“SOCIAL NETWORK ANALYSIS”

A PROJECT REPORT SUBMITTED IN THE PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE Of
BACHELOR OF TECHNOLOGY in PROCESS ENGINEERING WITH MBA

By:
SANJAY DEO(14214027)

Supervisor:
Gaurav Dixit
Assistant Professor, DOMS

DEPARTMENT OF MANAGEMENT STUDIES
INDIAN INSTITUTE OF TECHNOLOGY, ROORKEE

ABSTRACT

In this study, Social Network Analysis is performed on **'major_us_cities'** data.The dataset consists of information about major US cities such as population and location.
Analysis is done using Python Programming Language in Jupyter Notebook using advanced **NetworkX** library for graphical representation of networks.Explanation of various terms are done in Key Terms with code implementation too. These terms or metrics are very crucial to create readable networks. Code implementation is shown as snapshot while data is analysed through graphs and tables.A complex graph is simplified using various customization to provide easiness in visualization (using color encoding).
In the end conclusion is made along with suggestion to improve analysis in next stage.

INTRODUCTION

Social Network Analysis is the process of studying social structures through networks and graphs. The network is characterised in the term of nodes and edges that ties them. Examples of social structures commonly visualized through social network analysis include social media networks, information circulation,business networks,etc.
In business use of Social Network is increasing day by day due to increase in big data.Analysing Network is bit tedious work and one should know key terms before Network interpretation.Organizations like Law enforcement agencies,Social networking sites,Civil society operators and Network Operators use SNA like methods to optimize the structure and capacity of their networks.
Let's understand some Network type with few examples :

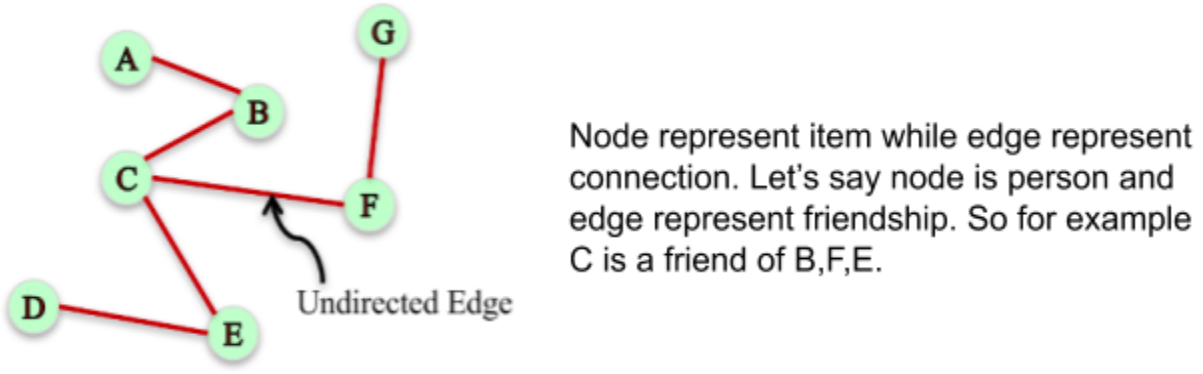
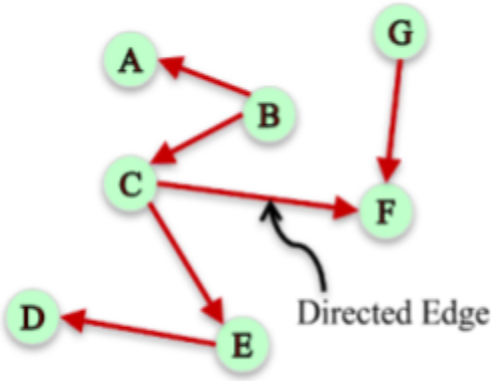
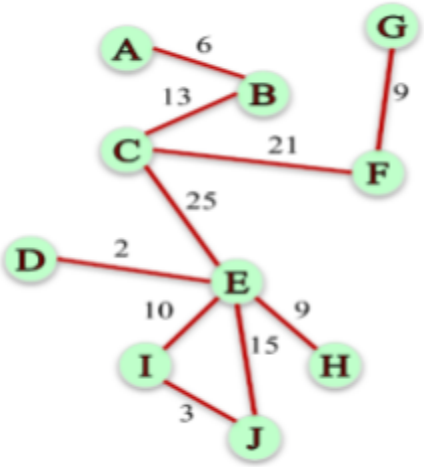


