from flask import Flask, render\_template, request

import pickle

import pandas as pd

import numpy as np

from sklearn.preprocessing import OneHotEncoder

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

# Load the model and OneHotEncoder

model = pickle.load(open("model.pkl", 'rb'))

ohe = pickle.load(open("ohe.pkl", 'rb')) # Load the OneHotEncoder

@app.route('/')

def welcome():

return render\_template('index.html')

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

def predict():

age = request.form['age']

gender = request.form['gender']

occupation = request.form['occupation']

line\_of\_work = request.form['line\_of\_work']

prefer = request.form['prefer']

certaindays\_hw = request.form['certaindays\_hw']

time\_bp = float(request.form['time\_bp'])

time\_dp = float(request.form['time\_dp'])

travel\_time = float(request.form['travel\_time'])

easeof\_online = float(request.form['easeof\_online'])

home\_env = float(request.form['home\_env'])

prod\_inc = float(request.form['prod\_inc'])

sleep\_bal = float(request.form['sleep\_bal'])

new\_skill = float(request.form['new\_skill'])

fam\_connect = float(request.form['fam\_connect'])

relaxed = float(request.form['relaxed'])

self\_time = float(request.form['self\_time'])

like\_hw = float(request.form['like\_hw'])

dislike\_hw = float(request.form['dislike\_hw'])

# Validate that the input values are not empty

if '' in [age, gender, occupation, line\_of\_work, prefer, certaindays\_hw]:

return render\_template('index.html', predict="Please fill all fields.")

# Create a DataFrame from the input data

input\_data = pd.DataFrame({

'age': [age],

'gender': [gender],

'occupation': [occupation],

'line\_of\_work': [line\_of\_work],

'prefer': [prefer],

'certaindays\_hw': [certaindays\_hw],

'time\_bp': [time\_bp],

'time\_dp': [time\_dp],

'travel\_time': [travel\_time],

'easeof\_online': [easeof\_online],

'home\_env': [home\_env],

'prod\_inc': [prod\_inc],

'sleep\_bal': [sleep\_bal],

'new\_skill': [new\_skill],

'fam\_connect': [fam\_connect],

'relaxed': [relaxed],

'self\_time': [self\_time],

'like\_hw': [like\_hw],

'dislike\_hw': [dislike\_hw]

})

# Transform categorical variables using the loaded OneHotEncoder

encoded\_data = ohe.transform(input\_data[['age', 'gender', 'occupation', 'line\_of\_work', 'prefer', 'certaindays\_hw']])

encoded\_data\_df = pd.DataFrame(encoded\_data, columns=ohe.get\_feature\_names\_out(['age', 'gender', 'occupation', 'line\_of\_work', 'prefer', 'certaindays\_hw']))

# Concatenate the encoded data with the rest of the input data

input\_features = pd.concat([encoded\_data\_df, input\_data[['time\_bp', 'time\_dp', 'travel\_time', 'easeof\_online', 'home\_env', 'prod\_inc', 'sleep\_bal', 'new\_skill', 'fam\_connect', 'relaxed', 'self\_time', 'like\_hw', 'dislike\_hw']]], axis=1)

# Make the prediction using the loaded model

prediction = model.predict(input\_features)

# Return the prediction result

return render\_template('index.html', predict=f"Predicted value: {prediction[0]}")

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

app.run(debug=True, port=1111)