Load Package

In [1]:
```
import pandas as pd
import numpy as np
import os
import networkx as nx
from node2vec import Node2Vec
```

In [2]:
```
print (os.getcwd())
os.chdir('D:/OneDrive/ASU/2021 Spring/Applied Project/ASU_Applied_Project_2021')
print (os.getcwd())
```

```
C:\Users\Jinhang Jiang\ASU - Adidas
D:\OneDrive\ASU\2021 Spring\Applied Project\ASU_Applied_Project_2021
```

Load Data and Explore

In [3]:
```
df = pd.read_csv("Data/FullData.csv")
#df = pd.read_csv("Data/networkanalysis_cum.csv")
```

In [4]:
```
df.rename(columns = {'author':'Usernames'}, inplace = True)
```

In [5]:
```
view=df.groupby(['Celebrity','Usernames']).size().reset_index(name='Freq')
view
```

Out[5]:

	Celebrity	Usernames	Freq
0	Adriene Mishler	-en-	4
1	Adriene Mishler	Alhana69	32
2	Adriene Mishler	AutoNewsAdmin	12
3	Adriene Mishler	AutoNewspaperAdmin	12
4	Adriene Mishler	Chachabooom	6
...	...	...	...
9597	iZone	yeroc1016	2
9598	iZone	yoossi_	8
9599	iZone	yunoohaddicted	1
9600	iZone	zacchi4k	2
9601	iZone	zer0a0	2

9602 rows × 3 columns

In [6]:
```
a=["Celebrity","Usernames"]
data = view[a]
data.shape
```

Out[6]: (9602, 2)

In [7]:
```
print(*data.Celebrity.unique(),sep="\n")
```

```
Adriene Mishler
Ally Love
BTS
Bad Bunny
Beyonce
BlackPink
Chinae Alexander
GFriend
Jerry Lorenzo
Karlie Kloss
Kerwin Frost
NCT
Naeun Son
Ninjas Hyper
Pharrell Williams
Seolhyun
Solar
Yara Shahidi
Zoe Saldana
iZone
```

In [8]:
```
data.shape
```

Out[8]: (9602, 2)

In [9]:
```
print("Number of Celebrities: %0.0f" %len(data.Celebrity.unique()))
print("Number of Users: %0.0f" %len(data.Usernames.unique()))
```

```
Number of Celebrities: 20
Number of Users: 8086
```

In [10]:
```
print("The percentage of unique values: {:.2%}".format(len(data.Usernames.unique())/len(data.Usernames)))
```

```
The percentage of unique values: 84.21%
```

In [11]:
```
data.Celebrity.value_counts()
```

Out[11]:
```
NCT                1769
BTS                1047
BlackPink           768
iZone               636
GFriend             592
Bad Bunny           491
Pharrell Williams   471
Jerry Lorenzo       465
Naeun Son           437
Zoe Saldana         399
Beyonce             398
Solar               359
Ally Love           346
Ninjas Hyper        310
Karlie Kloss        270
Yara Shahidi        220
Chinae Alexander    219
Kerwin Frost        215
Seolhyun            168
Adriene Mishler      22
Name: Celebrity, dtype: int64
```

Generate Adjacency Matrix

In [12]:
```
df_merge = data.merge(data, on='Usernames')
results = pd.crosstab(df_merge.Celebrity_x, df_merge.Celebrity_y)
np.fill_diagonal(results.values, 0)
network_table=results
network_table
```

Out[12]:

Celebrity_y / Celebrity_x	Adriene Mishler	Ally Love	BTS	Bad Bunny	Beyonce	BlackPink	Chinae Alexander	GFriend	Jerry Lorenzo	Karlie Kloss	Kerwin Frost	NCT	Naeun Son	Ninjas Hyper	Pharre Willian
Adriene Mishler	0	3	3	0	3	3	3	3	3	2	0	2	2	0	
Ally Love	3	0	6	0	5	6	6	7	2	5	0	5	3	0	
BTS	3	6	0	7	11	56	6	67	3	8	0	79	37	0	
Bad Bunny	0	0	7	0	6	4	2	8	6	4	1	8	1	0	
Beyonce	3	5	11	6	0	13	7	9	4	14	0	8	4	0	
BlackPink	3	6	56	4	13	0	9	81	4	9	0	73	58	0	
Chinae Alexander	3	6	6	2	7	9	0	12	4	7	1	10	4	0	
GFriend	3	7	67	8	9	81	12	0	4	7	1	105	77	0	
Jerry Lorenzo	3	2	3	6	4	4	4	4	0	1	2	6	2	0	
Karlie Kloss	2	5	8	4	14	9	7	7	1	0	0	6	3	1	
Kerwin Frost	0	0	0	1	0	0	1	1	2	0	0	3	0	0	
NCT	2	5	79	8	8	73	10	105	6	6	3	0	63	2	
Naeun Son	2	3	37	1	4	58	4	77	2	3	0	63	0	0	
Ninjas Hyper	0	0	0	0	0	0	0	0	0	1	0	2	0	0	
Pharrell Williams	3	5	10	9	13	9	4	6	8	7	3	10	4	2	
Seolhyun	2	3	23	0	4	43	3	49	2	1	0	48	40	0	
Solar	2	4	52	2	4	55	7	73	3	6	2	93	49	0	

