Week 2

Created	@February 10, 2025 3:29 PM

1. Some interesting facts :

- Take all the words in English language as nodes, and draw edge from one word to another if it is a synonym. The resultant graph is connected. Interesting part here is, we can even travel from node ‘beautiful’ to ‘ugly’.
- Take all websites/webpages as nodes and draw an edge from one node to another, if there is a hyperlink between them. This resultant graph is called “Web graph”. This was used by Google to rank the pages/websites for their search engine.
- Take each person as node, and draw edge between them if they are friends on social media. This graph can be used to predict who is the next probable friend of a person.

2. Network Datasets Format :

- csv
- GML
- PajekNet
- GraphML
- GEXF

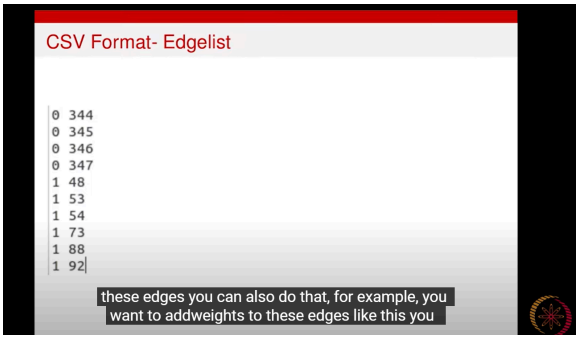
2.1 : CSV

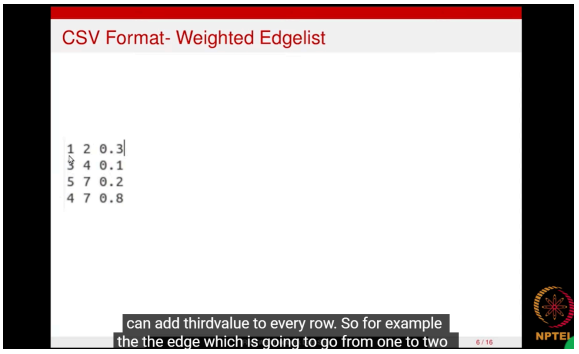
Comma Separated Values

Extension : .csv, .txt

Types : Edgelist, Adjacency list

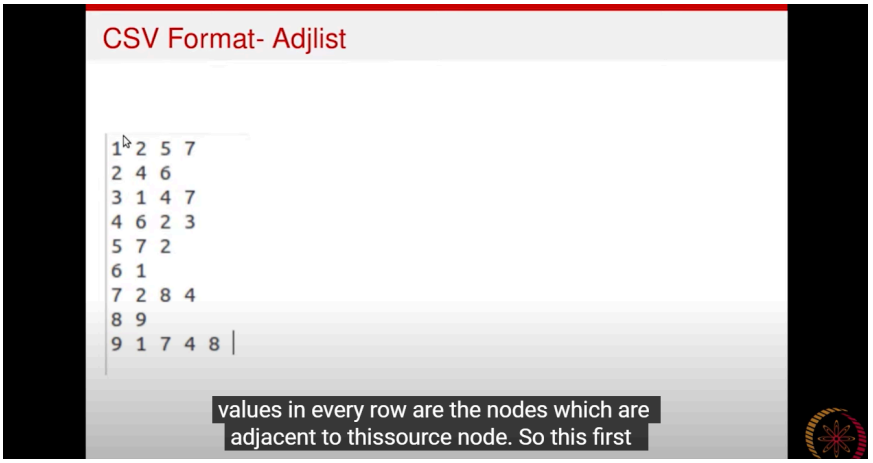
Edgelist :





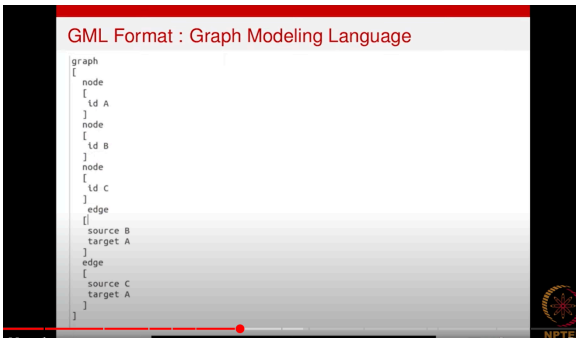
- Format : Sourcenode Targetnode
- It is simple and contains list of edges from source node to target node.
- It can also have weights associated with each edge .
- It cannot have any non-numeric weight, add any attribute, change any labels.

