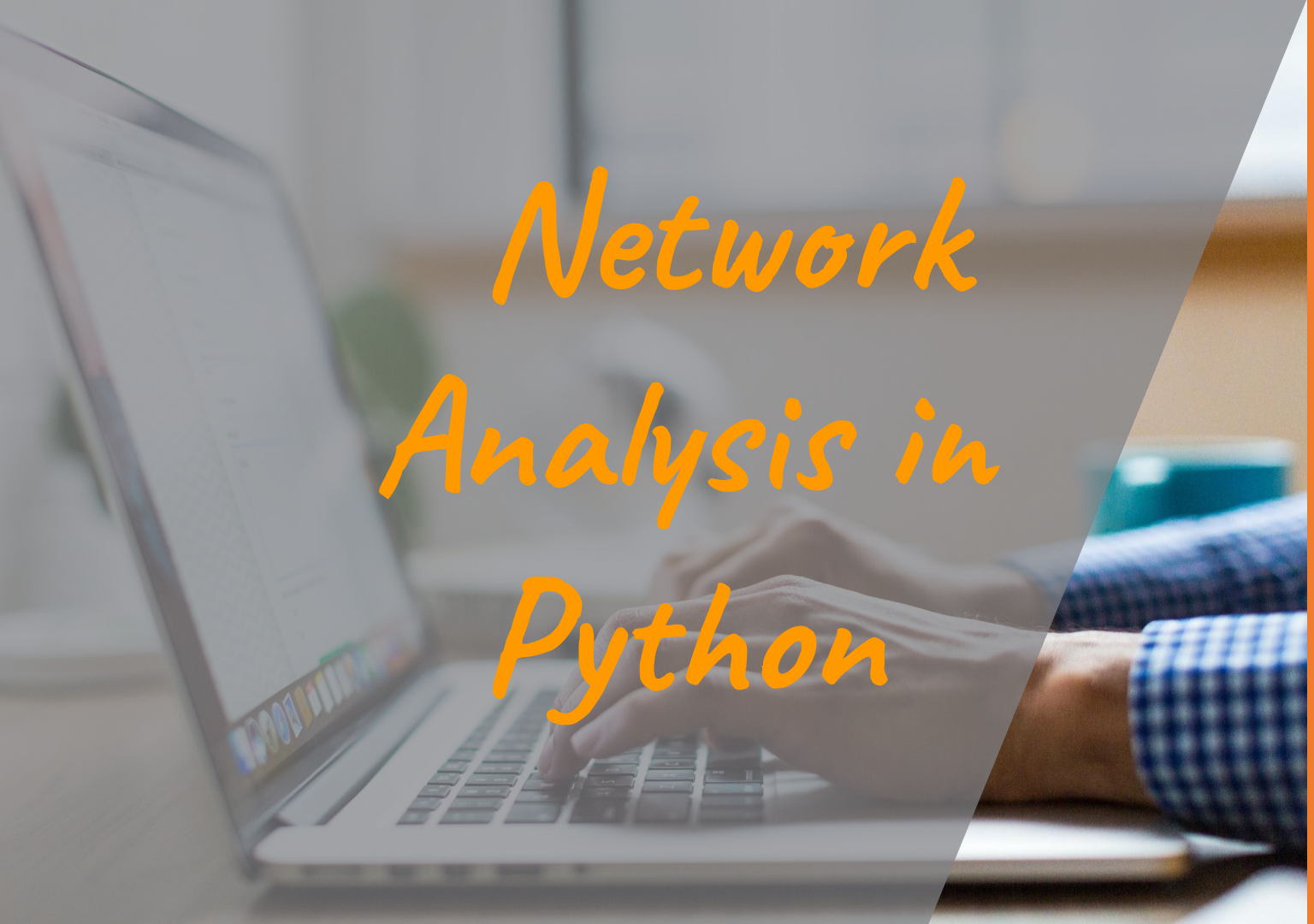


Network Analysis in Python

Agenda of this Video...

1. Introduction to the field of Network Analysis.
2. Real life use cases of Network analysis.
3. Network Analysis terminologies.
4. Types of Networks (or Graphs).
5. Visualization of a network.
6. Introduction of networkx library in python
7. Degree (neighbours) Importance of a node in the network (graph).
8. Degree centrality Importance of a node in the network (graph).
9. Intro to path Finding using networkx Library
10. BFS (Breadth First Search Algorithm) for path Finding
11. Task for you.

Introduction to the field of Network Analysis

Network analysis as the name suggests is the analysis of networks (also known as graphs).

Now, the real question is “What is a network?”

Network is a graph of interconnection between some objects or persons.

So, in network analysis, we study this graph for finding some meaningful relationship in the graph.

Real Life use cases....

