**TRƯỜNG ĐẠI HỌC ĐỒNG THÁP**

**KHOA SƯ PHẠM TOÁN – TIN**



**BÁO CÁO MÔN HỌC**

**LẬP TRÌNH WEB**

**ĐỀ TÀI**

**WEBSIDE BÁN ĐIỆN THOẠI**

**NGHÀNH: KHOA HỌC MÁY TÍNH**

**LỚP: ĐHCNTT19**

Đồng Tháp, tháng 6 năm 2022

**TRƯỜNG ĐẠI HỌC ĐỒNG THÁP**

**KHOA SƯ PHẠM TOÁN – TIN**



**BÁO CÁO MÔN HỌC**

**ĐỀ TÀI**

**WEBSIDE BÁN ĐIỆN THOẠI**

**NGHÀNH: KHOA HỌC MÁY TÍNH**

**LỚP: ĐHCNTT19**

LÊ NGUYỄN QUỐC NHẬT 0019410466

PHẠM TẤN MINH 0019410795

NGUYỄN HOÀNG THẢO 0019410300

GIẢNG VIÊN HƯỚNG DẪN: Ths. NGUYỄN THỊ THANH THẢO

MỤC LỤC

[PHẦN 1. GIỚI THIỆU 4](#_Toc105700205)

[1.1 Cơ sở lý thuyết 4](#_Toc105700206)

[1.2 Ví dụ các ứng dụng của giải thuật 6](#_Toc105700207)

[1.3 Chạy giải thuật trên weka 6](#_Toc105700208)

[PHẦN 2. ỨNG DỤNG GIẢI THUẬT XÂY DỰNG ỨNG DỤNG WEB 9](#_Toc105700209)

[2.1 Tìm hiểu giải thuật và tạo mô hình dự đoán 9](#_Toc105700210)

[2.2 Xây dựng giao diện ứng dụng Web 10](#_Toc105700211)

[2.3 Xây dựng ứng dụng mô hình dự báo 10](#_Toc105700212)

# GIỚI THIỆU

## Giới thiệu bài toán

Thế giới ngày một phát triển, đời sống con người càng hiện đại hóa, nhu cầu về đời sống vật chất và tinh thần của người ngày càng một cao và có nhiều đòi hỏi khắc khe hơn.

Hằng ngày mõi người trong chúng ta đều có nhunwngx nhu cầu thiết yếu như: ăn uống, mua sắm, xem phim, nghe nhạc, … để mua những sản phẩm mình muốn hay thưởng thức những bản nhạc mới nhất.

Nắm bắt được nhu cầu đó và nhận thấy sự phát triển vượt bậc của mạng Internet ở khắp nơi trên thới giới nói chung và trong nước nói riêng trong những năm gần đây việc ứng dụng công nghệ thông tin vào lĩnh vực tìm kiếm thông tin, truy cập dữ liệu từ xa không còn khó khăn. Người ta tận dụng những lợi ích mà lĩnh vực này mang lại.

Thương mại điện tử ngày nay là một xu thế đăng được phát triển mạnh mẽ và không ngừng.

Shopping online là công cụ đăng được nhiều người lựa chọn.

Vì vậy, nắm bắt được nhu cầu của người sử dụng mà nhiều giang hàng thương mại điện tử xuất hiện không ngừng để đáp ứng nhu cầu của người dùng

Có hai phương án để đáp ứng nhu cầu cho người dùng:

Một là, xây dựng một giang hàng thương mại điện tử.

Hai là, xây dựng một webside chuyên kinh doanh .

