Lecture Notes:Week 9

From JOCWiki

For Area Calculation - Don't Measure 1st way scipy.misc.imread is removed from sciPy ver 1.2 use import imageio imageio.imread("directory_of_image")

For image of map go to https://mapchart.net/india.html

For 6 degree of separation Facebookcombined.txt File Go to-->http://snap.stanford.edu/data/ego-Facebook.html

1. *dont forget to extract the file if it has Extention Facebookcombined.txt.{gz}*

---

2) Introduction to Networkx

Part 01 NetworkX

NetworkX is a Python language software package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. Pygraphviz is a Python interface to the Graphviz graph layout and visualization package. Python language data structures for graphs, digraphs, and multigraphs.

Create an empty graph with no nodes and no edges.

```
import networkx as nx
G=nx.Graph()
```

By definition, a Graph is a collection of nodes (vertices) along with identified pairs of nodes (called edges, links, etc). In NetworkX, nodes can be any hashable object e.g. a text string, an image, an XML object, another Graph, a customized node object, etc. (Note: Python's None object should not be used as a node as it determines whether optional function arguments have been assigned in many functions.)

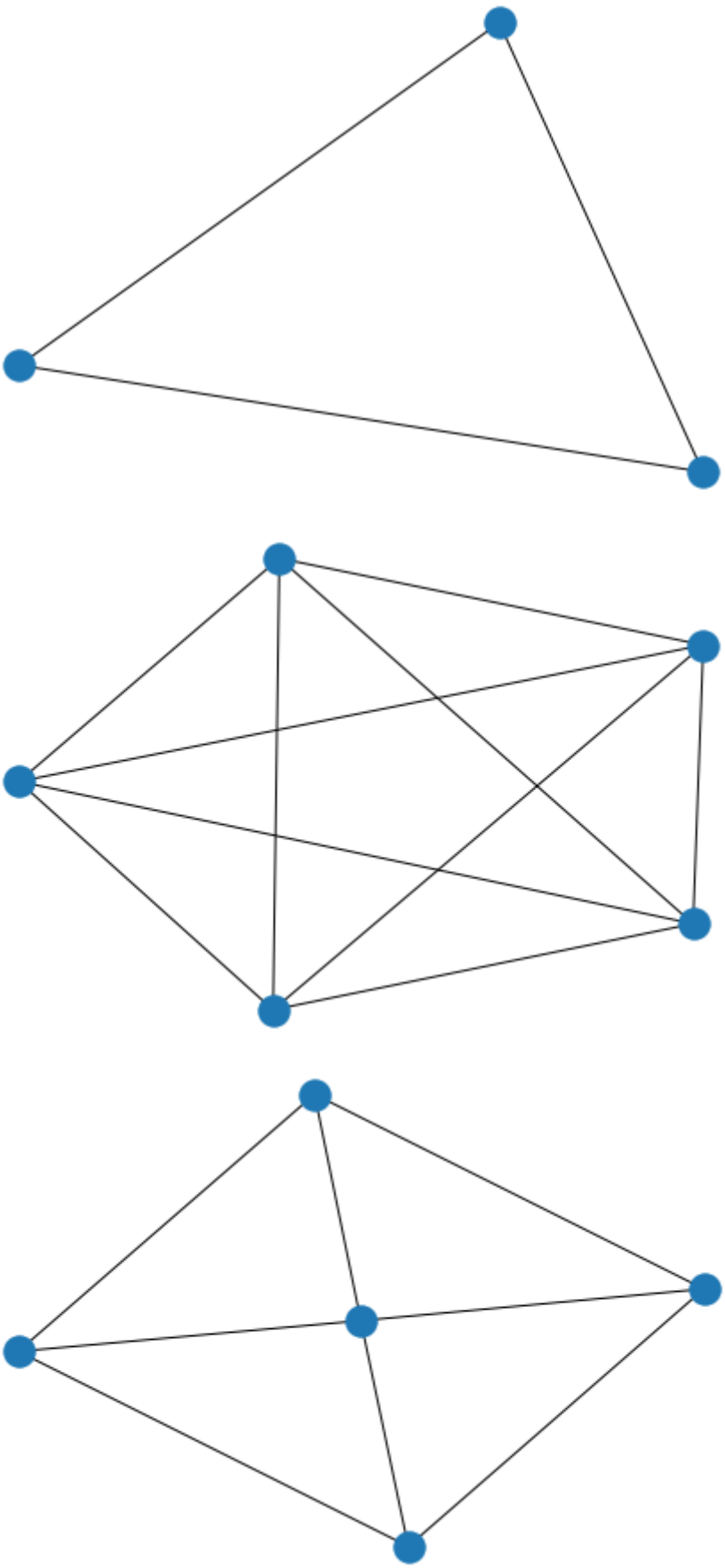
Sample Program to visualize a Structure

```
@author: Sharanjit Singh


import networkx as nx
import matplotlib.pyplot as plt


# METHOD 1
# Creating an emplty graph
G = nx.Graph()
# creating nodes in the graph
G.add_node(1)
G.add_node(2)
G.add_node(3)


(this is another method of creating nodes)
l = [1,2,3]
```

```
   G.add_nodes_from(l)

   # adding edges to the nodes (or connecting nodes together)
   G.add_edge(1,2)
   G.add_edge(2,3)
   G.add_edge(3,1)
   # printing the values of nodes and edges
   print(G.nodes())
   print('\n')
   print(G.edges())
   # printing the graph
   nx.draw(G)
   plt.show()



   # METHOD 2
   # networkx comes with some pre-defined graphs
   GC = nx.complete_graph(5)  # 5 is no. of nodes
   # printing the graph G
   nx.draw(GC)
   plt.show()



   # METHOD 3
   # we can plot some random graphs
   GR = nx.gnp_random_graph(5,.7)  # 5 is no. of nodes and .7 is the probabilit of
edges
   # printing the grapg GR
   nx.draw(GR)
   plt.show()
```

1. *Output of the Program*

```
   [1, 2, 3]


   [(1, 2), (1, 3), (2, 3)]
```



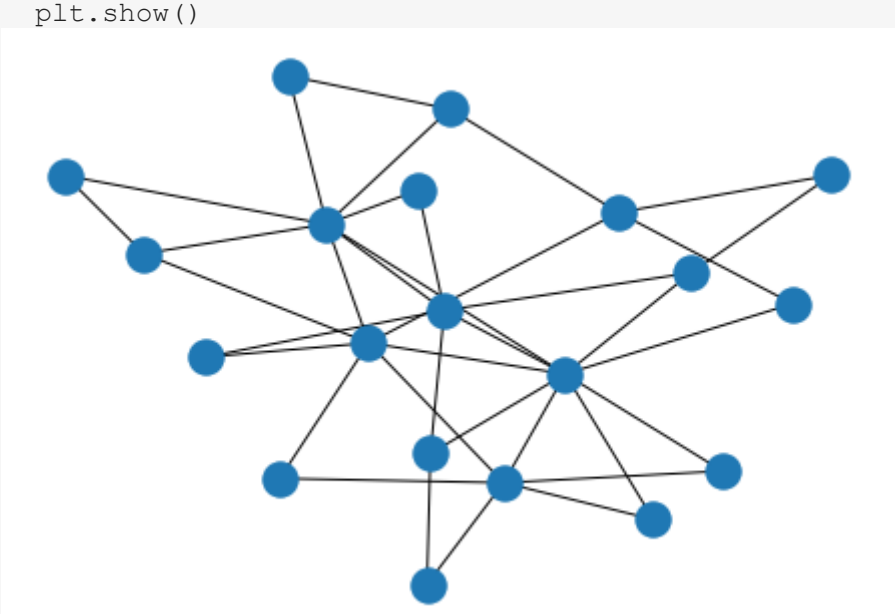
Part-02 Networkx

Some useful Commands :-

```
G.nodes()  # used to print the nodes
G.edges()  # used to print the edges
nodes = int(input())  # This will take input the node number as integer
print(G.degree(node))  # used to know the degree of a specific Node.
```

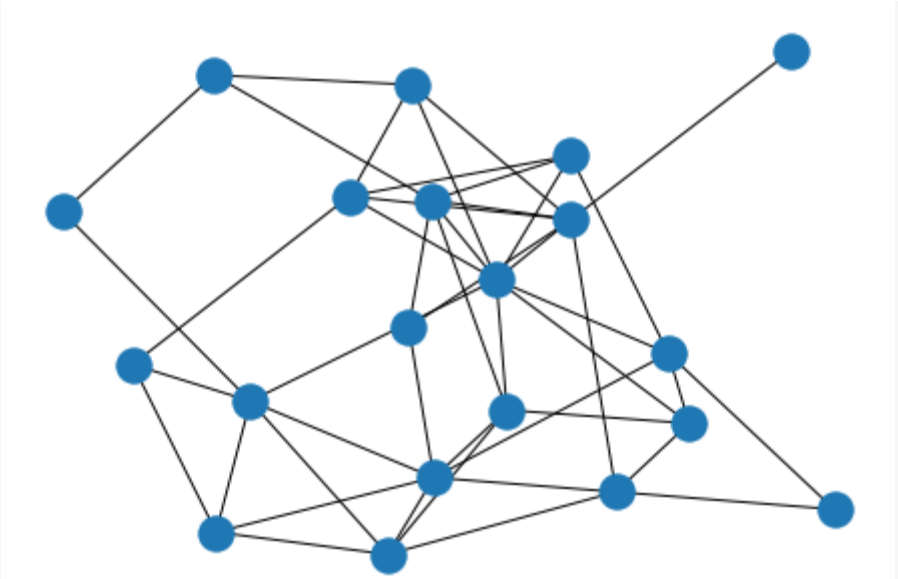
The node degree is the number of edges adjacent to the node. The weighted node degree is the sum of the edge weights for edges incident to that node. In this, we learn how to generate graphs in file format(gexf) and we will use a new tool gephi tool used to visualize the graph. Gephi is open-source software for visualizing and analyzing large network graphs. Gephi uses a 3D render engine to display graphs in real-time and speed up the exploration. You can use it to explore, analyze, spatialize, filter, clusterizes, manipulate and export all types of graphs. **Scale Free Graph**--*somewhat different from random graphs and the command is: G=nx.barabasi_albert_graph(n,m) It takes two parameters one is n-number of nodes and the other m is each node is coming up with m=2 edges.* **Sample Program to illustrate this**

```
import networkx as nx
import matplotlib.pyplot as plt
#create a scale-free model graph
G=nx.barabasi_albert_graph(20,2)
nx.draw(G)
plt.show()
```



the above is for scale free model and the following is for random graph

```
import networkx as nx
import matplotlib.pyplot as plt
#create a random graph
G=nx.gnp_random_graph(20,0.2)
nx.draw(G)
plt.show()
```



From the above two graphs, we can conclude that a new node attaches to an older node with more edges i.e degree which is of scale-free. This is known as preferential attachment.

Now we will see this graph in anew fromat.we will write this in a file in a gexf format. The command is as follows.

```
import networkx as nx
import matplotlib.pyplot as plt
#create a scale-free model graph
G=nx.barabasi_albert_graph(20,2)
nx.draw(G)
plt.show()
#write this in a gesf format
nx.write_gexf(G."analysis.gexf")
```

this is imported to a new tool gephi

_____ ----- _____

3) Introduction to NLTK

Samples of sentence tokenization:=

import nltk

text="Today is a great day. It is even better than yesterday. And yesterday was the best day ever."

from nltk.tokenize import sent_tokenize

tokens = sent_tokenize(text)

token_word = nltk.word_tokenize(text)

print(tokens)

OUTPUT _____

['Today is a great day.', 'It is even better than yesterday.', 'And yesterday was the best day ever.']

_____

print()

text = "Last night, I went to Mrs. Martinez's housewarming. It was a disaster."

tokens = sent_tokenize(text)

print(tokens)

_____

OUTPUT _____

["Last night, I went to Mrs. Martinez's housewarming.", 'It was a disaster.']

_____

print()

text = "சென்னை: கொரோனா காரணமாக உலகம் முழுக்க பாதிப்படையும் மக்களின் எண்ணிக்கை நாளுக்கு நாள் அதிகரித்து வருகிறது. உலகம் முழுக்க *12,72,737* பேர் கொரோனாவால் பாதிக்கப்பட்டு உள்ளனர்."

tokens = sent_tokenize(text)

print(tokens)

_____ OUTPUT

_____

['சென்னை: கொரோனா காரணமாக உலகம் முழுக்க பாதிப்படையும் மக்களின் எண்ணிக்கை நாளுக்கு நாள் அதிகரித்து வருகிறது.', 'உலகம் முழுக்க *12,72,737* பேர் கொரோனாவால் பாதிக்கப்பட்டு உள்ளனர்.*']*

_____