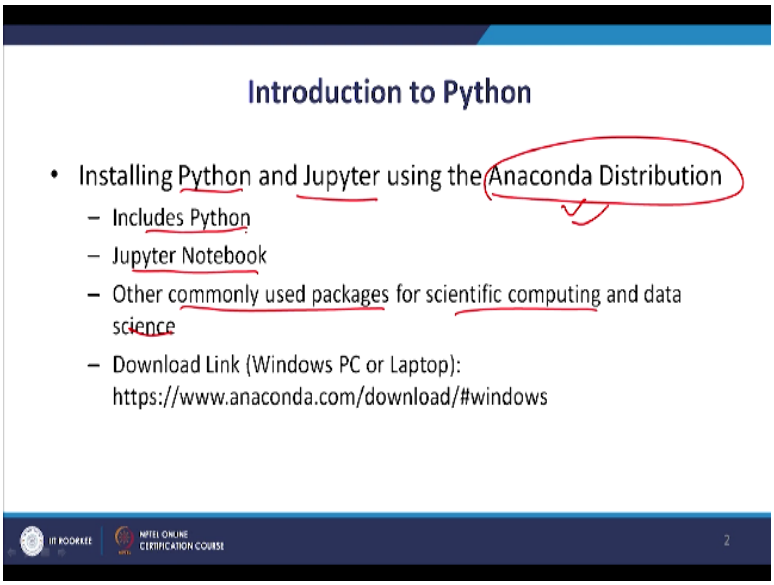
Business Analytics And Text Mining Modeling Using Python
Prof. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology-Roorkee

Lecture-04
Python for Analytics-Part I

Welcome to the course business analytics and text mining modeling using python. So in previous few lectures we have covered the first module that is the introductory part of this course. So we have covered the overview of text mining and now let us move to the next module, so that is for you know python for analytics. So we will start with introduction to python in this particular module, so this is our first lecture.

So first we will start with some of the installation aspect of this particular platform python platform.

(Refer Slide Time: 00:58)

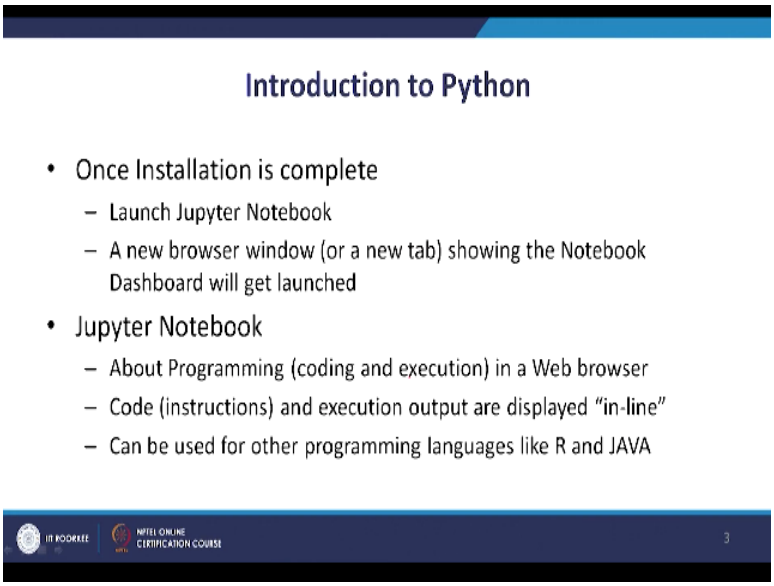


So first thing that you need to learn is that we are going to be using the anaconda distribution for our installation. So this particular distribution contains you know both python and Jupyter. So both python and Jupyter are going to be part of this installation process in one go. So we won't have to separately install these two things and in one go we will be doing this. So it includes python and Jupyter notebook.

And also some of the commonly used packages for scientific computing and data science. So those are also part of this distribution, so much of what we need is already part of this distribution and therefore you know we typically this has become one of the most popular distribution for data science purposes in the python platform. If we require any more packages, so we have certain functionality in building to python which will allow us for installation of additional packages if required.