🡪 Phương án 2 sẽ phù hợp hơn. Chính vì những lý do trên, nhóm em chọn đề tài *xây dựng Webside bán điện thoại di động.*

## Nhận xét bài toán

Bài toán thương mại điện tử là một bài toán lớn và phức tạp với nhiều model khác nhau như: tìm kiếm sản phẩm, đưa hàng vào giỏ hàng, đăng ký thành viên, các chức năng quản lý,…

Những khó khăn hiện tại của cửa hàng: Cùng với sự xuất hiện của đang dạng của các hãng điện thoại di động khác nhau và yêu cầu về mở rộng thị trường cũng như việc tiếp thị nhanh nhất đến khách hàng, cửa hàng phải đối mặt với những vấn đề khó giải quyết do cơ cấu củ không đáp ứng được:

* Khách hàng tốn thời gian đến trực tiếp đến các của hàng để xem và mua sản phẩm.
* Khả năng tiếp thị đến khách hàng thấp, không thu hút được nhiều khách hàng cho cửa hàng.
* Việc quản lý tồn kho không chặc chẽ, còn thiếu linh động dẫn đẻn khó khăn trong khâu quản lý và tình trạng thiếu thông tin cho khách hàng.

## Giải quyết bài toán

* **Sơ đồ thương mại Webside điện tử B2C Exchange:**

Diagram

Description automatically generated

Sơ đồ Webside B2C Exchange

* **Mô tả:**

**ADMIN** (Nhà quản trị): là quản trị trang Web, admin đang nhập vào tài khoản có toàn quyền sử dụng cấu hình và thêm sửa xóa cho toàn cơ sở dữ liệu, xem chi tiết các đơn hàng, quản lý thông tin danh mục, sản phẩm, nhà cung cấp, đơn đặt hàng, …

**Customer** (Khách hàng): là khách hàng cần mua sản phẩm trên mạng. Họ có thể chọn từng hãng điện thoại ưa thích. Họ dạo quanh cửa hàng mua từng sản phẩm của các hãng điện thoại khác nhau, chọn sản phẩm bỏ vào giỏ hàng, tạo đơn đặt hàng cho nhà quản trị. Customer là khách hàng thành viên đã đăng ký tài khoản và mật khẩu để tiện cho cho các giao dịch tiếp theo. Họ có thể thừa hưởng những chương trình khuyến mãi, giảm giá của cửa hàng (nếu có).

