Assignment 3

Professor: Ming Jiang

Student Name: Deepak Rajput **Program**: MS Applied Data Science

Do network analysis and answer questions.

Q1: Compare the structure of two networks using graph-level measurements

Answer: Graph-level measurements of both networks

a) Marriage network details:

Number of nodes: 16

Network density: 0.16667

Average clustering: 0.15

Counts of smaller graphs embedded in the network: 18

Structure analysis:

The marriage graph shows you there is a single largest component of connected nodes is Medici and other important nodes with just three or four connections around the edges. The number of connections or degree of network as the size of nodes, the visualization also shows that there are a few node which are more powerful and tied network more firmly.

Network density, is simply the ratio of actual edges in the network to all possible edges in the network. Network density gives you a quick sense of how closely knit your network is and its density value 0.1667. Only one node in network is not having connection with others.

[8]:
```
import csv
from networkx.algorithms import community
# Getting number of nodes of marriage_net
with open('./padgett_agent_agent[PADGM].csv', 'r') as marr_nodecsv:
    marr_nodereader = csv.reader(marr_nodecsv)
    marr_net_nodes = [n for n in marr_nodereader][1:]
    marr_node_names = [n[0] for n in marr_net_nodes]
    print(len(marr_node_names))
```

```
16
```

[11]:
```
density = nx.density(marr_net)
print("Network density:", density)
```

```
Network density: 0.16666666666666666
```

[13]:
```
marr_net_connection = nx.is_connected(marr_net)
print(marr_net_connection)
marr_net_degree = nx.average_clustering(marr_net)
print(marr_net_degree)
```

```
False
0.15
```

b) Business network details:
Number of nodes: 16
Network density: 0.125
Average clustering: 0.2979
Counts of smaller graphs embedded in the network: 14
Structure analysis:
The business graph shows you there is a single largest component of connected nodes is Medici and other nodes also having significant relation with more than three or four connections around the edges. The number of connections or degree of network as the size of nodes, the visualization also shows that there is only Medici node which is more powerful and tied network more firmly.
Network density, is showing that business network is loosely connected and 5 nodes out of 16 having no connection with the others.

[16]:
```
with open('./padgett_agent_agent[PADGB].csv', 'r') as busi_nodecsv:
    busi_nodereader = csv.reader(busi_nodecsv)
    busi_net_nodes = [n for n in busi_nodereader][1:]
    busi_node_names = [n[0] for n in busi_net_nodes]
    print(len(busi_node_names))
```

```
16
```

[17]:
```
busi_density = nx.density(business_net)
print("Network density:", busi_density)
```

```
Network density: 0.125
```

[20]:
```
busi_net_connection = nx.is_connected(business_net)
print(busi_net_connection)
busi_net_degree = nx.average_clustering(business_net)
print(busi_net_degree)
```

```
False
0.2979166666666666
```

Q2: What is the meaning of isolated nodes in each network?
Answer:
Marriage network has one isolated node i.e., PUCCI and Business network has 5 isolated nodes which are ACCIAIUOL, ALBIZZI, PUCCI, RIDOLFI, STROZZI
Isolated nodes indicated that they have no edges with other nodes in the network so not able to form any connection, relation or reachability with other nodes.

[22]:
```
marr_i_node = nx.Graph(marr_net)
marr_isolated_node = [point for point in marr_i_node.nodes() if marr_i_node.degree
print(marr_isolated_node)
```

```
['PUCCI']
```

[23]:
```
busi_i_node = nx.Graph(business_net)
busi_isolated_node = [point for point in busi_i_node.nodes() if busi_i_node.degree
print(busi_isolated_node)
```

```
['ACCIAIUOL', 'ALBIZZI', 'PUCCI', 'RIDOLFI', 'STROZZI']
```

Q3: Who is the most influential node in the Padgett Florentine Families marriage-level network?

i) Medici has highest degree is 6

ii) Medici has highest betweenness among all nodes 0.45 whereas 5 nodes out of 16 having 0 betweenness

iii) Medici has highest closeness

iv) Medici has highest eigen vector values

From above, we can drive that, Medici is the most powerful node in the marriage network