To download this particular package you will have to go visit this URL and for windows PC and laptop we have given this link, for other you know operating system also you can easily find the detail in the same link itself. So this is for the windows where you will get the appropriate executable or installation setup there and you can easily download and install. So once you are done with your installation.

(Refer Slide Time: 02:39)



So once installation is complete, so what we will do is we will launch you know Jupyter notebook so Jupyter notebook and you know once you install this whole you know package. So in the start button you would find in the newly installed item this Jupyter notebook button also there. So you can just click that and you will be able to install Jupyter notebook. So a console window first a console window will get opened which will have all the locks about how you know Jupyter notebook setup is started.

And then a new browser window you know the default browser whatever you might have in your

setup in your system that will get launched and a new tab in that will be showing the notebook dashboard. So that will get automatically launched, so once you click on the Jupyter notebook in the start button or the search box there in this taskbar. So you will be able to launch Jupyter notebook, a console window will open and in your default browser a new tab would be open where you will see the notebook dashboard.

So we will go through this process but before that let us discuss certain aspects about Jupyter notebook and Python platform in general. So let us start with Jupyter notebook so this is about you know programming in a web browser. So this when we say programming it includes both coding and execution. So typically we are familiar with doing you know our coding and execution in IDE based you know IDE based software.

All in our you know bass cell like UNIX or so other cells or ID based software, so typically we have been doing our programming for you know last two decades or so and those kind of interfaces and those kind of environments. However, Jupyter notebook is something which provides us the coding and execution facility in a web platform interface where we have the web browser and there we can you know code and install code and execute our code.

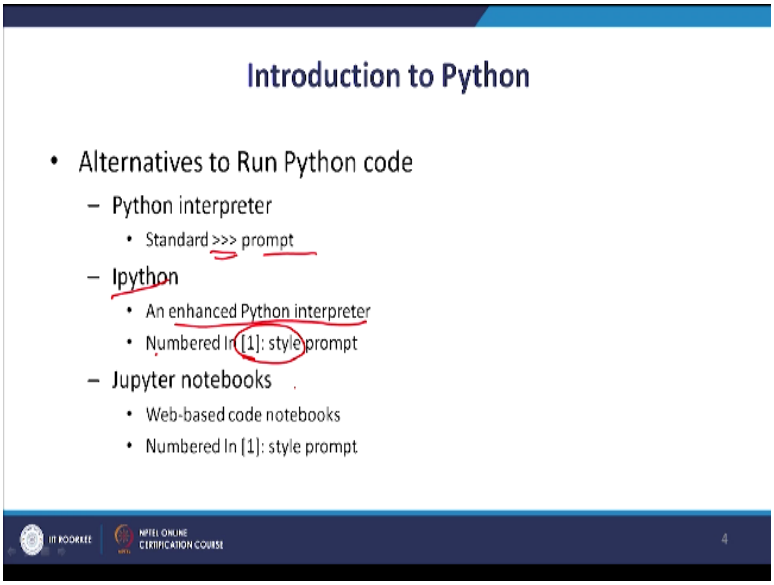
Another aspect about Jupyter notebook is that the code the set of instruction that we write and the execution output. So once we execute that code whatever output is going to be generated, all these things are going to be displayed in line. So you have in line you execute that and you will get the output, then you reach to the next line and then you execute that line of code and then you get the output.

So everything is in line in this particular environment, so as we have discussed that Jupyter notebook can also be used for other programming language like R and Java. So if someone is interested in you know using Jupyter notebook for their larger programming needs, larger application development, software development needs, then you know other programming languages have also been indicated.

So typically in Jupyter notebook you know different kernel engines are required for doing coding

and execution and doing programming in different programming languages. So that would be required and that would have to be installed. Now is Jupyter one question that might come into your mind if you are new to python platform, that what would be the other alternatives to run python code, is Jupyter notebook is the you know only remaining you know alternative to run Python code.

