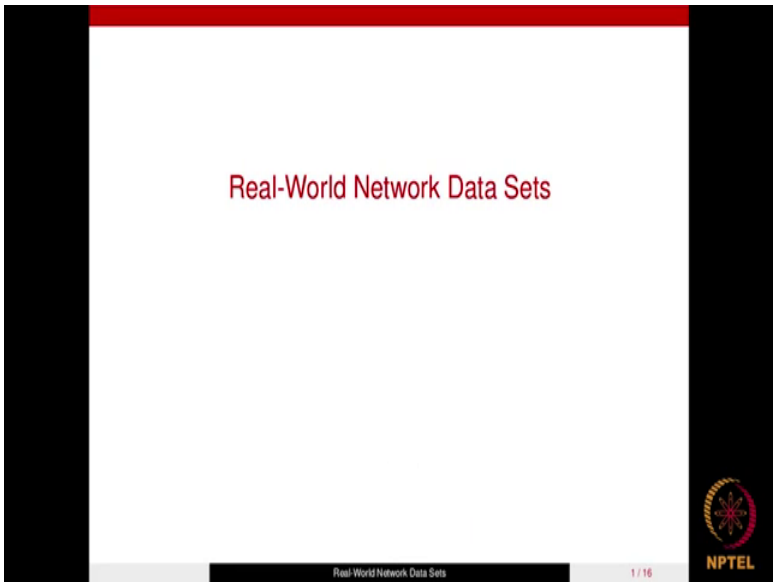
Social Networks
Prof. S. R. S. Iyengar
Department of Computer Science
Indian Institute of Technology, Ropar

Lecture – 19
Handling Real-world Network Datasets
Datasets: Different Formats

(Refer Slide Time: 00:05)

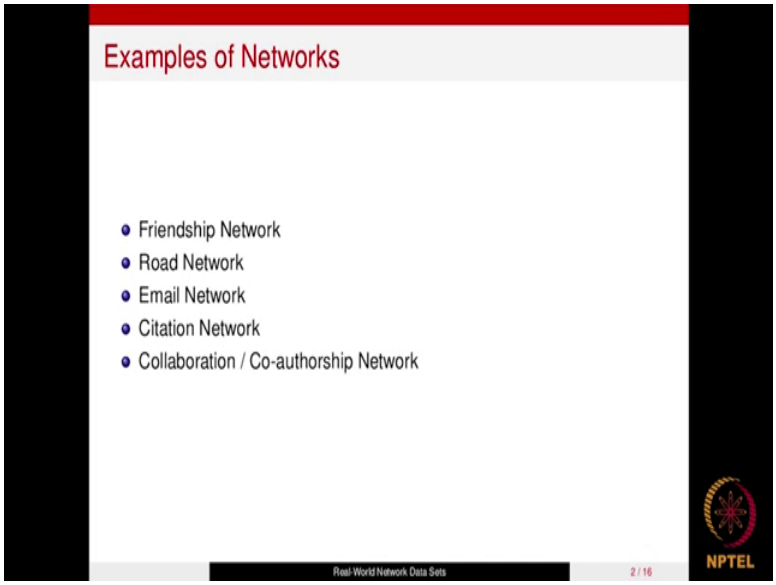


Hi everyone, so, after having got introduced to the course and after having seen the various packages that you can use for the energy the social networks, I am sure you might be all geared up to learn the concept from social networks. I would say if you only learn the concepts from this course and do not apply them to nearby situations, you may not be able to get an enough understanding. So, it is important that whatever knowledge you required you apply them to real world scenarios and for that purpose you obviously, need a datasets from the real world situations.

Now, how do we get that data? Fortunately there are various individuals and organizations who have gathered network datasets for different situations and topics and the good thing is that they have kept this data online publicly available for anyone to access and analyze more over these datasets are available in different formats. So, in this video, we are first going to take a look at the formats into which the datasets are available and after that we are going to look at the sources from which we can download these datasets and

then in the subsequent videos we are going to analyze these datasets which we will download from the internet.

