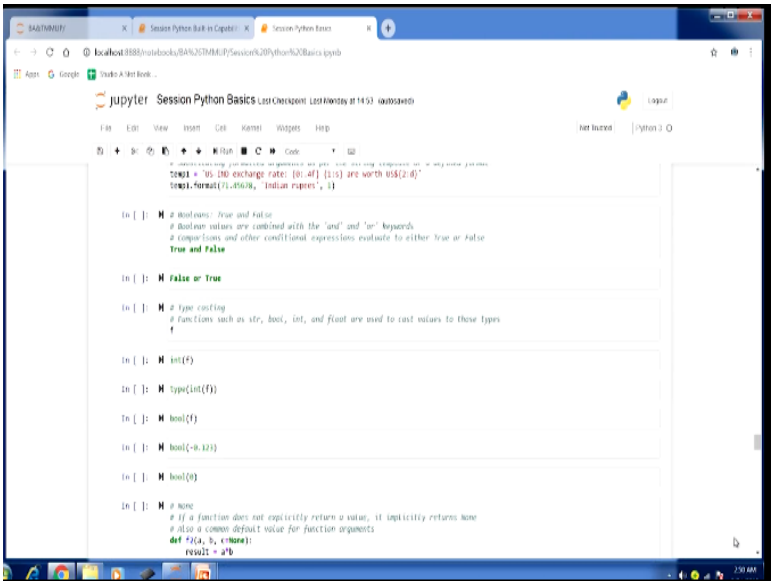
Business Analytics And Text Mining Modeling Using Python
Prof. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology-Roorkee

Lecture-08
Python for Analytics-Part V

Welcome to the course business analytics and a text mining modeling using python. So, in previous lecture we were discussing the some of the basic types the data structures that are available in python. So, let us continue our discussion from where we left in the previous lecture. So, we were talking about the boolean's and how they can be used.

(Refer Slide Time: 00:48)



Now we will move to the next part that is typecasting. So, we have certain functions available in python the built-in function, functions available in python such as STR, a bool, int, float, which can be used to cost values to those particular types. So, let us start with few examples in typecasting.

(Video Starts: 01:14)

So, we have this value F value that we had a variable that we had created earlier. So, I will just run this. So, you can see this is the value, this is a floating-point number the value is 1.2 3 4 5 6 7 8 9. So, this is the one that we created in the previous lecture. So, if I want to convert this value from a floating type to an integer type. So, I will have to just call this function int and within the

arguments I I will have to just pass this variable F.

So, if I run this here. So, you can see the value floating point number 1.23 and so on has been converted into an integer number now that is 1. So, in this fashion we can perform typecasting. So, of course there is going to be certain loss of information. For example in this case the floating point number and when we convert it into an integer number. So, the decimal points they would be lost here.

Similarly you know another way if we are looking at, if we want to find out the type of the number that is we just you know type casted. So, if we just run this you can see, now the you know once be typecast a number into a particular type and the type also changes to that particular type. For example we type casted floating point number to integer type. Now that number will also you know carry on that integer type itself the same thing is reflected when we use the type function to find out the data type of that variable.

Similarly we have another function bool which can be used to type cost any variable into a pool type. So, again we will use the same you know variable f to again convert this into a bool types so let us run this. So, you can see now you know this value. So, the value that we passed on to the bool function was a nonzero value right. So, therefore it has been converted into true. So, all the non zero values are going to be converted into true and only the you know 0 is going to be mapped to false.

All other nonzero values are going to be mapped to you know true. So, to display the same thing I am calling in another you know another cell. So, in this particular cell I am passing on a negative value -0.123 and if I run this you can see again true. So, this is any nonzero value this is also very small between -1 to 0 and if I just use a bool function this will also be converted into a you know true type right.

Similarly if I you know pass on the value 0, then I will get default. So, if I run this you can see here the result is false here. So, in this fashion we have these function which can actually be used to convert to cast one type into another. Now let us move on to the next you know aspect that is

this none. So, we have this none data type that is that typically has only a one unique instance of its own.

