:

```
print('Accuracy score of the training data by Logistic : ',accuracy)
```

Accuracy score of the training data by Logistic : 0.7402597402597403

- Đánh giá kết quả: Dựa trên kết quả đánh giá, đưa ra đánh giá về khả năng của mô hình KPDL trong giải quyết bài toán cụ thể. Xem xét hiệu suất, độ chính xác, độ tin cậy và các yếu tố khác để xác định xem mô hình KPDL có phù hợp với bài toán hay không. Dựa trên accuracy của mô hình KNN và Logistic Regression thì thấy sự chính xác của mô hình Logistic Regression cao hơn nên lựa chọn mô hình này để huấn luyện model.

Quá trình thực nghiệm và đánh giá sẽ cung cấp cho bạn thông tin quan trọng về khả năng và hiệu suất của mô hình KPDL trong giải quyết bài toán cụ thể. Từ đó, có thể đưa ra quyết định xem liệu mô hình KPDL có phù hợp và mang lại lợi ích trong bài toán hay không.





2.1.5. Cài đặt và triển khai Back-end.

Cài đặt và triển khai back-end là quá trình đưa hệ thống back-end của ứng dụng từ môi trường phát triển sang môi trường sản phẩm hoạt động thực tế. Dưới đây là các bước cơ bản để cài đặt back-end:

- Xác định môi trường triển khai: Đầu tiên, xác định môi trường triển khai back-end của bạn, bao gồm hệ điều hành, máy chủ và dịch vụ cần thiết. Sử dụng Framework Django của Python để xây dựng website dự đoán và Jupyter để train dataset tạo model dự đoán
- Tạo cấu hình môi trường triển khai: Cấu hình các biến môi trường, cổng kết nối và các tham số khác cần thiết cho hệ thống back-end. Đảm bảo rằng môi trường triển khai được cấu hình chính xác và bảo mật.
- Cài đặt các phụ thuộc: Cài đặt các thư viện, gói phần mềm hoặc các phụ thuộc khác cần thiết cho ứng dụng back-end. Sử dụng trình quản lý gói, các thư viện cần thiết (ví dụ: npm cho Node.js, pip cho Python) để cài đặt các phụ thuộc một cách tự động.
- Kiểm tra hệ thống: Thực hiện các bước kiểm tra và kiểm tra hệ thống back-end để đảm bảo rằng nó hoạt động một cách chính xác trong môi trường triển khai.
- Theo dõi và nâng cấp: Theo dõi hoạt động của hệ thống back-end, phát hiện lỗi và nâng cấp hệ thống khi cần thiết. Thực hiện các bản cập nhật, bản vá lỗi và cải tiến để duy trì và cải thiện hệ thống theo thời gian.

Triển khai Back-end:

- Khai phá dữ liệu, training model và dựa vào model để dự đoán kết quả bệnh

 diabetes.csv
 KPDL.ipynb
 README.md
 train_model.sav

- Back-end website nhận dữ liệu nhập vào từ client, gọi model được huấn luyện và đưa ra kết quả dự đoán

```

val1 = float(request.GET['n1'])
val2 = float(request.GET['n2'])
val3 = float(request.GET['n3'])
val4 = float(request.GET['n4'])
val5 = float(request.GET['n5'])
val6 = float(request.GET['n6'])
val7 = float(request.GET['n7'])
val8 = float(request.GET['n8'])

loaded_model = pickle.load(open('C:/Users/ADMIN/Desktop/KPDL/BTL/train_model.sav', 'rb'))

input_data = (val1, val2, val3, val4, val5, val6, val7, val8)

# changing the input_data to numpy array
input_data_as_numpy_array = np.asarray(input_data)

# reshape the array as we are predicting for one instance
input_data_reshaped = input_data_as_numpy_array.reshape(1, -1)

# pred = loaded_model.predict([[val1, val2, val3, val4, val5, val6, val7, val8]])
pred = loaded_model.predict(input_data_reshaped)

result2 = ""
if pred == [1] and val2 > 130:
    result2 = "Tiểu đường"
    advice = "Ăn uống theo chế độ, giảm tinh bột, ưu tiên ăn chay, bớt ăn đồ ngọt"
elif pred == [1] and val2 >= 110 and val2 <= 130:
    result2 = "Tiền tiểu đường"
    advice = "Cần ăn uống điều độ, giảm tinh bột, ăn nhiều rau củ ít tinh bột"
else:
    result2 = "Bình thường"
    advice = "Không bị bệnh, nên giữ chế độ ăn uống hiện tại và healthy hơn nếu có thể"

return render(request, 'predict.html', {"result2": result2, "advice": advice})

```

Quá trình cài đặt và triển khai back-end đòi hỏi sự cẩn thận và kiến thức về hệ thống và công nghệ sử dụng. Đảm bảo rằng bạn đã thực hiện các bước trên một cách cẩn thận và theo đúng quy trình để đảm bảo rằng hệ thống back-end của bạn hoạt động ổn định và hiệu quả.

