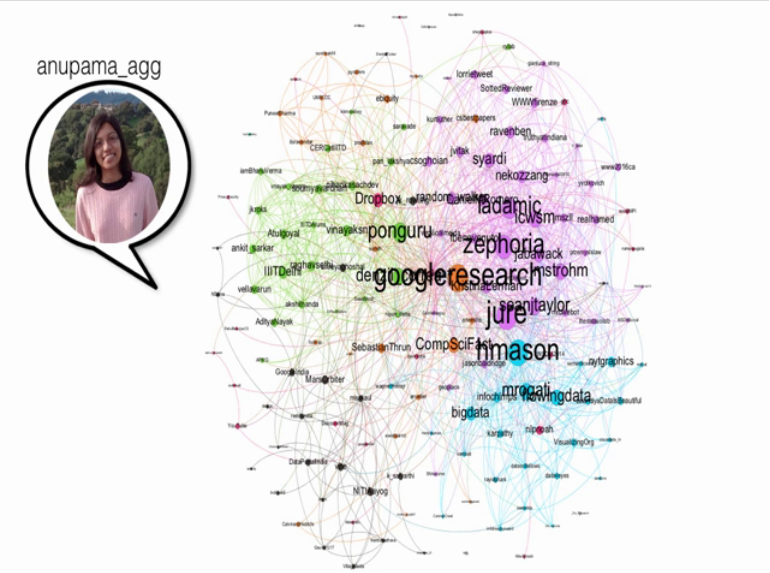
Privacy and Security in Online Social Networks
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 21
Tutorial 4 Social Network Analysis

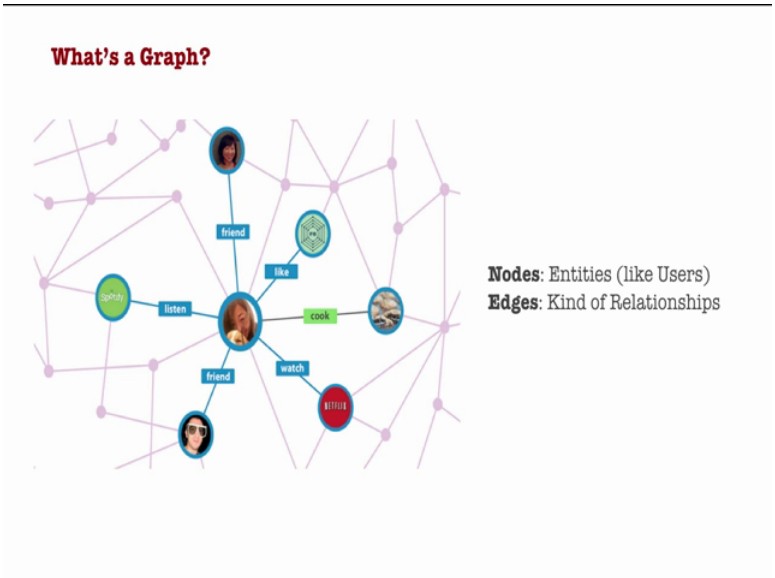
So far we know about various online social networks like Facebook and Twitter. In this tutorial, we will learn how to represent social media data in a graph format consisting of nodes and edges. We will also learn the basics of social network analysis.

(Refer Slide Time: 00:29)



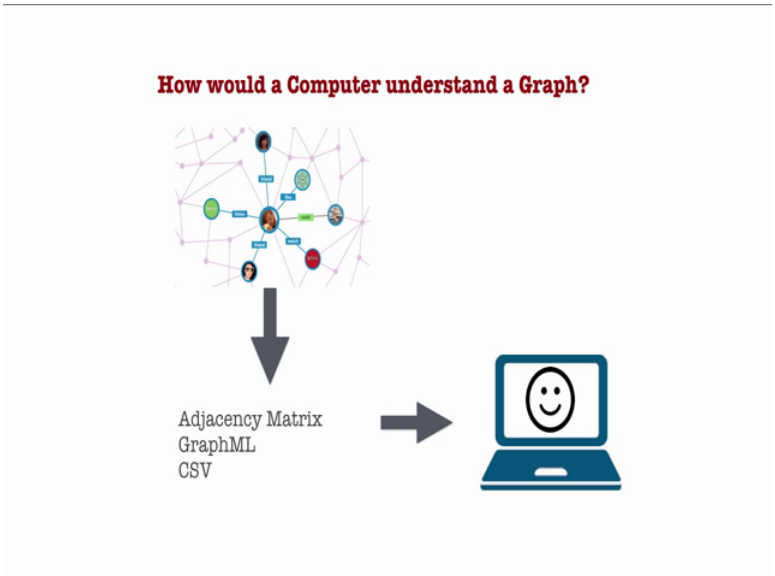
This is an example of my Twitter followees network graph. Nodes are my followees and an edge signifies that the node is following the other node.

(Refer Slide Time: 00:41)



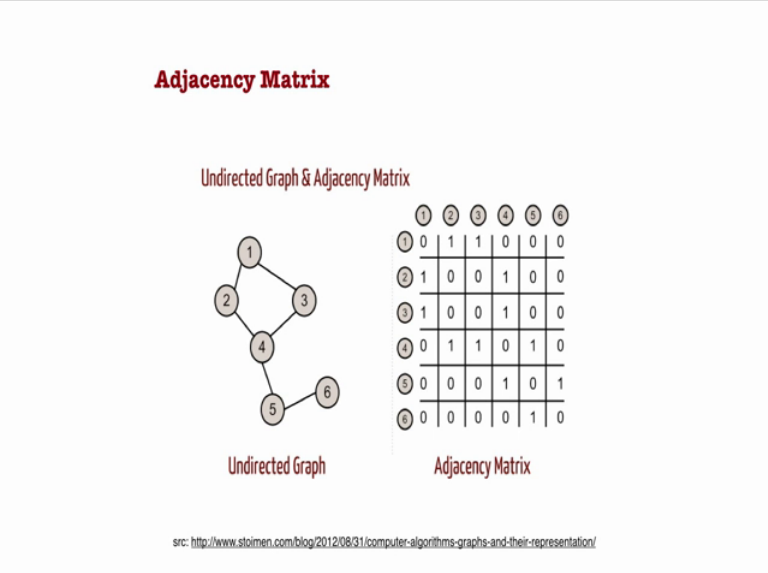
A graph is a data structure which consists of a finite set of nodes and edges. Nodes represent the entities of social network like users, pages or groups. Edges define the relationships between various nodes, for instance, a directed edge from user a to user b can mean that a follows b or an edge between a user a and the page can mean that user likes that particular page.

(Refer Slide Time: 01:11)



However, how would a computer understand such a node edge format? There are various ways to represent a node edge graph; some of the most widely used methods are adjacency matrix, graph ML format and CSV files. Let us look at what they are.

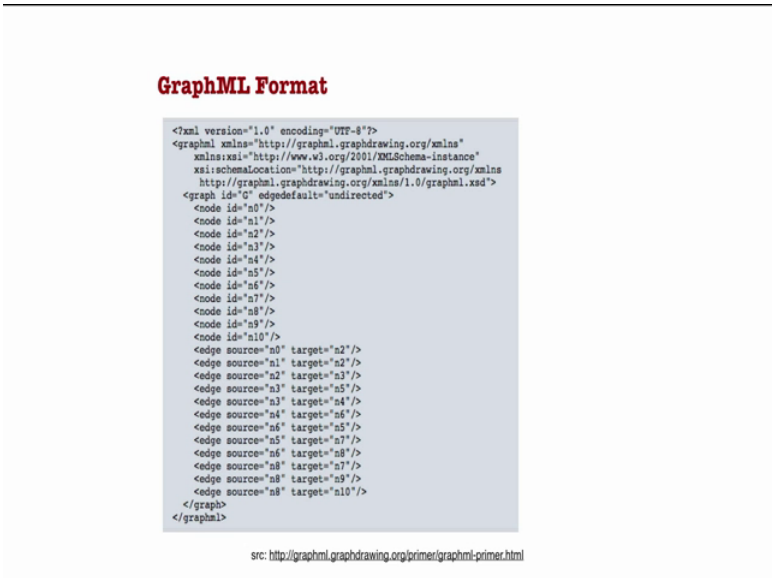
(Refer Slide Time: 01:31)



An adjacency matrix is a 2 dimensional square matrix whose size is equal to the number of nodes in the graph. In this particular example, since the graph has 6 nodes, the size of the corresponding adjacency matrix is 6 x 6, this is at intersection of ith row and jth column is 1, if an edge exist between nodes i and node j, otherwise 0. In this example, there is an edge from node 1 to node 2 and 3. Therefore, the cell at the intersection of first row and second column gets a 1.

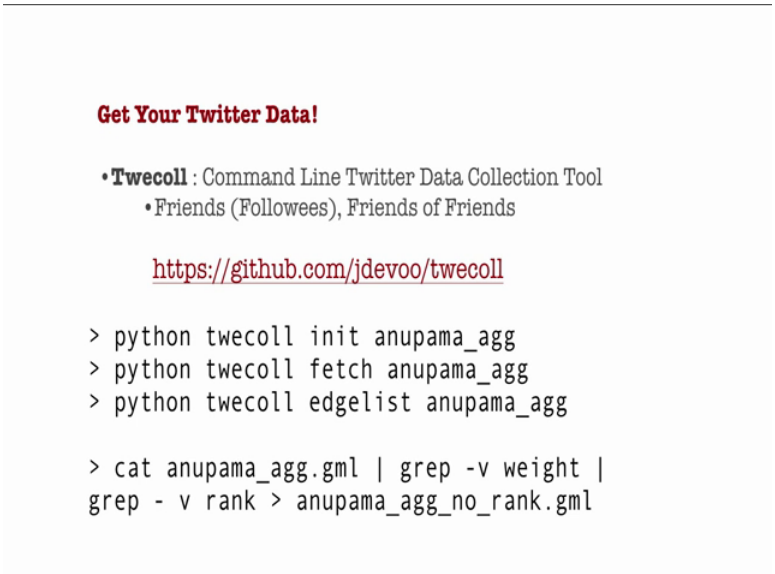
Similarly, the cell at the intersection of first row and third column also gets a 1. However, there is no edge between node 1 and node 4. Therefore, the cell at the intersection of first row and fourth column remain 0. The rest of the adjacency matrix is also filled in similar manner. Adjacency matrix can be very easy to construct using an array data structure in any programming language; however, if the input graph has high number of nodes and less edges then the resulting adjacency matrix can be very sparse and space consuming. Therefore, let us look at another way to represent a graph, graph ML format.

