|  |  |
| --- | --- |
| import pickle  import numpy as np  from flask import **Flask, request** | import joblib  import numpy as np  from flask import Flask , jsonify |
| model = None  app = Flask(\_\_name\_\_) | # 2 guines \_ \_ name \_ \_  app = Flask(\_\_name\_\_) |
| def load\_model():  global model  # model variable refers to the global variable  with open('iris\_trained\_model.pkl', 'rb') as f:  model = pickle.load(f) |  |
| @app.route('/')  def **home\_endpoint():**  return 'Hello World! |  |
| @app.route('/predict', methods=['POST'])  def **get\_prediction():**  # Works only for a single sample  if request.method == 'POST':  # Get data posted as a json  data = request.get\_json()  # converts shape from (4,) to (1, 4)  data = np.array(data)[np.newaxis, :]  prediction = model.predict(data)  return str(prediction[0]) | @app.route('/predict', methods=['GET'])  def predict():  X\_test = np.array([7.594444821,7.479555538,1.616463184,  1.53352356,0.796666503,0.635422587,  0.362012237,0.315963835,2.277026653])  # 1, -1 1 fila, -1 sin limite  prediction = model.predict(X\_test.reshape(1,-1))  return jsonify({'prediccion' : list(prediction)}) |
|  | @app.route('/predict', methods=['POST','GET'])  def predict\_lofic():  query = request.args  campo1 = float(query.get('campo1'))  campo2 = float(query.get('campo2'))  resultado = modelo.predict([campo1, campo2])  return jsonify({‘prediction’: resultado}) |
|  | @app.route('/predict', methods=['GET'])  def predict():  parser = reqparse.RequestParser()  parser.add\_argument('petal\_length')  parser.add\_argument('petal\_width')  parser.add\_argument('sepal\_length')  parser.add\_argument('sepal\_width')  args = parser.parse\_args() # creates dict  X\_new = np.fromiter(args.values(), dtype=float)  # convert input to array  resultado = {'Prediction': IRIS\_MODEL.predict([X\_new])[0]}  return resultado, 200 |
| if \_\_name\_\_ == '\_\_main\_\_':  load\_model() # load model one time  app.run(host='0.0.0.0', port=80) | if \_\_name\_\_ == "\_\_main\_\_":  model = joblib.load('./models/best\_model.pkl')  # model = joblib.load('./models/iris.mdl')  app.run(port=8080) |