2.2.Front-end

2.2.1. Phương pháp tiếp cận và giải quyết vấn đề cho ứng dụng Front-end.

- Xác định yêu cầu:

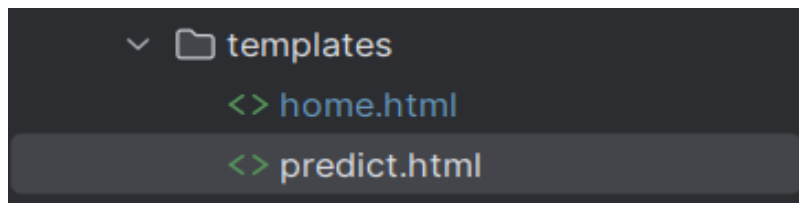
- Định rõ chức năng và yêu cầu của ứng dụng Front-end, bao gồm việc xác định các trang, thành phần giao diện, yêu cầu dữ liệu, và các tương tác người dùng.
- Sử dụng Django view để xử lý yêu cầu từ phía người dùng và gửi dữ liệu đến các template để hiển thị giao diện người dùng.
- Thiết kế giao diện người dùng:
 - Sử dụng Django template language để tạo ra các template HTML chứa giao diện người dùng.
 - Sử dụng CSS để tạo kiểu cho các thành phần giao diện, áp dụng các hiệu ứng và trang trí cho giao diện.

2.2.2. Công nghệ giải quyết phía Front-end.

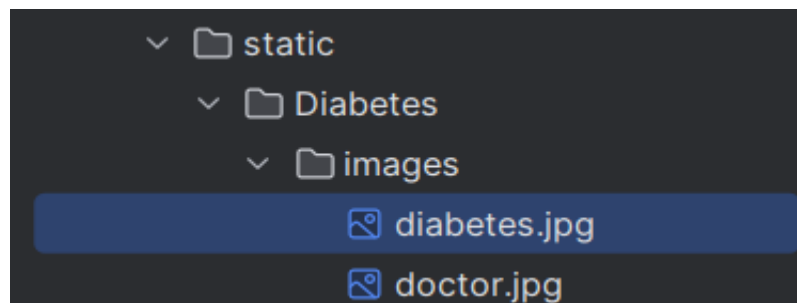
Django là một framework phát triển ứng dụng web sử dụng ngôn ngữ Python. Django cho phép xây dựng các trang web phức tạp và mạnh mẽ bằng cách kết hợp HTML, CSS và Python.

- HTML: Django hỗ trợ việc sử dụng các file HTML để xây dựng giao diện người dùng. Chúng ta có thể tạo các file HTML trong thư mục templates của dự án Django. Django sử dụng hệ thống template để kết hợp HTML với dữ liệu động từ Python.
- CSS: Django không có cách đặc biệt để sử dụng CSS. Chúng ta có thể tạo các file CSS trong thư mục static của dự án Django và sau đó liên kết chúng với các file HTML bằng cách sử dụng thẻ `<link>`. Ở trong dự án này sử dụng CSS Internal.

Cấu trúc thư mục HTML:

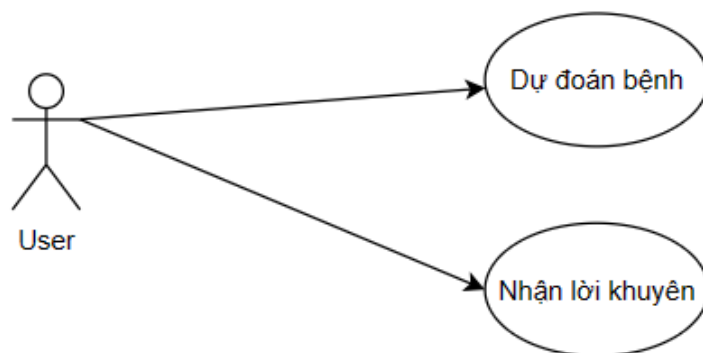


- Cấu trúc thư mục hình ảnh sử dụng trong dự án:



2.2.3. Phân tích thiết kế.

a. Use case



b. Scenario

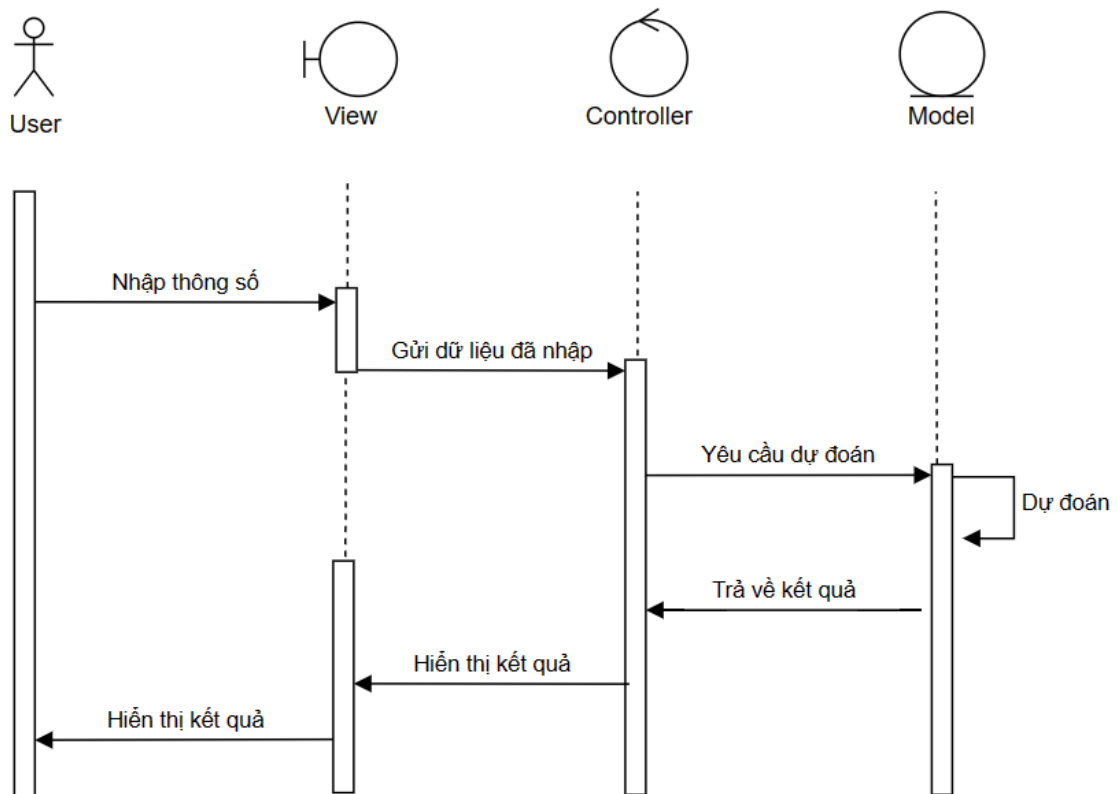
| | |
|---|--|
| Tên Use Case | Dự đoán bệnh |
| Tác nhân | Người dùng |
| Tiền điều kiện | Người dùng truy cập vào trang web |
| Đảm bảo thành công | Trang web hiển thị giao diện để người dùng nhập các thông số về bệnh và bấm vào nút “Dự đoán”.Kết quả được hiển thị phía dưới. |
| Chuỗi sự kiện chính: <ul style="list-style-type: none"> - Người dùng truy cập vào trang web - Hệ thống hiển thị các thông tin về các thông số để người dùng nhập - Người dùng click vào nút “Dự đoán” sau khi đã nhập đầy đủ các thông số - Hệ thống xử lý và hiển thị ra kết quả dự đoán | |
| Ngoại lệ: Không có | |

2.Đưa ra lời khuyên

| | |
|---|--|
| Tên Use Case | Nhận lời khuyên |
| Tác nhân | Người dùng |
| Tiền điều kiện | Người dùng click vào nút “Dự đoán” sau khi đã nhập đầy đủ các thông số |
| Đảm bảo thành công | Trang web hiển kết quả và lời khuyên được hiển thị phía dưới. |
| Chuỗi sự kiện chính: <ul style="list-style-type: none">- Người dùng các thông số và click vào nút dự đoán- Hệ thống xử lý và hiển thị ra kết quả dự đoán và lời khuyên | |
| Ngoại lệ: Không có | |

Sequence Diagram

Dự đoán bệnh tiêu đường



2.2.4. Cài đặt và triển khai Front-end.

a. Xây dựng giao diện

- Code HTML trang chủ

```

<body>
<div align="center">
<h1>
Chào mừng đến với hệ thống dự đoán bệnh tiêu đường
</h1>
<form action="predict">
<input type="submit" value="Dự đoán">
</form>
</div>
</body>
  