(Refer Slide Time: 06:20)



So that is not the case , you know classically people have been using python interpreter where a standard arrow based prompt is displayed, then any developers many analysts they prefer to use Ipython, so this Ipython is actually an enhanced python interpreter , so this is also being used. So in this interpreter the prompt is like numbered this kind of a style, prompt is there, numbered style prompt is there.

So this is also being used very popularly and if you have gone through once you go through your installation of anaconda distribution that we refer to than as part of that installation you will also get an ID that is a spyder. So in spite of you open that ID you will get a console which is actually Ipython console. So there also so you can use that also, so in the installation that we are asking you to perform in that itself you will get Ipython environment as well and of course with Jupyter notebooks.

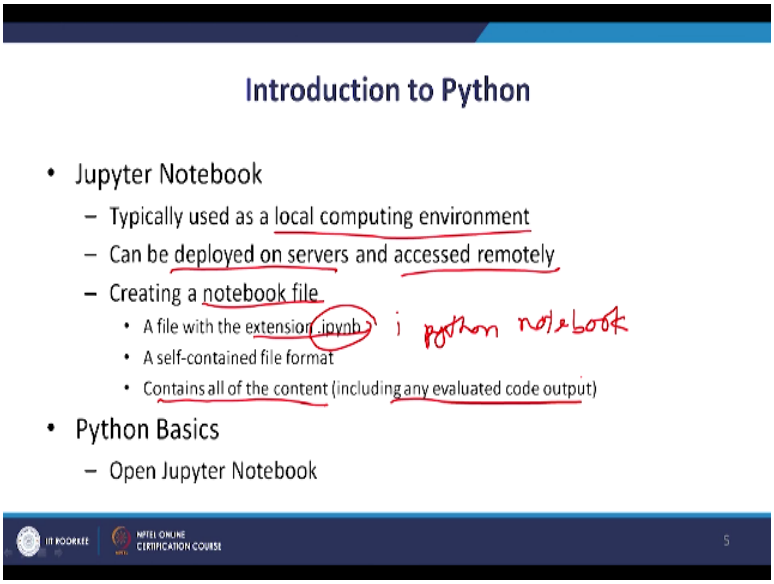
So this is web-based code notebooks and here also the prompt is quite similar to what is there in ipython. So actually Jupyter notebooks is part of the ipython project and therefore there are

similarities in these two platforms. So whatever you can actually execute an ipython you know environment you can also do that you know in Jupyter notebook and you know vice versa. So that is applicable in most of the scenarios.

So these 3 options are available, the classical vital interpreter, the standardized Python interpreter and ipython is there, then Jupyter notebook is there, two of them are very easily available to you and just click spyder if you click in the spyder ID that is provided with the anaconda distribution that we are talking about and you have to just click on the Jupyter you know notebook that is also part of the installation that we talked about.

So we will get two alternatives, however for this course we would be using Jupyter notebook. So let us understand few more features, few more aspects about Jupyter notebook. So typically Jupyter notebook is used as a local computing environment.

(Refer Slide Time: 08:44)



So why we are using these words local computing environment because as we discussed that Jupyter notebook is a you know web platforms we do our coding and execution in a web browser. So because of that automatically the client-server aspect those client-server you know terminologies they can also be used because as you understand that in a web browser we typically whatever is displayed in a web browser.

Whenever we type a URL and whatever is displayed that is actually downloaded from the server.

So whatever we enter there so that will that URL will you know identify the appropriate that will have the appropriate identification to contact the server and download the relevant page. So the same thing is here, so the local so the client-server kind of terms and therefore any other terminology associated with you know client-server environment that is also been used in this context.

So typically when we say typically used as a local computing environment there can be another alternative where we can remotely access the server part of Jupyter notebook, so that can be deployed on servers and accessed remotely. So you can run your server on the same machine, so when you are when you are done with your installation so a local host server is going to be running on your machine.

