Business Analytics And Text Mining Modeling Using Python

Prof. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology-Roorkee

Lecture-07
Python for Analytics-Part IV

Welcome to the course business analytics and text mining modeling using python. So, in previous few lectures we have a started our discussion on python and we are going through hands-on exercises using Jupyter notebook platform. So, what we will do we will quickly go through you know whatever we have discussed so far. So, we will quickly run through the lines of code that we have discussed in previous lectures.

And then we will pick up from where we left in the previous lecture. So, let us start. So, first thing we are required to load the library modules.

(Video Starts: 00:59)

So, you can see importing NumPy hnp. So, these aspects we have discussed also we need to configure. So, I will just run through some of the starting lines of code. So, that some of the variables that we initialize typically we would be using them later on also. So, that we will be able to refresh our memories ok but for the values of those variables and how they were slashed on that. However variable creation right you know tab-completion features.

So some of these things we are not trying out. So, you will see some error but that is not important , main focuses some of the functionality that will just to refresh your memory writing function and some of these things question operator, magic commands. So, some of these things we have already gone through listing of files and folder as you can see running python script looking at you know values that are part of that script you know loading a script into cells, all these things we have on matplotlib integration.

So, this we have gone through plotting using the same and then you know essential python programming concept that we have discussed, you know how the code is to be written and all that. So, that you have gone through, then how to separate multiple statements on a single line

though not recommended but is still. So, all those things we have talked about function and object you know calls method calls positional and keyword arguments, so all this we have gone through.

Variable assignment as you can see, passing objects, dynamic references, then the aspect related to typed language and how python is a strong type language on all those things we have gone through before. So, just quickly I will run through this part printing and some other functions important function for example is instance. So, these could be really useful .sometimes importing at a script and you know calling those functions and so those aspects also we have covered.

Then you know function calls as well, then we reach to the I think in the previous lecture we were into this binary operators. So, this is how we can perform .some of these operations addition, subtraction you know divide division, lower division just like you know this is more like C style interior division, that we discussed in the previous lecture raise to the power this is also you can see bitwise and you know bitwise all it was exclusive R companions and checks.

So, if you want to compare to back you know variables V1 and V2, V1 2 you know equals sign, V1 equal to V2 and not equal even not equal not operator we can use and we can find out. So, you can see V1=V2. So, that came out to not equal came out as false and less than or equal 2. So, for example these 2 value we can compare like this 18 less than or equal to 48.5. So, immediately we will get the results.

So, these are .some of the binary operator that part of the python. So, for example 18 greater than 48.5 eight you can see false. Similarly if you want to check whether 2 references refer to the same python object or not. So, this is something also we discussed in the previous lecture. So, we have these keywords is and is not. So, a is b and so in this fashion we can find out V1, V2. So, we know that V1 V2 are you know referring to the same object.

So, true V is not c, this also, will come out to be true, then we also, talked about the none object. So, here you can see non object this is you know in a sense python null value. So, there is only one instance of this object. So sometimes this is also, useful and many situation especially as we

will see when we are dealing with conditional expressions and a functional arguments you know especially when we are indicating the default values of you know common default values of those functional argument and none object is quite useful in that sense.

Now I think we covered up to this much, now let us just start our discussion on some of the other basics of python. So, next thing is mutable and immutable objects. So, you know we need to understand this what are the objects, what are the python objects that can actually be modified. So, when we say immutable objects essentially we are talking about objects or values that they contain the value other object that they may contain whether they can be modified or not.

They are mutable they can be modified and if they are immutable they cannot be modified. So, how do you know how these aspects are important and how they can be really useful while we are writing our block of code. So, this is important, so in terms of mutable objects some of the examples we have gained for example list they come by arrays and most user-defined functions. So, these are some of the examples of mutable objects.

So, for example here we have behind variable V3, V3 as a list on the first element then 5 then again a list is another element and I know having 2 elements, 9 and 18. So, if we just run this and if we want to you know because this is list. So, it is mutable as we discuss. So, we can change a particular element within this list. So you can see V3 2. So, I am going to change this particular element here and let us run this.