```

- Code CSS trang chủ

```

<style type="text/css">
div{
color: 'white'
}
h1{
color: 'white';
font-family: arial, sans-serif;
font-size: 60px;
font-weight: bold;
margin-top: 200px;
}
h2{
  
```

```

color: 'white';
font-family: arial, sans-serif;
font-size: 15px;
font-weight: bold;
margin-top: 400px;
}
body{
background-image:url("{% static 'Diabetes/images/puppy1.jpg' %}");
background-repeat: no-repeat;
background attachment:fixed;
background-size:cover;
}
input[type=submit]{
background-color: #4dc3ff;
border: 2px;
color: white;
padding: 16px 32px;
cursor: pointer;
margin-top: 15px;
}
</style>

```

Trang dự đoán

- Code HTML

```

<body>
<div align="center" class="main">
<a href="/" class="btn btn-primary" > Trở lại trang chủ </a>

<h1>Nhập các thông số kiểm tra: </h1>
<form action="result" method="">
<table>
<tr>
<td align="right">Pregnancies:</td>
<td align="left">
<input type="text" name="n1">
</td>
</tr>

<tr>
<td align="right">Glucose:</td>
<td align="left">
<input type="text" name="n2">
</td>
</tr>

<tr>
<td align="right">Blood Pressure:</td>
<td align="left">
<input type="text" name="n3">
</td>
</tr>

```

```

<tr>
<td align="right">Skin Thickness:</td>
<td align="left">
<input type="text" name="n4">
</td>
</tr>

<tr>
<td align="right">Insulin:</td>
<td align="left">
<input type="text" name="n5">
</td>
</tr>

<tr>
<td align="right">BMI:</td>
<td align="left">
<input type="text" name="n6">
</td>
</tr>

<tr>
<td align="right">Diabetes Pedigree function:</td>
<td align="left">
<input type="text" name="n7">
</td>
</tr>

<tr>
<td align="right">Age:</td>
<td align="left">
<input type="text" name="n8">
</td>
</tr>

</table>
<input type="submit" value="Dự đoán">
</form>
<span style="color:red;"> Kết quả:{{result2}}</span>
<br>
<span style="color:red;"> Lời khuyên:{{advice}}</span>
</div>

```

```

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.b
undle.min.js"
integrity="sha384-geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j
/68SIy3Te4Bkz" crossorigin="anonymous"></script>
</body>

```

- Code CSS

```

<style>
body{

```

```

background-image : url(" {% static 'Diabetes/images/diabetes.jpg' %}
");
background-repeat: no-repeat;
background attachment:fixed;
background-size:cover;
}
.main{
position: fixed;
top: 40px;
left: 510px;
width: 550px;
background-color: #ffffff;
border-radius: 10px;
align-items:center;
padding: 5%;

}
h1{
color: #0086b3;
font-size: 30px;
font-weight: bold;
}
input[type=submit]{
background-color: #4dc3ff;
border: 2px;
color: #ffffff;
padding: 8px 16px;
cursor:pointer;
margin-top: 15px;
}
</style>

```

b. Giao diện hệ thống

Trở lại trang chủ

Nhập các thông số kiểm tra:

Pregnancies:

Glucose:

Blood Pressure:

Skin Thickness:

Insulin:

BMI:

Diabetes Pedigree function:

Age:

Dự đoán

Kết quả:

Lời khuyên:

3. Kết luận

Bệnh tiểu đường đã trở thành một trong những nguyên nhân hàng đầu gây tử vong cho con người trong những thập kỷ gần đây. Tỷ lệ mắc bệnh tiểu đường liên tục gia tăng hàng năm do một số lý do bao gồm thói quen ăn uống, lối sống ít vận động và sự phổ biến của thực phẩm không lành mạnh. Mô hình dự đoán bệnh tiểu đường có thể đóng góp vào quá trình ra quyết định trong quản lý lâm sàng. Biết các yếu tố nguy cơ tiềm ẩn và xác định những người có nguy cơ cao trong giai đoạn đầu có thể hỗ trợ phòng ngừa bệnh tiểu đường. Mô hình dự đoán được lựa chọn là hồi quy logistic dựa trên thuật toán học máy là một trong những phương pháp phổ biến nhất. Đây là một thuật toán máy học đơn giản, có thể được sử dụng để sàng lọc bệnh tiểu đường mà không cần sử dụng phòng thí nghiệm.

Trong quá trình triển khai và thử nghiệm dự án vẫn còn xuất hiện những hạn chế. Trong chương trình thử nghiệm, độ chính xác của mô hình là 74%, tỷ lệ dự đoán sai của mô hình là 26%. Vì thế trong tương lai, nhóm sẽ tiếp tục cải tiến mô hình để đưa ra dự đoán chính xác nhất