(Refer Slide Time: 01:31)



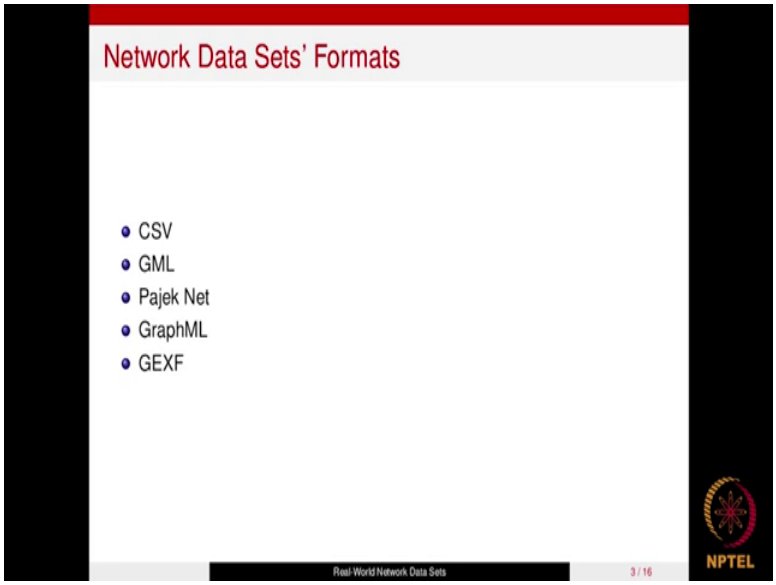
So, let us get started and before we get into the formats of these data sets I want you to take a look at the networks on which data is available of course, they are not limited we have data sets available on diverse range of topics, I am just going to give you a few examples of networks over here. So, we have friendship networks where the nodes are people and there will be a link from one node to the other node if these 2 nodes are if these 2 people are friends with each other similarly we have road networks where the nodes will be cities and there will be link from one node to the other node if these 2 cities are connected to a road.

So, usually these friendship networks and road networks are undirected networks and if you look at directed networks we have email network where the nodes will be the people and there will be a directed edge from node A to node B, if person is sent an email to person B, another example of a director network is citation network. So, you might be knowing that in scientific scenario if a paper uses some information from another paper then this paper sites the second paper. So, that is what information is captured by a citation network. So, this is of course, directed network.

Another example of a network from scientific domain is collaboration network. So, scientists collaborate with each other for some research project or some scientific

development. So, this sort of information is captured by collaboration or co authorship network which captures which 2 people collaborated with each other or co authored for a people. So, these where just few examples of networks we are going to come across a number of them in on different topics. So, as he said these networks are available in different formats.

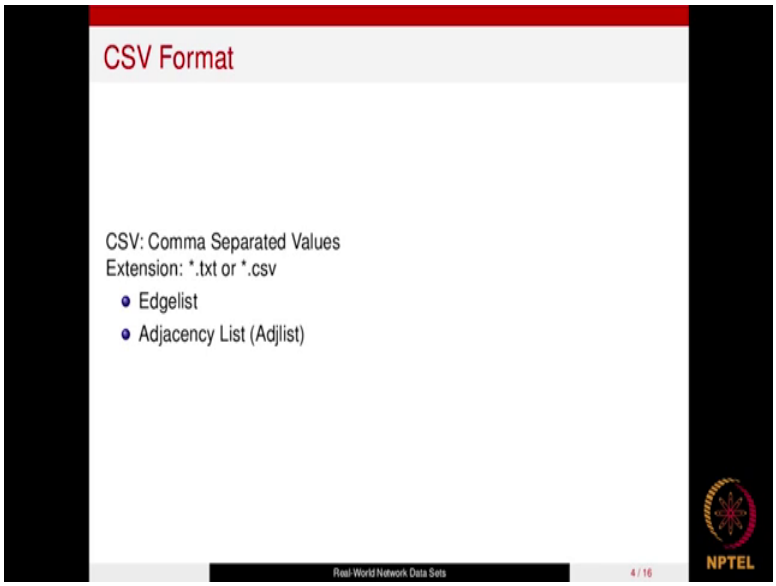
(Refer Slide Time: 03:27)



Let us take a look at these formats one by one. So, we have CSV format, we have GML format, we have Pajek format, we have GraphML format, we have GEXF format.

So, I am going to take these networks one by one, I am going to tell you the basic features of these formats and the basic structure that these that the files of these networks carry. So, apart from these formats you might also come across from two three more formats, but these 5 are the most commonly used formats when you download some data sets from internet that might most likely been one of these formats. So, we are going to take a look at these formats in detail will take them one by one.