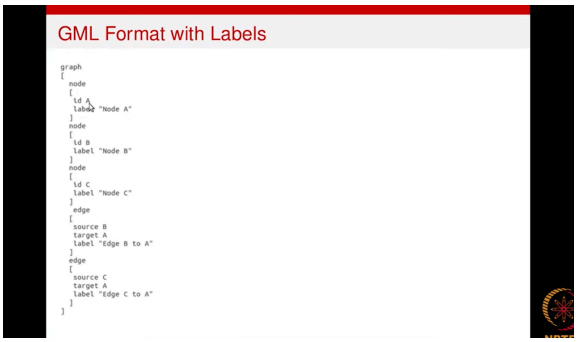
Adjaceny List :



- Format : Soucenode Neighbours
- It is simple and contains sourcenode and its neighbours in each row.
- It also cannot add any attribute , change any labels.

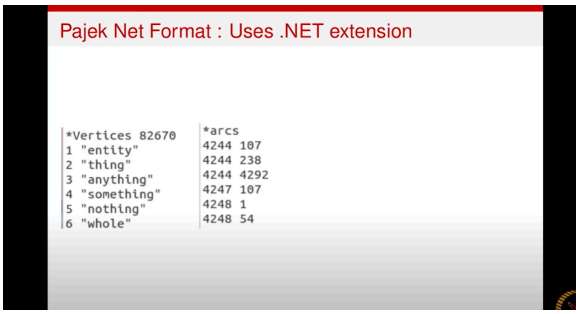
2.2 : GML

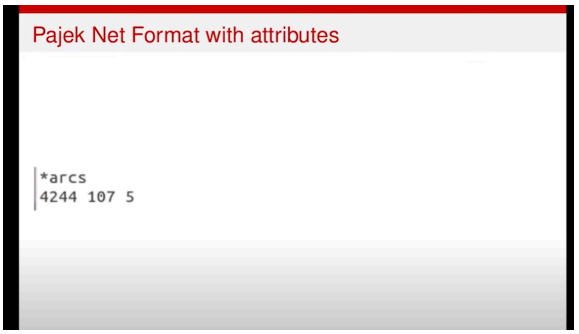




- Graph Modelling Language
- Most commonly used format for datasets.
- Each node , edge is defined with keyword and [ ] .
- Labels and attributes can also be added here.

2.3 : PajekNet

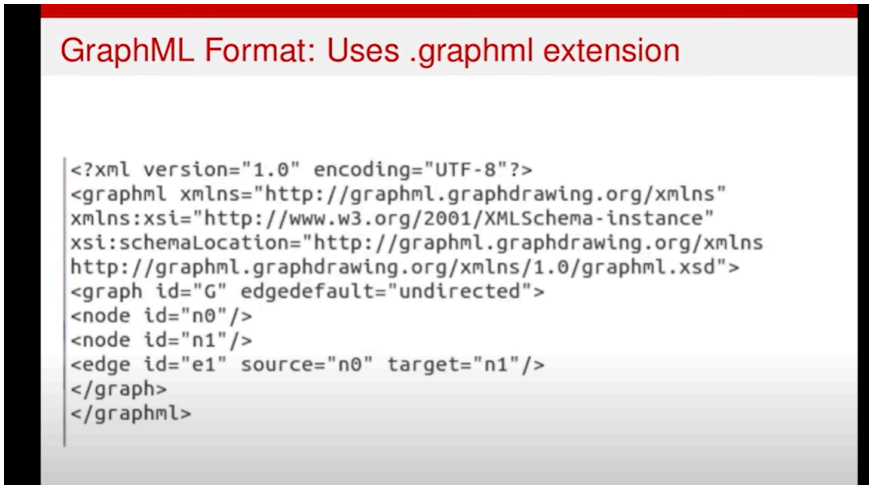




- Extension : .net , .paj
- Format : *Vertices no.of vertices , Then each row contains vertex number and label

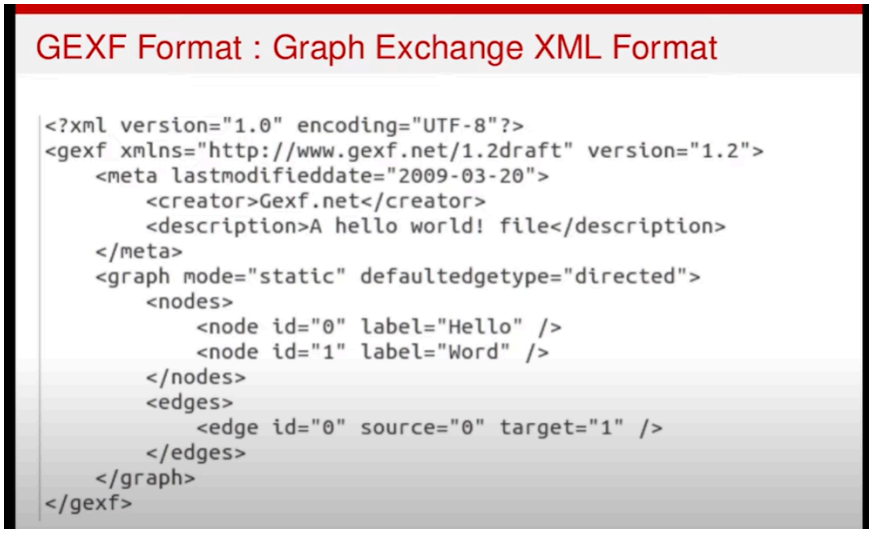
  * arcs/ * Edges , Then each row contains source node and target node and attribute