So, you can see the list is now changed from om 5 and the third element that was there, see here in the bracket notation when we are indicating the index as 2 it is actually you know third element there because it is the counting starts from 0 so 0, 1, 2. So, in that sense it is D when we say 2. So, it indicates the third element actually. So, you can see the third element has been changed, similarly we have .something a mutable object.

So some of the examples that will fall under this category are strings and tuples. So, these are immutable objects. So, here what I have done again I have initialized variable you can see a tuple here. So, values 1, 2 and you know third element itself is a tuple 3, 4 and let us run this and here

because tuple an immutable object. So, if you I try to change one of the elements. So, in the next line of code I am changing I am trying to change one of the element V4 1 that is second element.

So, if I want to change it to the text this is string 1, then I will get error because you can see tuple object does not support item assignment, because this being an immutable object. So, it is it does not support this kind of you know assignment this kind of modification. So, this is how we need to understand which kind of objects are mutable which are immutable and accordingly we are supposed to write our functions and other pieces of code too.

So, that they work well and these objects of course you know modifying any object of course these objects you know have their own side effect. So, when we are using them within a function and we are trying to manipulate them. You know it is better programming practice if we are not trying to change you know modify an existing object rather it would be preferable to read them as immutable , that is better programming practices.

Now let us move into the some of the built-in types. So, for handling numerical data strings, unit froth values and they send time these kind of objects. So, let us start, so, here we will first talk about numeric types. So, there are 2 main numeric types that we have in python environment, one is int i n t, that is for integer numbers and then float that is for you know decimal numbers. So, these are the 2 primary numeric types that we have in Python.

So, in the first example I have again you know initialize a variable I with this value 1, 2, 3 it is rightly long as well just to indicate that larger values arbitrary larger values can also be stored in this particular type. So, here I initialize as this large number 1 2 3 4 5 6 7 8 9. So, if I just run this is the output that I get, similarly you know a float I can initialize in this also, and I have taken a number of decimal places so, 1 point 2 3 4 5 6 8 9. So, if I just run this variable is going to be initialize with the well.

The floating you know these floating point numbers can also be you know these decimal numbers can also be expressed using scientific notations. So, I can have another variable effort and I can express the values the value of one the scientific notation for example 1.23 and e to the

power -4. So, if I run this I will get. So, this value is equivalent to this one point 0.0000123. So, in these formats we can display these value.

Now let us move to the next built-in type that is strings. So, in this we can write string literals whatever the text that we want to initialize as a string tribe, we can write that, we can type that within single quotes or you know within double quotes. So, these are the 2 main primary ways available to us. So, for example here I have a slides STR1 the first you know variable as you know within single quotes.

So, if I run this I will have this STR one any sliced and output you can say within double quotes you know within single quotes single quotes way and then another string variable str2 here I am using the double quotes. So similarly this also I can initialize in this fashion. Now sometimes you might require multi-line strings so with line breaks of course. So, in those situations we can use what is called triple quotes.

So, here you can see we can either go for you know triple single quotes or triple double quotes. So, in this fashion you can see here STR 3 and then triple double quotes and then I have text in first line and second line, third line in this fashion. So, in this fashion I can initialize this, if I run this so you can see, in the output as you can see that first line because this was a newline character is clearly there because no text is there.

And then I have the first line that is long string spanning and then the you know again the newline character then multiple lines, then newline character triple quotes way and then again newline character. So, in this if you look at this output if we I want to you know count the number of lines that are part of this particular string this multi-line string, then actually I can use the number of newline characters that are part of this string.

And in that using that I can actually count. So, just to display the same thing if I can use this count function count method. So, STR 3 and dot count and I am just passing this character newline correct you know character here if I just learn this you can see 4 is the output. So, there are 4 new line characters there. So, depending on that using this particular count I can further

calculate the number of multiple lines that are there depending on the way I define it.

So, in this fashion we can count the number of lines also. Now in terms of mutability python strings are immutable. So, how do we because in many times we will face the situation where we like to modify a particular string into something else. So how we are supposed to do this. So main thing is we can create a new string by replacing the set of characters that we want to actually modify.

So, we can look at the set of characters in the original string that we can modify and just replace those characters with the desired you know set of characters and replace it and create a new string object. So, here you can see in the example I have STR4 so, it says it is a string. So, if I run this I will get this is so, this is my was in the string. Now if I want to change this into a it is a longer string.

So, if I want to pick the particular index you can see I want to insert between a and string the longer what if I want to insert something there and if I try in this fashion STR 4/8 picking up that index here to point out that element and just want to insert this word it would not work, because again just like the previously we had done similar kind of thing that string object does not support item assignment.

So, because it being an immutable object. So, this modification is not allowed in python environment. So, how do we do it , what we do is for this object STR4 we can call the method dot replace and here you can see we will replace string with longer string. So, if I just run this I will get I will create a new string object STR5 and I will get my modified string.

So, here in the output you can see it is a longer string. So in this fashion I can you know modify and as I said earlier also this is better practice this is considered to be better programming practice to consider most of the objects as immutable and the code you know write your code in that fashion itself. Similarly if you want to convert our python objects to a string. So, other object types like integer or float you know if we want to convert them into a string.

So, that is allowed. So, for example if I want to create a string6 variable here I can use this STR function I can fire pass on I. If you remember I is the integer variable that we had created earlier and it had this value 1, 2, 3, 4, 5, 6, 7, 8, 9. So, if I just run this it will be converted into a string type. So, we are using this string STR function here, you can see the output it shows that it is within single quotes we have this value.

So now this integer has got converted into a string type. Similarly if I want to print this string variable str6 now so I will get you know this print in the appropriate format as you can see here are 1, 2, 3, 4, 5, 6, 7, 8, 9 without you know single quotes, so in this fashion I can print. Now let us move on, now another aspect here is that we can also consider string as a sequence of unicode characters, in that sense it can be treated just like any other sequences like lists and tuples.

So, certain modifications certain you know things can be done. So, let us take an example here if we had the STR5 we have already created this string variable str5, if we just use the list function and pass on this variable will see that we will be able to create a list of this and all the characters you can see if I run this you can see all the characters that are part of this that were part of this string, they are now a part of you know they have not become elements of this list.

So, it is a longer string you can see each of these characters IT then space and IS and then space a. So, all these characters of the string become elements of this list. So, in this fashion we can actually you know treat a string as any other sequence. There is another manipulation that can be performed it is called slicing. So, we can always slice a string.

So, this is the syntax you can see STR5 is the string variable that we have and within double brackets colon and 7. So, slicing would happen up to you know that character and so up to 7th character. So, you can see here if I done this. So, our original string that is STR5 it you know it is a longer string that would be sliced and we will just get it is a. So, in this fashion slicing is something that is very important in string processing.

And especially in the text mining context we would be using this a lot. So, we will have more on this in the coming lectures as well. So, this is another manipulation that we can do. Now another

aspect related to string processing is the escape character for this we use the backless character you know this one and if we are you know if we ever require to you know write a special characters like newline character/n if you want to make that as part of the string or pinned that stringm, then we can use these escape characters.

So, for example if we have this string STR7 you know within single '12'/34. So, if we try to run this so you can see in the output within single quotes we have you know 12 we have got all right, then backslash and then we have got we are getting some other you know a set of character x1c. So, it is not coming as expected we wanted to print it as s. So, for this you know we can use backslash character.

So, you can see in this string8, now I am using backslash for printing the other backslash that I want to originally print here. So, you can see now str8 within single quotes I have 1,2/ then again backslash 1/ is working as a escape character and then another second black slash and then 34. So, if I run this is what I get you know within you know single quotes 1, 2, and then // and then 3, 4.

If I want to print this I will get the appropriate print that I wanted in the first place 12 you know backslash 34. So, in this fashion you know whenever we are dealing with these special characters which might have some other purposes as well, then we can use the escape character that is backslash. Now here you see if I try to print you know this a without using the escape character here. So, I just run this you can see I am getting something else the output is truncated and I am just getting 12.

Similarly if I directly pass on this you know if I use the backslash character directly pass it on to the print, again I will get the appropriate output , few more examples here I am using you know backslash n. So, if I print you know without using the escape character then I will get this output you can see 12 on the first line and you know newline I get the 34. So, this is not why what I wanted in the first place.

So I have to use this escape character again. So, here if I run in the next line with the extra you

know backslash for this cape character then I get the appropriate output here you can see 12 and a newline character and 34. Similarly another example you know backspace correct here if I want to print it without the escape character then again you can see if I run it here, then you can see I wanted to print this within single quotes 12 and then backslash character then 34.

Now what has happened is you know because backslash character has the purpose of you know erasing one character, then you can see the output is coming out to be 1, 3, 4 and 2 has been erased, but I wanted originally for that I require the you know escape character. So, in the next line you can see and within single quotes I am passing 12 then backslash for the escape character and then backslash B and then 34.

If I run this I will get the output as I wanted, there might be a situation where we might be dealing with a lot of you know back slashes. So, in that case it might be cumbersome sometimes to type these many back slashes. So, we have another mechanism. So, what we do is we prefer the leading code of the string with R where this R stands for raw that means character should be printed interpreted as is printed accordingly.

So, I have any slides in other string variable here STR9 and you can see it has a lot of back slashes within the single quotes, this backslash has backslash slots, black backs R or backslash you know back slashes. So, if I run this you can see in the output some unwanted values are also printed you can see after this you know this double backslash and then has pend double backslash, lots double black slash of.

And then you can see x08. So, this is not something that we wanted in the first place. So, what I can do is in the string10 in the another instance you can see I am prefacing the whole listening within the single quotes with R. So, it will indicate the interpreter that it is to be interpreted as is. So if I run this you can see here now the output is coming as desired. So, everywhere I have the backs slashes.

So, if I want to you know use different function to get the appropriate output you can see print and STR9. If I do this then if I do the if I used the original we have a writing this without the you

know escape character STR9, then I will get this you can see some of the characters are you know missing. If I use the STR10 which has the you know R as the preface before this string then I will get the output as desired.

You can see this has lots of back slashes and we have just one backslash printed as desired. Now let us move on next thing is about adding 2 strings. So concatenation so, previously we have all defined str1 and str2. So, here we can add these 3 strings str1 and we can just use this + operator STR1+ and another string within single quotes space or space and then +operator and then STR2. So, in this fashion we can actually add strings or concatenate strings.

So, you can see within a single quotes, single quotes way or double quotes way. So, these 3 strings have been concatenated in this output. Now let us move on to the next thing that is a string formatting. So, string formatting has a larger scope, but here will display just one of the example later on in the coming lectures we will focus more on this. So, here we are taking this example where we might be required to substitute formatted arguments as per some template or you know some defined format.

So, based on the fine format we would like to format string, typically it might be you know arguments that we might want to pass on to function or similar kind of situation. So, here first we have defined a template. So, temp 1. So, in this you know within single quotes we are typing us IND exchange rate and the first term that is within curly braces is that first element is a floating wine number with up to 4 decimal places.

Then the second argument is you know string and then the if look at the third curly brace here that is third argument is to be exact integer. So, exact number. So, this is the template and this is what we want to print in this fashion. If I pass on this value tempo 1 dot format. So, dot format is the method that is to be used. So, here the first argument is 71.45678. So if I you know this is to be used only up to 4 decimal places.

Of course rounding and all that is to be applied and then the second argument is a string where we are saying Indian rupees and then third argument is integer 1. So, you can see our statement

would be like US IND exchange rate and then our first argument the value and then Indian rupees are worth and then again US dollar and the value. So, if I done this temp 1 dot form and I am passing values I run this.

So, in this fashion I get the formatted output US IND exchange rate is 71.4568 you can see this rounded the pass down value and you can compare this now Indian rupees are worth US 1 dollar. So, this is how we can do string formatting. So, the next billion type is boolean's. So, in this we have as we know 2 you know typical values true and false. So, typically they are combined with and or keywords and we use them comparisons and other conditional expressions.

Typically an evaluation leads to either you know true or false result. So, just to display these boolean aspect as well. So, few examples true and false, if I run this I will get false. If false or true if I run this I will get true. So, typically whenever we are using these boolean values the comparisons or other condition expression, this is how it will happen.

(Video Ends: 28:15)

So, at this point we would like to stop and in the next lecture we will pick up our discussion from this point and we will start discussion on type casting, thank you.

Keyword: Typecasting, strings, Text mining, Tuples, attributes, user- defined functions, magic commands