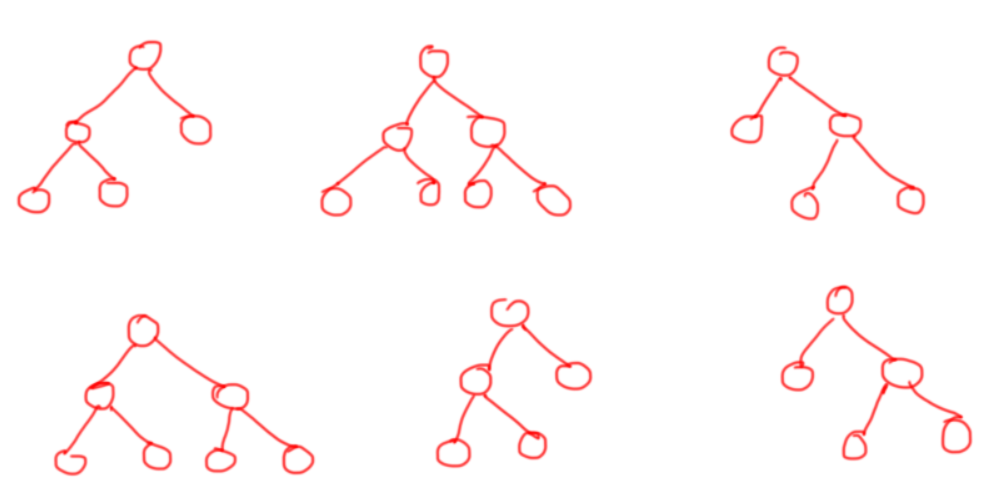
# ỨNG DỤNG GIẢI THUẬT XÂY DỰNG ỨNG DỤNG WEB

## Tìm hiểu giải thuật và tạo mô hình dự đoán

**Giới thiệu về thuật toán Random Forest**

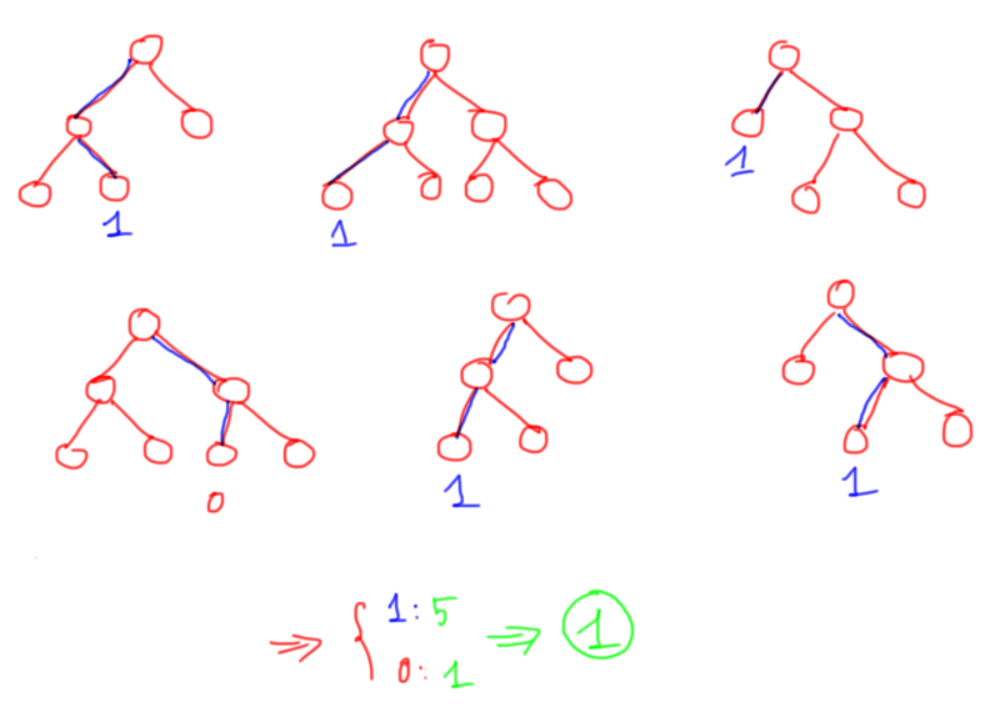
Random là ngẫu nhiên, Forest là rừng, nên ở thuật toán Random Forest mình sẽ xây dựng nhiều cây quyết định bằng thuật toán Decision Tree, tuy nhiên mỗi cây quyết định sẽ khác nhau (có yếu tố random). Sau đó kết quả dự đoán được tổng hợp từ các cây quyết định.

Ở bước huấn luyện chúng ta sẽ xây dựng nhiều cây quyết định, các cây quyết định có thể khác nhau (phần sau mình sẽ nói mỗi cây được xây dựng như thế nào).



Hình Bước huấn luyện xây dựng cây quyết định

Sau đó ở bước dự đoán, với một dữ liệu mới, thì ở mỗi cây quyết định mình sẽ đi từ trên xuống theo các node điều kiện để được các dự đoán, sau đó kết quả cuối cùng được tổng hợp từ kết quả của các cây quyết định.