Fig 1



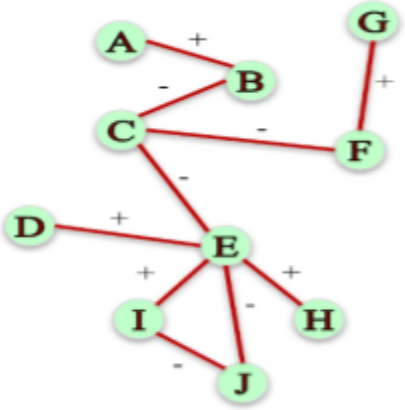
In this Directed Network Directed edges have some meaning. Let's say Node represents person and directed edges represents email send from person X to Y. So,C receives email from B while sends to F and E.

Fig 2



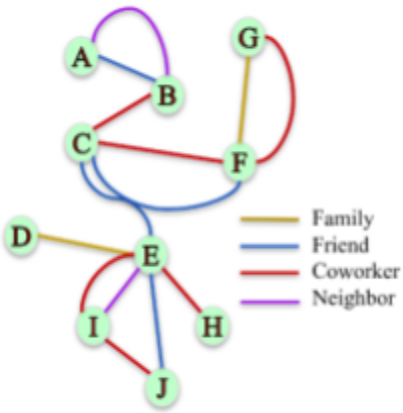
Such kind of Networks are called weighted Network.In this figure let's say Node represents Person and Edge represents having lunch together. While weight on edge represents how many times they have lunch together.

Fig 3



In this Network nodes represent person and edges represent social connection between two person. And sign(+ or -) of edges represent connection type(friend or enemy).

Fig 4



In this figure nodes represent person and edge color represents type of social connection.

Fig 6

KEY TERMS

Symmetric Networks

In figure 1 where undirected edges represents two way relation that is if A is related to B then B is related to A. Such networks are Symmetric Networks.

Asymmetric Networks

In figure 2 , edges have direction which signifies one way relation so if A is related to B then vice versa is not true. So relationship is not symmetric hence called Asymmetric Networks.

Weighted Networks

If we assign weight to edges where weight signifies magnitude of relation then such networks are called Weighted Networks.

Multigraph

In a figure 6 where attributes are assigned to edges and signifies some type. Such Networks are called Multigraph.

Degree

Defined as number of connection a node has. Node with highest Degree is considered to be crucial node for spread of information.

Clustering Coefficient

In Social Network Analysis, association of nodes are measured using clustered coefficient and is the fraction of pairs of the node's connections that are connected with each other. There exists two versions of Clustering Coefficient: 1. Global and 2. Local. Overall indication of clustering is measured by Global Clustering Coefficient while embeddedness of single nodes is indicated by Local version.
For global version:

$$C = \frac{\text{number of closed triplets}}{\text{number of all triplets (open and closed)}}$$

For local version:

$$C_i = \frac{|\{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E\}|}{k_i(k_i - 1)}.$$

Eccentricity

For a particular node, eccentricity is defined as largest distance between that node and all other nodes. For disconnected graph eccentricity for all nodes will be infinite.

Degree Centrality

For a particular node, measured value of number of connection in the network. It helps in identification of most influential people of social network, key infrastructure node and super spreader of disease.

Eigenvector Centrality

Just like Google's PageRank and the Katz centrality, Eigenvector Centrality helps in identification of most influential node by using concept of relative ranking score measured for all nodes in network.
The relative centrality score is given by:

$$x_v = \frac{1}{\lambda} \sum_{t \in M(v)} x_t = \frac{1}{\lambda} \sum_{t \in G} a_{v,t} x_t$$

where lambda is a constant and M(v) is set of neighbors of v.
The basic concept for Eigenvector centrality is that it increases importance of node if the node is connected with other important nodes.