So, using this program you know data type none can be useful in many situations in python. So, for example you know one thing that we have mentioned here is that if a function does not explicitly returns a value, then it implicitly returns none. So, that is that is in willed into a you know typically the way functions are defined in python. So, let us take an example here and also another situation could be that in a common default value for most of the functional arguments.

So, apart from the return value you know where this none is going to be used. Sometimes you know the arguments they require default values they are also none can be used right. So, that would be the default value for those arguments. So, one example that we are giving here is we are you know defining this function F2. So, here I have you know three arguments ABC and you can see the default value for C is none.

So, this is one of the common default value and within the function the first line that we have there is result. So, we are multiplying the first argument a with b. So, a*b, if c is none we have another conditional logic running here, then result is going to be result +1. So, and then we are returning the result. So, if I just run this so the function f2 is now defined and you can see. Now in the first instance of calling this function at 2 I am just passing on 2 arguments here.

First value is 1, the second argument and its value is 2. So, if I just run this even you can see for the third argument that is c default value is already given there that is none. So, it will run quite well. So, you can see we got the output as 3, because you know first 2 arguments those were multiplied. So, we got that value. So now moving on, if we look at the type of this data type none. So, we can just run this type function and pass on none as the argument.

And we will find out the type of you know data type of this particular you know value. So, you can see none type. So, that is why when I was caught talking about none I was also referring to as a data type, had its own data type is it is called none type. So, whenever we are using none so, it has its own type also and unique instance. Now let us move on. So, next thing that we would

like to cover is dates and times.

So, you know we have certain types you know basic you know built-in types in python. So, for example date time date and time. So, these are 3 typical types. So, date time is typically used more often especially in the analytics context. So, for if before you can actually use these types we will have to you know import these modules. So, from date time you can see here from date time we will have to import these module date time, date and time.

If I run this I will have access to all the functions that are part of these modules. Now let us first start with date, time. So, date time is a function that we can use to you know to actually create a date, time object here. So, here you can see first thing that I am doing here I am passing on all the relevant arguments here. So, you can see first I am passing on the year and then the month and then the you know day and then hours, minutes and seconds.

So, this whole thing date and time both you know that gives the full information in terms of indicating the date of that particular day and also the current time that is there or whatever time that we would like to convert into a date time object. So, if I run this, this is the value I get we created this you know object here. So, if I want to access one of you know these details here. So, you can see the dt the date time object that I have just created I can you know call these methods here.

So, dt.date to access the day information if I just run this you can see 29. So, this is the date is 29h here, if I want to find out the month so I will have to type dt.month. So, month method will give me the you know details. So, this is a month number 10 and then if I want to access the year then you can see here 2019. Similarly for minute, hour and second. So, in this fashion 30 minutes, 20th hour and 21 second.

So, each of those you know elements of this state time object can be accessed using these particular methods. Now let us move on to the next object that we have. So, next object is data object. So, here you can see here I am using a date method here. So, if I want to find out you know just the date part of you know we just saw the date time object. Now if I want to you know

just look at the date object then I can use this you know within the date time object.

I can call this method and we will get the date object as you can see here only the date part is shown here. So, this is the object that we get and if I am just interested in the time part of it then I can call this dot time method here. So, I will get the time object you can see here. So, this was about date time date and time objects now let us move on to the some more aspects about string. So, formatting date time as a string.

So, typically there are you know various ways we can you know format a date time string a date time object as a string. So, here you can see what I am doing is date time object dot I am calling this function strf time. So, this is for you know STR for a string F for formatting time. So, formatting date time object as a string here and you can see I am just trying to format using these percentage and then you know forward slash and then percentage capital Y.

So, these are to indicate which particular piece of information I would like to you know have here .So, if I just learn this so, you can see the dt value the dt you know variable that we had and only the month and year information is extracted here for you know is formatted and you can see fully year 2019. So, Capital Y actually stands for the full year, if you want to use just 2 digit year number then you will have to use percentage small y.

So, in this fashion we can you know all the time we can format and extract the kind of you know date time information that we require now let us move on to the next thing some scenarios might require us to actually you know pass string into a date time object. So, for this again we are taking this example here. So, for this we are calling this method str time. So, this is for parsing the string into this you know date time object.