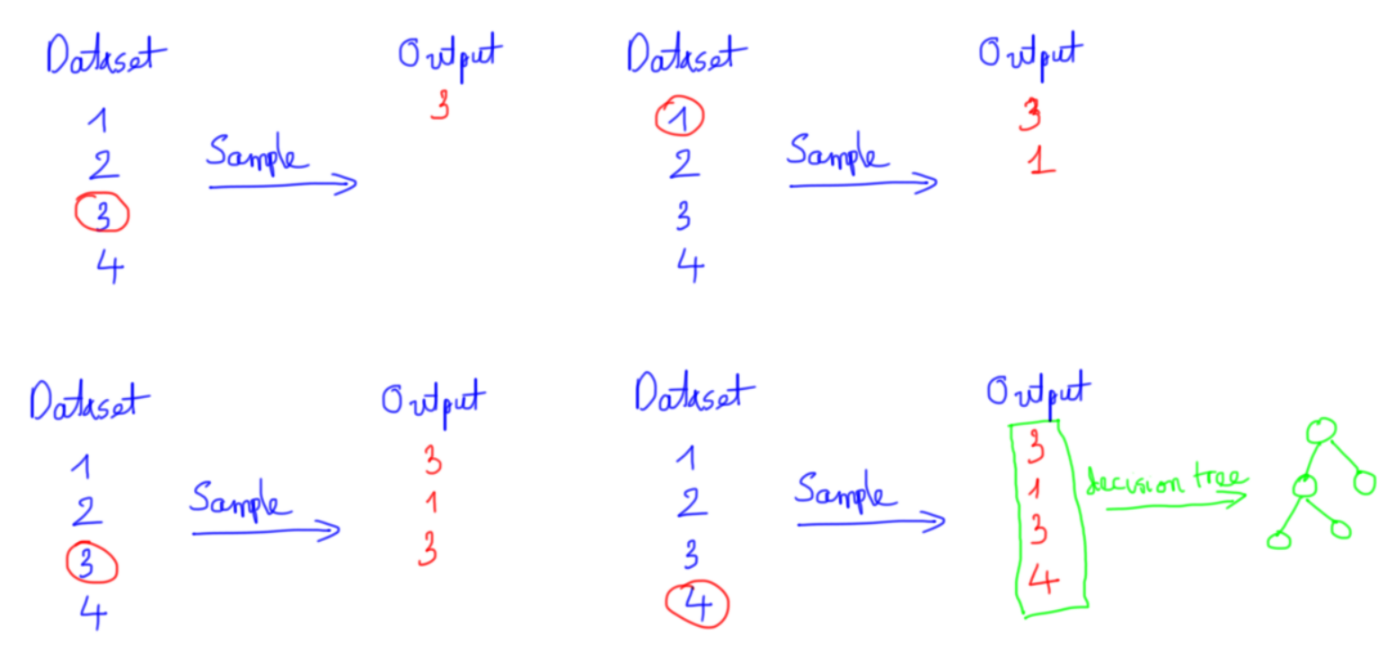
Hình Tổng hợp kết quả từ các cây quyết định

**Xây dựng thuật toán Random Forest**

Giả sử bộ dữ liệu của mình có n dữ liệu (sample) và mỗi dữ liệu có d thuộc tính (feature).

Để xây dựng mỗi cây quyết định mình sẽ làm như sau:

**Bước 1:** Lấy ngẫu nhiên n dữ liệu từ bộ dữ liệu với kĩ thuật Bootstrapping, hay còn gọi là random sampling with replacement. Tức khi mình sample được 1 dữ liệu thì mình không bỏ dữ liệu đấy ra mà vẫn giữ lại trong tập dữ liệu ban đầu, rồi tiếp tục sample cho tới khi sample đủ n dữ liệu. Khi dùng kĩ thuật này thì tập n dữ liệu mới của mình có thể có những dữ liệu bị trùng nhau.



Hình Minh hoạ bước 1 của thuật toán Random Forest

**Bước 2:** Sau khi sample được n dữ liệu từ bước 1 thì mình chọn ngẫu nhiên ở k thuộc tính (k < n). Giờ mình được bộ dữ liệu mới gồm n dữ liệu và mỗi dữ liệu có k thuộc tính.

**Bước 3:** Dùng thuật toán Decision Tree để xây dựng cây quyết định với bộ dữ liệu ở bước 2.

Do quá trính xây dựng mỗi cây quyết định đều có yếu tố ngẫu nhiên (random) nên kết quả là các cây quyết định trong thuật toán Random Forest có thể khác nhau.

