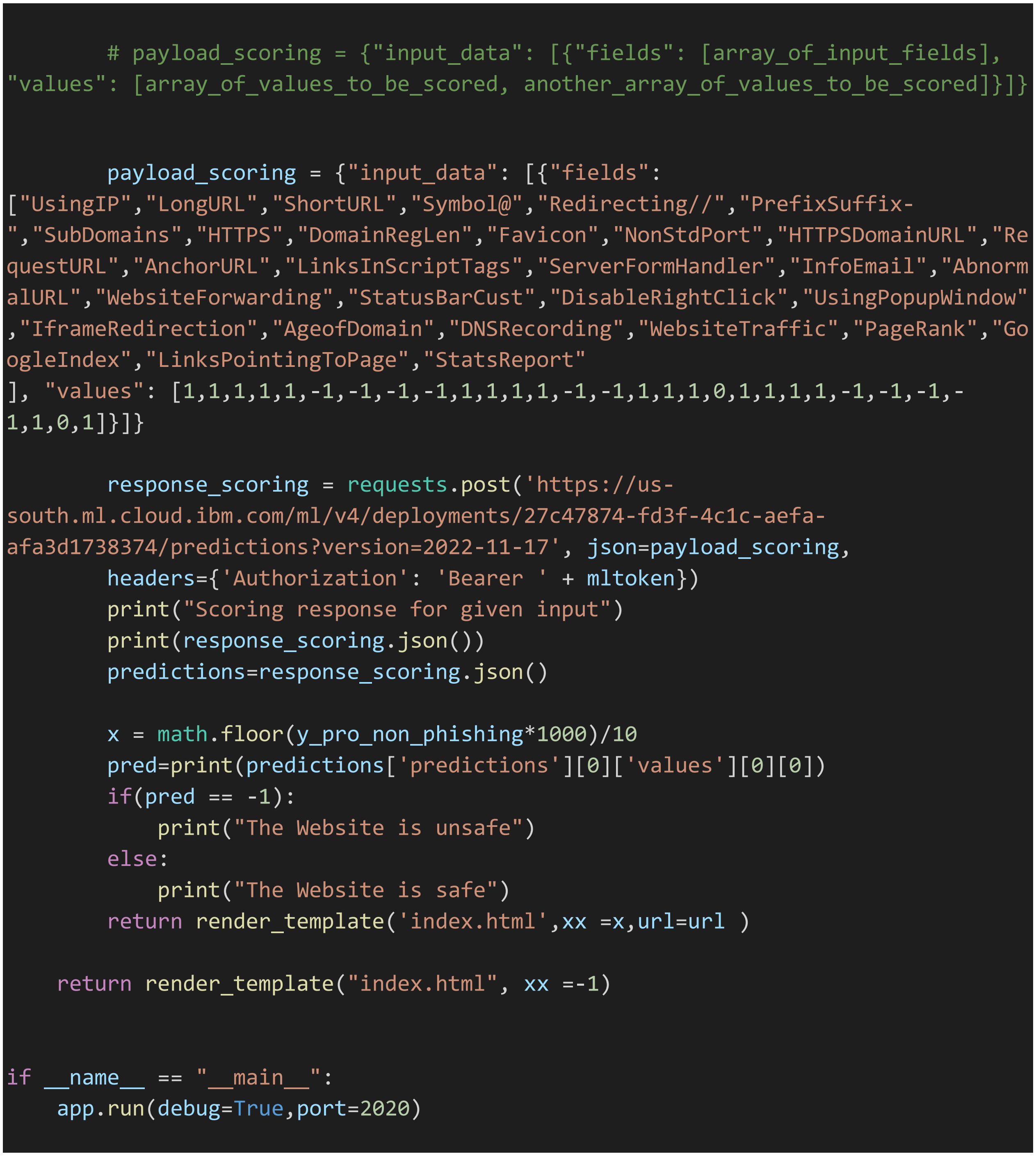
SCORING ENDPOINT

scoring_endpoint.py

```
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
import math

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

API_KEY = "<YOUR_API_KEY>"
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__, template_folder="templates")

@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]
        #0 - unsafe
        #1 - safe
        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": [{"fields": [array_of_input_fields], "values": [array_of_values_to_be_scored, another_array_of_values_to_be_scored]}]}


        payload_scoring = {"input_data": [{"fields": ["UsingIP","LongURL","ShortURL","Symbol@","Redirecting//","PrefixSuffix-","SubDomains","HTTPS","DomainRegLen","Favicon","NonStdPort","HTTPSDomainURL","RequestURL","AnchorURL","LinksInScriptTags","ServerFormHandler","InfoEmail","AbnormalURL","WebsiteForwarding","StatusBarCust","DisableRightClick","UsingPopupWindow","IframeRedirection","AgeofDomain","DNSRecording","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,-1,1,1,0,1]}]}

        response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/27c47874-fd3f-4c1c-aefa-afa3d1738374/predictions?version=2022-11-17', json=payload_scoring,
        headers={'Authorization': 'Bearer ' + mltoken})
        print("Scoring response for given input")
        print(response_scoring.json())
        predictions=response_scoring.json()

        x = math.floor(y_pro_non_phishing*1000)/10