(Refer Slide Time: 02:53)



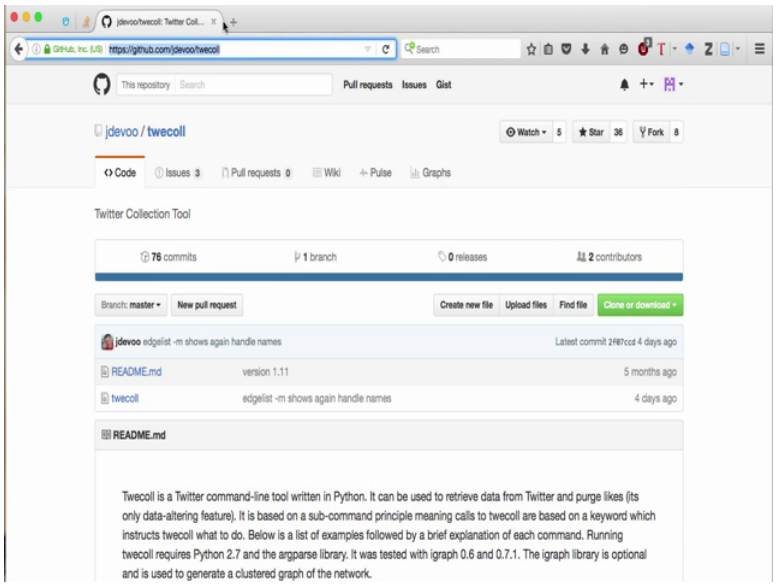
Graph ML is an xml file format for graphs. It consists of an xml file containing a graph element within which is an unordered sequence of node and edge elements. Each node element should have a distinct id attribute and each edge element has source and target attributes that identify the end points of an edge between two nodes, in this example, we have a graph with 11 nodes that node ids n 0 to n 10. The first edge element signifies that there exists an edge between node n 0 and n 2. Now, we have learnt how to collect your own twitter following network graph in graph ML format.

(Refer Slide Time: 03:42)



We will be using twecoll, a command line tool to get twitter data in graph ML format. Using twecoll, we will collect our followees information which is also called friends and friends of friends information. Let us see how that works. Go to this particular URL and let us fetch the code of twecoll.

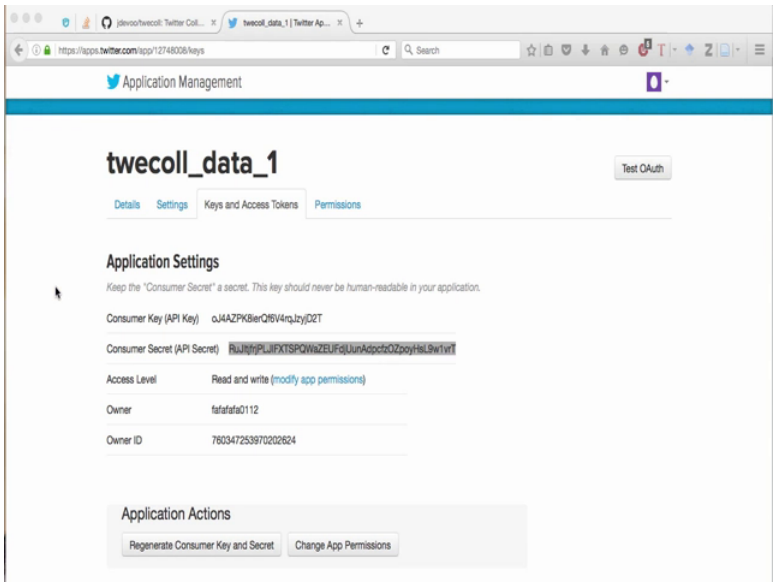
(Refer Slide Time: 04:07)



Download the code in a folder and unzip it or clone this git repo to get the code. Now, go to the terminal and navigate to where you have downloaded the code. I have downloaded it to a folder name NPTEL tutorial and I will go there. Now, we will start collecting data as first step we will initialize the program to authorize a twitter app, run the command python twecoll init, followed by the user name.

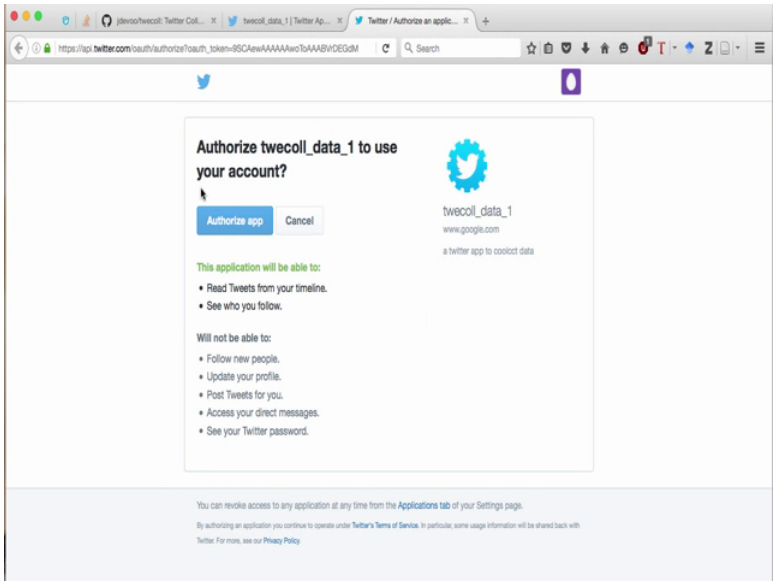
In this case, I am using a dummy user name for the demo purpose. The program will prompt you to go to apps dot twitter dot com and enter the consumer key of a Twitter application. Let us go back to the browser and navigate to apps dot twitter dot com.

(Refer Slide Time: 05:08)



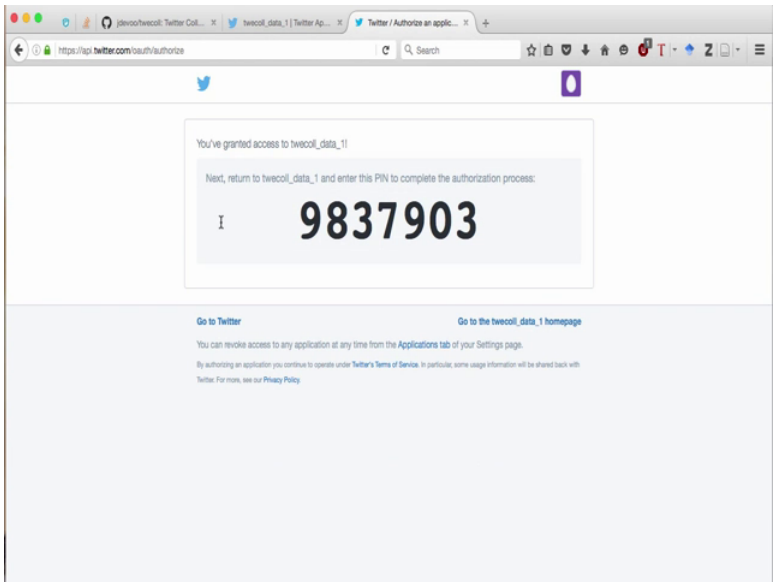
Now, I have already created an app, you can either use a previously created app or use a new one, go to the keys and access token and get hold of the consumer key. Then it will also ask for the consumer secret. Next, it will generate a link which will ask you to enter a pin, copy paste that URL into a browser and you will be redirected to the Twitter application authorization page.

(Refer Slide Time: 05:52)



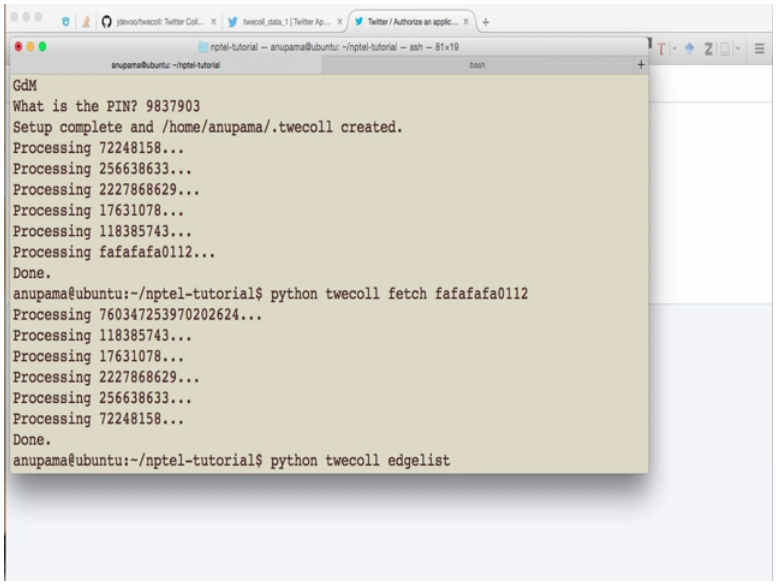
Authorize the app and get hold of the pin.

(Refer Slide Time: 05:58)



Copy, paste it in the terminal and your initialization will be complete. Next, it will start processing the ids of the followees of your twitter account.

(Refer Slide Time: 06:01)



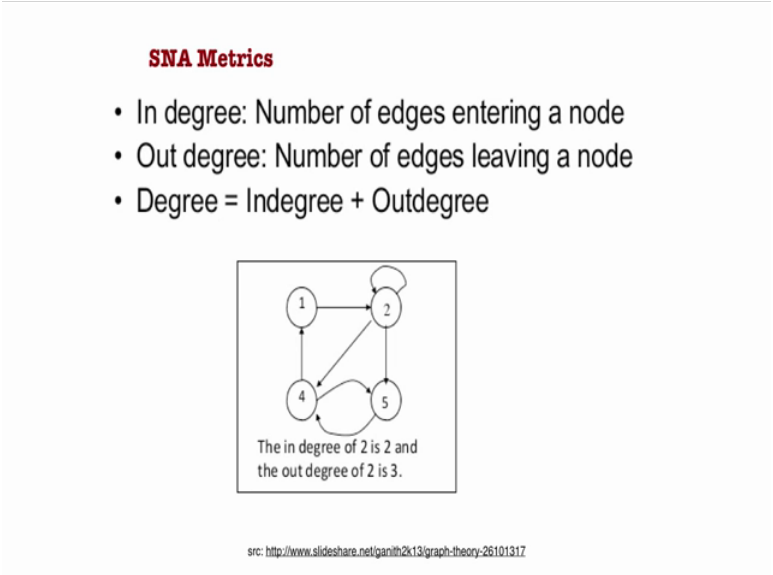
Now, we will collect the friends of friends' information of your twitter account. To do that, fire the command python tweecoll fetch followed by a user name. This will start getting the list of followees of your own followers.

Next, we want to build the edges between the followees and their followees, to do that we will run the command python tweecoll edgelist followed by the user name.

(Refer Slide Time: 06:55)

This command will generate the edges between your followees and their followees. Do not worry, if you get a notification which says no module named I graph, you can list the files in the folder and you should be able to see a file named, your user name followed by the extension GML. This is the file which contains the information of your followees and their followees and we will use Gephi tool to generate a network graph of a twitter data.

