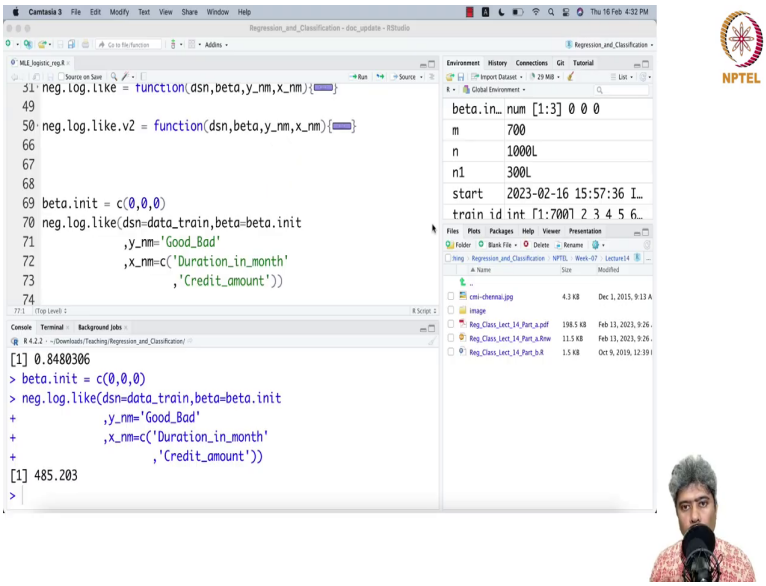
Predictive Analytics - Regression and Classification
Prof. Sourish Das
Department of Mathematics
Chennai Mathematical Institute

Lecture - 39
Hands on with R: Measure Time performance of R code

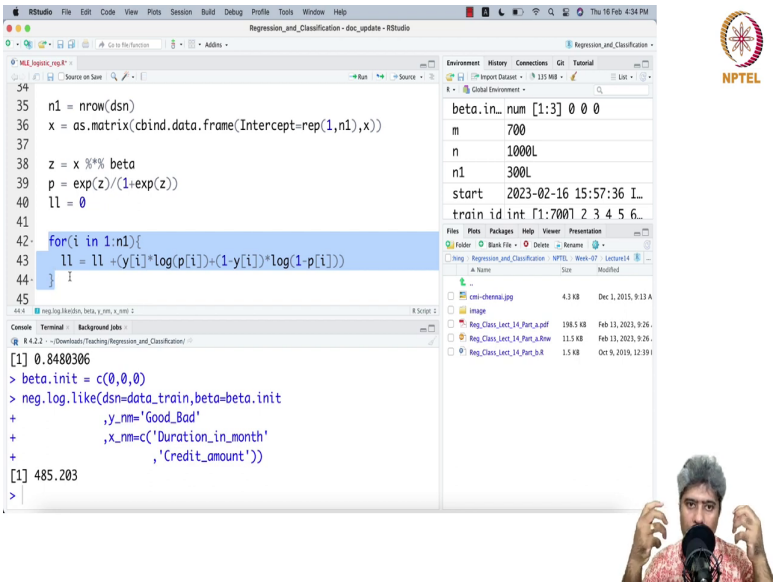
Hello all, welcome back to the part D of the lecture 11. I am going to do a small demo on time complexity of R and of the code that we wrote last in last video.

(Refer Slide Time: 00:32)



So, this is if you are from Computer Science or Engineering background and if you have taken a course on algorithms, then you understand this issue of time complexity that if you write a code how much time it takes to complete the code. That is the typical time complexity. And if you see I want to show you a demo here.

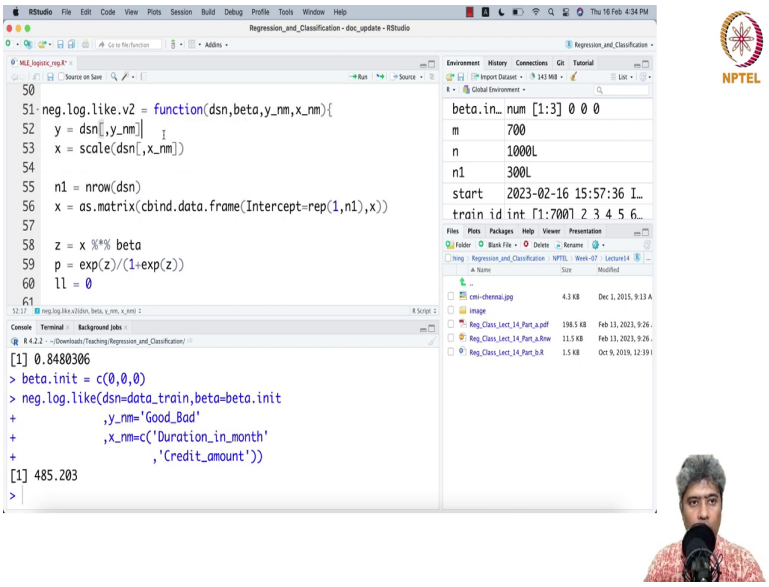
(Refer Slide Time: 01:12)