]:
```
from operator import itemgetter
degree_dict = dict(marr_net.degree(marr_net.nodes()))
nx.set_node_attributes(marr_net, degree_dict, 'degree')
sorted_degree = sorted(degree_dict.items(), key=itemgetter(1), reverse=True)
print("All 16 nodes by degree:")
for d in sorted_degree[:16]:
    print(d)
```

```
All 16 nodes by degree:
('MEDICI', 6)
('GUADAGNI', 4)
('STROZZI', 4)
('ALBIZZI', 3)
('BISCHERI', 3)
('CASTELLAN', 3)
('PERUZZI', 3)
('RIDOLFI', 3)
('TORNABUON', 3)
('BARBADORI', 2)
('SALVIATI', 2)
('ACCIAIUOL', 1)
('GINORI', 1)
('LAMBERTES', 1)
('PAZZI', 1)
('PUCCI', 0)
```

2]:
```
m_betweenness_dict = nx.betweenness_centrality(marr_net)
m_betweenness_dict
```

2]:
```
{'ACCIAIUOL': 0.0,
 'ALBIZZI': 0.18412698412698414,
 'BARBADORI': 0.08095238095238096,
 'BISCHERI': 0.09047619047619049,
 'CASTELLAN': 0.04761904761904762,
 'GINORI': 0.0,
 'GUADAGNI': 0.22063492063492063,
 'LAMBERTES': 0.0,
 'MEDICI': 0.45238095238095244,
 'PAZZI': 0.0,
 'PERUZZI': 0.01904761904761905,
 'PUCCI': 0.0,
 'RIDOLFI': 0.09841269841269841,
 'SALVIATI': 0.12380952380952381,
 'STROZZI': 0.0888888888888889,
 'TORNABUON': 0.07936507936507937}
```

5]:
```
m_closeness_dict = nx.closeness_centrality(marr_net)
m_cl = sorted(m_closeness_dict.items(), key=itemgetter(1))
print(m_cl)
```

```
[('PUCCI', 0.0), ('PAZZI', 0.26666666666666666), ('LAMBERTES', 0.3038759689922481), ('GINORI', 0.3111111111111111), ('ACCIAIUOL', 0.343859649122807), ('PERUZZI', 0.343859649122807), ('CASTELLAN', 0.362962962962963), ('SALVIATI', 0.362962962962963), ('BISCHERI', 0.37333333333333335), ('BARBADORI', 0.4083333333333333), ('STROZZI', 0.4083333333333333), ('GUADAGNI', 0.4355555555555556), ('ALBIZZI', 0.4505747126436782), ('TORNABUON', 0.4505747126436782), ('RIDOLFI', 0.4666666666666667), ('MEDICI', 0.5226666666666667)]
```

9]:
```
m_eigenvector_dict = nx.eigenvector_centrality(marr_net)
m_eig = sorted(m_eigenvector_dict.items(), key = itemgetter(1))
print(m_eig)
```

```
[('PUCCI', 1.5210930780184965e-24), ('PAZZI', 0.044814939703863084), ('GINORI', 0.0749245316027793), ('LAMBERTES', 0.08879253113499551), ('ACCIAIUOL', 0.1321573195285342), ('SALVIATI', 0.14592084164171834), ('BARBADORI', 0.21170574706479847), ('ALBIZZI', 0.2439605296754477), ('CASTELLAN', 0.25902003784235145), ('PERUZZI', 0.2757224374104833), ('BISCHERI', 0.2827943958713356), ('GUADAGNI', 0.2891171573265014), ('TORNABUON', 0.325846704169574), ('RIDOLFI', 0.3415544259074365), ('STROZZI', 0.3559730326460451), ('MEDICI', 0.4303154258349923)]
```

Q4: Who is the most influential node in the Padgett Florentine Families business-level network?

i) Medici has highest degree is 5 and barbadori, lambertes and Peruzzi has 4 degree

ii) Barbadori and Bishcheri has highest betweenness whereas Medici is on third place and 10 nodes among 16 are having 0 betweenness

iii) Barbadori has highest closeness whereas Medici and Peruzzi are on second place

iv) Peruzzi has highest eigen vector values and Barbadori having more value than Medici

v) From above, we can drive that, Barbadori is the most powerful node in the business network

