Load Package

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

In [7]:
```
print (os.getcwd())
os.chdir('D:/OneDrive/ASU/2021 Spring/Applied Project/ASU_Applied_Project_2021/Data')
print (os.getcwd())
```

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

Load Data and Explore

In [9]:
```
data = pd.read_csv("networkanalysis1.csv")
```

In [10]:
```
data.head(6)
```

Out[10]:

	Celebrity	Usernames
0	Kerwin Frost	jamiedevlin999
1	Kerwin Frost	neighborgang
2	Kerwin Frost	jothvm
3	Kerwin Frost	nostylist2900
4	Kerwin Frost	New_Age_Dryer
5	Kerwin Frost	janspirit

In [11]:
```
print(*data.Celebrity.unique(),sep="\n")
```

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

In [6]:
```
data.shape
```

Out[6]: (9764, 2)

In [7]:
```
print("Number of Celebrities: %0.0f" %len(data.Celebrity.unique()))
print("Number of Users: %0.0f" %len(data.Usernames.unique()))
```

```
Number of Celebrities: 12
Number of Users: 1505
```

In [8]:
```
print("The percentage of unique values: %0.4f" %(len(data.Usernames.unique())/len(data.Usernames)))
```

```
The percentage of unique values: 0.1541
```

In [14]:
```
data.Celebrity.value_counts()
```

Out[14]:
```
BadBunny             1505
Ninjas Hyper         1326
BlackPink            1190
Adriene Mishler      1035
Pharrell Williams    1023
naeun                 860
Yara Shahidi          804
karlie kloss          698
ZoeSaldana            605
beyonce               428
James Bond            260
kerwinfrost            30
Name: Celebrity, dtype: int64
```

In [124]:
```
matrix["BlackPink"].where((matrix["BlackPink"]==1)&(matrix["Pharrell Williams"]==1)&(matrix["karlie kloss"]==1)).dropna()
```

Out[124]:
```
Usernames
12e64m3                1.0
12jonboy12             1.0
18hundreds             1.0
1mi_K                  1.0
20stocktwits____       1.0
                      ...
youlostthewarwehrs     1.0
yourlocalmagicalgirl   1.0
zagreux                1.0
zanzibar_eggs          1.0
zeaNN1                 1.0
Name: BlackPink, Length: 698, dtype: float64
```

Generate Adjacency Matrix

In [71]:
```
#Create matrix
matrix = pd.get_dummies(data.set_index('Usernames')['Celebrity'].astype(str)).max(level=0).sort_index()
```

In [72]:
```
matrix.iloc[0:5,0:5]
```

Out[72]:

Usernames	Adriene Mishler	BadBunny	BlackPink	James Bond	Ninjas Hyper
-Fashion-News-	1	1	1	0	1
-SODANK-	0	1	0	0	0
-Sportswear-	1	1	1	0	1
-en-	1	1	1	0	1
12e64m3	1	1	1	0	1

In [73]:
```
#matrix.to_csv("dummy_matrix.csv")
```

In [38]:
```
cel_matrix = np.asmatrix(matrix)
cel_matrix_transpose = cel_matrix.transpose()
final_matrix = cel_matrix_transpose.dot(cel_matrix)

network_table = pd.DataFrame(final_matrix)
print(network_table.iloc[0:5,0:5])
print(network_table.shape)
```

```
     0    1    2  3    4
0   11   11   11  4   11
1   11  225  166  4   46
2   11  166  166  4  166
3    4    4    4  4    4
4   11   46  166  4   46
(12, 12)
```

In [39]:
```
## append index name
Celebrity = list(data.Celebrity.unique())
Celebrity.sort()

network_table.index = Celebrity
network_table.columns = Celebrity
network_table
```

Out[39]:

	Adriene Mishler	BadBunny	BlackPink	James Bond	Ninjas Hyper	Pharrell Williams	Yara Shahidi	ZoeSaldana	beyonce	karlie kloss	kerwinfrost	naeun
Adriene Mishler	11	11	11	4	11	255	36	93	172	186	30	92
BadBunny	11	225	166	4	46	255	36	93	172	186	30	92
BlackPink	11	166	166	4	166	255	36	93	172	186	30	92
James Bond	4	4	4	4	4	4	4	4	4	4	30	4
Ninjas Hyper	11	46	166	4	46	255	36	93	172	186	30	92
Pharrell Williams	255	255	255	4	255	255	36	93	172	186	30	92
Yara Shahidi	36	36	36	4	36	36	36	93	172	186	30	36
ZoeSaldana	93	93	93	4	93	93	93	93	172	93	30	93
beyonce	172	172	172	4	172	172	172	172	172	172	30	172
karlie kloss	186	186	186	4	186	186	186	93	172	186	30	186
kerwinfrost	30	30	30	30	30	30	30	30	30	30	30	30
naeun	92	92	92	4	92	92	36	93	172	186	30	92

In [52]:
```
#matrix.to_csv('frequency_matrix1.csv')
#network_table.to_csv('network_table1.csv')
```

Fit NetworkX

In [57]:
```
#network_table = pd.read_csv('network_table1.csv', index_col=0)
```

In [58]:
```
network_table.head(3)
```

Out[58]:

	Adriene Mishler	BadBunny	BlackPink	James Bond	Ninjas Hyper	Pharrell Williams	Yara Shahidi	ZoeSaldana	beyonce	karlie kloss	kerwinfrost	naeun
Adriene Mishler	11	11	11	4	11	255	36	93	172	186	30	92
BadBunny	11	225	166	4	46	255	36	93	172	186	30	92
BlackPink	11	166	166	4	166	255	36	93	172	186	30	92

In [63]:
```
#pip install --upgrade networkx
```

```
Collecting networkx
  Downloading https://files.pythonhosted.org/packages/9b/cd/dc52755d30ba41c60243235460961fc28022e5b6731f16c268667625baea/networkx-2.5-py3-none-any.whl (https://files.pythonhosted.org/packages/9b/cd/dc52755d30ba41c60243235460961fc28022e5b6731f16c268667625baea/networkx-2.5-py3-none-any.whl) (1.6MB)
Requirement already satisfied, skipping upgrade: decorator>=4.3.0 in d:\1dataanalytics\python\anaconda3\lib\site-packages (from networkx) (4.4.0)
Installing collected packages: networkx
  Found existing installation: networkx 2.3
    Uninstalling networkx-2.3:
      Successfully uninstalled networkx-2.3
Successfully installed networkx-2.5
Note: you may need to restart the kernel to use updated packages.
```

In [68]:
```
graph=nx.from_numpy_matrix(np.matrix(network_table))
```

In [74]:
```
node2vec = Node2Vec(graph, dimensions=25, walk_length=10, num_walks=300, workers=4)
model = node2vec.fit(window=10, min_count=1)
```

```
Computing transition probabilities: 100%|████████████████████████████████████████████████████| 12/12 [00:00<00:00, 2005.72it/s]
```