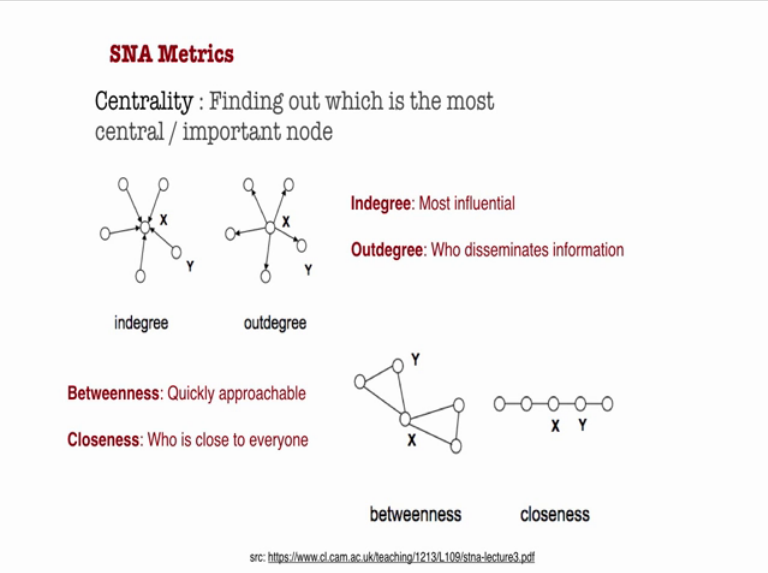
(Refer Slide Time: 07:30)



Before we move on to graph visualization, let us first look at the basics of social network analysis. The most commonly use SNA metric is degree, in a directed graph, in degree is

equal to the number of edges entering a node. In this example, for node 2, edges are entering it from node 1 and there is a self loop from itself therefore, it is in degree is two. Out degree equals to the number of edges leaving a node in this example, edges are going away from node 2 to 4 and node 2 to 5, node two also has a self loop. Therefore, the out degree of node 2 is 3. Total degree of a graph is calculated by summing the in degree and out degree.

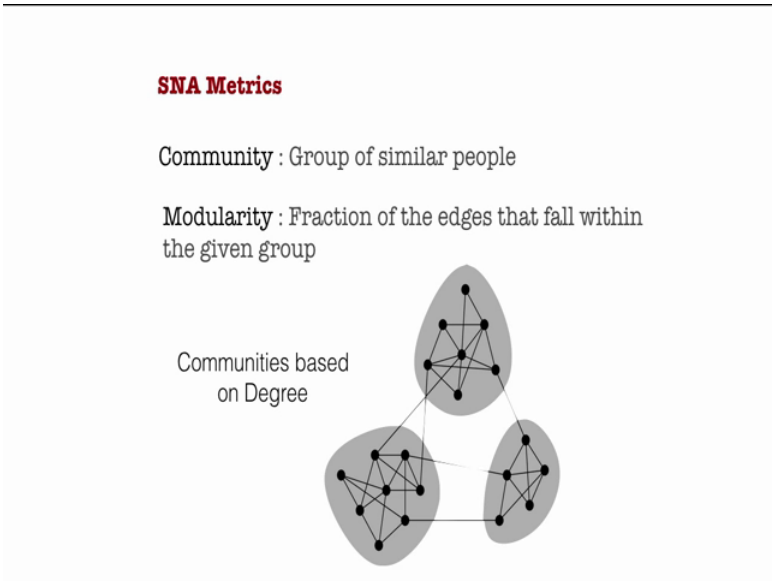
(Refer Slide Time: 08:22)



One of the other useful SNA metrics is centrality that is finding out which is the most central or important node. There can be various ways to define centrality. Let us look at them one by one; in degree centrality finds the node with highest in degree. It can signify the most influential node or in case of Twitter follower graph, the user with highest number of followers.

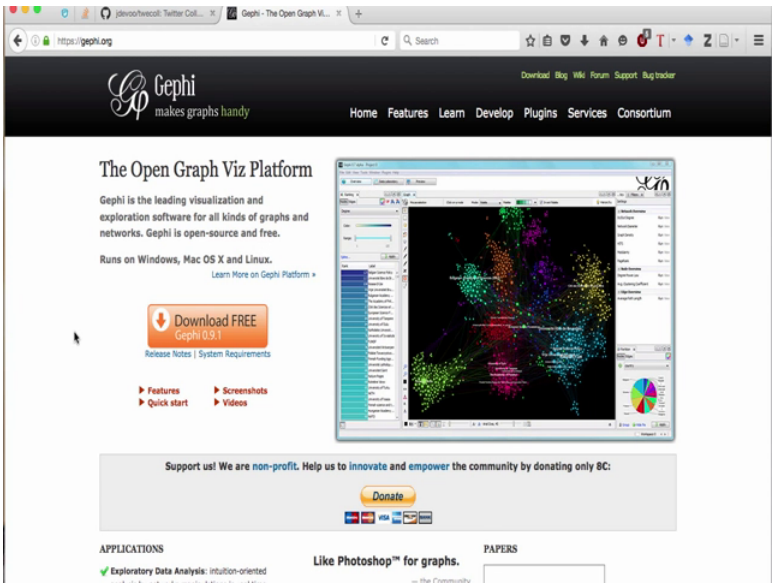
Out degree centrality helps in locating the node whose out degree is the highest, other ways to measure centrality are betweenness and closeness. Betweenness centrality is equal to the number of shorter paths from all vertices to all others that pass through that node, closeness centrality helps to find the node with the lowest total distance from all other nodes.

