**deploy**

mục đính chính của ML là tạo ra một mô hình có thể được sử dụng để dự đoán. Sau khi cleaning data, select features để có được một tập dữ liệu sạch và có các feature cần thiết liên quan đến kết quả, ta sẽ sử dụng tập dữ liệu đó để huấn luyện model. Trước khi huấn luyện model thì cần chọn ra một model được đánh giá cao nhất, sau đó tiến hành quá trình huấn luyện model và lưu lại model để sử dụng cho việc dự đoán. Sau khi lưu lại model, model được deploy để người dùng có thể sử dụng.

**server.py**

import pickle

from flask import Flask, request, json, jsonify

import numpy as np

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

# ---the filename of the saved model---

filename = 'diabetes.sav'

# ---load the saved model---

loaded\_model = pickle.load(open(filename, 'rb'))

@app.route('/diabetes/v1/predict', methods=['POST'])

def predict():

    # ---get the features to predict---

    features = request.json

    # ---create the features list for prediction---

    features\_list = [features["Glucose"], features["BMI"], features["Age"]]

    # ---get the prediction class---

    prediction = loaded\_model.predict([features\_list])

    # ---dự đoán xác suất(probabilities)---

    confidence = loaded\_model.predict\_proba([features\_list])

    # ---tính toán để phản hồi lại client---

    response = {}

    response['prediction'] = int(prediction[0])

    response['confidence'] = str(round(np.amax(confidence[0]) \* 100, 2)) #độ tin cậy

    return jsonify(response)

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

    app.run(host='0.0.0.0', port=5000)

**client.py**

import json

import requests

def predict\_diabetes(BMI, Age, Glucose):

    url = 'http://127.0.0.1:5000/diabetes/v1/predict'

    data = {"BMI": BMI, "Age": Age, "Glucose": Glucose}

    data\_json = json.dumps(data)

    headers = {'Content-type': 'application/json'}

    response = requests.post(url, data=data\_json, headers=headers)

    result = json.loads(response.text)

    return result

if \_\_name\_\_ == "\_\_main\_\_":

    predictions = predict\_diabetes(30, 40, 80)

    print("Diabetic" if predictions["prediction"] == 1 else "NotDiabetic")

    print("Confidence: " + predictions["confidence"] + "%")

output:

