

Intro to Data Science

Assignment # 03

**Wajahat Masood
Abdullah Bin Abdur Rauf
Saif Ur Rehman**

SP19-BSE-003
SP19-BSE-007
SP19-BSE-013

BSE 7A

TO: Dr. Hikmat Ullah

WORKFLOW

Data Extraction

Data is extracted from Facebook about friend Circle.

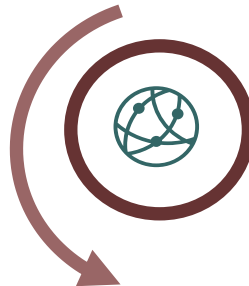


Import Data Set

Manual cleaning txt data & Import in Code

Betweenness Centrality

Betweenness centrality is a measure of centrality in a graph based on shortest paths.



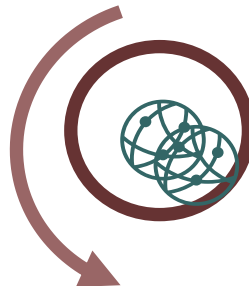
Closeness Centrality

Closeness centrality is a measure of the average shortest distance from each vertex to each other vertex.



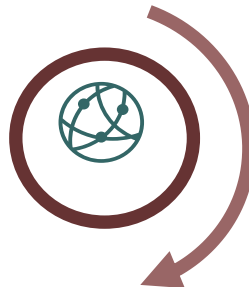
Clustering

Clustering is the task of grouping a set of objects in such a way that objects in the same group are more like each other.



Eccentricity

The maximum distance of one vertex from other vertex is eccentricity.



Degree Centrality

The degree centrality of a node is simply its degree the number of edges it has.

Degree Graph

The degree of a vertex of a graph is the number of edges that are incident to the vertex.

Average Clustering

Compute average clustering for nodes in the container.

TOOLS WE USE

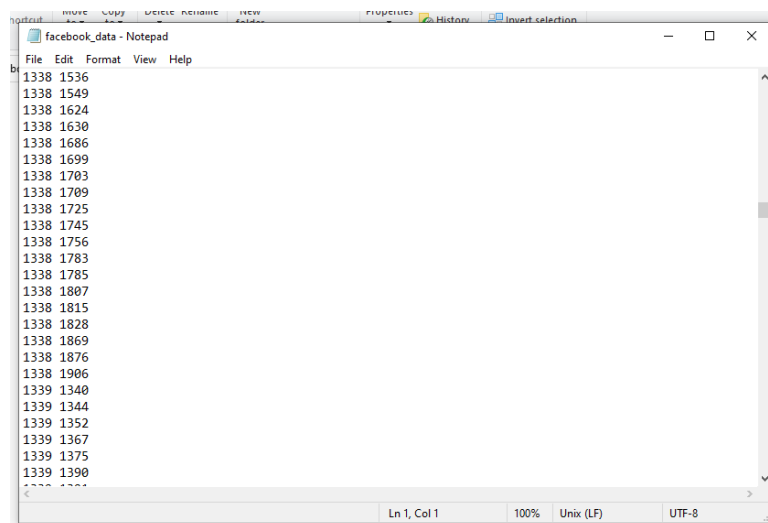


Google Colab

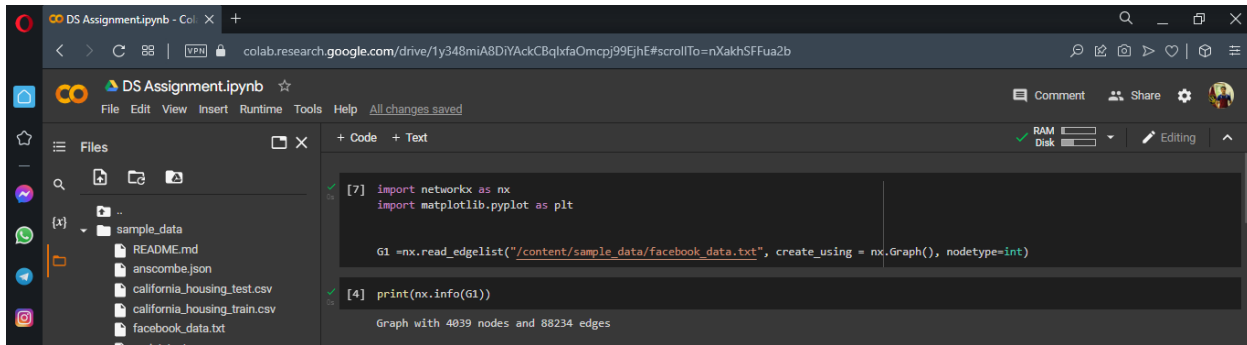


DETAIL OF DATA

We gather Data from Facebook friend's circle. We Extract the data and clean it first and then we start our work.



Import Facebook Dataset



The image shows a Google Colab notebook titled "DS Assignment.ipynb". The left sidebar displays a file explorer with a folder named "sample_data" containing files like "README.md", "anscombe.json", "california_housing_test.csv", "california_housing_train.csv", "facebook_data.txt", and "mnist_test.csv". The main code area shows two code cells. The first cell imports the necessary libraries and reads the Facebook dataset into a graph object G1. The second cell prints the graph's information.

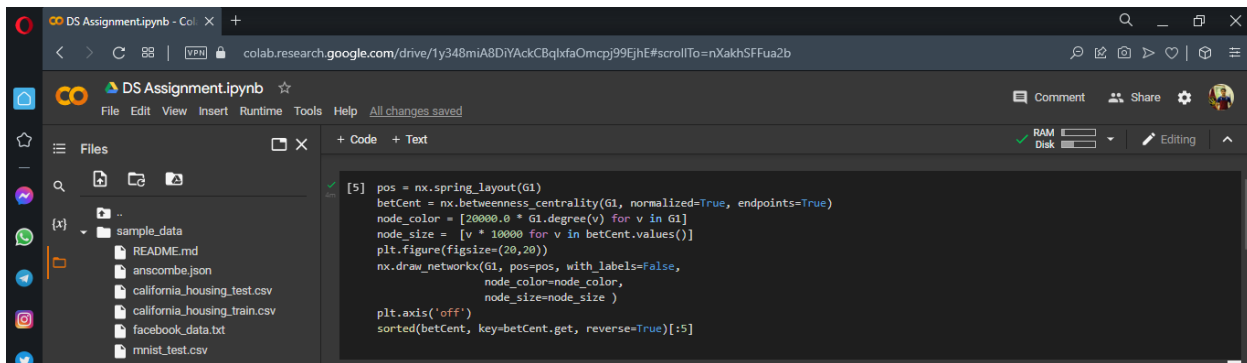
```
[7] import networkx as nx
import matplotlib.pyplot as plt

G1 = nx.read_edgelist("/content/sample_data/facebook_data.txt", create_using = nx.Graph(), nodetype=int)

[4] print(nx.info(G1))

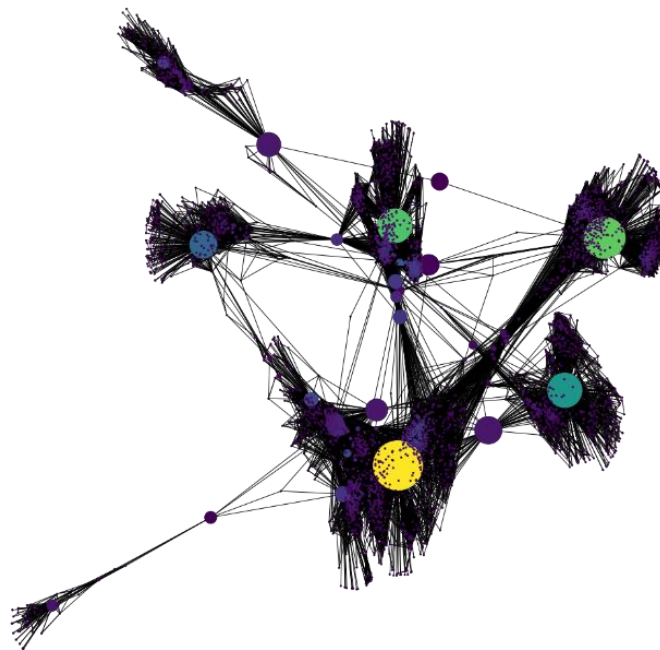
Graph with 4039 nodes and 88234 edges
```

Betweenness Centrality



The image shows the same Google Colab notebook with a third code cell. This cell calculates the betweenness centrality for each node in the graph, creates a spring layout, and visualizes the network. Nodes are colored based on their degree and sized based on their betweenness centrality. The axes are turned off for a cleaner visualization.

```
[5] pos = nx.spring_layout(G1)
betCent = nx.betweenness_centrality(G1, normalized=True, endpoints=True)
node_color = [20000.0 * G1.degree(v) for v in G1]
node_size = [v * 10000 for v in betCent.values()]
plt.figure(figsize=(20,20))
nx.draw_networkx(G1, pos=pos, with_labels=False,
                 node_color=node_color,
                 node_size=node_size )
plt.axis('off')
sorted(betCent, key=betaCent.get, reverse=True)[:5]
```



Degree Centrality

```
DS Assignment.ipynb - Colab
colab.research.google.com/drive/1y348mIA8DiYAckCBqlxfA0mcpj99EjhE#scrollTo=9ri97ATDt8nN

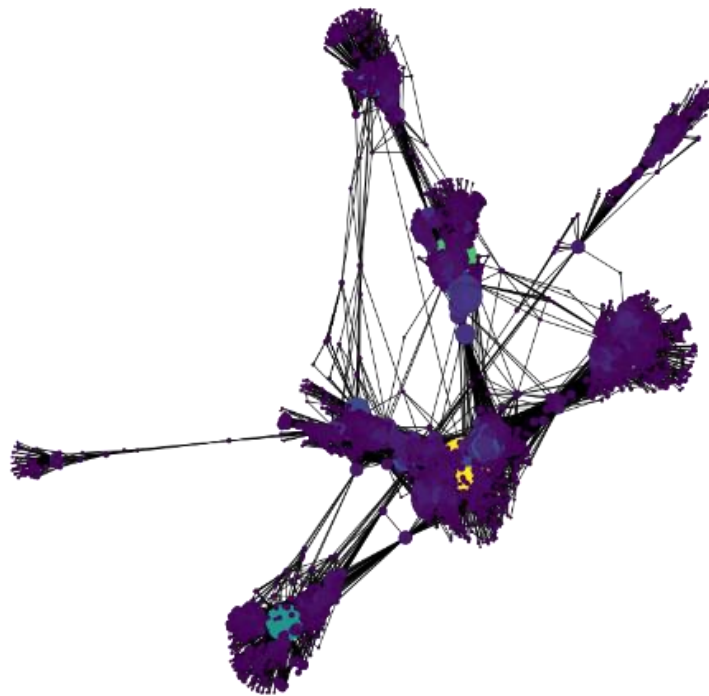
DS Assignment.ipynb
File Edit View Insert Runtime Tools Help All changes saved

Files
sample_data
  README.md
  anscombe.json
  california_housing_test.csv
  california_housing_train.csv
  facebook_data.txt
  mnist_test.csv

+ Code + Text
pos = nx.spring_layout(G1)
degCent = nx.degree_centrality(G1)
node_color = [20000,0 * G1.degree(v) for v in G1]
node_size = [v * 10000 for v in degCent.values()]
plt.figure(figsize=(15,15))
nx.draw_networkx(G1, pos=pos, with_labels=False,
                 node_color=node_color,
                 node_size=node_size)

plt.axis('off')

sorted(degCent, key=degCent.get, reverse=True)[:5]
```



Sorted

```
[9] sorted(degCent, key=degCent.get, reverse=True)[:5]

[107, 1684, 1912, 3437, 0]
```

