**Integrating flask with IBM Cloud**

In [3]:

*#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')

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","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,**-**1,1,0,1]]}]}

response\_scoring **=** requests**.**post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/084b5c52-f617-40ef-a0e8-3e6cf79ae447/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

\* Restarting with watchdog (windowsapi)

An exception has occurred, use %tb to see the full traceback.

**SystemExit:** 1