2.4 : GraphML



- ML denotes XML, it uses various tags like in html.
- <graph id=”g” edgedefault = “undirected”>
- <node id=”n0”>
- <edge id=”e1” soucenode=”n0” targetnode=”n1”>

2.5 : GEXF



- Graph Exchange XML Format
- Similar to GraphML but more cleaner than it.

3. Analyzing Network Datasets :

```
import networkx as nx
import matplotlib.pyplot as plt

G = nx.read_edgelist('dataset path') # reads edgelist dataset and return a grap
print(nx.info(G))  # show basic details of the graph
print(nx.number_of_nodes(G)) # return no.of nodes
print(nx.number_of_edges(G)) # return no.of edges
print(nx.is_directed(G)) # return True or False

G = nx.read_pajek('dataset path') # to read a pajek dataset
G = nx.read_gml('dataset path') # to read gml dataset
G = nx.read_graphml('dataset path') # to read graphml dataset
G = nx.read_gexf('dataset path') # to read gexf dataset

nx.draw(G) # to draw the graph
nx.draw_circular(G) # nodes will be arranged circularly
plt.show() # to see the output
```

Degree :

The degree of the node is the total number of edges it has.

For undirected graph - degree

For directed graph - indegree, outdegree

Density :

Number of edges / Total possible edges

Density value of complete graph is 1.0

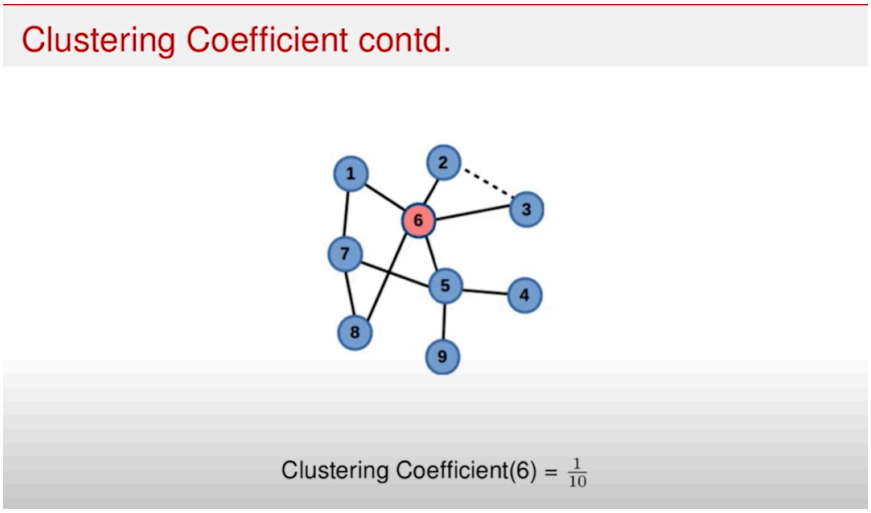
Total possible edges = n(n-1) / 2 [n - number of nodes]

Clustering Coefficient :

Actual edges between neighbours / Total possible edges between neighbours

It is calculated for each node.

Clustering coefficient of a graph is average of all nodes.



Here , consider Node 6 , its neighbours are 1,2,3,5,8

Actual edges between neighbours = 1 (from 2 to 3)

Total possible edges between neighbours = 10 [n*(n-1)/2]

so, 1/10

Diameter :

Take shortest path from each node to each other node. Diameter is the longest path in this.

In other words. it is the maximum of all shortest paths from one node to other.

```
import networkx as nx
import matplotlib.pyplot as plt

G = nx.read_gml('dataset path')
nx.degree(G)
# returns dictionary with node : degree

nx.density(G)
# returns density of graph

nx.clustering(G)
# returns dictionary with node : clustering coefficient
nx.average_clustering(G)
# returns average clustering coefficient
```

```
nx.diameter(G)
#returns the diameter of graph
```

4. Emergence of connectedness :

If there is a graph with “n” nodes and if we add “n logn” edges uniformly at random the probability that a node will be isolated is 1/n^2.

So, For 100 nodes after 100log100 edges, probability that a node will be isolated = 1/10,000

That is, probability the graph is not connected = 1/10,000

Number of edges required to make a graph connected = n logn

More specifically, it is order of nlogn