Celebrity_y	Adriene Mishler	Ally Love	BTS	Bad Bunny	Beyonce	BlackPink	Chinae Alexander	GFriend	Jerry Lorenzo	Karlie Kloss	Kerwin Frost	NCT	Naeun Son	Ninjas Hyper	Pharre Willian
Celebrity_x															
Yara Shahidi	4	7	11	5	14	9	11	10	7	13	1	7	3	0	
Zoe Saldana	3	5	11	6	398	13	7	9	4	14	0	8	4	0	
iZone	0	2	66	2	2	83	3	132	0	4	1	119	66	0	

Fit NetworkX

In [13]:
```
#graph=nx.from_numpy_matrix(np_matrix)
graph=nx.from_pandas_adjacency(network_table)
print(nx.info(graph))
```

```
Name:
Type: Graph
Number of nodes: 20
Number of edges: 160
Average degree:  16.0000
```

In [14]:
```
setup = Node2Vec(graph,dimensions=128, walk_length=80, num_walks=10, workers=4)
model = setup.fit(window=3, min_count=1)
```

Computing transition probabilities: 100% 20/20 [00:00<00:00, 371.38it/s]

In [21]:
```
#vocab, vectors = model.wv.key_to_index, model.wv.get_normed_vectors()
vocab, vectors = model.wv.vocab, model.wv.vectors

# get node name and embedding vector index.
name_index = np.array([(v[0], v[1].index) for v in vocab.items()]) #.index

# init dataframe using embedding vectors and set index as node name
node2vec_output = pd.DataFrame(vectors[name_index[:,1].astype(int)])
node2vec_output.index = name_index[:,0]
```

In [22]:
```
node2vec_output
```

Out[22]:

	0	1	2	3	4	5	6	7	8	9	...	118	
Yara Shahidi	0.053552	0.189677	-0.043327	0.022491	0.010450	-0.113112	0.149638	0.051164	0.016838	0.034390	...	-0.102222	0.14
Karlie Kloss	0.051904	0.177485	-0.054276	0.043238	0.024365	-0.113049	0.163114	0.057656	0.049426	0.015410	...	-0.099046	0.14
Beyonce	0.029752	0.082568	-0.090873	0.155293	0.088061	-0.089937	0.217620	0.049874	0.180861	-0.093390	...	-0.062997	0.13
Zoe Saldana	0.032947	0.069968	-0.092983	0.163731	0.090817	-0.090103	0.230716	0.058234	0.196180	-0.102949	...	-0.055807	0.14
Jerry Lorenzo	0.050763	0.190235	-0.046107	0.015539	0.011177	-0.107821	0.143505	0.051352	0.013760	0.041947	...	-0.106001	0.14
Bad Bunny	0.054179	0.184971	-0.045436	0.023568	0.016585	-0.105328	0.154039	0.052558	0.033800	0.032711	...	-0.100281	0.14
Pharrell Williams	0.051125	0.192679	-0.051619	0.024307	0.012218	-0.111619	0.149696	0.050759	0.024161	0.039750	...	-0.107991	0.14
Adriene Mishler	0.060439	0.215986	-0.037208	-0.007543	0.000579	-0.114378	0.134917	0.055638	-0.011346	0.067408	...	-0.117593	0.14
GFriend	0.064089	0.263153	-0.011491	-0.075562	-0.031373	-0.117432	0.098997	0.048920	-0.099513	0.134986	...	-0.137297	0.14
NCT	0.059169	0.242793	-0.021216	-0.059780	-0.018873	-0.105840	0.103179	0.051860	-0.066403	0.116612	...	-0.128453	0.14
Naeun Son	0.063408	0.270511	-0.016875	-0.082024	-0.035985	-0.116578	0.095683	0.055159	-0.106042	0.143063	...	-0.138281	0.14
iZone	0.070267	0.261283	-0.027577	-0.066787	-0.027232	-0.110564	0.110685	0.056290	-0.079122	0.126205	...	-0.132967	0.14
Solar	0.059244	0.255737	-0.018719	-0.060526	-0.028128	-0.118357	0.107575	0.048653	-0.077215	0.120171	...	-0.130219	0.14
BTS	0.065392	0.255524	-0.016285	-0.068807	-0.026739	-0.115633	0.098802	0.050647	-0.089366	0.129963	...	-0.132988	0.14
BlackPink	0.057522	0.239329	-0.026674	-0.044764	-0.020123	-0.114947	0.112668	0.047096	-0.062766	0.100850	...	-0.129076	0.13
Seolhyun	0.062871	0.249710	-0.015839	-0.066646	-0.031080	-0.115572	0.096846	0.050296	-0.088199	0.122852	...	-0.133226	0.14
Chinae Alexander	0.053750	0.203156	-0.041079	0.000885	0.009370	-0.114402	0.141757	0.055749	-0.001031	0.053910	...	-0.117284	0.14
Ninjas Hyper	0.054550	0.196651	-0.037055	0.006643	0.003169	-0.112036	0.142111	0.047020	-0.000995	0.049646	...	-0.111295	0.13
Ally Love	0.048764	0.187265	-0.048454	0.021132	0.018969	-0.108276	0.146847	0.049539	0.020219	0.039426	...	-0.103673	0.14

	0	1	2	3	4	5	6	7	8	9	...	118	
Kerwin Frost	0.054078	0.221494	-0.033171	-0.016507	-0.008847	-0.117103	0.132296	0.048451	-0.028856	0.078175	...	-0.121301	0.14

20 rows × 128 columns

In [23]:
```
node2vec_output.shape
```

Out[23]: (20, 128)

In [24]:
```
node2vec_output.to_csv("Data/node2vec_mar29.csv")
```

In [ ]:
```

```