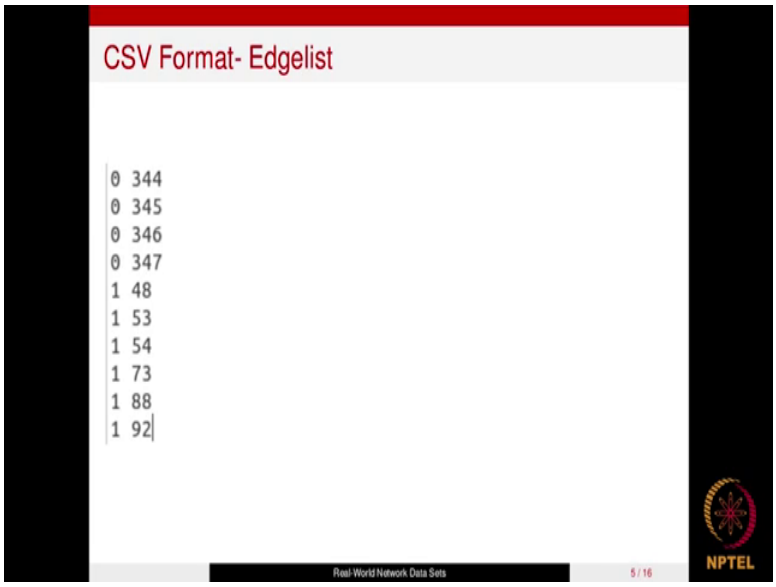
(Refer Slide Time: 04:16)



Let us see; what is CSV format? So, we might be knowing that CSV stands for comma separated values this file will be in form having the extension either dot txt or dot csv.

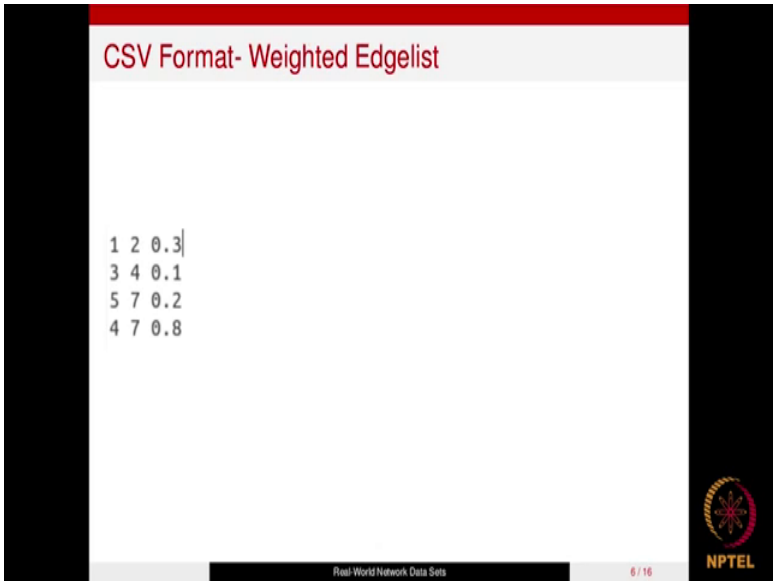
Now, CSV format file can have 2 more types it could either be in Edgelist list format or it could be in adjacency list format. So, if you are from graph background you might know what an adjacency list means and what an Edgelist mean any anyway I am going to give an example of both these formats.

(Refer Slide Time: 04:54)



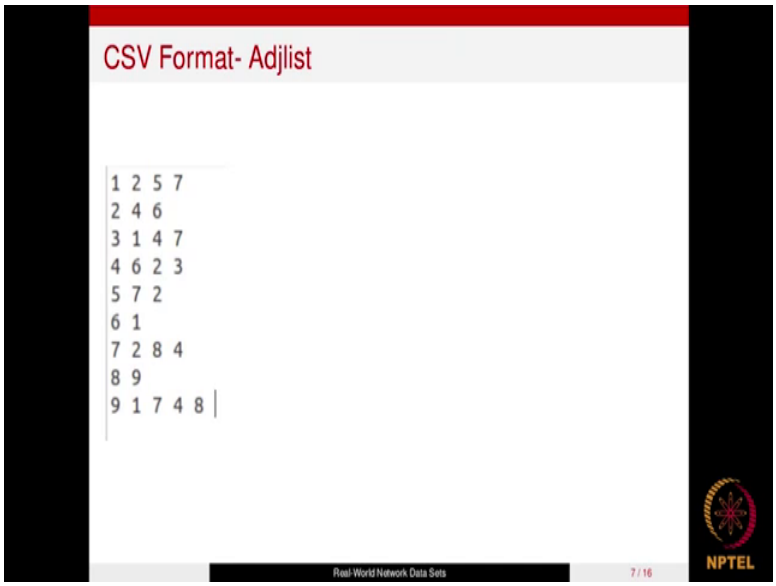
So, let us see what is an edge list format you can look at the simplicity of this file this is a snapshot from one of the files which is in CSV format which is in Edgelist type here every row contains 2 values first value is the source node and the second value is the target node. So, this indicates the row indicates that is going to be an edge from 0 to 344, 0 to 345, 0 to 346 and so on.

(Refer Slide Time: 05:38)



So, basically the file will contain a list of the edges and in case you want to add some weight to these edges you can also do that for example, you want to add weights to these edges like this you can add third value to every row. So, for example, the edge which is going to go from 1 to 2 should have a 8.3. So, you can add it like this; however, there is one limitation that the edge list format carries and that you cannot add non numeric weights to the to the edges you cannot change the labels you cannot change any you cannot add an attribute to the nodes and edges. So, that is a limitation that this format carries. So, there is a trade. So, you can see that there is a trade of between the simplicity and the flexibility that a format provides.

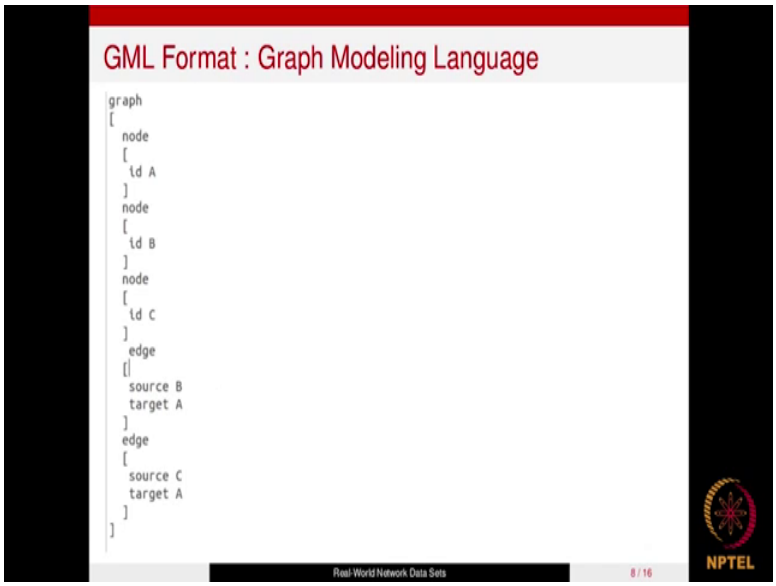
(Refer Slide Time: 06:22)



Let us now take a look at the adjacency list format. So, this is a snapshot from a file which is in adjacency list format you see every row is having more than 2 values, now what is this indicate the first value in every row is basically a source node and the subsequent values in every row are the nodes which are adjacent to this source node. So, this first row indicate that there is going to be an edge from one to 2 this going to be an edge from 1 to 5 and 1 to 7 and if you go to the second row it indicates that there is going to be an edge from 2 to 4, 2 to 6, 3 to 1, 3 to 4 and 3 to 7 and so on.

So, basically every row tells all the nodes that are adjacent to a given node. So, this is the adjacency list format again you can see the simplicity that this format provides; however, you as I already told you this format does not give you much flexibility for example, you just cannot add any sort of attribute or label to the nodes or edges if we want to add some color attribute to the nodes or edges you cannot do that. So, it all depends on your requirement.