Closeness Centrality

```
DS Assignment.ipynb - Colab
colab.research.google.com/drive/1y348miA8DiYAckCBqlxfA0mcpj99EjhE#scrollTo=GAjZKkmbuFWZ

DS Assignment.ipynb
File Edit View Insert Runtime Tools Help All changes saved

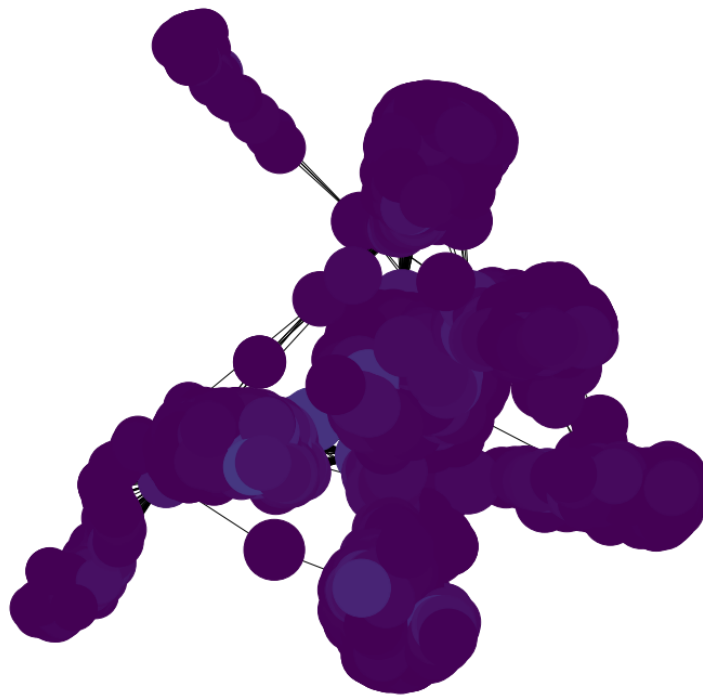
Files
Connecting to a runtime to enable file browsing.

{x}

+ Code + Text
Allocating Editing

pos = nx.spring_layout(G1)
cloCent = nx.closeness_centrality(G1)
node_color = [20000.0 * G1.degree(v) for v in G1]
node_size = [v * 10000 for v in cloCent.values()]
plt.figure(figsize=(13,13))
nx.draw_networkx(G1, pos=pos, with_labels=False,
                 node_color=node_color,
                 node_size=node_size)

plt.axis('off')
sorted(cloCent, key=cloCent.get, reverse=True)[:5]
```

