# -\*- coding: utf-8 -\*-

import numpy as np

import pickle

import joblib

import matplotlib

import matplotlib.pyplot as plt

import time

import pandas

import os

from sklearn import \*

from flask import Flask,request,jsonify,render\_template,redirect,url\_for

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

# return render\_template('index.html')

app = Flask(\_\_name\_\_, static\_folder='static')

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

scale= pickle.load(open("scale.pkl","rb"))

encoder = pickle.load(open("encoder.pkl","rb"))

@app.route('/')

def home():

return render\_template('index.html')

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

def predict():

inp\_feature = [x for x in request.form.values()]

inp\_feature=inp\_feature[:17]

print(inp\_feature)

feature\_values = [np.array(inp\_feature)]

names = [['Location', 'MinTemp', 'MaxTemp', 'Rainfall', 'WindGustSpeed',

'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm',

'Pressure9am', 'Pressure3pm', 'Temp9am', 'Temp3pm', 'RainToday',

'WindGustDir', 'WindDir9am', 'WindDir3pm']]

data = pandas.DataFrame(feature\_values,columns=names)

print(data)

data = scale.fit\_transform(data)

print(data)

data = pandas.DataFrame(data,columns=names)

print(data)

prediction = model.predict(data)

#pred\_prob = model.predict\_proba(data)

print(prediction)

if prediction == "Yes":

return render\_template("yes.html")

else:

return render\_template("no.html")

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

app.run()