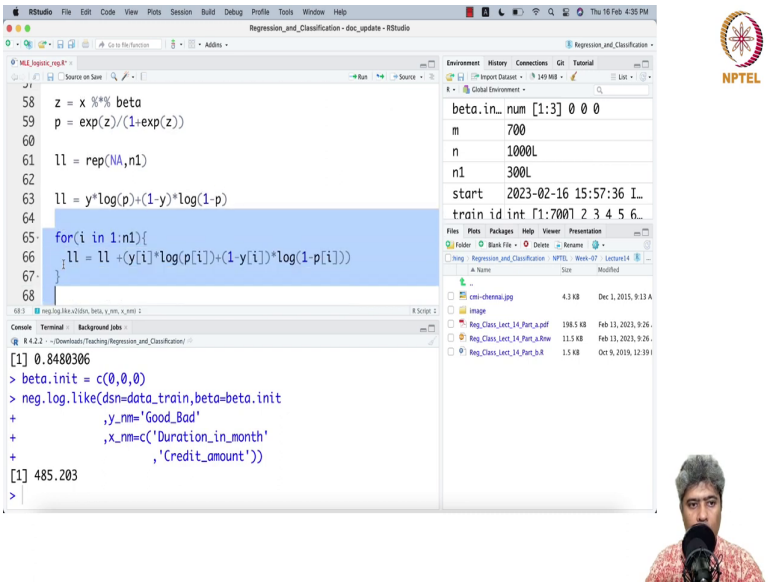
So, you see if you have this is the negative log likelihood, I function that I wrote and here I wrote a for loop to compute the log likelihood value. Now, what happens is when in my previous video you have heard that I am saying that if you write for loop in R then it might be not efficient way to write it, instead we should write it in a vector form. What happens is R is a high level language and its compiler is in C and the compiler creates the machine language.

So, the whole thing gets in computed in C language, but as a result what happens if you running. So, it is R is often called R both R in Python are often called interpretable language. Means, if you write a for loop every time the loop is running it is going back to C doing the computation coming back. So, that is where the for loop is. Now, if you say run let me run copy this function and I will show you this could be actually a painful it could slow your computation to good extent.

(Refer Slide Time: 02:30)

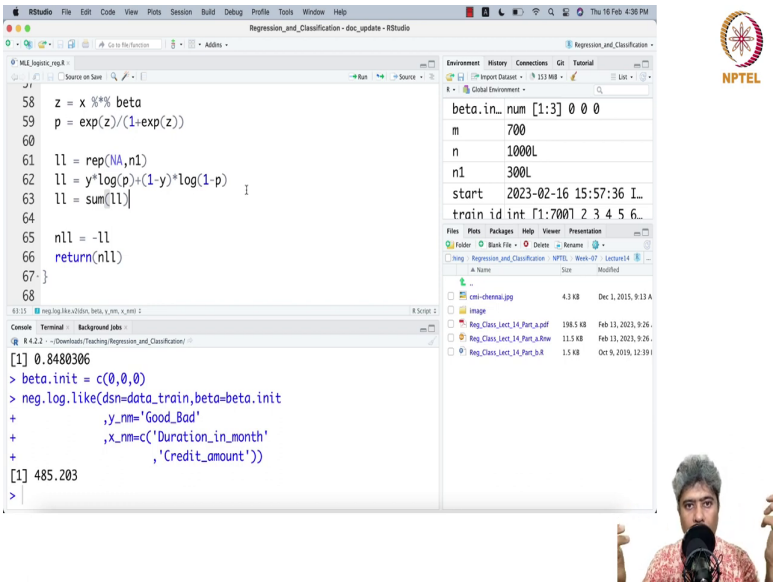


(Refer Slide Time: 02:41)



Let me write a v2 version and v2 version will be instead of. So, basically what I want is instead of ll 0 I will just say replicate NA comma n1, n1 is the yeah and then ll equals to y times log p plus 1 minus y; 1 minus y times log 1 minus p and I do not need this piece of line and then I need to do ll equal to just some ll.

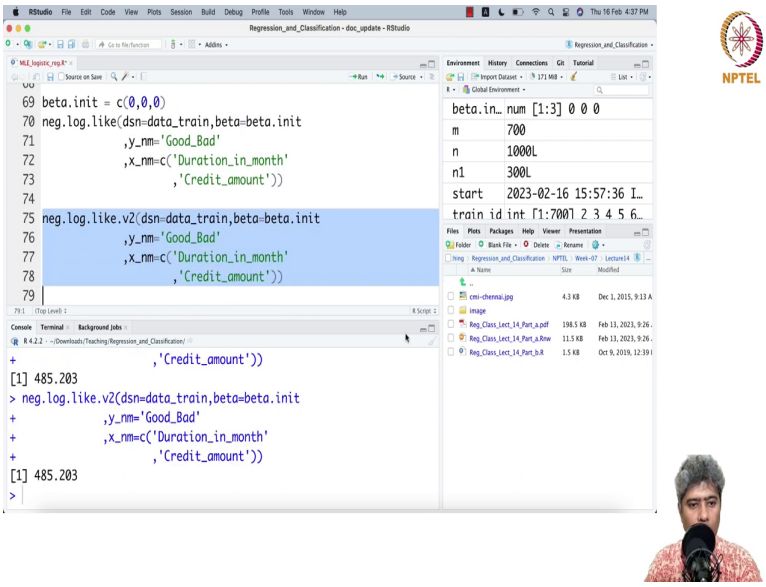
(Refer Slide Time: 03:24)



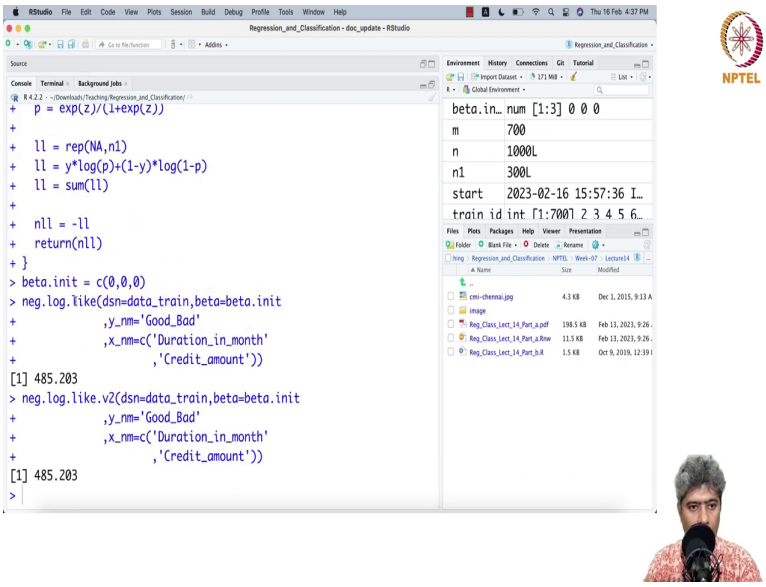
That is if I write this if I just replace that two line with for loop with this vectorize this is called vectorization of code, ok. So, you avoid the. So, I am just doing vector operation here. So, if I do this then it suppose to work much better, it is supposed to do quick job. So, now I have in the negative log likelihood is the way I wrote the function previously and negative log likely log like dot v2 is the next one which I vectorized.

So, so what I am going to do or let me just run this code both of them and this is what I am getting from the older version.