        pred=print(predictions['predictions'][0]['values'][0][0])
        if(pred == -1):
            print("The Website is unsafe")
        else:
            print("The Website is safe")
        return render_template('index.html',xx =x,url=url )

    return render_template("index.html", xx =-1)


if __name__ == "__main__":
    app.run(debug=True,port=2020)
```

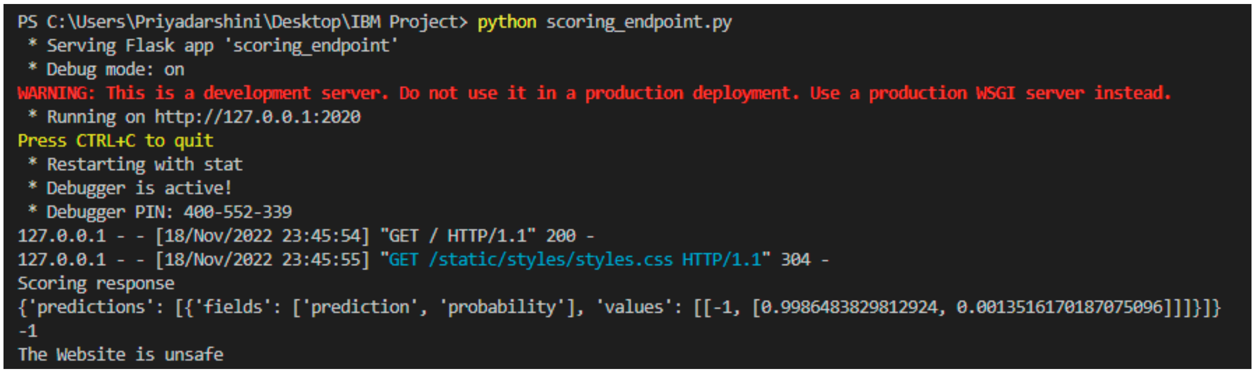
Given Input:

["UsingIP","LongURL","ShortURL","Symbol@","Redirecting//","PrefixSuffix-","SubDomains","HTTPS","DomainRegLen","Favicon","NonStdPort","HTTPSDomainURL","RequestURL","AnchorURL","LinksInScriptTags","ServerFormHandler","InfoEmail","AbnormalURL","WebsiteForwarding","StatusBarCust","DisableRightClick","UsingPopupWindow","IframeRedirectio n","AgeofDomain","DNSRecording","WebsiteTraffic","PageRank","GoogleIndex","LinksPointingToPage","StatsReport"

]

[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]

Given Output:



```
PS C:\Users\Priyadarshini\Desktop\IBM Project> python scoring_endpoint.py
 * Serving Flask app 'scoring_endpoint'
 * 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
 * Debugger is active!
 * Debugger PIN: 400-552-339
127.0.0.1 - - [18/Nov/2022 23:45:54] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [18/Nov/2022 23:45:55] "GET /static/styles/styles.css HTTP/1.1" 304 -
Scoring response
{'predictions': [{'fields': ['prediction', 'probability'], 'values': [[-1, [0.9986483829812924, 0.0013516170187075096]]]}]}
-1
The Website is unsafe
```