(Refer Slide Time: 07:38)

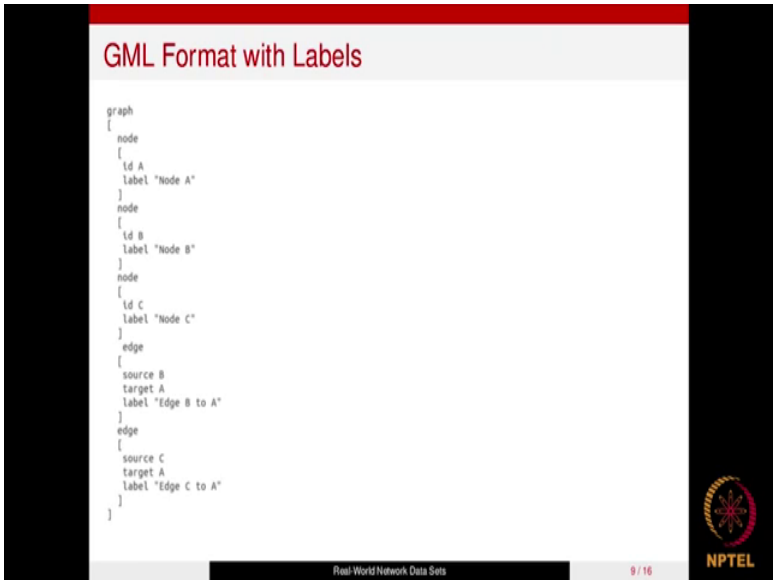


Let us go to the next format the next format is GML format with stand for graph modeling language now this is one of the most commonly used format for network datasets the reasons include the flexibility that this format provides when it comes to labeling or assigning attributes to the nodes and edges the also this format is not very complex.

Let me explain you the format of this file. So, you start with the graph keyword and when you start the square bracket and inside that you have node keyword and when you again this square bracket and we write the keyword id and then you write the id of that node we close the node and so on you keep adding the nodes. So, if there are some n nodes in the network they are going to be n nodes keyword once the nodes are done you start with the edge keyword. So, again you start the edge square bracket and then you write the source keyword and this is the source node the idea of the source node and then you write the target keyword and then the idea of the target node and we close it and so on.

So, if there are n edges there will be m edge keywords like this that is that the simplest format of a file which is in GML format in case you want to add some labels to these nodes and edges you can do that as well in this format.

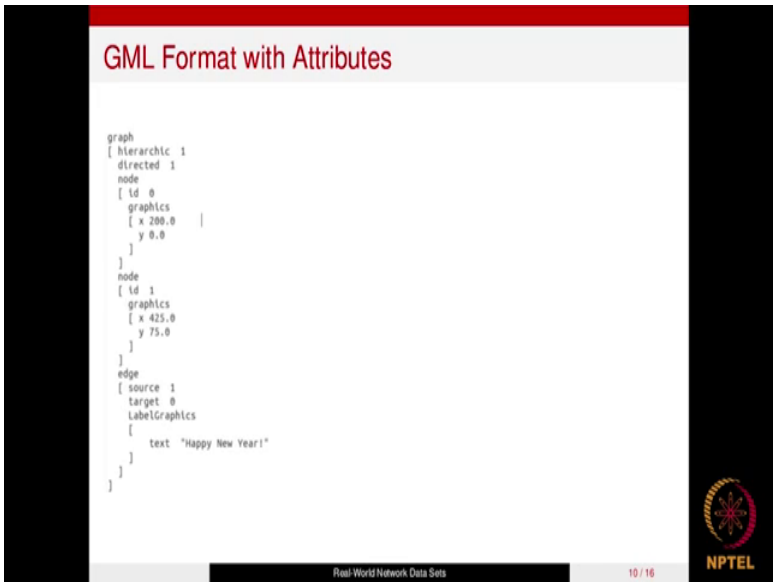
(Refer Slide Time: 09:09)



Let me show you an example for that. So, here you see we are adding a label to this node A. So, you can see this syntax you go inside the node keyword and there you write this keyword label and then you write the label of that node which is node a in this case similarly to write the label of the next node you go inside that in you write the label keyword and then you write the label that is node B and so on you do this for all the nodes individually and similarly you can do the labeling for the edges again you use the label keyword and then you write the label for the edge A which is edge B to A in this case and so on.

So, this is how to add labels and in case you want to add some attributes to the nodes and edges you can do that as well let me show you an example for that.