(Refer Slide Time: 09:18)



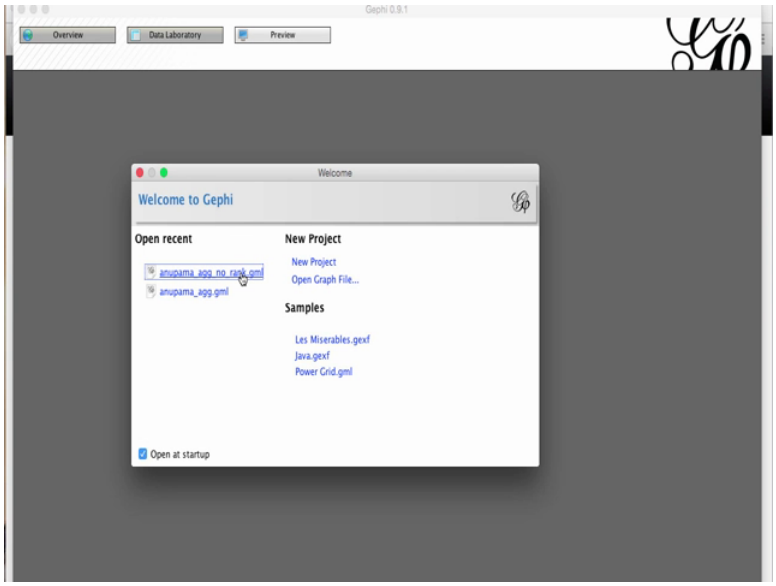
Let us also briefly look at community in a graph. A community is a group of similar or strongly connective nodes. The measure to define the strength of a community is modularity, which means the fraction of edges that fall within the given group. Now, we will look at a tool called Gephi for graph visualization.

(Refer Slide Time: 09:39)



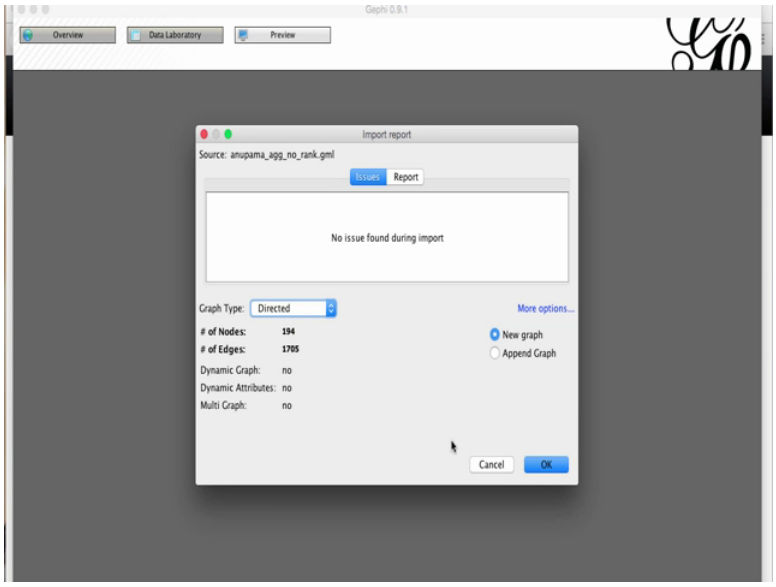
Now, go to Gephi dot org and download the Gephi installer for the operating system you are using. I have already downloaded and installed Gephi on my machine.

(Refer Slide Time: 09:53)



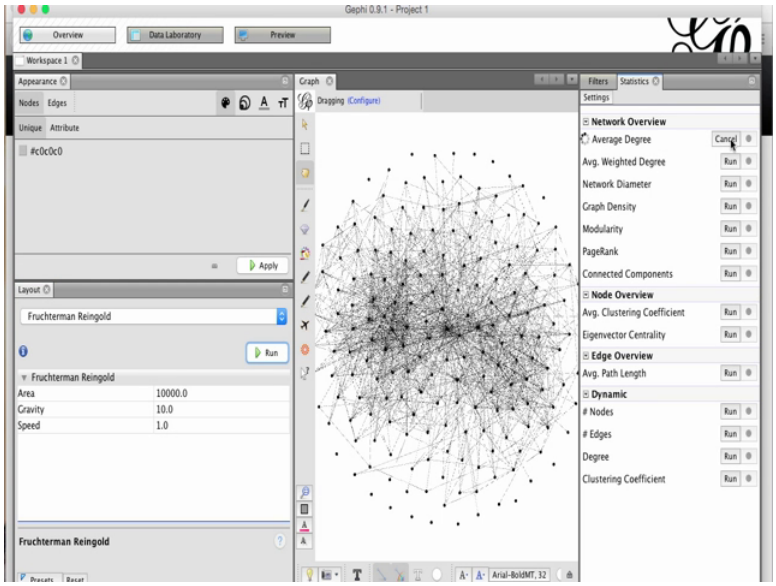
Start Gephi and open the graph ML file which you previously created.

(Refer Slide Time: 10:05)



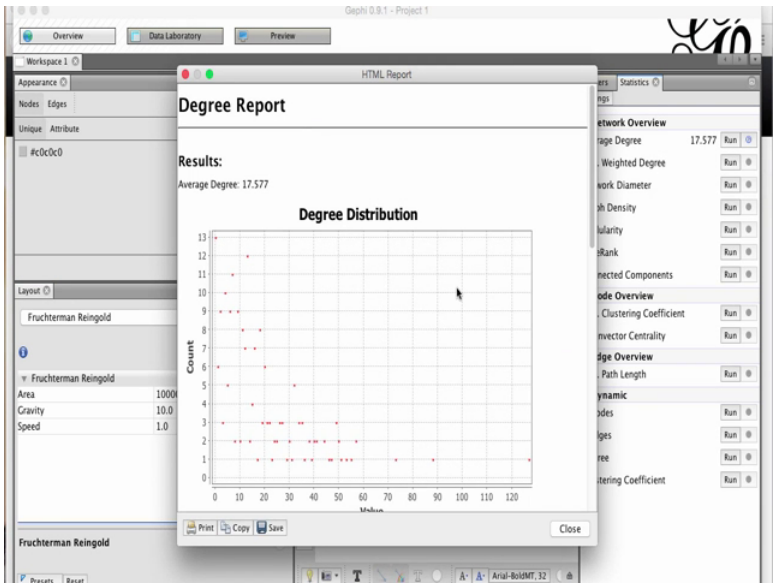
This will show some of the statistics about the graph which is the number of nodes and the number of edges, click on OK and soon your graph will be loaded. If you do not see anything, click on the overview.