So, here you can see we have first argument is the string to the other 19, 10 and 18 and then we are formatting we are parsing this into in this particular of fashion we have given our format formatting expression here %y then %m and then %d. If I run this you can see I am able to create a in a date time object here where the you can see the data information 2019, 10 and 18 is as per the date time object format and then 00 to indicate you know hours and minute that information.

So, in this fashion we can actually any string if it is in that you know you know using the formatting expression we can you know convert it into the appropriate you know date time object as well. So, both ways. So, using these methods strf time and str time we can you know format the string both ways format our pass string from a string to a daytime object or daytime object restraints both way kind of formatting is can be done.

Now if you want to modify a daytime object. So, next thing is about the same replacing fields of a date time object So, here we can use this replaced method. So, first argument that in this example we are passing minute. So, you can see here these are named arguments here. So, you can see minute 0 and seconds 0. So, if i just run this. So, I would be replacing that information here. So here you can see in the rate time object I am able to replace those details.

So, minute and second information is gone, now in place of minute you just see the 0. Now similarly sometimes we might require to difference to date time objects. So, that is also possible. So, this particular you know when we difference to date time objects what we actually get is a date time dot time delta type. So, it is like the difference between 2 time and the deltas for that. So, the delta difference the delta of that time that we will have.

So, for that we have this type also in python a date time dot time delta type. So, we can use this date time function you know let us to define another object here. So, dt2. So, let us learn this now we have dt2-dt, so, if we want to difference will get this delta value. So, let us run this. So, you can see here you can see the difference between these 2 you know date time objects, date time variables and you can see the differences in a number of days and seconds.

So you can see in this fashion that d2 dt2 the second variable that we created it is actually these many days behind the dt variable dt object that we have. So, in this fashion we can have even the difference also difference of date time objects. So, if you want to confirm the type of this product or time delta type. So, you can use the type function again here. So, if I just you know run this. So, you can see a type is very well reflected here date time.time delta.

So, this is about few things that we can do with the date time objects. So, now next thing is sifted date time. So is it possible to is it is also possible to shift a particular date time. So, here if we look at the dt value here. So, the current dt value is this much. So, if I add the delta value you know that is date time.time delta time. So, if I add that value. So, in a sense I am sifting the you know date time, sometimes in some situation this could be really useful especially in the matrix context.

So, if I you know add these to dt+this delta a value that I have. So, I will I will have in result would be that I have shifted. So, in a sense this value has been lagged by that delta value that we had. So, this is about this is so these are a few scenarios few things that we can use to work with the built-in data types that we have in python. Now let us move to the next important aspect that is control flow.

So, we have certain constructs in python also just like in another programming language which can be used for control flow as well. So, we will start with the conditional logic. So, in conditional logic we typically have these keywords if, elif and else. So, as you can understand these are based on the conditional logic. So, if a particular condition a is going to be checked you know.

And if it is found to be true then the following block is going to be executed, if it is not found to be true then if there are any other conditional you know logic blocks then those checks would be performed and the following blocks would be exhibited if final checks were found to be true as for the you know the statement there. So, in this fashion based on the weather a certain you know condition is evaluated to be true certain blocks are executed.

So**,** this is what we typically refer as control for in a sense the flow of execution is being controlled by certain checks, only if certain checks are evaluated as per the return value desired answer, only then that particular flow of code, only then that particular you know a following block of code is going to be you know allowed. So, let us understand the same thing with you examples.

So, if I have this variable X=-1 and I can have this you know if conditional logic here. So, if we X less than 0, then you can see let us understand the semantics here also, if X less than 0 then we type a colon here. So, after that whatever is the following block it would be indented and that is going to be considered as the following block and it is going to be executed. So, if the conditional expression after following the if keyword in this case X less than 0 is found it is evaluated to be true.

Then the following block that is the print statement that we have that is going to be executed. So, you know let us run this. So, you can see because we had initialized the X variable as -1 which is a you know negative value. So, therefore the conditional expression that we had X less than 0 that was evaluated to be true and therefore we have printed this is statement here. Now let us take another example in this case we have initialized ss2 and now we have we are using all the keywords that we have under this conditional logic.

If you can see there is block for if, there is block for elif, then another block for elif, then another block for else. So, the difference is first we start with if. So, if that particular you know conditional expression is true that part is going to be executed and other parts are not going to be executed. Now if so, happens that if was not found to be true then we moved to the next you know conditional logic that is typically you know written using elif.

So, again after elif also in the elif block also, will have a conditional expression again that is going to be evaluated if it is found to be true, then the following block is going to be executed So, in this version we might have more than one elif blocks. So, if none of them are evaluated to be true then we can have a one else block. So, this particular else block is like a catch-all block. So, if all the preceding blocks and the conditional expression that were there in those proceedings locks.

If they were found to be false, then we can have this sketch all block which is the else block where we can you know we can have a default kind of for all depending on the scenario depending on the situation we can have a set of instruction on a set of lines of code which we would like to execute in a catch-all sense. So, here what we are doing is we have initialized xs2

and we are in the if block we are have we have the conditional expression X less than 0.

So, of course this is going to be evaluated as false. So, the following you know block following statement print it's negative is not going to be executed, it is not going to be printed then we will move to the next block which is a elif block. So, here we are looking at X==0. So, of course this is also not true. So, the following block print equal to 0 we will also not be executed, then we will move to the next block that is elif where the we are checking whether X is lying between 0 and 5.

So, 0 less than x less than 5 and since this is true. So, this particular block is going to be executed. So, the fourth log that is else block we do not you know you know we do not need to event each to that block because we found one elif block which where the conditional expression was found to be true and therefore will execute the following block or following you know lines of code. So, let us run this you can see positive but is smaller and 5.

So, we have successfully printed this. So, it is important. So, whenever we have these kind of scenarios that after a certain lines of code we you know we have that you know conditional some based on certain condition you would like to execute something. So, we can all the time use these conditional logics using if, elif and else keywords. Now let us move on to the you know next thing. So, we can have a you know compound condition.

So, and R so these are keywords that are typically used. So, these conditions are typically evaluated from left to right and you know we will sort circuit also. So, we will see example. So, here we have two variables a and b. So, let us define them initialize them and then c and d. So, now we have these 4 variables and we have compound condition where we are comparing first expression conditional expression is a less than b.

And then we have these keyword and and R in this case we are using R and we are comparing in the second conditional expression c greater than and d. So these kind of compound condition also, we can have. So, in this fashion we can write them. So, here in this example I have written if a less than b or c greater than d then print this. So, if any of these 2 conditional expressions are

evaluated to be true then the following line of code is going to be executed let us run this.

So, because if the values that we have any slides for these variables you can see 15 a is 15, V 17. So a is less than V. So, this is found to be true and if we look at the c and d. So, c is 18, d is 14. So, c is greater than these this is also found to be true. So, in this particular case you know first conditional expression that a is less than b and the second conditional expression that c is greater than d, both are true. So, true are true.

So, it is you know anyway it is going to be executed. So, you can see reached here that particular you know massage is printed. Now similarly you know just like compound condition we might have situation where we will have to make chain companions. So, in those situations also you know can be very well coded here. So, for example I can compare like this 1 less than 2 greater than 3 less than 4. So, values are there.

So, there itself I can check whether this is true or not. So, you can see 1 is less than 2 this is you know true, then 2 greater than 3 which is not true, then 3 less than 4 this is true. So, even if one of them is found to be false then the overall you know the result is going to be overall evaluation is going to be false. So, if I run this you can see the messages the output is false. So, if there are situations where we need to make these you know chain comparisons.

So, in this fashion we can type our code. Now let us move to the next aspect next you know construct that we have in python loops, which is also there in most of the programming language. So, what is the main idea behind loops. So, typically it is used to iterate over a collection. So, collection can be any list or tuple. So, few you know we have not gone into a detail of you know these you know sequence or data type list or tuple will go in more detail in the coming lectures.

However you know these kind of you know sequences or collection list or tuple. So, if we want to you know in loop typically you would like to perform certain number of iterations. So, these collection or sequence types can be used to you know control or to you know facilitate that you know those number of iteration. So now we will learn about I know loops what kind of loops are available in python.

So, let us start with far. So, which is typically the more popular you know loop that we typically use. So, here you can see on the first statement for keyword and then x in this particular list 1 2 3. So, till you know all the values are iterated. So, this is the list 1 2 3 till X you know it rates over this particular collection, this particular list will keep on executing the following you know block. In this case just one statement print x. So, we will keep on printing the X.

So, essentially what we are saying is will be printing all the elements of this particular list. So, let us run this, you can see 1, 2 and 3. So, one by one each of these elements you know in that particular list 1, 2, 3, have been printed. So, we have been able to iterate over this particular list sequence. Now there might be certain situations you know where we will have to advanced to next iterations.

We will have to bypass you know that particular you know iteration. So, for that we have a keyword continue. So, in that case if you know certain evaluation certain you know let us say conditional check that we have you know within that you know for loop or any other loop for that matter and if that is you know found to be you know true then we can use the continue keyword to advance over to the next iterations.

So, let us understand that through an example. So, let us say we have this sequence. So, this is a list and then we have another variable I. So, we are initializing at 0, then we are running over for loop. So, for X in sequence. So, all the values are going to be iterated here and within the you know following block for the for loop we have the conditional expression you know conditional block as well if X is none, then we will continue.

So, whenever you know in the iterations that we are running for X in that particular sequence, whenever X is found to be none will you know we are using continue keyword that means will advance that iteration. So, the following lines of code after this particular logic this particular block are not going to be executed. So, in a sense we have suspended the execution of following lines of code after the continue keyword and we have moved on to the next iteration.

So, let us run this. So, after we run you can see that value of I comes out to be 12 because if you see here we are just adding on you know whatever value of x you know that we had earlier and then we are adding that into the I values. So, eventually the final I value is this. So, if we had not encountered none which is the fourth element in the list here in the sequence here, then we would we might have gotten you know higher value for I.

So, the next thing is about loop is the break statement the break keyword. So, sometimes this situation might require us to exit from a loop altogether. So, let us understand this you know scenario also through an example. So, here we have the sequence 0 1 2 3 4 5 5 and here you know we have initialized I as 0 and in the for loop that you know I am running here x in this sequence whenever then we have the conditional expression, whenever X is found to be greater than 4 I am using the that is found to be truth and I am using the break keyword.

So, whenever the x reaches that particle value I will exit out of the loop. So, not just that the following statements would not be executed over all other remaining iterations of the loop they would also not be executed, when we use the continuing keyword. So, that particular iteration the following you know statements following lines of code are not executed and we move to the next iteration, but when we use the break keyword we exit out of the loop that means all the remaining iterations along with the current iteration that is going on.

And the following lines of code all of that would not be executed. So, let us run this, now let us find out the value of I. So, you can see here it is 10. So, because here at the moment with is this 5 value that is here immediately you know that break would be applied because X is greater than 4 and then we exit out of the loop. So, that is why again the value of I is on the lower side here. Now next scenario where we can use the break statement is that, sometimes we might be running you know nested loops.

So, break can also be so, whenever we and while we are running nested loops there might be scenario where we would like to exit from the inner loop. So, there also a break statement can be you know that keyword can be really useful. So, in this case you know same thing we can understand through this example. So, I am running 2 loops here. So, first loop is for I in range 5

and for J in range 5.

So, range 5 will actually give me those number of you know you know iterations there. So, whenever if J greater than I. So, whenever the value of J becomes greater than I would like to break out of that loop and you know after that we would like to you know within the inner loop you would like to print the combination of I and J as well. So, if I run this loop. So, first the outer loop would be run starting from 0 to 4 and when the outer loop is reaching here.

So, within that loop and the value of J becomes 5. So, it will be greater than I. So, we will exit from there. So, you can see 4, So, the whole thing has been you know printed here. So, next thing that we would like to discuss is the while loop.

(Video Ends: 33:10)

So, we will stop at this point and in the next lecture will start our discussion from the while loop, thank you.

Keywords: for loop, while loop, else if loop, conditional expression, nested loop, break statement, elif block, iterations.