(Refer Slide Time: 04:24)

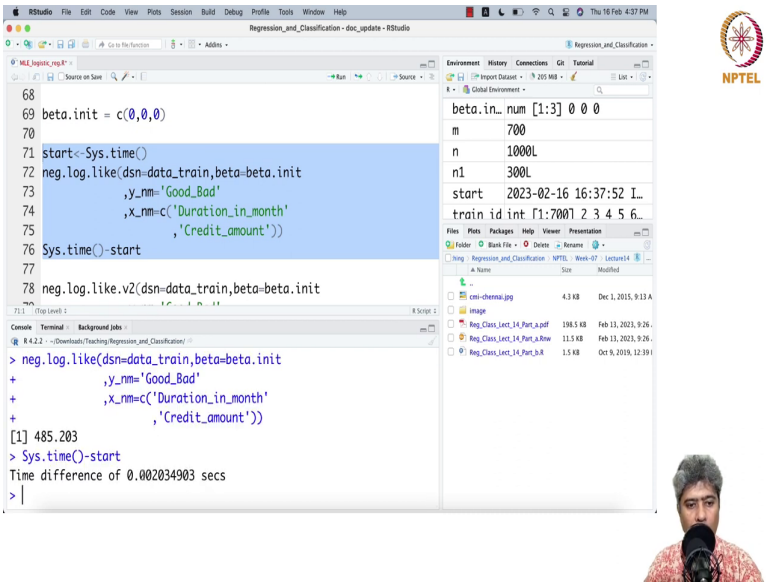


(Refer Slide Time: 04:40)



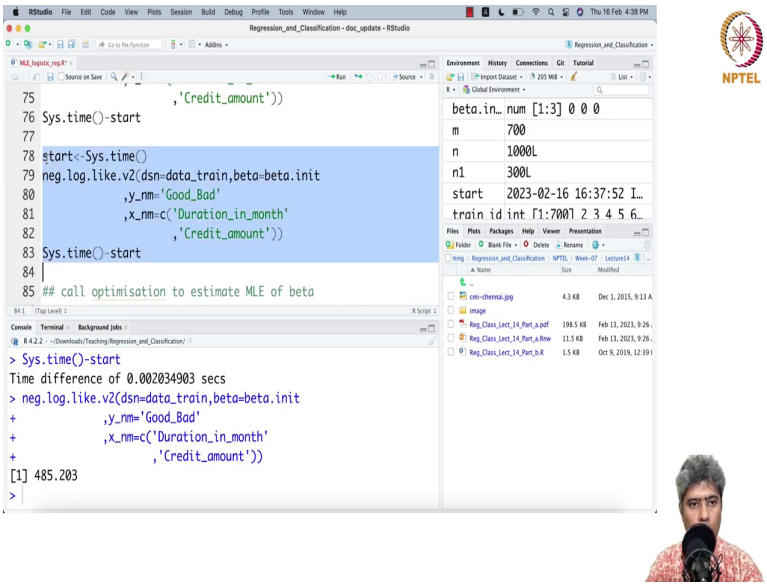
Now, first I am if I it suppose to give me the exact same value for my newer version also the v2 1 dot v2 as well, right. Yeah, you see here I called the older version which give me 485.203 and this is my newer version which is also give me the exact same value. So, it returns the exact same value. But how much time it takes?

(Refer Slide Time: 04:57)

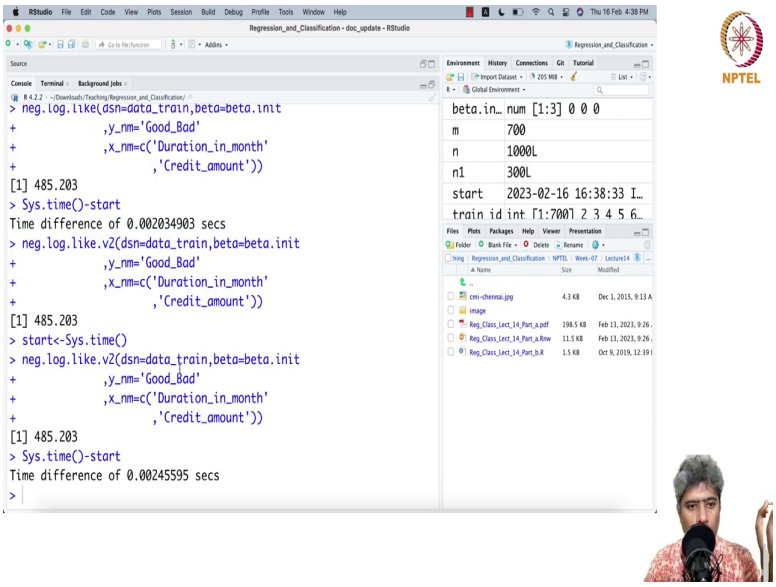


Let me just do that start Sys dot time and Sys dot time minus start s t a r t, alright. So, if I run the older version. So, it is taking 0.00203 and similarly if I taking if I write this one of the (Refer Time: 05:40) sorry, but that s t a r t. So, this is the one I am going to do and then yeah.

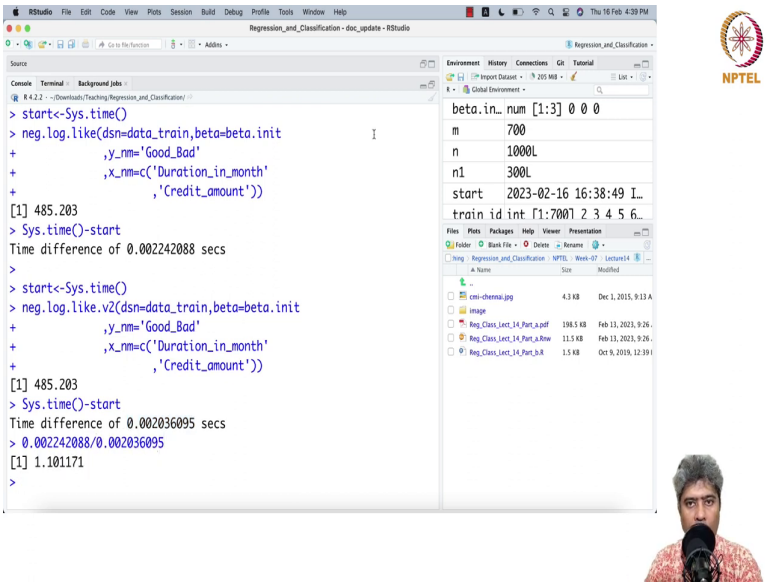
(Refer Slide Time: 06:01)



(Refer Slide Time: 06:13)

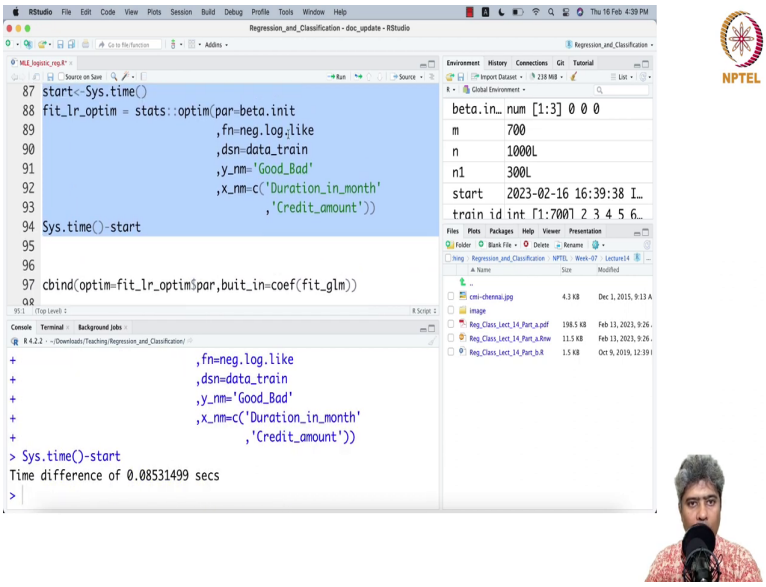


(Refer Slide Time: 06:31)



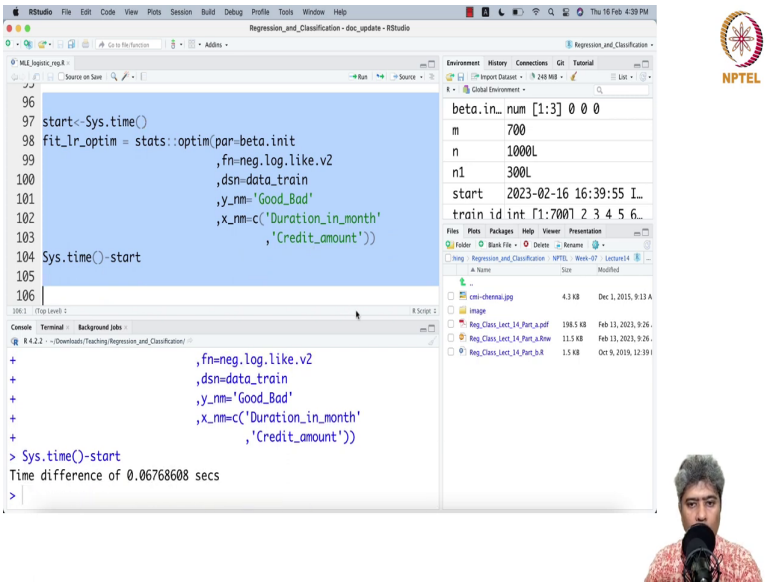
So, if I run this. So, let me just run it once more. So, the first one alright let me see how much time it is taking. So, this is the old one and this is the new one. So, this is the old one divided by new one. So, it is looks like the new one is slightly faster. So, what I will do and same way what we can do is we can do the optimization will do.

(Refer Slide Time: 06:55)

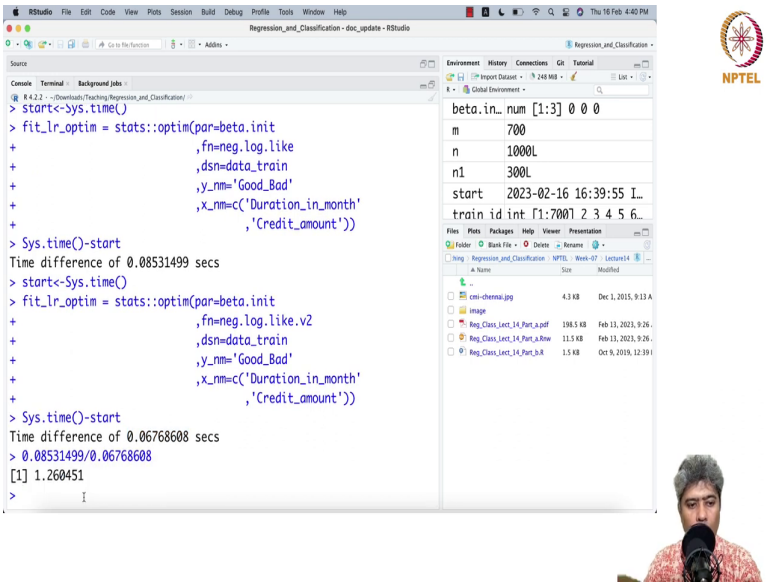


So, let me just do the optimization with the old version and the new version. So, an optimization will have to do some computation additional computation. So, if I just run this with. So, it takes about 0.8 08 second with the older version.

(Refer Slide Time: 07:22)



(Refer Slide Time: 07:34)



And if I just run the same thing with the newer version it takes little faster it is little faster now clearly 0.067. So, about; so, you know this divided by this one is 1.26 almost 25 percent faster. So, 25 percent faster is this means if in this way you can make your code efficient.

Now, you can tell me that ok sir why if I am whether one code is doing it 0.08 second or 0.06 second do, I really care in this case. So, yes if your data is small probably you do not care that much, but as a developer you always have aspire for writing code for very big data you want to handle very big data very big models right. Now imagine you have such a big data that your one code you have written two code; one code or two programmer has written two code.

One programmer's code take 8.5 seconds another programmer's code take 6.7 seconds which code 8.5 minutes and you know 8.5 hours this could be like 8.5 hours and this could be 6.7 hours who naturally the same code it will run the same code will run on very big large data

also like you know millions of data points. Here you have only 10 1000 data point actually 700 data points on which you are trying to train this model.

So, over if, but if you have millions of data points it might take 8.5 hours or 6.7 hours. Now if you are running it on the cloud; that means, and cloud services typically charge you by minutes sometimes even by seconds. So, you are reducing your model training cost by almost 25 percent if you use this code the older code it will cost you 125 dollars and if you use the efficient code it will cost you 100 dollars.

So; that means, you are saving 25 dollars in training very big large data on cloud and there is another aspect to it is; that means, your algorithm has less carbon footprint because these cloud machines take like you know burn takes lot of electricity and those electricity are typically has you know at least in country like India uses lot of fossil fuel.

So; that means, effectively a efficient code means less carbon footprint. So, you better write your code which runs faster and gives you code as accurate as possible. So, this time complexity unfortunately we statistic students our statistic students do not get to read much, but it is extremely important to study algorithms and time complexity, space complexity all these things. So, yeah.

So, this was a small video I thought I should show you guys that it is extremely important to write an efficient important efficient code everyone should watch this video and everyone should you know understand these concepts of time complexity.

Thank you very much, see you in the next video.