And your web browser would actually be taking services from that local server and running in your system or your Jupyter notebook can be installed in a server and you can start accessing remotely. So these kind of you know alternatives are available as far as Jupyter notebook is concerned and some of the terminology that we deeply use in client-server computing and that is you know part of this setup.

Now as far as creating our first file you know that is referred typically referred as a notebook file you know when we create this file this will be a file with extension ipynb, so that is Ipython notebook. So this is the simple name of this you know and this you know extension, so that file is going to be created in your computer system and as we have talked about this file is a self-contained file format.
So what we mean by that, so it can contain all kind of all kind of content that could be there that could be required to execute to you know write the code or execute the code. So all that is going to be contained in this kind of file format. So the same thing is mentioned that contains all of the content including any evaluated code output. So this is the kind of file that we would be creating. So the next thing is that we will be moving into the Python basics.

So for that you know we will have to open Jupyter notebook, so let us start so first we will find out from the start button where we would see that our installation is there in the start button.

(Video Starts: 12:06)

If we click here you would see that a Jupyter notebook in recently added items the Jupyter notebook is very clearly displayed here. So this is something that we can click and immediately launch the Jupyter notebook, we go down and see where our installation folder is created this is our anaconda 3 and you can see what other software's are part of this installation anaconda navigator.

So lot this will give you details about a lot more software related things that are part of you know anaconda. So I have just clicked that so once it runs through and launches itself you will see what our different packages software's alternatives that are available as part of this distribution.

So let us wait it will launch itself, so you can see here, so in this lecture I am going through all this from the next you know lecture onwards we will have everything launched before, so let us go through this it will take you know for the first time we will take slightly more time to load, you can see here. So you get this message, so just have to say ok and do not show again. So you can see what other things are part of this Anaconda navigator it can be Jupyter lab, Jupyter notebook, you know this QT console, spyder.

So all these things are part of this, so in this installation you can access ipython console if you just want this or you can use the IDE, in the IDE itself also you will have one console that is actually ipython console. So that is there, other things are also there you can see Rstudio is also there, but this has to be installed. So support for other programming language is also there and you can see these 4 items they are already installed with this.

So let us close this, now let us take a look at the other installation that we have, so this is anaconda prompt then we have Jupyter notebook and then we have a spyder. So anything that you want you can launch from here. So now we will start with Jupyter notebook because that is something, that is the platform that we will be using in this course. So let us click this, so you would see all the locks in this window are being generated.

You can see Jupyter notebook is running at HTTP local host colon 8888 this is port, so you can see when I was talking about the client-server environment, so you can see this is the URL so this is being you know this is being loaded in your browser. So once this is launched so you can see your default browser is launched and this is the dashboard objector that is there. So once you click this prompt will open.

And this you will get the log, how this whole set up is taking place, so from here itself you can see this is a server client setup and you know this is local host. So local computing environment is something that we are using. The same thing could have easily been into a you know actual server and accessing the Jupyter through remote you know access, so that kind of setup could also be used.

So in this you can see different files folders are there, which are by default there is a you know in your computers users directory, you know all this is going to be created. So there we are going to you know click in the appropriate folder for us. So I have created this folder I have already created this folder BA and business analytics and text mining modeling using Python. So for coming lectures we will be using this folder.

You can create your own, so here you can see in here in this place you can see a new button is there a new Python 3, so if you know click on this you would be creating a Jupyter notebook file if you want to create a text file or folder or terminal, so to create a folder you can click on this and you can create your own folder just like this one. So this I have already created so let us click on this.

So here you can see I already have two files the session on Python basics, so this ipynb file is already there and then I have another Python script testscript . py. So this file is also there, so let us start, let us open these file session Python basics and much of what we have to do we will be discussing in this code itself. So once I click this it will get launched. So this is our notebook file and for this purpose of this discussion that we require in this lecture.

