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/networkanalysis3.csv")
#df = pd.read_csv("Data/networkanalysis_cum.csv")
```

In [4]:
```
view=df.groupby(['Celebrity','Usernames']).size().reset_index(name='Freq')
view
```

Out[4]:

	Celebrity	Usernames	Freq
0	ALLY LOVE	-en-	12
1	ALLY LOVE	-lastmanstan-	3
2	ALLY LOVE	0LoveRainbow0	6
3	ALLY LOVE	444strega	6
4	ALLY LOVE	AHealthyDoseofFran	3
...	...	...	...
3625	naeun	whoatethechips	3
3626	naeun	yoossi_	9
3627	naeun	yukyakyuk	3
3628	naeun	yulje	3
3629	naeun	zhuzhuH	108

3630 rows × 3 columns

In [5]:
```
a=["Celebrity","Usernames"]
data = view[a]
data.shape
```

Out[5]: (3630, 2)

In [6]:
```
print(*data.Celebrity.unique(),sep="\n")
```

```
ALLY LOVE
Adriene Mishler
BadBunny
Beyonce
BlackPink
CHINAE ALEXANDER
JERRY LORENZO
Karlie Kloss
Kerwin Frost
NinjasHyper
Pharrell Williams
Yara Sayeh Shahidi
Zoe Saldana
naeun
```

In [7]:
```
data.shape
```

Out[7]: (3630, 2)

In [8]:
```
print("Number of Celebrities: %0.0f" %len(data.Celebrity.unique()))
print("Number of Users: %0.0f" %len(data.Usernames.unique()))
```

```
Number of Celebrities: 14
Number of Users: 3425
```

In [9]:
```
print("The percentage of unique values: {:.2%}".format(len(data.Usernames.unique())/len(data.Usernames)))
```

```
The percentage of unique values: 94.35%
```

In [10]:
```
data.Celebrity.value_counts()
```

Out[10]:
```
Pharrell Williams     480
Zoe Saldana           389
Beyonce               365
BadBunny              327
NinjasHyper           305
Karlie Kloss          277
BlackPink             265
CHINAE ALEXANDER      227
Yara Sayeh Shahidi    215
Kerwin Frost          213
ALLY LOVE             192
JERRY LORENZO         182
naeun                 171
Adriene Mishler        22
Name: Celebrity, dtype: int64
```

Generate Adjacency Matrix

In [15]:
```
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[15]:

Celebrity_x \ Celebrity_y	ALLY LOVE	Adriene Mishler	BadBunny	Beyonce	BlackPink	CHINAE ALEXANDER	JERRY LORENZO	Karlie Kloss	Kerwin Frost	NinjasHyper	Pharrell Williams	Yara Sayeh Shahidi	S
ALLY LOVE	0	3	0	5	0	6	2	5	0	0	5	6	
Adriene Mishler	3	0	0	4	0	3	3	2	0	0	3	4	
BadBunny	0	0	0	2	0	2	1	3	1	0	7	4	
Beyonce	5	4	2	0	4	4	3	14	0	0	11	14	
BlackPink	0	0	0	4	0	1	0	1	0	0	1	0	
CHINAE ALEXANDER	6	3	2	4	1	0	2	8	2	0	4	11	
JERRY LORENZO	2	3	1	3	0	2	0	1	1	0	3	5	
Karlie Kloss	5	2	3	14	1	8	1	0	0	1	8	14	
Kerwin Frost	0	0	1	0	0	2	1	0	0	0	3	1	
NinjasHyper	0	0	0	0	0	0	0	1	0	0	2	0	
Pharrell Williams	5	3	7	11	1	4	3	8	3	2	0	20	
Yara Sayeh Shahidi	6	4	4	14	0	11	5	14	1	0	20	0	
Zoe Saldana	3	1	0	17	1	7	1	34	0	0	12	21	
naeun	2	2	0	3	9	3	2	0	0	0	2	2	

Fit NetworkX

In [22]:
```
#graph=nx.from_numpy_matrix(np_matrix)
graph=nx.from_pandas_adjacency(network_table)
print(nx.info(graph))
```

```
Name:
Type: Graph
Number of nodes: 14
Number of edges: 62
Average degree:   8.8571
```

In [41]:
```
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%                    14/14 [00:00<00:00, 250.64it/s]
```

In [44]:
```
vocab, vectors = model.wv.key_to_index, model.wv.get_normed_vectors()

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

# 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 [45]:
```
node2vec_output
```

Out[45]:

	0	1	2	3	4	5	6	7	8	9	...	118	
Yara Sayeh Shahidi	-0.003286	-0.070792	0.167736	-0.039911	-0.063836	-0.176614	-0.068094	0.073181	-0.006206	0.007760	...	0.006158	0.062
Zoe Saldana	-0.005929	-0.066382	0.174871	-0.044243	-0.060025	-0.175628	-0.069812	0.065397	-0.001876	0.010284	...	0.008574	0.056
Karlie Kloss	-0.007259	-0.069894	0.159697	-0.046346	-0.055861	-0.174254	-0.064817	0.066350	-0.007426	0.017489	...	0.006919	0.057
Beyonce	-0.007066	-0.074266	0.170209	-0.038664	-0.061298	-0.178285	-0.075713	0.063692	-0.007694	0.006151	...	0.006051	0.053
Pharrell Williams	-0.010451	-0.077603	0.165588	-0.053034	-0.062961	-0.175358	-0.073871	0.063925	-0.008048	0.008029	...	0.012530	0.067
CHINAE ALEXANDER	-0.000960	-0.074873	0.174687	-0.049085	-0.057355	-0.167624	-0.077288	0.059697	-0.007173	0.007589	...	0.007286	0.065
ALLY LOVE	-0.006096	-0.074635	0.169478	-0.046892	-0.055666	-0.172641	-0.075114	0.067598	0.003490	0.006106	...	0.009459	0.059
Adriene Mishler	-0.001750	-0.068323	0.174162	-0.050409	-0.061787	-0.178034	-0.083894	0.073024	-0.001600	0.002224	...	0.012437	0.067
naeun	-0.009100	-0.068870	0.164279	-0.050991	-0.053158	-0.171834	-0.079704	0.070538	0.008204	0.017245	...	0.010918	0.072
JERRY LORENZO	0.002592	-0.067762	0.160887	-0.044696	-0.058286	-0.170634	-0.077246	0.068003	-0.000681	0.010006	...	0.014246	0.069
BadBunny	-0.010707	-0.067867	0.165615	-0.037078	-0.063117	-0.174590	-0.067844	0.065353	-0.007149	0.010220	...	0.005186	0.070
BlackPink	0.003623	-0.078971	0.170526	-0.057810	-0.061833	-0.179775	-0.081970	0.075963	0.010587	0.008379	...	0.008818	0.073
Kerwin Frost	0.002061	-0.076593	0.160140	-0.038977	-0.053158	-0.177616	-0.079682	0.074541	-0.000138	0.014703	...	0.011344	0.067
NinjasHyper	-0.004717	-0.069545	0.163903	-0.040718	-0.054932	-0.170066	-0.069856	0.073028	-0.002397	0.007002	...	0.012723	0.064

14 rows × 128 columns

In [46]:
```
node2vec_output.shape
```

Out[46]: (14, 128)

In [47]:
```
node2vec_output.to_csv("Data/node2vec_mar27.csv")
```

In [ ]:
```

```