(Refer Slide Time: 10:28)



And a graph will initialize, in case your graph is very dense you can change the layout of the graph. From the left panel click on the drop down menu and choose the different layout. Let us choose the reingold algorithm and run it, you can stop it after a while. Now, let us customize this graph to make it look prettier. Before doing that, notice that on your right, you will be able to see a lot of network statistics, for instance, the average degree, average weighted degree, network diameter, etcetera, you can click on run to generate the reports.

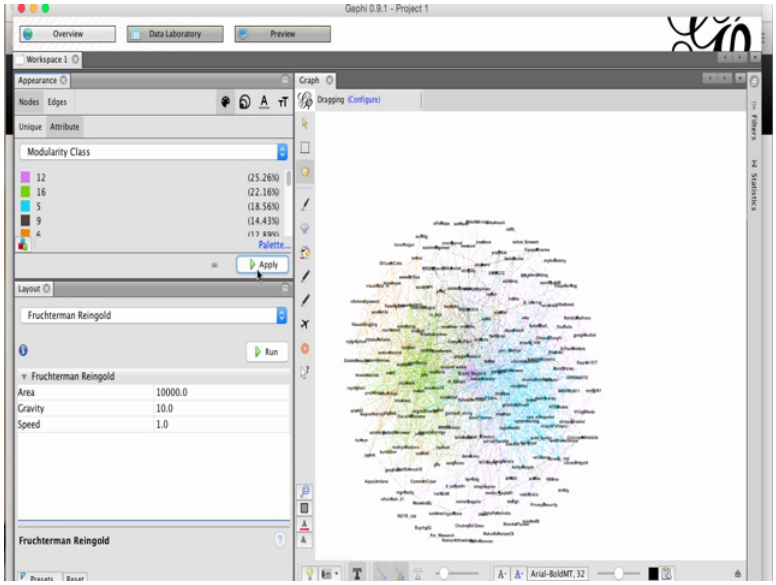
(Refer Slide Time: 11:21)



For instance, this is the degree distribution of my Twitter followee graph. You can generate similar statistics; we will also run the modularity to understand that what are the numbers of communities in our twitter followee graph?

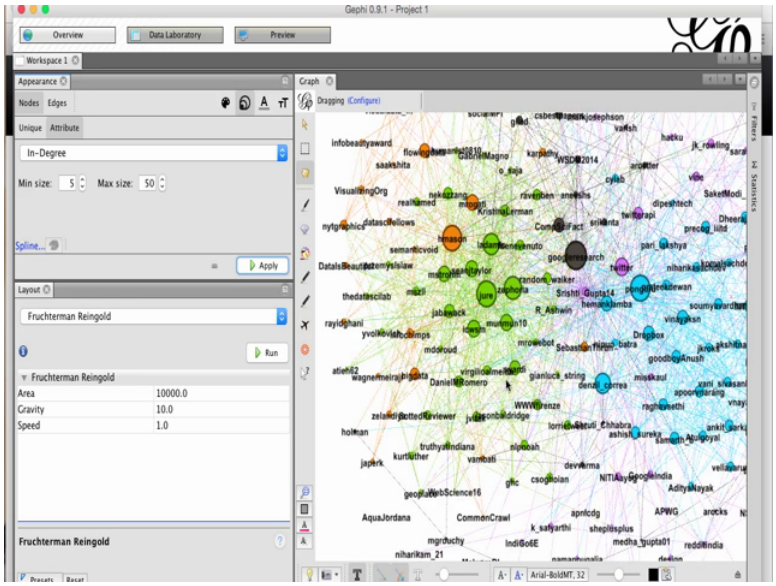
Next, let us customize this graph, the nodes are our followees, but we are not able to see their labels. Therefore, click on 't' to generate the labels, we can also customize the color of the nodes and edges. In the left panel choose the color pallet, select nodes, click on attributes and then select modularity class.

(Refer Slide Time: 12:26)



Click on apply; we can even customize the size of the nodes. Click on the size pallet nodes and then attributes and choose an attribute. Let us select in degree, so that we can get a sense of which are the most important nodes in a graph, click on apply this will change the color and size of the nodes.

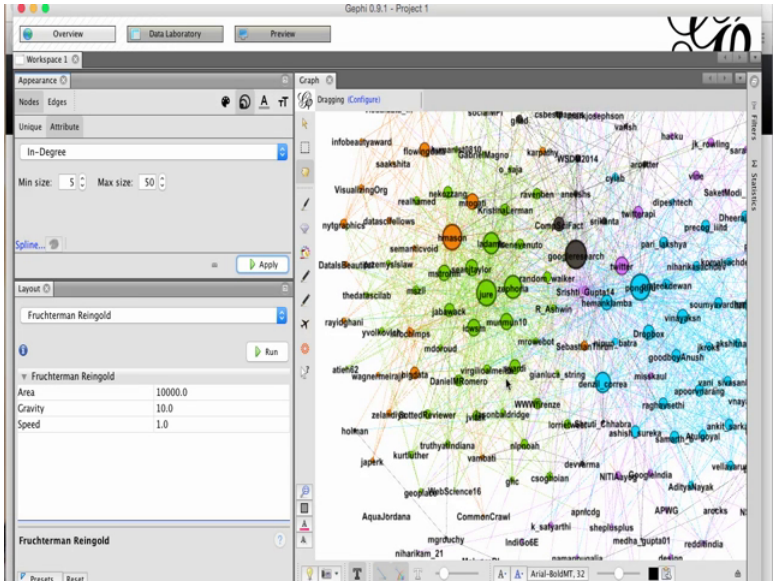
(Refer Slide Time: 12:57)



You can zoom in and see that, which are the most important nodes in your graph and what are the different colors according to the communities which they belong to, we can further customize this graph by changing the appearance of the edges. Let us go to preview.

