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

                    # NOTE: you must manually set API_KEY below using information
                    retrieved from your IBM Cloud account.
                    API_KEY = "" #hidden because of security reasons
                    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.config['MYSQL_HOST'] = 'localhost'
                    app.config['MYSQL_USER'] = 'root'
                    app.config['MYSQL_PASSWORD'] = ''
                    app.config['MYSQL_DB'] = 'report'

                    mysql = MySQL(app)

                    xgb = pickle.load(open("XGBoostClassifier.pkl", "rb"))

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

                            url = request.form["url"]
                            obj = FeatureExtraction(url)

                            x = np.array(obj.getFeaturesList()).reshape(1,13)
                            print(x)
```

```
        t=[obj.getFeaturesList()]
        print("t")
        print(t)
        # NOTE: manually define and pass the array(s) of values to be
scored in the next line
        payload_scoring = {"input_data": [{"fields":
[['f0','f1','f2','f3','f4','f5','f6','f7','f8','f9','f10','f11','f12'
]], "values": t}]}

        response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/859ae568-d692-4958-9dbe-
60431a8a0918/predictions?version=2022-11-11', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})
        print("Scoring response")
        print(response_scoring.json())

        y_pred =xgb.predict(x)[0]
        print(y_pred)
        y_pro_phishing = xgb.predict_proba(x)[0,0]
        print(y_pro_phishing)
        y_pro_non_phishing = xgb.predict_proba(x)[0,1]
        print(y_pro_non_phishing)

        if(y_pro_phishing*100<60):
            msg="Trick or Treat?\n Look at this tweet.\n This site is
elite!\n"
            flag=1
        else:
            msg="Trick or Treat?\n Don't get deceit.\n This site is
creep!\n"
            flag=-1

        return render_template('result.html', msg=msg, url=url,
val=flag)

    return render_template("index.html")

@app.route("/report", methods=["GET", "POST"])
def report():
    if request.method == 'GET':
        return render_template("contact.html")

    if request.method == 'POST':
        name = request.form['name']
        email = request.form['email']
        query = request.form['query']
        cursor = mysql.connection.cursor()
        cursor.execute(''' INSERT INTO responses
VALUES(%s,%s,%s)''',(name,email,query))
        mysql.connection.commit()
        cursor.close()
        return render_template("alert.html")


if __name__ == '__main__':
    app.run(debug=True)
```