That I have already coded into you know this particular file, you look at this name of this file if

you want to change it you can click it here and you can see this window box will open and you can rename your file, then you would see that a lot more options file, edit, view, insert, cell, kernel, widgets cell. All these are available, so we will be learning all these you know controls as we go along in this lecture and coming lectures.

Similarly there are other options as you can see, the most important right now is to understand this box, this particular box is where we this is the first line of code for us where we can use has to do you know add our comments and then we can write our line sub codes here and if you want to execute this we will have to just click on the we have to select this box which is already selected as you can see in green.

And then we will have to click on this run or alternatively what can be done is that we can press shift you know enter, so it will run and the control will shift to the next, you know next line of code. So in this fashion we can we would be you know executing this code, so if you want more help in terms of understanding keyboard shortcuts, so here in the help section you can see keyboard shortcuts so here a number of key board shortcuts are displayed.

For the edit mode or in the command mode, so there are two types of modes that are available command mode, so you will have to press escape to enable this mode and default mode is edit mode, so you have to just press enter to enable. So when we click a particular you know line, so we are we go into the edit mode it is equivalent to clicking up it is equivalent to pressing enter.

And when we press escape then we will go into command mode, so here you can see shift enter that I talked about this is to run a cell and select below. So that cell that line that cell is going to be executed and then control would be a move to the below cell. So this is the window from where you can learn about the different keyboard shortcuts that are available. The first thing that we typically do in any you know script any coding script that we load required library modules.

So in this case as I have discussed that you know will be using numpy that is for numerical python so we use certain abbreviation there is this norm for using certain abbreviation in python so we are loading this import numpy as NP and typically it is considered that whatever is

required only that should be loaded and not everything from python should be loaded into the environment. So as we require certain functionality we would be loading the appropriate library module.

So this is for loading this library once this is done we would like to set up certain you know parameters to configure our output environment, so one of those things you can see number of digits here precision is four default is eight so we would just like to display values up to four decimal point we don't want to look at you know values up to eight decimal points this is more readable more understandable.

So for that purpose you can have your own setting then similarly to a wide scientific notation printing this is again to increase the readability of the code and output will remember everything is going to be displayed in the same file our code and output everything. So this is the function that we would be using NP this is the numpy library . set underscore print option, so we are setting print option using this function.

Where first argument precision the default value is eight, we are specifying as four and then sub press is true, this is to suppress the scientific notation printing and if we just you know run this or press shift enter then this is going to be executed. So we will have to trust this, so this option is there because this is being run in a web environment. So trust issues could be there security aspects are important for to keep our systems safe.

So we will have to click on this, once it is tested then we will be able to run, so once we have done this it is done so let us run this part of code and once this is set we can test out our first line of python. So this is first line of python code is print hello world let us run this you can see hello world was already there so it got printed again and you would see that these numbers will keep on changing.

So as I execute these lines again and again, so this number is going to be dynamically changing to indicate the sequence of execution. So if I run it again so the sequence will change and then if I run this part again then you know this number will also change. So it will always indicate the

sequence of execution. Now if we look at the variable creation the way variable creation is done in Python.

So one variable and one more variable I have used I am using these variable names purposefully, so we will see that so let us run this, so once this is done we can talk about the you know certain tab-completion features. So these could be really useful in the sense when we are dealing with long variable names function names and other objects arguments, file paths, then you know if we can use the tab feature.

So we can type one or few characters, let us say on and then we type tab so you can see it you know drop-down box is visible, so here we can select the variable which we want and it will be selected and then we can execute and you can see twenty four. So in this fashion tab completion feature is going to be really useful. Now other way to use tab completion feature is to access the methods and attributes.

So let us first create this variable, so let us run this, now you can see what I am trying to do is I will use the . tab, so when we use . then after that as per the syntax in python we can use different methods and attributes. So for that first type . and then tab and you can see a drop-down box is created, so whatever function that I want to use again you known use that here. So for example clear.