1. Social media network analytics for platform such as Facebook, Twitter etc. How do you think facebook recommends you friends?
2. Citation network for research topics
3. Analysing Transportation network (Useful for companies like Uber, Ola etc)
4. Employee network which might be useful for the HR in taking some decisions.
5. It is also used for marketing of products.

Network analysis terminologies...

1. Graph = Graph is nothing but a interconnection of various object which are represented as nodes.
2. Node = Each individual object in a graph (or network) is known as node and each node can have "metadata" associated with it.
3. Edge = Edge is a line or arrow which signify the actual connection between two nodes. Edge can also have "metadata" associated with it.

For example, we can have a network (graph) of all facebook users. In this graph, each node will represent the user while the friendship between two users will be known as the edge. Each user can be metadata associated with it like name of the user, age of the user etc. The metadata will be common for both user such as the date on which the they became friends.

Types of networks (graphs)

Networks can be many types but two of those are listed here..

a) Directed Graph

b) Undirected Graph

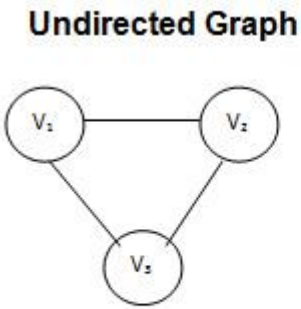


Figure 1: An Undirected Graph

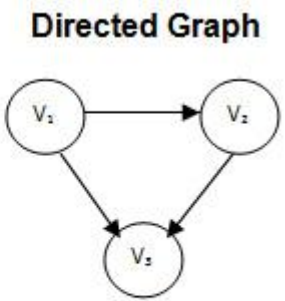
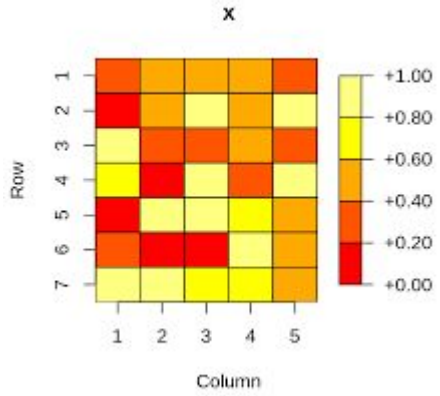


Figure 2: A Directed Graph

Visualization of a network

1. Network Diagram where each object (node) is represented as a circle and each relationship is represented as arrow or line depending upon whether the network is directional or not.
2. Matrix Plot: It is a plot where each node is written in columns and as well as in rows and then each cell is filled in if there is a connection between those two nodes.



Networkx library for doing network analysis

```
#Step. 1 Import the networkx library.
import networkx as nx

#Step. 2 Create a Graph object
graph = nx.Graph() # non directional graph

#Step.3 Add nodes one by one
graph.add_node(node1)
graph.add_node(node2)

# or Add nodes in one go
graph.add_nodes_from(pass_a_list_with_all_nodes)

# Add edges one by one
graph.add_edge(edge1)
graph.add_edge(edge2)


# or Add all the adges in one go
graph.add_edges_from(pass_a_list_with_all_edges)

# Adding the metadata of any node or edge by
graph.node[name_of_node][name_of_metadata] = value_of_metdata
graph.edge[name_of_node][name_of_metadata] = value_of_metdata

# Draw the network
graph.draw() and plt.show()
```

Importance of a node (degree)

I want to answer a questions like “Which node is the most important node in the network?” and also “How important it is in comparison to other nodes?”

There are various metrics of importance out there but one of the most intuitive and most easy to understand is “Number of neighbours a node has”

Higher the number, higher the importance.

Let us see how to check the importance of a node in networkx library

Importance of a node continued (degree centrality)

What we just seen was an example of “degree” of a node?

Degree of a node is defined as the number of neighbors a node has.

Degree centrality is another metric which is computed after dividing the degree with the maximum number of node that a node could possibly have.

Path finding problem

What else can we do in network analysis apart from node importance that we just discussed?

There is one more concept which is very popular in transportation networks and that problem is to find the shortest path in a network.

So, the question is..

“How can i reach from node 1 to node 2 in a minimum number of steps?”

BFS algo for Path finding

BFS algo is one of the earliest algorithms for path finding and it quite intuitive.

So, let us say we want to reach from node A to node B, we will start from node A and look at its neighbors and ask

a) Is node B present? If yes, stop else
   i) Loop over all the neighbours of node A and check their neighbors and this steps keeps going recursively until we hopefully get to node B

Task for you.

1. Make a hypothetical Network of Social Media Platform and make some 50 + nodes for people with their metadata. You could do this manually or get some data of people from Internet.
2. After that, define some edges between these people. Try to define at least 20 + edges.
3. After that, try to compute the degree and degree centrality.
4. Try implementing the BFS algo and try to find some shortest paths. Does they tell you anything about the node in general?

Thank you