Thuật toán Random Forest sẽ bao gồm nhiều cây quyết định, mỗi cây được xây dựng dùng thuật toán Decision Tree trên tập dữ liệu khác nhau và dùng tập thuộc tính khác nhau. Sau đó kết quả dự đoán của thuật toán Random Forest sẽ được tổng hợp từ các cây quyết định.

Khi dùng thuật toán Random Forest, mình hay để ý các thuộc tính như: số lượng cây quyết định sẽ xây dựng, số lượng thuộc tính dùng để xây dựng cây. Ngoài ra, vẫn có các thuộc tính của thuật toán Decision Tree để xây dựng cây như độ sâu tối đa, số phần tử tối thiểu trong 1 node để có thể tách.

## Xây dựng giao diện ứng dụng Web

* Về mô hình dự báo của nhóm, nhóm quyết định chọn mô hình dự báo bệnh tiểu đường. Giao diện của web do nhóm thiết kế bao gồm các thông tin: Số lần mang thai, isulin, nồng độ glucose, chỉ số BMI, huyết áp, chức năng phả hệ tiểu đường, độ dày của da, tuổi.

+ Số lần mang thai: là sổ lần mà người dự đoán mang thai, giá trị tối đa là 17 giá trị tối thiểu là 0 vì có thể người dự đoán là nam nên số lần mang thai bằng không là trường hợp riêng biệt.

+ Isulin: là một loại hormone từ các tế bào đảo tụy ở tuyến tụy tiết ra. Chúng có tác dụng chuyển hóa các chất carbohydrate trong cơ thể. Insulin còn có tác dụng đến việc chuyển hóa các mô mỡ và gan thành loại năng lượng ATP để cung cấp cho hoạt động của cơ thể. Giá trị tối đa là 846 và giá trị tối thiểu là 0.

+ Nồng độ glucose: Glucose trong máu là lượng đường trong máu hay còn có tên gọi khác là “đường huyết”. Glucose trong máu rất quan trọng, nó giúp cung cấp năng lượng cho các hoạt động của cơ thể. Khi glucose trong máu không đủ là nguyên nhân vì sao chúng ta hay cảm thấy mệt mỏi, chóng mặt, có khi bị ngất vì lượng đường trong máu không đủ dẫn đến dẫn đến “hạ đường huyết”.

Với bệnh nhân bị bệnh tiểu đường hay còn gọi là đái tháo đường thì glucose trong máu lúc đói khoảng 7mmol/L trở lên. Còn nếu mức Glucose trong lúc đói rơi vào khoảng từ 6,1-7 mmol/L thì có thể bạn đang nằm trong giai đoạn bị rối loạn đường huyết lúc đói. Nói cách khác thì đây là giai đoạn tiền tiểu đường.

Có khoảng 40% người có chỉ số glucose như này sẽ mắc bệnh tiểu đường trong 4 – 5 năm sau. Vì vậy nếu chỉ số xét nghiệm glucose trong máu của bạn nằm trong khoảng trên thì bạn nên thăm khám với bác sĩ sớm để được tư vấn. Mặt khác có biện pháp điều chỉnh hàm lượng glucose trong máu sao cho phù hợp. Cần ttránh để bệnh nặng rồi mới điều trị khi đó vừa tốn kém chi phí mà hiệu quả lại không cao.

+ Chỉ số BMI: BMIlà viết tắt của Body Mass Index được gọi là chỉ số khối cơ thể. Người phát minh ra công thức này là nhà khoa học người Bỉ có tên là Adolphe Quetelet.

Chỉ số BMI là cách tính dựa vào các thông tin chi tiết về hình dáng, cân nặng, chiều cao để đưa ra kết quả cuối cùng, từ đó xác định người đó có bị thừa cân, béo phì, suy dinh dưỡng hay không.