(Refer Slide Time: 10:00)

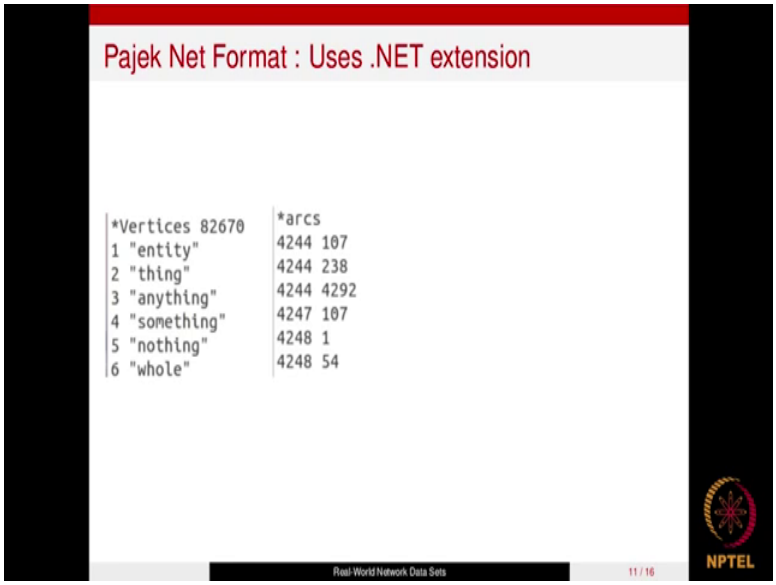


Here you see this is a graph; this is the starting with graph and inside this, there are 2 attributes here which are the graph attribute. So, apart from nodes node attributes and edge attributes there are graph attributes as well in this case there are 2 attributes hierarchic and directed and these are the values of these attributes to said the node attributes you go inside the node and when you start this attribute that you want to assign and then the name of that attribute and the value of that attribute inside the square brackets. So, this is the syntax. So, basically here they are assigning x and y positioning to the node.

Similarly, for the next node they are assigning x and y positions of the second node and here for adding an attribute to the edge again you start another attribute and start square brackets and then assign the value of that attribute now here at this point let me tell you one thing in case you are just making use of the data set download from the internet you might not need to go into the details whatever I am just now explaining. So, only in case you want to create your own datasets only in that case you should know these integrate details that I am telling you otherwise you should not worry about these details I just want to you to be aware of the syntax. So, that in case you come across some file should be able to mix sense sort of it you should be able to know what these a labels and keywords mean.

In other cases, you just want to download the files from internet, you might not need to go into all these details, let us go to the next format now.

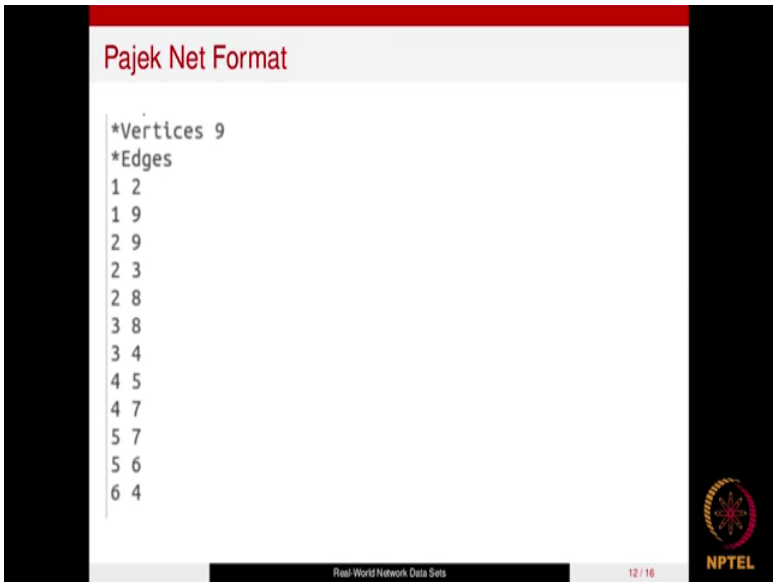
(Refer Slide Time: 11:46)



We have Pajik format which many people use for network datasets this kind of format has extension dot net and let me tell you that in some cases you might also find networks with an extension dot Paj that also indicates a Pajik format network. So, it does not make any difference. So, you have either dot net or dot PNG format networks and the way the way of handling and the syntax is completely the same. So, let me show you the syntax of the file that is a structure of the file. So, you will start with the keyword star vertices and after that will be a number return which basically mean a number of vertices in the graph or the network after that you would have on every row you have these numbers one 2 up to n and after the number you would have some string written which basically indicates the label of that node.