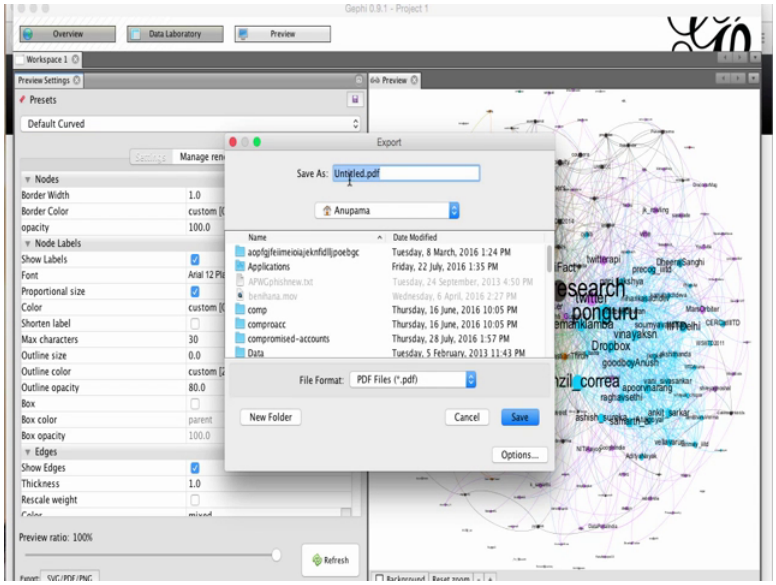
In case, here you do not see any window tab and then click on the preview settings, again click on window and click on the preview. Now, from the presets select default curved and press refresh.

(Refer Slide Time: 13:35)



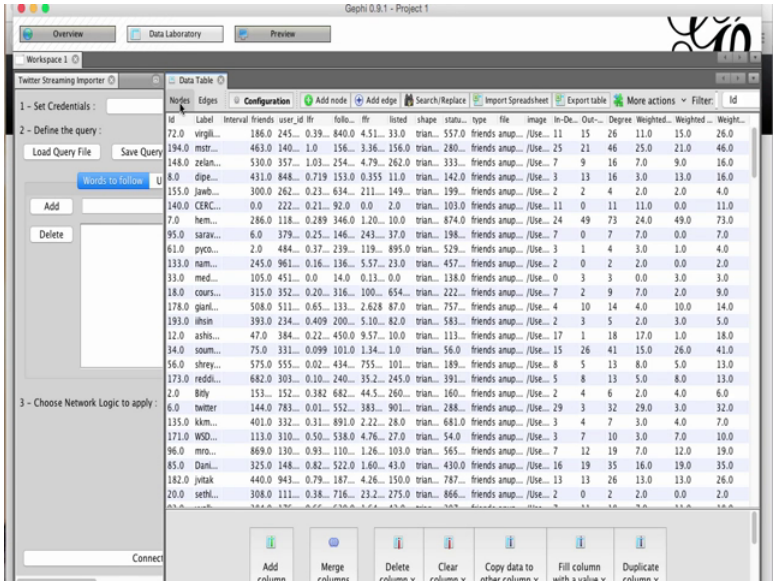
This will make the edges curved instead of straight. Now, you can save this graph as svg, pdf or png format by exporting it from the left panel and clicking on export.

(Refer Slide Time: 13:50)



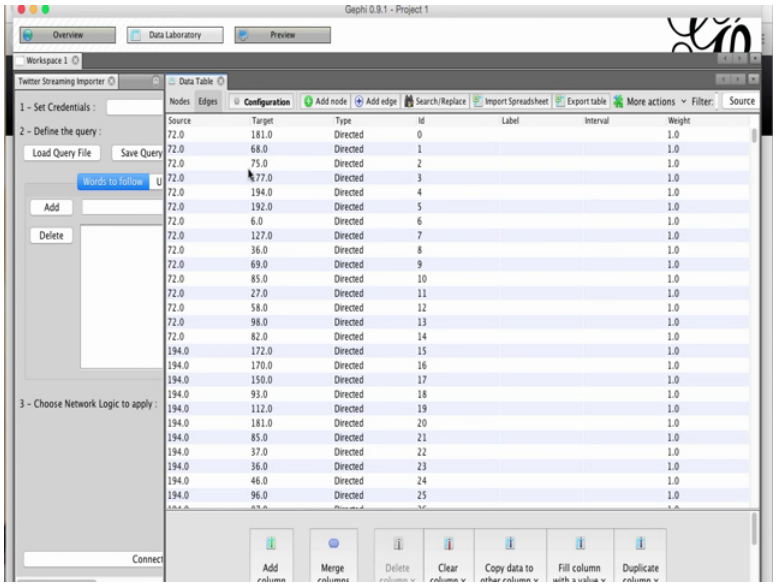
You can also browse the data which is used to generate this graph by going to data laboratory, click on the data laboratory tab.

(Refer Slide Time: 04:14)



This will show you the list of the nodes, edges and the relationships between them. There will be two tabs, nodes and edges. The nodes tab will list the information about the nodes and the edges tab will list the source target information as we stored it in the gml file.

(Refer Slide Time: 04:31)



Now, we know how to generate a network graph from social media data.