In [82]:
```
model.wv.save_word2vec_format('embedding')
```

In [76]:
```
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()])

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

In [92]:
```
df = df.sort_index(ascending=True)
df
```

Out[92]:

	0	1	2	3	4	5	6	7	8	9	...	15	16	
0	-0.147744	-0.044176	0.065672	-0.083068	-0.294284	0.342981	0.184983	-0.237668	0.223541	0.081853	...	-0.161684	0.379380	-0.00
1	-0.163335	-0.066436	0.052841	-0.114052	-0.319886	0.359776	0.173201	-0.256655	0.205670	0.079700	...	-0.160427	0.385589	0.01
10	-0.168566	-0.035804	0.098199	-0.129922	-0.360240	0.346373	0.188753	-0.252003	0.193143	0.093038	...	-0.155007	0.396961	0.01
11	-0.149148	-0.059372	0.066437	-0.085810	-0.300096	0.354894	0.169167	-0.266838	0.240197	0.083486	...	-0.180132	0.362748	-0.00
2	-0.159323	-0.042151	0.084771	-0.099135	-0.318472	0.362776	0.176587	-0.273824	0.238587	0.097037	...	-0.156165	0.377907	0.02
3	-0.180591	-0.036817	0.059722	-0.131494	-0.343669	0.364086	0.160926	-0.249468	0.192522	0.103723	...	-0.163323	0.381578	0.01
4	-0.142373	-0.066504	0.063518	-0.087927	-0.326281	0.340159	0.167958	-0.248896	0.206247	0.102187	...	-0.165456	0.372962	-0.00
5	-0.137536	-0.075668	0.055149	-0.105118	-0.317359	0.348143	0.163557	-0.249705	0.239935	0.107772	...	-0.161702	0.365878	0.01
6	-0.120312	-0.063577	0.092431	-0.119435	-0.311107	0.323934	0.184652	-0.245627	0.242171	0.104827	...	-0.135402	0.354035	0.00
7	-0.147220	-0.047069	0.068712	-0.114014	-0.333816	0.333292	0.183427	-0.263661	0.252079	0.091213	...	-0.145265	0.371257	-0.01
8	-0.167371	-0.071309	0.053822	-0.086711	-0.287004	0.363799	0.163352	-0.263727	0.207472	0.091727	...	-0.186760	0.390325	0.00
9	-0.168127	-0.057711	0.071768	-0.089614	-0.323656	0.364120	0.159911	-0.267953	0.231975	0.111026	...	-0.175510	0.380983	0.01

12 rows × 25 columns

In [95]:
```
celebrity_names = ['Adriene Mishler','BadBunny','kerwinfrost','naeun',
 'BlackPink','James Bond','Ninjas Hyper','Pharrell Williams','Yara Shahidi',
 'ZoeSaldana','beyonce','karlie kloss']
```

In [96]:
```
df.index = celebrity_names
```

In [97]:
```
df
```

Out[97]:

	0	1	2	3	4	5	6	7	8	9	...	15	
Adriene Mishler	-0.147744	-0.044176	0.065672	-0.083068	-0.294284	0.342981	0.184983	-0.237668	0.223541	0.081853	...	-0.161684	0.3793
BadBunny	-0.163335	-0.066436	0.052841	-0.114052	-0.319886	0.359776	0.173201	-0.256655	0.205670	0.079700	...	-0.160427	0.3855
kerwinfrost	-0.168566	-0.035804	0.098199	-0.129922	-0.360240	0.346373	0.188753	-0.252003	0.193143	0.093038	...	-0.155007	0.3969
naeun	-0.149148	-0.059372	0.066437	-0.085810	-0.300096	0.354894	0.169167	-0.266838	0.240197	0.083486	...	-0.180132	0.3627
BlackPink	-0.159323	-0.042151	0.084771	-0.099135	-0.318472	0.362776	0.176587	-0.273824	0.238587	0.097037	...	-0.156165	0.3779
James Bond	-0.180591	-0.036817	0.059722	-0.131494	-0.343669	0.364086	0.160926	-0.249468	0.192522	0.103723	...	-0.163323	0.3815
Ninjas Hyper	-0.142373	-0.066504	0.063518	-0.087927	-0.326281	0.340159	0.167958	-0.248896	0.206247	0.102187	...	-0.165456	0.3729
Pharrell Williams	-0.137536	-0.075668	0.055149	-0.105118	-0.317359	0.348143	0.163557	-0.249705	0.239935	0.107772	...	-0.161702	0.3658
Yara Shahidi	-0.120312	-0.063577	0.092431	-0.119435	-0.311107	0.323934	0.184652	-0.245627	0.242171	0.104827	...	-0.135402	0.3540
ZoeSaldana	-0.147220	-0.047069	0.068712	-0.114014	-0.333816	0.333292	0.183427	-0.263661	0.252079	0.091213	...	-0.145265	0.3712
beyonce	-0.167371	-0.071309	0.053822	-0.086711	-0.287004	0.363799	0.163352	-0.263727	0.207472	0.091727	...	-0.186760	0.3903
karlie kloss	-0.168127	-0.057711	0.071768	-0.089614	-0.323656	0.364120	0.159911	-0.267953	0.231975	0.111026	...	-0.175510	0.3809

12 rows × 25 columns

In [99]:
```
#df.to_csv("embedding1.csv")
```