Business Analytics And Text Mining Modeling Using Python

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

Lecture-10
Built-in Capabilities of Python-II

Welcome to the course business ethics and text mining modeling using python. So in previous lecture we started our discussion on some of the built-in capabilities that are there in python environment. So let us quickly go through some of the things you know that we discussed in the previous lecture. So we will run some of these commands, so that you know one thing is that will be able to recap of what we covered in the previous lecture and also as we do you know some of the variables we would be using again in this lecture as well.

(Video Starts: 00:55)

So first thing we talked about data structures that is tuple list dicts and sets, so let us start with this, so we went through tuple, tuples here a few example I will just run through this and in the meantime you can you know look at the commands and other examples that we have gone through. So accessing how to modify you know and in-place modification concatenation. So I think in the previous lecture we stopped somewhere here. So we were talking about the concatenation.

So we can use the plus operator as we discussed in the previous lecture. So for example if we have these tuples tup5 and tup4, so we can just you know write this line up tup5 + tup4 and you can see in the output both these tuples have been concatenated. Similarly we were talking about there is another way to concatenate this is involving repetition of elements. So we have a tuple which is having let us element 1 0 -1 and you would like to add few more you know elements with the same values that means we would like to repeat them over and over again multiple times.

So that we can do through this particular you know mechanism where we can type tup3 and then asterisk operator and then 3 so in a sense we are multiplying all the elements we are reproducing those elements. So let us run this you can see tup-3 had 3 elements 1 0 and -1 and then they have been repeated thrice, so we have 3 groups of these 3 elements. So in this fashion also we can

perform concatenation when the repetition is involved.

Then next thing that we are discussing is about unpacking tuples. So you can see in the left hand side we have a b c as tuple and you know if you want to assign values to these you know these elements a b c on the right hand side we can have any initialized tuple for example tup1 we have initialised. So if I just run this and if I want to find out the value for c you can see here we have 10. So in this version we can unpack tuple and some of the you know variables some of the elements can be assigned values you know using this unpacking.

Similarly if we have nested tuple also and there also we can kind of want to do similar kind of thing. So another example here you can see we have one tuple xyz and another tuple pqr and then we have tuple of these 2 tuples and on the right hand side we had previously created you know nested tuple. So if I run this you know individual elements of these tuples they would be assigned those values.

So if I run again for q you can see the value 6 has been you know assigned to this particular element. Similarly we talked about swapping, swapping variable names is quite easy in a one-line code sentence you can do your swapping which typically usually we do in other programming language by writing 3 sentences x y then y z and in that fashion, here using tuple this for a data structure in one go we can achieve this. So we can see here on the right hand side in this you know statement we have a b c this is the tuple.

Now if we want to you know swap a with c then what we have to do is in the left hand side you can see we are typing c b a, so in a sense a you know a and c have been swept and if we run this then the values would be variable names would be scrapped and you can see if I run this then c is displaying the value that was recorded in a. So in a sense we have achieved swap. Now let us move to the next aspect that is iterating over sequences of tuples or lists.

So this can also be easily done, so let us say we have sequence like this, so in this example we have the sequence this is a list with 3 tuple elements 1 2 3 then 4 5 6, then 7 8 9 and here we are running a loop, so you can see how we can iterate over this sequence. So in the loop you can see

your x y z this tuple we will be iterating over the sequence which is again consisting of this is a list of you know 3 tuples having 3 elements each.

So in this fashion any block of code where we require to iterate over you know 2 3 variables or 2 3 variables are involved and some iteration has to be done we can use loop and then you know tuple. So in just you know for loop and within that just one line and we are easily able to iterate this which is for other programming language this is this will take few more lines to achieve. so if I run this you can see the print statement that we have. So till now we have not discussed print you know function in more detail with that will do in coming lectures.

But right now you can see that we are trying to print you know this is statement you know multiple times. So if I run this you can see for the first tuple we have got these values assigned x 1 y2 z3. Similarly the second print it has x4 y5 z6, third print so in this fashion we can you know iterate over sequences up tups or lists. There might be certain scenarios which will require us to return multiple values from a function.

So that can be easily done again using tuples, so here in this example I have defined a function f it takes 3 variables as arguments xyz and then we are doing you know certain computation here x you know x is being assigned the multiplication of x*y and y is being assigned the multiplied values of you know y and z and similarly for z z*x. So here again we are returning these 3 values x y z.

The updated you know values of these 3 variables, so again you can see in a tuple in the form of tuple data structure we are able to you know return this para. So once we define this function you can see in the next line we are you know assigning the return values to a new tuple d e f. So this and the appropriate values, the updated values x y z within the function once they are computed, they are going to be assigned to this tuple and the elements you know and that is d e and f.

So if I run this and you know let us see the values you can see 60 12 and 120. So those computations are being performed and we get the updated values here. Let us move forward, now you know some situation will require us to plucking a few elements from a tuple. So we might not want to

use all of the elements that are part of a tuple rather we would like to create a you know new tuple are just use hue elements from a tuple.

So that is something that can be done for this we need to use this asterisk operator. So let us take this example of this tuple6 where we have these 7 elements so 1 2 3 4 5 6 7 and you know if we are interested in you know just you know plucking the first 2 elements. So in this case first 2 elements in this tuple 6 or 1 and 2. So we can write our code in this fashion you can see a b and then asterisk rest. So rest is not a this is not a special keyword it is just the remaining you know elements we are giving them as you know this name rest.

So nothing is special about this name the main thing is this operator. So asterisk operator and then the name you know for the remaining elements of the tuple. So in this fashion we can formulate this particular you know code expression a b and then a, b, asterisk rest, now what will happen is the first 2 values first 2 elements from this tuple6 they would be assigned to in these 2 elements a and b. So if I run this you can see values of a b 1 and 2 they are me appropriately plugged from the existing tuple that is tup6.

Now if you want to look at what is there in rest, so that also something that we can you know run and find out. So you can see in the rest tup out so the in the rest you know what we have is a list of the remaining a you know elements 3 4 5 6 7, now it is not you know sometimes you might require to just pluck the last you know elements from the tuple so that can also be done, here you can see the code expression has been modified slightly.

Now in the initial last 2 elements I mentioned c and d and the you know the remaining the part of the tuple I am denoting by asterisk rest 1. So in this fashion we can pluck out the last 2 elements of the tuple. So let us run this and let us find out the values of c and d. So you can see 6 and 7. So in this fashion so whether from the start or from the end we can pluck out the elements that we want.

Now again we want to find out what is the part of rest once you can see a list of these you know remaining 5 elements starting from 1 2 3 4 5. Now some situation might also require us you know

to you know plug the elements in the middle, so that can also be done especially if you know from there that asterisk variable has to be you know expressed at which point. So here in this example we can say that we can see that a,b and then asterisk rest to then comma c. If I run this so you can see rest 2.

And if I run this so 3 4 5 6 is here and the value of 6 is 7, so in this fashion also you know we can plug the be required elements from a tuple. Now let us move on now let us talk about you know just one tuple methods that we will discuss right now. So if we want to count the you know number of you know elements that are part of a tuple, so you can see tup4 we had previously created and in this count method what is happening is we are counting how many times you know this particular value m is appearing.

So if you want to count that particular you know value m how many times it is appearing in that tuple, so tup4 dot count so this will give us the output. So here if we run this you can see tup4. count and you know m is appearing twice. So we go back to our previous result where we had created the tup4, so you can immediately verify this, so we can see here, let us go back here. So you can see in this place output 6 that we can see here you can see you know m is coming over you know twice 1 and 2 here.

So that is why that you know in this fashion we can find out the element a particular element how many times it is appearing in a tuple so count method will give us that particular output. Let us move on, so now we come to the next built in structure that is part of python that is list. So what is a list about, this is a variable-length mutable sequence of python objects. So mutability immutability we have discussed before also. So tuple is an immutable python you know object while list is a mutable one.

So therefore of course the you know it will have variable length to allow this to happen. Now how we can create a list so this we have been doing from the start of our discussion on python. So we can use the bracket operator and within that comma separated values. So we can create a list with a comma separated sequence of values and using brackets operator. So let us take this example again we want 1,3, non, 5, true. So different kind of data types we can you know combine in a list

you know combined in a comma separated sequence and in this fashion we can create a list.

This is something that we have been doing all along while our discussion on python, so you can see this list has been created. Now it is also you know possible to you know convert some of the python objects into a list, for this we need to use this function called list so here you can see that example that we are giving is we are converting it tuple into a list. So if we look at list and tuple you know now you are able to see and we have gone through the tuple aspects as well and now we are discussing list also.

Previous lectures also we have whenever seen the usage of these 2 data structures, so you can see both are sequences and semantically the way they are to be you know manipulated worked with in python they are very similar and can be used interchangeably. So a list function can easily convert this tuple6 into a new python object that could be a list. So here we are using list function.

So V2 list tup6 so this will create a new python objects, so we can see that is tuple 6 has been converted into a list. Now accessing list element that is something that we have been doing all along, so again bracket operator, so here V2 and in the bracket we can specify the you know the element you know the index that we want to access, remember the most of these sequences in python they are 0 indexed.

So when we say you know V2 4 so it is actually the fifth element that we want to access. So if I run this I will get the value of 5th element in this you know list V2 and that is actually 5. Now we can modify because this is a mutable lists or mutable python object as we have discussed before. So we can modify any of these you know any of the element of this particular list V2. So let us take this one V2 4 this you know 5th element you know indexed by the value 4.

So we can change that from 5 to a string value that is omkar, so if we I run this you can see the list has changed. Now we have the elements as 1 2 3 4 and omkar and then 6 7. So in this fashion we can modify the individual elements of a list as well because list being the mutable python objects. We have talked about you know a range function also.

So range function if you can see here in this next example so we can you know if you want to so the range function will typically materialize when we are using it with a list or in a for loops here you can see if I run this range 6, so 6 values starting from 0 we get a list made of those 6 elements starting from 0 1 2 3 4 5. Now let us talk about you know adding and removing elements in a list.

So if we undo you know let us first talk about adding so at the end of the list so we can use this function append, so that we have used before also. So V2 is a list dot append is the method and om is the you know new element that we would like to add into this list. So if I run this and you can compare the you know previous you know elements previous values of elements in the v2 and you can see the last at the end of this list a new element has been added om.

If you want to add an element at a specific location in the list that can also be done. So we can use the insert function here, so in the next line of code you can see V2 dot insert and then in the in the parentheses we are passing on 2 arguments, the first argument is telling the you know a specific location the index of the specific location where this new element is to be you know inserted. So you can see namah you would like to insert here.

So if I run this you can see second element has been second elements which was earlier which was earlier to now has been changed to this you know a string value namah, so in this fashion we can modify our add elements in a list. Now let us move to the you know removing aspect so if we want to remove a particular element at a particular index, so we can use the pop you know method here.

So V2 is the list V2 dot pop the method it will have access to all the you know objects that are within that list. So V2 dot pop and we are giving the you know index. So in this case I am taking 2 here. So 2 means that this is index 2 that means third element is supposed to be removed. So if I run this line of code you can see the popped element value is 2 this is actually the third element in the list as you can confirm from the previous output number 37.

So and we can again have a look at the values that are part of this list right now, o V2 you can see one then namah and then 2 is gone and then 3 4 and other values. Now it is also possible you know to remove you know some of these elements by value, now whenever we are trying to remove an

element from a list by value there can be more than one instances of that value occurring in that list. So therefore if you use this remove method here, so the first instance is going to be removed.

So you can see here V2 dot remove and 6 so here if we look at it the third particular you know element the first element having value 6 is going to be removed. So if I run this you can see 1 namah 3 4 om car and after that we had 6 as you can confirm from output 39 and you can see an alts gone, let us move on now sometimes we might be required to find out whether a particular value is part of a list or not.

So how do we perform this kind of check, so we have 2 key words you know in python in and not in and not in, so they can tell us whether a particular you know value is present in the list or not. So let us see 2 in V2, so as you remember that we have popped out 2 before, so if I done this will get false because 2 is not part of this list. Similarly om not in V2 but we can see from the output 40 that om is there a part of the list.

So this again should also come out as false, now just like we talked about the concatenation of tuple list can also be concatenated using the plus operator. So V1 and V2, these 2 lists we have already created. So let us apply this plus operator V1+V2 now you would see if I run this the output 43 you can see all the elements of you know V1 and V2 they have been concatenated in that particular order itself.

Let us move on now sometimes you know we would require adding multiple elements in a list, so that can also be done for this we can use the extend method. So V1 is the list that we have already have so we can use this extend method so we can write V1 dot extend and then within parentheses we can you know pass on the list which will consist all the elements you know that we would like to add. So here you can see that we are adding om namah siva these 3 you know this new list consisting of these 3 elements.

And if I run this then you would see that the existing list would be extended by these 3 additional elements. So if I find out the value you can see that 1 3 9 5 so these 3 elements have been extended. Now let us move on to the next aspect, so this is about sorting the elements of a list. So for this

also we have a sort method that is available to us. So let us take this example we of this list V3. So it has these elements these values 19 75 53 23 and 7 and then 11.

So we can use sort function to produce a you know a list with sorted values so V3 dot sort and we do not need to pass an argument because the you know the list is to be sorted the elements of this list are to be sorted now if we you know do this apply sort method on this you can see in the output 46 you can see the elements have been sorted. So we have first 7 then 11, then 19, then 23, then we have 53 and then we have 75.

So in this fashion we can perform sorting the elements of a list. Now sometimes you know we would like to sort the elements of a list using a short key and this particular key can be function you know which will return values to be used for sorting. So this can also be done, so let us take an example here again, so in this case let us take the example of this list V4 and here we can see that you know we have 4 elements which are part of this list shivay om namah and chanting.

And if we want to so one thing is we are taking the elements here or string values and so probably you know we would like to in more often than not we would like to sort these elements by length of those string. So when we are talking about sorting elements using a you know short key. So here this key can be a function this function can be length in particular and this can be used to find out the length of these element strings.

And based on the length of these element string values we can generate a sorted list here, so what we are doing here is in the next line of code first let me initially the list V4, now what we are doing is V4 and then we call in short function and then we are in the arguments we are passing just one argument that is key this is named you know argument here and key and length is the function that we would like to use as a sort key.

So if I you know run this and access the updated list here you can see now this the list of these string values have been sorted accordingly according to the length of those string values. So om with the just 2 character comes first and followed by namah having 5 character then followed by you know shivay which is also having you know which is having 6 characters and then chanting

which is having I know more than 6 character.

So in this fashion also we can perform you know sorting, sometimes in a string processing I will see in later lectures that this kind of thing would be required. So as you can see it can be easily done. Now let us talk about you know sometimes you would like to maintain a particular sorted list, but we would also be required so why would have to maintain a sorted list we would be you know adding new elements to the sorted list.

But you would like to keep that sorted all long, so next thing is that we would like discusses inserting a new element into a sorted list. So how to keep lists sorted and still be able to you know insert new elements. So we can use this bisect module functions here, so first we need to import this module bisect and this will give us the this function you know this function bisect here and it will allow us you know based on the index of the location we can insert the particular new element.

So here let us take the example of this v3, so you can see V3 is already sorted, so you can see in the output number 46 7 11 19 23 53 and 75. So in this if you would like to insert this you know you know new element which is having value 17. So it should be between 11 and 19, so we use this function you know bisect you know here taught by set so if I done this so it is giving me the you know index location where this new element having value 17 is to be inserted.

So index 2 means you know after you know it should be the third element in that list. So you can see after 7 and 11 it can be inserted. Now you know this module also gives us the actual function which would be more practical to you know to insert the list in the appropriate place. So the next line of code is about the same, insert the element at an appropriate index into the sorted list.

So for this we can use this function bisect or in sort and you can see V3 is the less than 17, if I run this you would see that this new element 17 have been appropriately inserted into this sorted list. So now we what we have in the output 50 is 7 11 then 17 and then 19 23 53 and 75. So the earlier function bisect it gave us the index where this new element is to be inserted and the in sort you know function is you know rather method is giving us the you know the shorted list after performing the insertion.

(Video Ends: 28:20)

So we would like to stop at this point and in the next lecture we will pick up our discussion from here only and we will talk about slicing, slicing is something we have briefly discussed before we will talk about slightly you know inside more detail in the next lecture, thank you.

Keywords: Swapping, Concatenation, Scrapping, Slicing, tupple.