Betweenness Centrality

In shortest path between two nodes , Betweenness Centrality quantifies number of times a particular node occurs in bridge between two nodes. Node with highest Between Centrality plays vital role in social influencer and help in information transmission in fastest possible rate. Betweenness Centrality is represented by :

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where σ_{st} represents total number of shortest path from s to t.

PageRank Centrality

For PageRank Centrality formula turns into :

$$x_i = \alpha \sum_j a_{ji} \frac{x_j}{L(j)} + \frac{1-\alpha}{N},$$

where

$$L(j) = \sum_i a_{ji}$$

Finding above metrics is done in library called Networkx that provides easy implementation for all metric calculation.

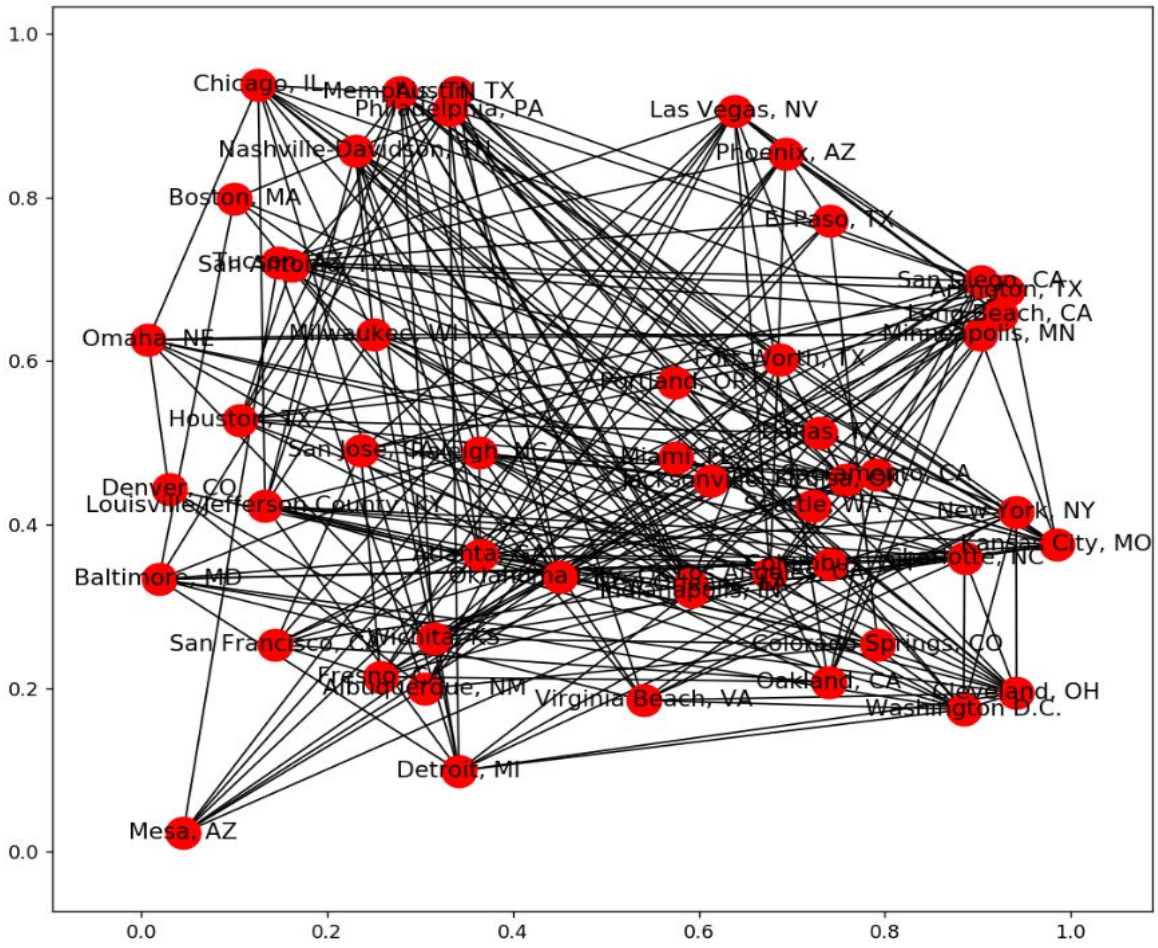
ANALYSIS

Dataset Information

Dataset contains information about major US cities stored in graph object as compressed form. It contains 51 nodes which represent city name with location and population information. And total number of edges are 235 which represents weighted connection between them.

```
Type: Graph
Number of nodes: 51
Number of edges: 235
Average degree:   9.2157
```