]:
```
b_degree_dict = dict(business_net.degree(business_net.nodes()))
nx.set_node_attributes(business_net, b_degree_dict, 'degree')
b_sorted_degree = sorted(b_degree_dict.items(), key=itemgetter(1), reverse=True)
print("All 16 nodes by degree:")
for e in b_sorted_degree[:16]:
    print(e)
```

```
All 16 nodes by degree:
('MEDICI', 5)
('BARBADORI', 4)
('LAMBERTES', 4)
('PERUZZI', 4)
('BISCHERI', 3)
('CASTELLAN', 3)
('GINORI', 2)
('GUADAGNI', 2)
('PAZZI', 1)
('SALVIATI', 1)
('TORNABUON', 1)
('ACCIAIUOL', 0)
('ALBIZZI', 0)
('PUCCI', 0)
('RIDOLFI', 0)
('STROZZI', 0)
```

[33]:
```
b_betweenness_dict = nx.betweenness_centrality(business_net)
b_betweenness_dict
```

[33]:
```
{'ACCIAIUOL': 0.0,
 'ALBIZZI': 0.0,
 'BARBADORI': 0.2380952380952381,
 'BISCHERI': 0.02380952380952381,
 'CASTELLAN': 0.04761904761904762,
 'GINORI': 0.0,
 'GUADAGNI': 0.0,
 'LAMBERTES': 0.057142857142857134,
 'MEDICI': 0.2285714285714286,
 'PAZZI': 0.0,
 'PERUZZI': 0.1285714285714286,
 'PUCCI': 0.0,
 'RIDOLFI': 0.0,
 'SALVIATI': 0.0,
 'STROZZI': 0.0,
 'TORNABUON': 0.0}
```

7]:
```
b_closeness_dict = nx.closeness_centrality(business_net)
b_cl = sorted(b_closeness_dict.items(), key=itemgetter(1))
print(b_cl)
```

```
[('ACCIAIUOL', 0.0), ('ALBIZZI', 0.0), ('PUCCI', 0.0), ('RIDOLFI', 0.0), ('STROZZ
I', 0.0), ('GUADAGNI', 0.20833333333333331), ('PAZZI', 0.23809523809523808), ('SA
LVIATI', 0.23809523809523808), ('TORNABUON', 0.23809523809523808), ('BISCHERI',
0.26666666666666666), ('LAMBERTES', 0.2777777777777778), ('GINORI', 0.30303030303
0303), ('CASTELLAN', 0.3333333333333333), ('MEDICI', 0.3508771929824561), ('PERUZ
ZI', 0.3508771929824561), ('BARBADORI', 0.39215686274509803)]
```

[40]:
```
b_eigenvector_dict = nx.eigenvector_centrality(business_net)
b_eig = sorted(b_eigenvector_dict.items(), key = itemgetter(1))
print(b_eig)
```

```
[('ACCIAIUOL', 7.406827552057626e-28), ('ALBIZZI', 7.406827552057626e-28), ('PUCC
I', 7.406827552057626e-28), ('RIDOLFI', 7.406827552057626e-28), ('STROZZI', 7.406
827552057626e-28), ('PAZZI', 0.07275338532647235), ('SALVIATI', 0.072753385326472
35), ('TORNABUON', 0.07275338532647235), ('GINORI', 0.19051895536421606), ('GUADA
GNI', 0.23505765303813655), ('MEDICI', 0.24111147325311608), ('BISCHERI', 0.34418
960249072633), ('BARBADORI', 0.3902899486872091), ('CASTELLAN', 0.391030002640222
4), ('LAMBERTES', 0.43482864636112956), ('PERUZZI', 0.4708114043450975)]
```

[ ]:

Q5: Based on the findings in Q3 and Q4, are there any associations between the marriage network and business network?

hints: any marriage relationships influence the business relationships? any business relationships influence marriage relationships?

Answer: Yes, as Barbadori is most powerful in business network so marriage connection between Medici and Barbadori influence whole network connection of business.
So we can say marriage between Barbadori and Medici influence business.