So, you see that for every row you have every node return and once all the nodes are done you start with the information about the edges by writing with a star arcs or star edges this will basically come below it. So, every row contains the source node and the target node and so on.

(Refer Slide Time: 13:16)

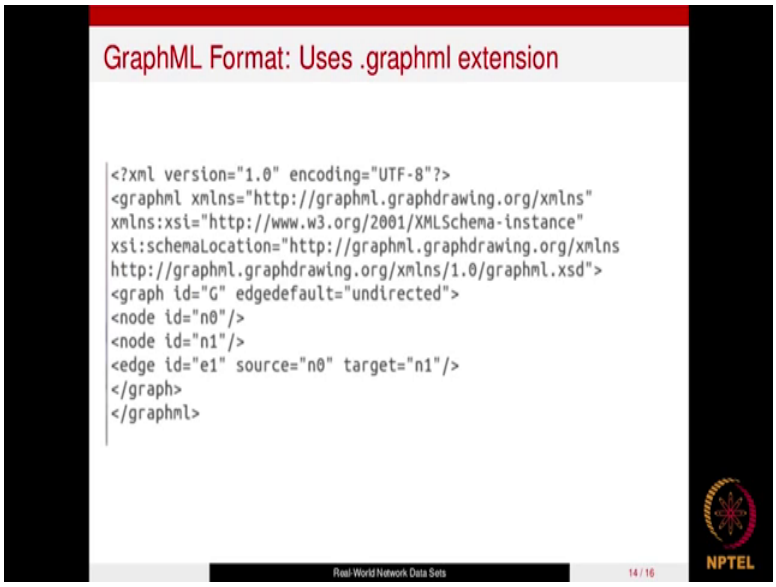


So, this is the basic syntax of a Pajik file and there is one more thing that I should tell you in case there is no labeling assign to the vertices of the network you just do not write anything below this vertices keywords because that does not make any sense you just write the there is no need to write the nodes consecutively one to n. So, you just leave that blank you just write the number of vertices that the graph contains and afterwards you start with the details on the edges and that is the same and in case you want to add some attribute to the edges you can do that as well.

(Refer Slide Time: 13:49)

Let me show you an example. So, here this is a source node and the target node and then you have value written this value basically is an attribute for this edge. So, in this case it is sort of a weight assign to this edge. So, that was about the Pajik format.

(Refer Slide Time: 14:08)

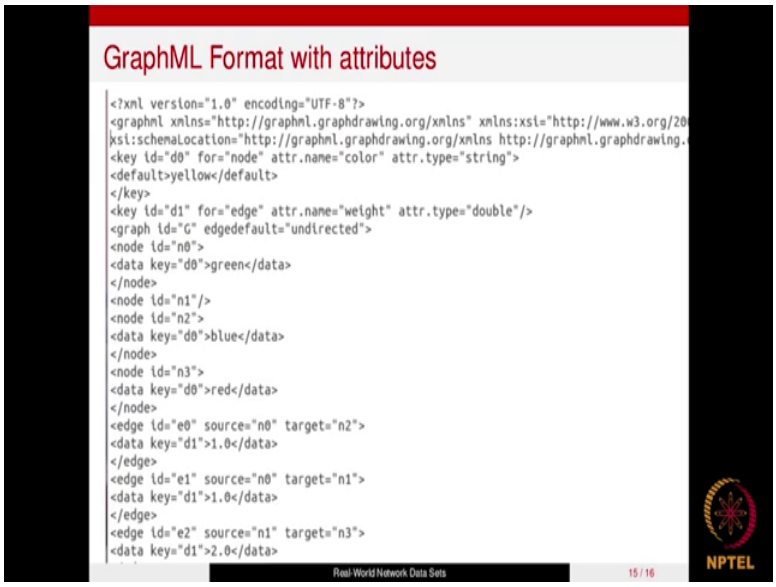


So, the next format is graph ml format ml here stands for xml basically you might be knowing what xml format is this basically provides the hierarchical structure and it makes use of various tags just like html and that is how you store the details in the form of tags. So, in a graph ml file you would have a number of tags let me quickly tell you about these tag. So, initially there will be an xml tag and after that you would have this graph ml tag and an inside the graph ml you would have a graph tag right and after the graph tag you would have number of node tags once the node tags are done you would have number of edge tags. So, this is the basic structure.