Graph Visualization



The graph is visualized using Networkx **draw_network()** function :

```
#plotting graph
plt.figure(figsize=(10,9))
pos = nx.random_layout(G)
nx.draw_networkx(G, pos)
```

Creating features from Network

Features from nodes

```
df['clustering'] = pd.Series(nx.clustering(G))
df['degree'] = pd.Series(G.degree())
df
```

	location	population	clustering	degree
El Paso, TX	(-106, 31)	674433	0.700000	5
Long Beach, CA	(-118, 33)	469428	0.745455	11
Dallas, TX	(-96, 32)	1257676	0.763636	11
Oakland, CA	(-122, 37)	406253	1.000000	8
Albuquerque, NM	(-106, 35)	556495	0.523810	7
Baltimore, MD	(-76, 39)	622104	0.800000	10
Raleigh, NC	(-78, 35)	431746	0.615385	13
Mesa, AZ	(-111, 33)	457587	0.750000	8
Arlington, TX	(-97, 32)	379577	0.763636	11
Sacramento, CA	(-121, 38)	479686	0.777778	9
Wichita, KS	(-97, 37)	386552	0.622222	10
Tucson, AZ	(-110, 32)	526116	0.750000	8

We can see from table that each node generates a city with respective **clustering coefficient and degree** measures. The values are generated by using Networkx built in functions .
Before generating this table many intermediate steps are performed like:
Converting Dictionary into dataframe
Finding Population and location feature

Features from edges

```
df['preferential attachment'] = [i[2] for i in nx.preferential_attachment(G, df.index)]
df['Common Neighbors'] = df.index.map(lambda city: len(list(nx.common_neighbors(G, city[0], city[1]))))
df
```

	weight	preferential attachment	Common Neighbors
(El Paso, TX, Albuquerque, NM)	367.885844	35	4
(El Paso, TX, Mesa, AZ)	536.256660	40	3
(El Paso, TX, Tucson, AZ)	425.413867	40	3
(El Paso, TX, Phoenix, AZ)	558.783570	45	3
(El Paso, TX, Colorado Springs, CO)	797.751712	30	1
(Long Beach, CA, Oakland, CA)	579.582999	88	7
(Long Beach, CA, Mesa, AZ)	590.156204	88	5
(Long Beach, CA, Sacramento, CA)	611.064979	99	7
(Long Beach, CA, Tucson, AZ)	698.656667	88	5
(Long Beach, CA, San Jose, CA)	518.233061	88	7
(Long Beach, CA, Fresno, CA)	360.470458	99	8
(Long Beach, CA, San Diego, CA)	151.450082	121	10

We can see from the table two new features generated for respective edges are **Preferential Attachment and Common Neighbors.** The value are generated using weight as input in builtin functions.

Our main focus was to tackle find **Maximum Degree value Node** for assumed problem and is done by using finding degree centrality for each node and selecting node with maxing degree centrality value. This was done using **degree_centrality()** function from Networkx.
The following is assumed problem for analysis :

Question : Suppose we are launched a new product and concerned about its delivery to more and more cities. Unfortunately we can launch in only one city.Luckily, we can deliver product to nearest neighbor cities as well. What is the most successful way to deliver product from above 'G' Network ?

Solution : Finding city which has maximum degree measure will be required city.

RESULT

‘Columbus, OH’ is the city with has highest degree centrality and most influencing city in the Network.

Code Implementation :

```
def answer_two():
    degree = nx.degree_centrality(G)
    return max(degree.keys(), key=lambda x:degree[x])
answer_two()
```

```
'Columbus, OH'
```

REFERENCES:

https://en.wikipedia.org/wiki/Social_network
https://www.coursera.org/learn/python-social-network-analysis
https://www.datacamp.com/community/tutorials/social-network-analysis-python
https://bookdown.org/chen/snaEd/intro.html