```
from
urllib.pa
rse
import
urlparse
            import ipaddress
            import re
            import requests
            import whois
            from datetime import datetime

            class FeatureExtraction:
                features=[]
                def __init__(self,url):
                    self.features=[]
                    self.url = url

                    #Address bar based features
                    self.features.append(self.having_IPhaving_IP_Address())
                    self.features.append(self.URLURL_Length())
                    self.features.append(self.Shortining_Service())
                    self.features.append(self.having_At_Symbol())
                    self.features.append(self.double_slash_redirecting())
                    self.features.append(self.Prefix_Suffix())
                    self.features.append(self.HTTPS_token())

                    # HTML & Javascript based features
                    try:
                        self.response = requests.get(url)
                    except:
                        self.response = ""

                    self.features.append(self.on_mouseover())
                    self.features.append(self.RightClick())
                    self.features.append(self.popUpWidnow())
                    self.features.append(self.Iframe())

                    #Domain based features
                    dns = -1
                    try:
                        self.domain_name = whois.whois(urlparse(url).netloc)
                    except:
                        dns = 1

                    self.features.append(1 if dns == 1 else self.age_of_domain())
                    self.features.append(dns)
```

```
     # 1.UsingIp
    def having_IPhaving_IP_Address(self):
        #print("IP")
        try:
            ipaddress.ip_address(self.url)
            print("IP")
            return -1
        except:
            print("IP except")
            return 1

    # 2.longUrl
    def URLURL_Length(self):
        #print("Length")
        if len(self.url) < 54:
            return 1
        else:
            return -1

    # 3.shortUrl
    def Shortining_Service(self):
        #print("short")
        shortening_services =
r"bit\.ly|goo\.gl|shorte\.st|go2l\.ink|x\.co|ow\.ly|t\.co|tinyurl|tr\.im|is
\.gd|cli\.gs|" \

r"yfrog\.com|migre\.me|ff\.im|tiny\.cc|url4\.eu|twit\.ac|su\.pr|twurl\.nl|s
nipurl\.com|" \

r"short\.to|BudURL\.com|ping\.fm|post\.ly|Just\.as|bkite\.com|snipr\.com|fi
c\.kr|loopt\.us|" \

r"doiop\.com|short\.ie|kl\.am|wp\.me|rubyurl\.com|om\.ly|to\.ly|bit\.do|t\.
co|lnkd\.in|db\.tt|" \

r"qr\.ae|adf\.ly|goo\.gl|bitly\.com|cur\.lv|tinyurl\.com|ow\.ly|bit\.ly|ity
\.im|q\.gs|is\.gd|" \

r"po\.st|bc\.vc|twitthis\.com|u\.to|j\.mp|buzurl\.com|cutt\.us|u\.bb|yourls
\.org|x\.co|" \

r"prettylinkpro\.com|scrnch\.me|filoops\.info|vzturl\.com|qr\.net|1url\.com
|tweez\.me|v\.gd|" \
                      r"tr\.im|link\.zip\.net"
        match=re.search(shortening_services,self.url)
```

```
    if match:
        return -1
    else:
        return 1

# 4.Symbol@
def having_At_Symbol(self):
    #print("at")
    if "@" in self.url:
        return -1
    else:
        return 1

# 5.Redirecting//
def double_slash_redirecting(self):
    #print("//")
    pos = self.url.rfind('//')
    if pos > 6:
        if pos > 7:
            return -1
        else:
            return 1
    else:
        return 1

# 6.prefixSuffix
def Prefix_Suffix(self):
    #print("prefix")
    if '-' in urlparse(self.url).netloc:
        return -1
    else:
        return 1

#HTTPS token
def HTTPS_token(self):
    #print("https")
    domain = urlparse(self.url).netloc
    if 'https' in domain:
        return -1
    else:
        return 1

def on_mouseover(self):
    #print("mouse")
    try:
        if re.findall("", self.response.text):
```

```
                return -1
            else:
                return 1
        except:
             return -1

    def RightClick(self):
        #print("right")
        if self.response == "":
            return -1
        else:
            if re.findall(r"event.button ?== ?2", self.response.text):
                return 1
            else:
                return -1

    # 11. UsingPopupWindow
    def popUpWidnow(self):
        #print("popup")
        try:
            if re.findall(r"alert\(", self.response.text):
                return 1
            else:
                return -1
        except:
             return -1

    # 12. IframeRedirection
    def Iframe(self):
        #print("iframe")
        try:
            if re.findall(r"[<iframe>|<frameBorder>]", self.response.text):
                return 1
            else:
                return -1
        except:
             return -1

    # 13.Survival time of domain: The difference between termination time
and creation time (Domain_Age)
    def age_of_domain(self):
        #print("age")
        creation_date = self.domain_name.creation_date
        expiration_date = self.domain_name.expiration_date
        if (isinstance(creation_date,str) or
isinstance(expiration_date,str)):
```

```
            try:
                creation_date = datetime.strptime(creation_date,'%Y-%m-%d')
                expiration_date = datetime.strptime(expiration_date,"%Y-%m-
%d")
            except:
                return -1
        if ((expiration_date is None) or (creation_date is None)):
            return -1
        elif ((type(expiration_date) is list) or (type(creation_date) is
list)):
            return -1
        else:
            ageofdomain = abs((expiration_date - creation_date).days)
            if ((ageofdomain/30) < 6):
                return -1
            else:
                return 1

    def getFeaturesList(self):
        print(self.features)
        return self.features
```