Now, let me get back here to the graph ml tag here you see some namespaces details basically you see some metadata you might be knowing the advantages of using namespaces and all that they allow different users to use different names and as long as I making use of different namespaces. So, I do not think you need to get into all that this is just a metadata about this network you can start from here you see a graph tag and then you have an attribute of graph which is edge default which is undirected and then you have node you have 2 nodes here and you have one edge here. So, this is a source and target keyword that we used inside the edge tag this is the basic structure of a graph ml

file you might you might see the structure you might feel that structure is little complex, but the flexibility that it provides in terms of assigning attributes to the nodes and edges is quite better than asking by 2 other formats.

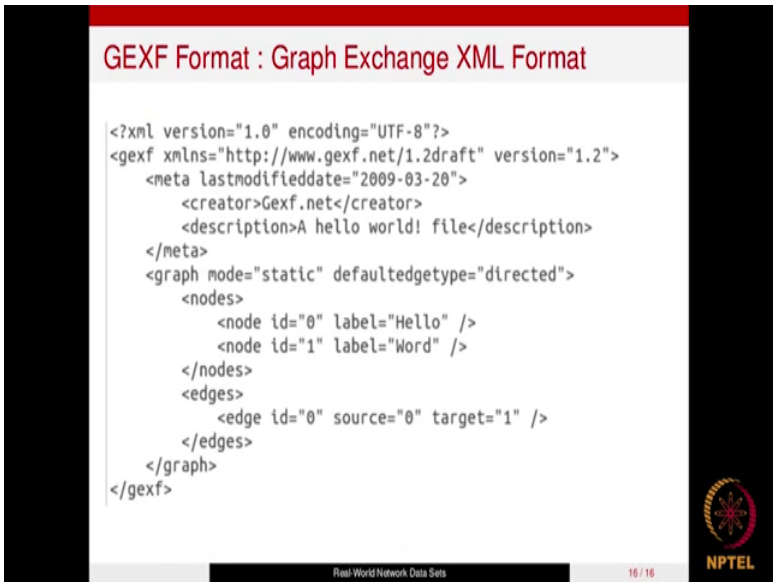
(Refer Slide Time: 16:02)



So, let us take a look at one more example from graph ml where there assigning attributes to the nodes and edges. So, after you done with this GraphML tag you start a key tag this key tag for example, is for nodes and this is for assigning colors and after that you have another key tag which is for edges and that is for assigning weights and once you done with the key tags you start the graph tag as usual and when you want to assign an attribute to a node you start a data tag and inside this data tag you have you make use of this key that is d naught you are making use of this key and you are assigning green color to this node in the next one you are assigning blue color to this node and red color and so on.

Similarly, you are assigning a weight 1.0 to the edge by making use of this d 1 key, you assigning an weight 1.0 and so on. So, this is how you assign attributes to the nodes and edges. So, that was about the GraphML format.

(Refer Slide Time: 17:07)



Let us go on to the last format in our list which is GEXF format this stands for graph exchange xml format this format was basically created by Gephi people Gephi is an open source software which is used for visualizing and analyzing social networks. So, this format is again inspired from xml as you can see it is composed of many xml tags. So, you start with an xml tag and we start with GEXF tag and inside that you have this meta tag to store the meta data about the network and then you start the graph tag inside the graph tag you have nodes tag and inside the nodes tag you have all the nodes and then you start the edges tag inside the edges tag you have the edge tag.

So, the format is little similar if it actually quite similar to graph ml format the previous one this is a little cleaner in terms of assigning various attributes to the nodes and edges. So, there is one thing that I would like to point out here. So, I want to say that do not worry if you not understanding anything out of this video because the aim of this video was just to introduce to you the main formats into which the datasets are available and in case you are not creating your own dataset you might not able to me to you know get into also integrate details in case you are just using the datasets downloaded from the internet you might not need to yourself write all these details that I just explain to you.

So, the aim was just for you to know you now get introduced to these statements and in case you get file you would be able to make some sense out of it that was your only aim. So, do not worry if you do not understand these integrate details.