



## Lab 3: Getting the stargazers of a GitHub Repository and create a network

NAME : Suriya S

225229140

### Step-1: Install PyGithub requests and get information about your Github Profile

```
In [1]: import requests
        from pprint import pprint
        username = 'suriya68'
        url = f'https://api.github.com/users/{username}'
        user_data = requests.get(url).json()
        pprint(user_data)
```

```
{'avatar_url': 'https://avatars.githubusercontent.com/u/127677511?v=4',
  'bio': None,
  'blog': '',
  'company': None,
  'created_at': '2023-03-12T15:19:22Z',
  'email': None,
  'events_url': 'https://api.github.com/users/suriya68/events{/privacy}',
  'followers': 0,
  'followers_url': 'https://api.github.com/users/suriya68/followers',
  'following': 0,
  'following_url': 'https://api.github.com/users/suriya68/following{/other_user}',
  'gists_url': 'https://api.github.com/users/suriya68/gists{/gist_id}',
  'gravatar_id': '',
  'hireable': None,
  'html_url': 'https://github.com/suriya68',
  'id': 127677511,
  'location': None,
  'login': 'suriya68',
  'name': None,
  'node_id': 'U_kgDOB5w0Rw',
  'organizations_url': 'https://api.github.com/users/suriya68/orgs',
  'public_gists': 0,
  'public_repos': 2,
  'received_events_url': 'https://api.github.com/users/suriya68/received_events',
  'repos_url': 'https://api.github.com/users/suriya68/repos',
  'site_admin': False,
  'starred_url': 'https://api.github.com/users/suriya68/starred{/owner}/{/repo}',
  'subscriptions_url': 'https://api.github.com/users/suriya68/subscriptions',
  'twitter_username': None,
  'type': 'User',
  'updated_at': '2023-07-20T03:32:07Z',
  'url': 'https://api.github.com/users/suriya68'}
```

## Step-2: Getting Public repositories of a user

```
In [11]: !pip install pygithub
```

```
Requirement already satisfied: pygithub in c:\users\sweth\downloads\nlp\lib\site-packages (1.59.0)  
Requirement already satisfied: requests>=2.14.0 in c:\users\sweth\downloads\nlp\lib\site-packages (from pygithub) (2.28.1)  
Requirement already satisfied: pynacl>=1.4.0 in c:\users\sweth\downloads\nlp\lib\site-packages (from pygithub) (1.5.0)  
Requirement already satisfied: deprecated in c:\users\sweth\downloads\nlp\lib\site-packages (from pygithub) (1.2.14)  
Requirement already satisfied: pyjwt[crypto]>=2.4.0 in c:\users\sweth\downloads\nlp\lib\site-packages (from pygithub) (2.4.0)  
Requirement already satisfied: cryptography>=3.3.1 in c:\users\sweth\downloads\nlp\lib\site-packages (from pyjwt[crypto]>=2.4.0->pygithub) (37.0.1)  
Requirement already satisfied: cffi>=1.4.1 in c:\users\sweth\downloads\nlp\lib\site-packages (from pynacl>=1.4.0->pygithub) (1.15.1)  
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\sweth\downloads\nlp\lib\site-packages (from requests>=2.14.0->pygithub) (2.0.4)  
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\sweth\downloads\nlp\lib\site-packages (from requests>=2.14.0->pygithub) (1.26.11)  
Requirement already satisfied: certifi>=2017.4.17 in c:\users\sweth\downloads\nlp\lib\site-packages (from requests>=2.14.0->pygithub) (2022.9.14)  
Requirement already satisfied: idna<4,>=2.5 in c:\users\sweth\downloads\nlp\lib\site-packages (from requests>=2.14.0->pygithub) (3.3)  
Requirement already satisfied: wrapt<2,>=1.10 in c:\users\sweth\downloads\nlp\lib\site-packages (from deprecated->pygithub) (1.14.1)  
Requirement already satisfied: pycparser in c:\users\sweth\downloads\nlp\lib\site-packages (from cffi>=1.4.1->pynacl>=1.4.0->pygithub) (2.21)
```

```
In [2]: import base64
        from github import Github
        from pprint import pprint
        username="suriya68"
        g=Github()
        user=g.get_user(username)
        for repo in user.get_repos():
            print(repo)
```

```
Repository(full_name="suriya68/CAR-PRICE-PREDICTION")
Repository(full_name="suriya68/credit-card-fraud-detection")
```