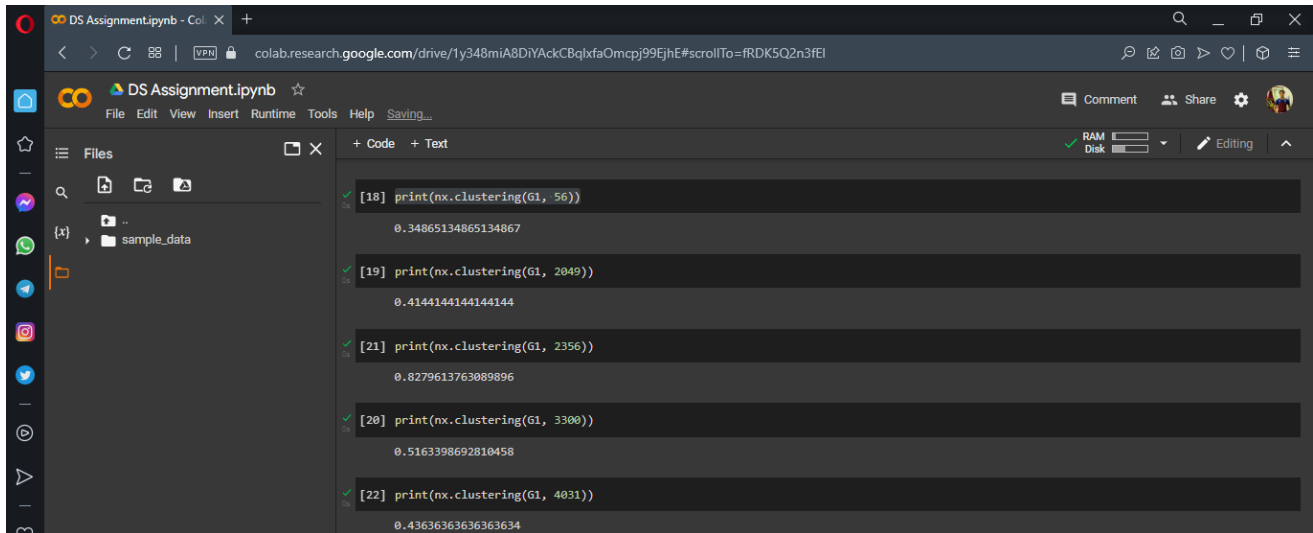


Degree Graph

```
[ ] max(x for x,y in nx.degree(G1))
```

4038

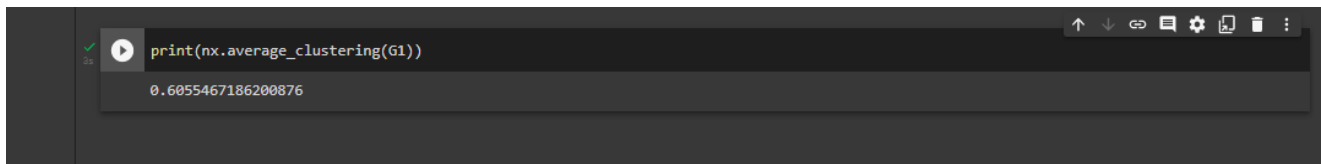
Clustering for 5 Different Nodes



A Google Colab notebook titled "DS Assignment.ipynb" showing five code cells, each performing a clustering operation on a graph G1. The left sidebar shows a file explorer with a folder named "sample_data". The top toolbar indicates RAM and Disk usage. The code cells and their outputs are as follows:

Cell Index	Code	Output
[18]	<code>print(nx.clustering(G1, 56))</code>	0.34865134865134867
[19]	<code>print(nx.clustering(G1, 2049))</code>	0.4144144144144144
[21]	<code>print(nx.clustering(G1, 2356))</code>	0.8279613763089896
[20]	<code>print(nx.clustering(G1, 3300))</code>	0.5163398692810458
[22]	<code>print(nx.clustering(G1, 4031))</code>	0.436363636363634

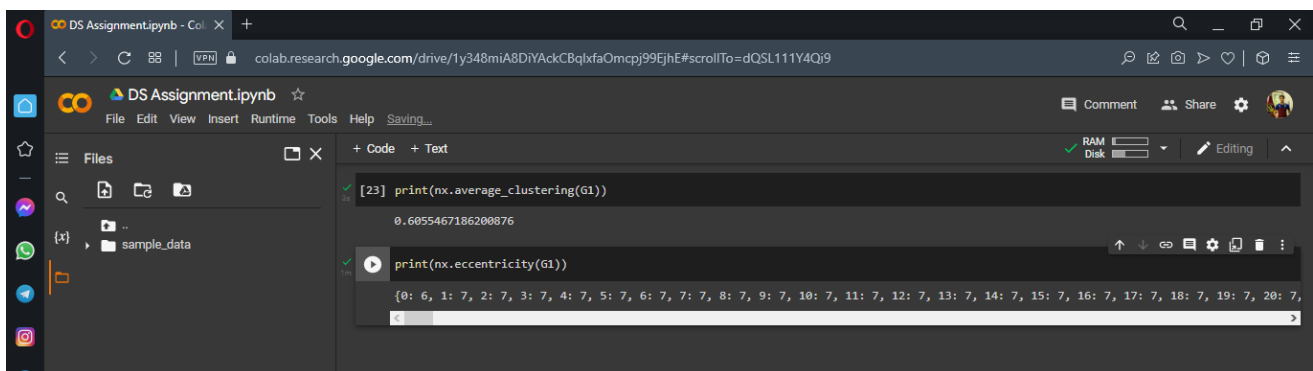
Average Clustering



A Google Colab notebook showing a single code cell that calculates the average clustering coefficient for a graph G1. The code and its output are:

Cell Index	Code	Output
[24]	<code>print(nx.average_clustering(G1))</code>	0.6055467186200876

Eccentricity



A Google Colab notebook showing two code cells. The first cell calculates the average clustering coefficient, and the second cell calculates the eccentricity of a graph G1. The code and its outputs are:

Cell Index	Code	Output
[23]	<code>print(nx.average_clustering(G1))</code>	0.6055467186200876
[24]	<code>print(nx.eccentricity(G1))</code>	{0: 6, 1: 7, 2: 7, 3: 7, 4: 7, 5: 7, 6: 7, 7: 7, 8: 7, 9: 7, 10: 7, 11: 7, 12: 7, 13: 7, 14: 7, 15: 7, 16: 7, 17: 7, 18: 7, 19: 7, 20: 7, ...}

THE END