Now there are a few other things that we can do in python environment is that to see info about an object. So for example if Vone is somewhere and variable that we have already created, so we just used the this question operator question mark operator here after Vone, it can be used before Vone also, but this is one way to use it and we just press shirt and enter, then what will happen this box will open up a pop-up and you can see that some description about this particular variable this particular object is given.

That this is a list and this is the string length is three and other details, so all those are going to be visible here. Now we want to write a function a user-defined function here, so def is the keyword that we can use def F and three arguments XYZ and this is the value that you would like to

written, so this is just for demonstration that we want to display that I have written this code. So again let us shift enter and now this has been executed.

Now there is another if we want to display the source code of this function we can use the name of the function and then use double question mark notation to display this. So if we again press shift enter so you would see that the code here source it has been displayed in this pop-up window. So in this fashion we can easily look at the source code of it and you have defined function.

Now there might be certain situations where we would like to search the namespace , so for that we might be requiring we might be using certain keyword and then we would be using the wild-card or you know character to find out what other options are available in this. So for this NPS there were you know the importer library NumPy and then . an asterisk wildcard character then load an asterisk.

So this expression asterisk load asterisk ah with this we would like to match all the objects that are available in the name space for that to do the match we are using the question mark you know operator. So just press shift enter and you would see that again a pop-up window is open where we can see np . underscore loader and np.load and np . loads, np . loadtxt. So all these different you know objects that are available in the namespace they have been displayed.

Similarly there are you know certain special commands for to execute common tasks and control to you know control the behavior of the system itself. So these are not built into python itself these are special commands that can be used to manage the whole you know whole code setup that we are using. So these commands are quite similar to what we have in what we might be familiar with the best commands in Unix.

So for example a package management tool related command tip, so first we all have to type this exclamation marks and paper and list, so this will give us all the you know packages, so if we press shift and enter, so you can see this asterisk, this is indicating that this particular line of code, this particular cell is executing or waiting to get execute, so you can see this now the

output so it took some time.

Because this list is quite long, so as we have discussed that runtime can be slightly on the higher side in python. So you can see this, so all the packages have been listed. Similarly there are certain other magic commands some of them are referred as line magic which could be really useful. So first we need to what are these magic command, so far this we have this command LS magic.

So this has to be first we are supposed to type this %operator and after that %character then LS magic. So once we run this you can see all the available line magics, these are the available magic commands and available cell magic. So these are the available cell magic commands, so that are available. You can also see that if one message is displayed here that auto magic is on.

So %prefix is not needed for line magics, so however for a better practice we might you know continue to use this %sign, because sometimes we might create some variable or some object in our code and therefore to avoid that situation %using %character is better. So we run this %PWD then we will get our current working directory in details.

So we just press shift enter, so you can see this has changed, now you can see users ET cell and the name of the folder. So this is our directory, now we want to list the files and folders in this directory we can use a %LS. So you will get all the details here you can see a first file that we have seen ipeynv session python basics and tessa script rpy. So these files are present here. Now there is another important you know this magic command is that if you want to run a Python script from Jupyter notebook.

So we can use %run command and we can enter the name of the argument here so and then we can just press shift enter and it will run and once it has done it has certain variable that a part of this we can learn and find out their values. For example C is seven point five, result is you know this value. So all these are part of this script that we will see just a moment. This is how if we want to run a code which is outside of this file in some other python scripts in this fashion we can run that.

So what was the code in this file, so that also we can do using %load you know command, so this %load is and we give the argument testscript.py, so we just run this, so let us run this and you can see immediately the code of that fourth code of that file testscript.py, is displayed and you can see C is mentioned is seven point five here.

The same was you know displayed in this output in the nineteen cell, you can see this line magics you know they take arguments in the same line itself, however cell magics they execute the whole cell, the line magic they execute the line, cell magic execute the whole cell. So that is one different.

(Video Ends: 32:27)

So next that we want to discuss is integration with data visualization, so that is something that will take up in the next lecture. So at this point we would like to stop, thank you.

Keywords: IDE, Ipython, QT console, Anaconda, Namespace, integration, visualization, UNIX