import pickle  
import pycaret  
import pandas as pd  
from pycaret.classification import load\_model, predict\_model  
  
import numpy as np  
from flask import Flask, request, jsonify, render\_template  
  
# create flask app  
app = Flask(\_\_name\_\_)  
  
# Load the pickle model  
model = load\_model("best-model")  
  
  
@app.route("/")  
def Home():  
 return render\_template("index.html")  
  
  
@app.route("/predict", methods=["POST"])  
def predict():  
 float\_features = [float(x) for x in request.form.values()]  
 features = pd.DataFrame([np.array(float\_features)],columns=['Length', 'Diameter', 'Height', 'Whole\_weight', 'Shucked\_weight',  
 'Viscera\_weight', 'Shell\_weight', 'Rings'])  
 prediction = predict\_model(model,features)  
  
 return render\_template("index.html", prediction\_text="The abalone sex is {}".format(prediction['prediction\_label'][0]))  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 app.run(debug=True)

import pickle  
import pycaret  
import pandas as pd  
from pycaret.classification import load\_model, predict\_model  
  
import numpy as np  
from flask import Flask, request, render\_template  
  
# create flask app  
app = Flask(\_\_name\_\_)  
  
# Load the pickle model  
model = load\_model("best-model")  
  
  
@app.route("/")  
def Home():  
 return render\_template("index.html")  
  
  
@app.route("/predict", methods=["POST"])  
def predict():  
 float\_features = [float(x) for x in request.form.values()]  
 features = pd.DataFrame([np.array(float\_features)],columns=['Length', 'Diameter', 'Height', 'Whole\_weight', 'Shucked\_weight',  
 'Viscera\_weight', 'Shell\_weight', 'Rings'])  
 prediction = predict\_model(model,features)  
  
 return render\_template("index.html", prediction\_text="The abalone sex is {}".format(prediction['prediction\_label'][0]))  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 app.run(debug=True)