```
#importing required libraries

from flask import Flask, request, render_template
import numpy as np
import pandas as pd
from sklearn import metrics
import warnings
import pickle
import requests
warnings.filterwarnings('ignore')
from feature import FeatureExtraction

file = open("model.pkl","rb")
gbc = pickle.load(file)
file.close()

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "cWGD5yTjEpEGtqPpvHPDBElN5eXFS7eh2JRDyUWhySMW"
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"]

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



app = Flask(__name__)

@app.route("/", methods=["GET", "POST"])
def index():
    if request.method == "POST":

        url = request.form["url"]
        obj = FeatureExtraction(url)
        x = np.array(obj.getFeaturesList()).reshape(1,30)

        y_pred =gbc.predict(x)[0]
        #1 is safe
        #-1 is unsafe
        y_pro_phishing = gbc.predict_proba(x)[0,0]
        y_pro_non_phishing = gbc.predict_proba(x)[0,1]
        # if(y_pred ==1 ):
        pred = "It is {0:.2f} % safe to go ".format(y_pro_phishing*100)
        payload_scoring = {"input_data": [{"field": [["UsingIP","LongURL","ShortURL","Symbol@","Redirecting//",
"PrefixSuffix-","SubDomains","HTTPS","DomainRegLen","Favicon","NonStdPort","HTTPSDomainURL","Reque
stURL","AnchorURL","LinksInScriptTags","ServerFormHandler","InfoEmail","AbnormalURL","WebsiteForwardi
ng","StatusBarCust","DisableRightClick","UsingPopupWindow","IframeRedirection","AgeofDomain","DNSRecor
ding","WebsiteTraffic","PageRank","GoogleIndex","LinksPointingToPage","StatsReport"
]], "values": [[1,1,1,1,1,-1,-1,-1,-1,1,1,1,1,-1,-1,1,1,1,0,1,1,1,1,-1,-1,-1,-1,1,0,1]]}]}
        response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/084b5c52-f617-40ef-a
0e8-3e6cf79ae447/predictions?version=2022-11-06', json=payload_scoring,
        headers={'Authorization': 'Bearer ' + mltoken})
        print("Scoring response")
        predictions=response_scoring.json()
#print(predictions)
        pred=print(predictions['predictions'][0]['values'][0][0])
        return render_template('index.html',xx =round(y_pro_non_phishing,2),url=url )
    return render_template("index.html", xx =-1)



if __name__ == "__main__":
    app.run(debug=True,port=2020)
 * Serving Flask app '__main__' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server
instead.
 * Running on http://127.0.0.1:2020
Press CTRL+C to quit
 * Restarting with stat
An exception has occurred, use %tb to see the full traceback.

SystemExit: 1
```