from pyexpat import features, model

import numpy

import pickle

from sklearn.preprocessing import LabelEncoder

import pandas as pd

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

import requests

import json

# NOTE: you must manually set API\_KEY below using information retrieved from your IBM Cloud account.

API\_KEY = "PQBr9MBF7mFuSh2VVLfOE-liIA04VH-h5VEk8EfjFIuw"

token\_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API\_KEY, "grant\_type": 'urn:ibm:params:oauth:grant-type:apikey'})

mltoken = token\_response.json()["access\_token"]

print("ML Token",mltoken)

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

# Declare a Flask app

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

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

@app.route('/')

def home():

return render\_template("home.html")

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

def chance():

return render\_template("chance.html")

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

def nochance():

return render\_template("noChance.html")

@app.route('/help/')

def help():

return render\_template("help.html")

@app.route('/contact/')

def contact():

return render\_template("contact.html")

@app.route('/about/')

def about():

return render\_template("about.html")

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

def predict():

res = " "

# If a form is submitted

if request.method == "POST":

Location = request.form.get('Location')

MinTemp = request.form['MinTemp']

MaxTemp = request.form['MaxTemp']

Rainfall = request.form['Rainfall']

WindGustSpeed = request.form['WindGustSpeed']

WindSpeed9am = request.form['WindSpeed9am']

WindSpeed3pm = request.form['WindSpeed3pm']

Humidity9am = request.form['Humidity9am']

Humidity3pm = request.form['Humidity3pm']

Pressure9am = request.form['Pressure9am']

Pressure3pm = request.form['Pressure3pm']

Temp9am = request.form['Temp9am']

Temp3pm = request.form['Temp3pm']

RainToday = request.form.get('RainToday')

WindGustDir = request.form.get('WindGustDir')

WindDir9am = request.form.get('WindDir9am')

WindDir3pm = request.form.get('WindDir3pm')

new\_row = {'Location':Location,'MinTemp':MinTemp,'MaxTemp':MaxTemp,'Rainfall':Rainfall,'WindGustSpeed':WindGustSpeed,'WindSpeed9am':WindSpeed9am,'WindSpeed3pm':WindSpeed3pm,'Humidity9am':Humidity9am,'Humidity3pm':Humidity3pm,'Pressure9am':Pressure9am,'Pressure3pm':Pressure3pm,'Temp9am':Temp9am,'Temp3pm':Temp3pm,'RainToday':RainToday,'WindGustDir':WindGustDir,'WindDir9am':WindDir9am,'WindDir3pm':WindDir3pm}

print(new\_row)

new\_df = pd.DataFrame(columns=['Location','MinTemp','MaxTemp','Rainfall','WindGustSpeed','WindSpeed9am','WindSpeed3pm','Humidity9am','Humidity3pm','Pressure9am','Pressure3pm','Temp9am','Temp3pm','RainToday','WindGustDir','WindDir9am','WindDir3pm'])

new\_df = new\_df.append(new\_row,ignore\_index=True)

labeled = new\_df[['Location','MinTemp','MaxTemp','Rainfall','WindGustSpeed','WindSpeed9am','WindSpeed3pm','Humidity9am','Humidity3pm','Pressure9am','Pressure3pm','Temp9am','Temp3pm','RainToday','WindGustDir','WindDir9am','WindDir3pm']]

X = labeled.values

print(X)

payload\_scoring = {"input\_data": [{"field": [['Location','MinTemp','MaxTemp','Rainfall','WindGustSpeed','WindSpeed9am','WindSpeed3pm','Humidity9am','Humidity3pm','Pressure9am','Pressure3pm','Temp9am','Temp3pm','RainyToday','WindGustDir','WindDir9am','WindDir3pm']], "values": X.tolist()}]}

response\_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/73230b85-51ea-45db-baa7-e86b5d528fbe/predictions?version=2022-11-14', json=payload\_scoring,headers={'Authorization': 'Bearer ' + mltoken})

print("Scoring response")

predictions = response\_scoring.json()

print(predictions)

output = predictions['predictions'][0]['values'][0][0]

print(output)

else:

output = ""

if output == 1:

return redirect(url\_for('chance'))

elif output == 0:

return redirect(url\_for('nochance'))

return render\_template("index.html", output = res)

#Running the app

if \_\_name\_\_== "\_\_\_main\_\_\_":

app.run(debug = True,host='0.0.0.0',port=80)