### Step-3: Querying for stargazers of a particular repository

```
In [3]: from github import Github
        ACCESS_TOKEN="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"
        USER="ptwobrussell"
        REPO="Mining-the-Social-Web"
        #REPO="Mining-the-Social-Web-2nd-Edition"
        client=Github(ACCESS_TOKEN, per_page=100)
        user=client.get_user(USER)
        repo=user.get_repo(REPO)
        stargazers=[s for s in repo.get_stargazers()]
        print("Number of stargazers", len(stargazers))
```

```
Number of stargazers 1210
```

## Step-4: Constructing an ego graph of a repository and its stargazers

```
In [4]: import networkx as nx
g=nx.DiGraph()
g.add_node(repo.name+"(repo)",type='repo',lang=repo.language,owner=user.login)
for sg in stargazers:
    g.add_node(sg.login+"(user)",type='user')
    g.add_edge(sg.login+"(user)",repo.name+"(repo)",type='gazes')
```

## Step-5: Perform handy graph operations

```
In [5]: print(nx.info(g))
print(g.nodes['Mining-the-Social-Web(repo)'])
print(g.nodes['ptwobrussell(user)'])
print(g['ptwobrussell(user)']['Mining-the-Social-Web(repo)'])
print(g['ptwobrussell(user)'])
print(g['Mining-the-Social-Web(repo)'])
print(g.in_edges(['ptwobrussell(user)']))
print(g.out_edges(['ptwobrussell(user)']))
print(g.in_edges(['Mining-the-Social-Web(repo)']))
print(g.out_edges(['Mining-the-Social-Web(repo)']))
```

DiGraph with 1211 nodes and 1210 edges

{'type': 'repo', 'lang': 'JavaScript', 'owner': 'ptwobrussell'}

{'type': 'user'}

{'type': 'gazes'}

{'Mining-the-Social-Web(repo)': {'type': 'gazes'}}

{}

[]

[('ptwobrussell(user)', 'Mining-the-Social-Web(repo)')]

[('rdempsey(user)', 'Mining-the-Social-Web(repo)'), ('prb(user)', 'Mining-the-Social-Web(repo)'), ('mcro  
ydon(user)', 'Mining-the-Social-Web(repo)'), ('twleung(user)', 'Mining-the-Social-Web(repo)'), ('kevinch  
iu(user)', 'Mining-the-Social-Web(repo)'), ('nikolay(user)', 'Mining-the-Social-Web(repo)'), ('tswicegoo  
d(user)', 'Mining-the-Social-Web(repo)'), ('ngpestelos(user)', 'Mining-the-Social-Web(repo)'), ('darron  
(user)', 'Mining-the-Social-Web(repo)'), ('brunojm(user)', 'Mining-the-Social-Web(repo)'), ('rgaidot(use  
r)', 'Mining-the-Social-Web(repo)'), ('openweb(user)', 'Mining-the-Social-Web(repo)'), ('shanlalit(use  
r)', 'Mining-the-Social-Web(repo)'), ('hoffmann(user)', 'Mining-the-Social-Web(repo)'), ('nacht(user)',  
'Mining-the-Social-Web(repo)'), ('hectoregm(user)', 'Mining-the-Social-Web(repo)'), ('tzuryby(user)', 'M  
ining-the-Social-Web(repo)'), ('marksands(user)', 'Mining-the-Social-Web(repo)'), ('wbzyl(user)', 'Minin  
g-the-Social-Web(repo)'), ('sou(user)', 'Mining-the-Social-Web(repo)'), ('magnum(user)', 'Mining-the-Soc  
ial-Web(repo)'), ('suzuki(user)', 'Mining-the-Social-Web(repo)'), ('tertsch(user)', 'Mining-the-Social-W

In [ ]:

