



PLAGIARISM SCAN REPORT



0%
Plagiarised



100%
Unique

Date 2021-04-09

Words 514

Characters 7368

Content Checked For Plagiarism

```
import pandas as pd
df1 = pd.read_csv("users/followers(ap).csv")
df1_public = df1[df1['isPrivate'] == False]
df1_public['profileUrl'].head()
print(len(df1_public['profileUrl']))
followers = set(df1_public['profileUrl'])
print(len(followers))

df2 = pd.read_csv('users/following(ap).csv')
df2_public = df2[df2['isPrivate'] == False]
df2_public['profileUrl'].head()

print(len(df2_public['profileUrl']))
following = set(df2_public['profileUrl'])
print(len(following))

best_friends = following & followers
print(len(best_friends))
df_bf = pd.DataFrame(best_friends, columns = ['profileUrl'])
print(df_bf.shape)

print(df_bf.head())
df_bf.to_csv('users/best_friends.csv', index = False)

import pandas as pd
from tqdm import tqdm

df_bf = pd.read_csv('users/best_friends.csv')
print(df_bf.shape)
df_bf.head()

df_bf.isnull().sum()
bestfriends = []
for i in df_bf['profileUrl']:
    bestfriends.append(i)
len(bestfriends)
```

```

filenames = []
for i in df_bf['profileUrl']:
    filename = i[26:]+"(following).csv"
    filenames.append(filename)
len(filenames)

df_all = pd.read_csv('following/'+filenames[0])
shape = []
shape.append(df_all.shape[0])
for i in range(1,len(filenames)):
    df_temp = pd.read_csv('following/'+filenames[i])
    shape.append(df_temp.shape[0])
    df_all = df_all.append(df_temp,ignore_index=True)

print(sum(shape))
print(df_all.shape)

df_all.head()

df = df_all[['query','profileUrl']]
df.columns = ['user','profileUrl']
print(df.shape)
df.head()

df[df['user'].isnull() == True]

df.dropna(inplace = True)
df = df.reset_index(drop=True)
df.shape

df[df['user']=='false']

df = df[df['user']!='false']
df = df.reset_index(drop=True)
df.shape

print(len(set(df['profileUrl'])|set(df['user'])))

df.to_csv('datasets/followingmerged.csv',header = False, index = False)

edges = []
for i in tqdm(range(len(df['profileUrl']))):
    temp = [df['profileUrl'][i],df['user'][i]]
    #print(temp)
    if temp not in edges:
        edges.append(temp)
print(len(edges))

df[df['profileUrl']=='https://www.instagram.com/aruntrendzzz']

df_filtered_followers = pd.merge(df,df_bf,how = 'inner')
df_filtered_followers

set(df_bf['profileUrl'])-(set(df_filtered_followers['user'])|set(df_filtered_followers['profileUrl']))

```

```

print(set(df[df['user'] == 'https://www.instagram.com/_deen_rifa._']['profileUrl'])&set(df_bf['profileUrl']))
print(set(df[df['user'] == 'https://www.instagram.com/black_pearl_sri']['profileUrl'])&set(df_bf['profileUrl']))
print(set(df[df['user'] == 'https://www.instagram.com/pavithra_kumar_t']['profileUrl'])&set(df_bf['profileUrl']))
print(set(df[df['user'] == 'https://www.instagram.com/prabu414']['profileUrl'])&set(df_bf['profileUrl']))
print(set(df[df['user'] == 'https://www.instagram.com/sridhar.362']['profileUrl'])&set(df_bf['profileUrl']))

df_temp = pd.DataFrame({'user':
['https://www.instagram.com/_deen_rifa._','https://www.instagram.com/black_pearl_sri','https://www.instagram.com/pavithra

df_temp
df_filtered_followers = df_filtered_followers.append(df_temp,ignore_index=True)
df_filtered_followers.shape

len(set(df_filtered_followers['user'])|set(df_filtered_followers['profileUrl']))

df_filtered_followers.to_csv('datasets/bestfriendsgraph-following.csv',header = False, index = False)

import pandas as pd
import networkx as nx
import igraph as ig
import matplotlib.pyplot as plt
from tqdm import tqdm

df_filtered = pd.read_csv('datasets/bestfriendsgraph-following.csv',header = None)
print('shape:',df_filtered.shape)
print(df_filtered.isnull().sum())

df_filtered.head()

df_filtered.dropna(inplace=True)

edges_filtered = set()
for i in range(len(df_filtered[0])):
    edges_filtered.add((df_filtered[0][i],df_filtered[1][i]))
len(edges_filtered)

g_filtered = ig.Graph.TupleList(edges_filtered, directed = True)
ig.plot(g_filtered)

#creating networkx graph
g_nx = nx.DiGraph()
g_nx.add_edges_from(edges_filtered)
print(nx.info(g_nx))
plt.figure(figsize =(25, 25))
nx.draw_networkx(g_nx,with_labels=False)

NSCC = nx.number_strongly_connected_components(g_nx)
SCC = nx.strongly_connected_components(g_nx)
for i in SCC:
    print(i)

largest = g_filtered.clusters().giant()
print('NODES:', largest.vcount())
print('EDGES:',largest.ecount())
ig.plot(largest)

```

```

d_filtered = g_filtered.community_edge_betweenness()
ig.plot(d_filtered,orientation = 'bottom-top')

comm_filtered = d_filtered.as_clustering()
print(comm_filtered)

Q_filtered = g_filtered.modularity(comm_filtered)
print(Q_filtered)

hubs, authorities = nx.hits(g_nx, max_iter = 50, normalized = True)

h = []
for i in hubs:
    h.append([i,hubs[i]])
h.sort(reverse = True, key=lambda x:x[1])
print('HUBS:')
for i in h[:5]:
    print(i)

a = []
for i in authorities:
    a.append([i,authorities[i]])
a.sort(reverse = True, key=lambda x:x[1])
print("\nAUTHORITIES:")
for i in a[:5]:
    print(i)

pr = nx.pagerank(g_nx)
pagerank = []
for i in pr:
    pagerank.append([i,pr[i]])
pagerank.sort(reverse = True, key = lambda x:x[1])
print('PAGERANK:')
for i in pagerank[:5]:
    print(i)

indegree = nx.in_degree_centrality(g_nx)
indegree_centrality = []
for i in indegree:
    indegree_centrality.append([i,indegree[i]])
indegree_centrality.sort(reverse = True, key = lambda x:x[1])
for i in indegree_centrality[:5]:
    print(i)

outdegree = nx.out_degree_centrality(g_nx)
outdegree_centrality = []
for i in outdegree:
    outdegree_centrality.append([i,outdegree[i]])
outdegree_centrality.sort(reverse = True, key = lambda x:x[1])
for i in outdegree_centrality[:5]:
    print(i)

betweenness = nx.betweenness_centrality(g_nx)

```

```

betweenness centrality = []
for i in betweenness:
    betweenness centrality.append([i,betweenness[i]])
betweenness centrality.sort(reverse = True, key = lambda x:x[1])

for i in betweenness centrality[:5]:
    print(i)

closeness = nx.closeness centrality(g_nx)
closeness centrality = []
for i in closeness:
    closeness centrality.append([i,closeness[i]])
closeness centrality.sort(reverse = True, key = lambda x:x[1])
for i in closeness centrality[:5]:
    print(i)

df_following_merged = pd.read_csv('datasets/followingmerged.csv',header = None)
df_following_merged.head()

fof = list(df_following_merged[1])
fof_set = set(fof)
print(len(fof),len(fof_set))

df_following = pd.read_csv('users/following(ap).csv')
df_following.head()
df_following = df_following[['profileUrl']]
df_following.columns = ['profileUrl']
df_following.head()

following_set = set(df_following['profileUrl'])
following_set.add('https://www.instagram.com/aruntrendzzz')
len(following_set)

fof_set = fof_set - following_set
len(fof_set)

dict_following = {}
for i in fof_set:
    dict_following[i] = 0

for i in fof:
    if i in dict_following:
        dict_following[i] += 1

following_freq_list = list(dict_following.items())
following_freq_list.sort(reverse = True,key = lambda x:x[1])

for i in following_freq_list[:10]:
    print(i)

```

Matched Source

