

อธิบาย Code app.py

render_template, request, redirect, url_for, jsonify และ import psycopg2 เชื่อมต่อกับฐานข้อมูล PostgreSQL และ import function predict_iris จากไฟล์ iris_predict

สร้าง object Flask ขึ้นมากำหนดหน้าแรกด้วย @app.route('/') จะเรียกใช้ฟังก์ชัน index()

ในฟังก์ชัน index() เปิดการเชื่อมต่อกับฐานข้อมูล PostgreSQL และทำการดึงข้อมูลทั้งหมดจากตาราง irispredict แล้วส่งข้อมูลไปยัง index.html เพื่อแสดงผล

กำหนดเส้นทางสำหรับการทำนาย (@app.route('/predict', methods=['POST'])) โดยฟังก์ชัน predict() จะถูกเรียกเมื่อมีการส่งข้อมูลผ่าน POST รับข้อมูลจากฟอร์มและทำการทำนาย ด้วยฟังก์ชัน predict_iris() จากนั้นจะทำการเพิ่มข้อมูลที่ทำนายได้เข้าไปในฐานข้อมูล และทำการดึงข้อมูลทั้งหมดอีกครั้งเพื่อส่งไปยัง index.html เพื่อแสดงผลรวมทั้งหมด

เงื่อนไข if __name__ == '__main__': เป็นจริง แสดงว่าโปรแกรมถูกเรียกโดยตรง จะทำการเรียกใช้งาน app.run() ซึ่งจะเริ่มเซิร์ฟเวอร์ Flask บน localhost ที่พอร์ต 8888

อธิบาย Code iris_predict.py

load_data(): ฟังก์ชันใช้งานกับ scikit-learn เพื่อโหลดข้อมูล iris โดยใช้ฟังก์ชัน load_iris() และส่งคืน features (X) และ labels (y)

train_model(X, y): ฟังก์ชันใช้งานกับ scikit-learn เพื่อฝึกโมเดล Logistic Regression ด้วยข้อมูลที่โหลดมา โดยแบ่งข้อมูลเป็นชุดฝึกและชุดทดสอบด้วยฟังก์ชัน train_test_split() และสร้างโมเดลด้วย Logistic Regression โดยกำหนดพารามิเตอร์ max_iter=1000 เพื่อให้การฝึกโมเดลเข้าสู่การประมวลผลที่มีการวนรอบมากพอ

predict_iris(features): ฟังก์ชันโมเดล Logistic Regression เพื่อการทำนายประเภทของดอก iris จาก โดยทำการโหลดข้อมูลและฝึกโมเดลด้วยฟังก์ชัน load_data() และ train_model() ตามลำดับ จากนั้นทำการทำนายประเภท iris จากคุณสมบัติที่กำหนด และส่งคืนผลลัพธ์เป็นชื่อของประเภทดอก iris ที่ทำนายได้ เช่น 'setosa', 'versicolor' หรือ 'virginica'