In [1]:

```
import pandas as pd
```

In [2]:

```
data = pd.read_csv('diabetes.csv')
```

In [3]:

```
data
```

Out[3]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFu
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	
2	8	183	64	0	0	23.3	
3	1	89	66	23	94	28.1	
4	0	137	40	35	168	43.1	
...	...	...	...	...	...	...	
763	10	101	76	48	180	32.9	
764	2	122	70	27	0	36.8	
765	5	121	72	23	112	26.2	
766	1	126	60	0	0	30.1	
767	1	93	70	31	0	30.4	

768 rows × 9 columns

In [5]:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

In [6]:

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(data.drop('Outcome', axis=1), data['
```

In [7]:

```
# Create a decision tree classifier and fit it to the training data
clf = DecisionTreeClassifier(random_state=42)
clf.fit(X_train, y_train)
```

Out[7]:

```
DecisionTreeClassifier(random_state=42)
```

In [8]:

```
# Make predictions on the testing data
y_pred = clf.predict(X_test)
```

In [9]:

```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.7467532467532467
```

Pickle

In [11]:

```
import pickle
```

In [12]:

```
with open('decision_tree.pkl', 'wb') as f:
    pickle.dump(clf, f)
```

Flask

In [1]:

```
from flask import Flask, render_template, request
import pickle
import numpy as np

# Load the trained decision tree model
model = pickle.load(open('decision_tree.pkl', 'rb'))

# Create a Flask app
app = Flask(__name__)

# Define the route for the index page
@app.route('/')
def index():
    return render_template('index.html')

# Define the route for the prediction
@app.route('/predict', methods=['POST'])
def predict():
    # Get the form data
    pregnancies = int(request.form['Pregnancies'])
    glucose = int(request.form['Glucose'])
    blood_pressure = int(request.form['BloodPressure'])
    skin_thickness = int(request.form['SkinThickness'])
    insulin = int(request.form['Insulin'])
    bmi = float(request.form['BMI'])
    diabetes_pedigree_function = float(request.form['DiabetesPedigreeFunction'])
    age = int(request.form['Age'])

    # Put the form data in a numpy array in the same order as the columns in the trainin
    data = np.array([[pregnancies, glucose, blood_pressure, skin_thickness, insulin, bmi

    # Make a prediction using the trained decision tree model
    prediction = model.predict(data)[0]

    # Return the prediction to the user
    return render_template('index.html', prediction=prediction)

if __name__ == '__main__':
    app.run(debug=True)
```

```
 * Serving Flask app "__main__" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production d
eployment.
   Use a production WSGI server instead.
 * Debug mode: on

 * Restarting with watchdog (windowsapi)
```

```
An exception has occurred, use %tb to see the full traceback.

SystemExit: 1



C:\Users\arulk\anaconda3\lib\site-packages\IPython\core\interactiveshell.
py:3377: UserWarning: To exit: use 'exit', 'quit', or Ctrl-D.
  warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)
```

HTML : From

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Diabetes Prediction Form</title>
  </head>
  <body>
    <h1>Diabetes Prediction Form</h1>
    <form action="{{ url_for('predict') }}" method="post">
      <label for="Pregnancies">Pregnancies:</label>
      <input type="number" name="Pregnancies" id="Pregnancies" required>
      <br>
      <label for="Glucose">Glucose:</label>
      <input type="number" name="Glucose" id="Glucose" required>
      <br>
      <label for="BloodPressure">Blood Pressure:</label>
      <input type="number" name="BloodPressure" id="BloodPressure" required>
      <br>
      <label for="SkinThickness">Skin Thickness:</label>
      <input type="number" name="SkinThickness" id="SkinThickness" required>
      <br>
      <label for="Insulin">Insulin:</label>
      <input type="number" name="Insulin" id="Insulin" required>
      <br>
      <label for="BMI">BMI:</label>
      <input type="number" step="0.01" name="BMI" id="BMI" required>
      <br>
      <label for="DiabetesPedigreeFunction">Diabetes Pedigree Function:</label>
      <input type="number" step="0.01" name="DiabetesPedigreeFunction"
id="DiabetesPedigreeFunction" required>
      <br>
      <label for="Age">Age:</label>
      <input type="number" name="Age" id="Age" required>
      <br>
      <input type="submit" value="Predict">
    </form>
  </body>
</html>
```

HTML : Result

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Diabetes Prediction Result</title>
  </head>
```

```
  <body>
    <h1>Diabetes Prediction Result</h1>
    {% if prediction is not None %}
      {% if prediction == 1 %}
        <p>You have diabetes.</p>
      {% else %}
        <p>You don't have diabetes.</p>
      {% endif %}
    {% else %}
      <p>Oops! Something went wrong. Please try again.</p>
    {% endif %}
    <p><a href="/">Back to form</a></p>
  </body>
</html>
```