+ Huyết áp: Huyết áp là áp lực cần thiết tác động lên thành của động mạch để đưa máu từ tim đến nuôi dưỡng các mô trong cơ thể. Huyết áp được tạo ra do lực co bóp của cơ tim và sức cản của thành động mạch.

+ Độ dày của da: Độ dày của da được xác định về mặt di truyền và có thể được phân loại thành một trong các các loại sau: mỏng, trung bình, dày và u mô thừa.

## Xây dựng ứng dụng mô hình dự báo

### Thu thập dữ liệu

Dữ liệu được thu thập chủ yếu từ các nguồn sau:

- Các triệu chứng lâm sàng được trích từ cẩm nang DSM-5.

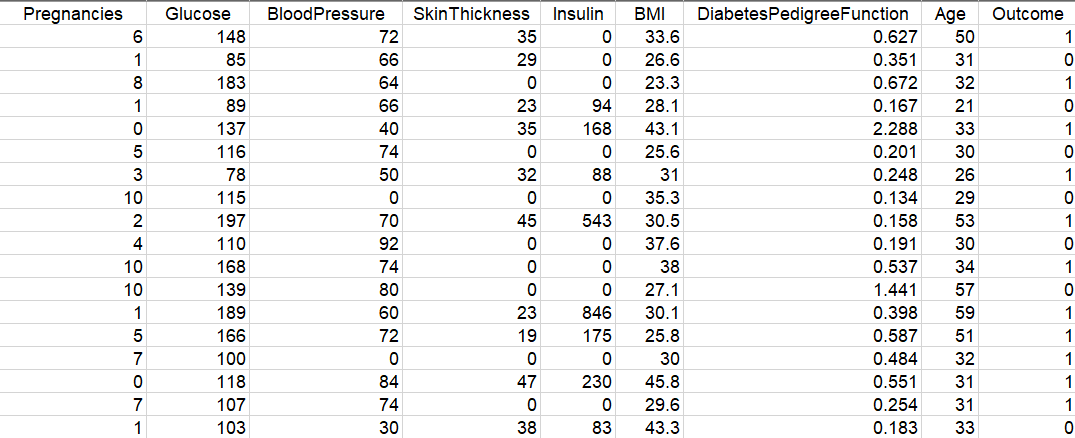
- Các tài liệu y học chuyên nghành, các bài báo chuyên môn.

- Ý kiến của một số bác sĩ hoặc các nhà tâm lý đang làm công

việc điều trị bệnh nhân cũng như trong công tác nghiên cứu về y

khoa.

### Xây dựng mô hình dự đoán bệnh tiểu đường. *a. Tập dữ liệu*

******

*Hình Tập dữ liệu diabetes.csv*

Ngôn ngữ lập trình: Python

Giải thuật: Rừng ngẫu nhiên

Chạy file trên Windows

Để chạy ứng dụng chúng ta sẽ chạy 2 dòng lên trên terminal như sau:

|  |
| --- |
| .\venv\Scripts\activate  python .\manage.py runserver |

**Xây dựng mô hình Random Forest sử dụng Python**

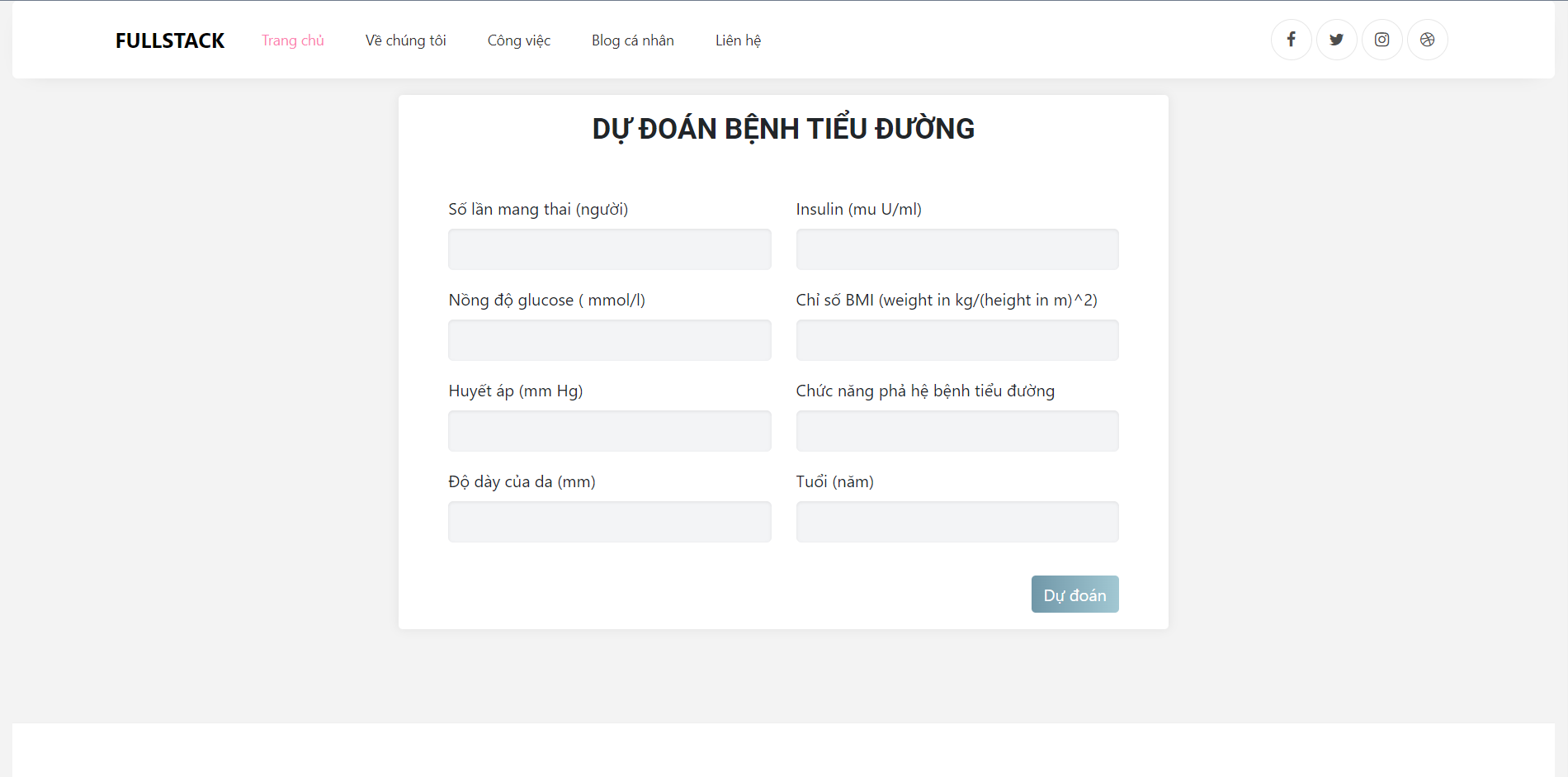
|  |
| --- |
| *import* pickle  *import* numpy *as* np *import* seaborn *as* sns *from* matplotlib *import* pyplot *as* plt *import* pandas *as* pd *# Import train\_test\_split function from* sklearn.model\_selection *import* train\_test\_split *# Import Random Forest Model from* sklearn.ensemble *import* RandomForestClassifier *# Import scikit-learn metrics module for accuracy calculation from* sklearn *import* metrics *from* sklearn.metrics *import* mean\_squared\_error  *# Load dataset* data\_url = 'diabetes.csv' raw\_df = pd.read\_csv(data\_url, header=9)  data = raw\_df.to\_numpy()[:, :8] target = raw\_df.to\_numpy()[:, -1]  dia = pd.DataFrame(data, columns=["Pregnancies", "Glucose", "BloodPressure", "SkinThickness", "Insulin", "BMI",  "DiabetesPedigreeFunction", "Age"]) dia['Outcome'] = target  X = dia.drop('Outcome', axis=1) y = dia['Outcome']  clf = RandomForestClassifier(n\_estimators=300, random\_state=21) clf.fit(X, y) pickle.dump(clf, *open*('model.pkl','wb')) |

**Xây dựng ứng dụng web kết hợp mô hình Randmon Forest với web frameword Django.**

|  |
| --- |
| *import* os  *from* .forms *import* DiabetesForm *from* rest\_framework *import* viewsets *from* django.core *import* serializers *from* rest\_framework.response *import* Response *from* rest\_framework *import* status *from* django.http *import* JsonResponse, HttpResponse *from* .models *import* Diabetes *from* .serializer *import* CustomerSerializers  *import* pickle *import* json *import* numpy *as* np *from* sklearn *import* preprocessing *import* pandas *as* pd *from* django.shortcuts *import* render, redirect *from* django.contrib *import* messages   *class* CustomerView(viewsets.ModelViewSet):  queryset = Diabetes.objects.all()  serializer\_class = CustomerSerializers   *def* status(df):  *try*:  PROJECT\_ROOT = os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_)))  model = pickle.load(*open*(os.path.join(PROJECT\_ROOT, "predict\_diabetes", "model.pkl"), 'rb'))   y\_pred = model.predict(df)  result = "Bạn bị bệnh tiểu đường" *if* y\_pred == 1 *else* "Bạn không bị bệnh tiểu đường"  *return* {'result': result, 'pred': y\_pred[0]}  *except ValueError as* e:  *return* Response(e.args[0], status.HTTP\_400\_BAD\_REQUEST)   *def* FormView(request):   *if* request.method == 'POST':  form = DiabetesForm(request.POST *or None*)   *if* form.is\_valid():  pregnancies = form.cleaned\_data['pregnancies']  glucose = form.cleaned\_data['glucose']  blood\_pressure = form.cleaned\_data['blood\_pressure']  skin\_thickness = form.cleaned\_data['skin\_thickness']  insulin = form.cleaned\_data['insulin']  BMI = form.cleaned\_data['BMI']  diabetes\_pedigree\_function = form.cleaned\_data['diabetes\_pedigree\_function']  age = form.cleaned\_data['age']  df = pd.DataFrame({'Pregnancies': [pregnancies], 'Glucose': [glucose], 'BloodPressure': [blood\_pressure],  'SkinThickness': [skin\_thickness], 'Insulin': [insulin], 'BMI': [BMI],  'DiabetesPedigreeFunction': [diabetes\_pedigree\_function], 'Age': [age]})  result = status(df)  *return* HttpResponse(json.dumps(result, ensure\_ascii=*True*))   form = DiabetesForm()  *return* render(request, 'form.html', {'form': form, 'title': 'Dự đoán bệnh tiểu đường'})   *# return HttpResponse(PROJECT\_ROOT)* |

**Demo ứng dựng dự đoán bệnh tiểu đường**

Ứng dụng mô hình dự đoán bệnh tiểu đường với 8 thuộc tính liên quan nhất về bệnh tiểu đường, người dùng cung cấp các thông tin vào các ô input để trang web dự đoán bệnh tiểu đường.

****

Hình Giao diện tổng thể website dự đoán bệnh tiểu đường



Hình Giao diện trang web đã dự đoán với các thông tin nhập vào.

Từ các thông số mà người dùng nhập vào trang web sẽ tiến hành phân tích các thông tin đó và tiến hành truyền vào các tham số cho mô hình. Mô hình sẽ tiến hành dự đoán và cho ra kết quả dự đoán phù hợp nhất từ đó giúp người dùng có các biện pháp bảo vệ bản thân, sức khoẻ tránh nguy cơ bị mất bệnh tiểu đường