simple linear regression

This is a super important algorithm because the techniques that we are going to learn in this will also be applicable in deep learning, when you're probably learning the first neural network, which is called an artificial neural network. Right. So it is super important we try to understand this and remember this machine learning algorithm. I'll try to break down this into separate, separate modules so that you can understand it in a better way. Now what kind of problem statements do we solve with simple linear regression as the name suggests, regression. So definitely in supervised machine learning technique, if you have a regression problem statement, we can definitely solve it with the help of simple linear regression. Now what exactly simple linear regression algorithm are all about. Let's say that I have a specific and easy data set. Okay. Let's say I have a data set. And this particular data set has features like age and probably weight. Or instead of writing age and weight, let's basically just write something like weight and height okay. Now, as I said, regression problem statement. Obviously I will be having some data points. Let's say, uh, if the weight is somewhere around 74kg, my height may be one 70cm. If the weight is 80kg, the height may be one 80 centimeter. If the weight is somewhere around 75kg, the height may be one 75.5cm. Let's say this kind of data set I have okay. Now in this particular data set, our main aim is that we need to train a model. And this specific model whenever we give our new weight. New weight, it should be able to predict the height. Okay. This should be able to predict the height. And now you just see this okay. So this feature is basically our independent features. Independent features, and this feature is specifically our output or dependent feature. Okay dependent feature. So this is what we are planning to do. And this model that we are going to train is with the help of simple linear regression. Okay. Simple linear regression. Now why do we say it as simple linear regression. Understand over here how many input features we have. Over here we just have one input feature and one output feature. Whenever we just have just one input feature, then we say it as simple linear regression. If we probably have multiple input feature, then we can basically call it as multiple linear regression. But it is very important to just understand simple linear regression. Then automatically whatever terminologies and maths that you probably will learn in this will get applied to multiple linear regression also. Okay. So in short, uh, what are we trying to do? Uh, as I said, for this specific model, we will train it. We will train it with this specific data. Okay. And later on this model, whenever it takes new, it should be able to predict the height. Okay. If I really want to show you in a. Geometric way. What are we trying to do? Let's say this is my weight and this is my height. Okay. Now with respect to this particular data set, let me plot some points. Let's say there are some points which is plotted like this okay. So with the help of simple linear regression, what we are trying to do is that we'll try to create a best fit line. Okay. And this best fit line will actually help us to do the prediction for the new weight. So let's say how does the prediction happen. Let's say once we get this best fit line okay. And this best fit line should be created in such a way that you know that the distance between the true true points, the true points basically means these are my true, uh, output. Right? These are all my true output. One 70cm, one 80cm, one 75.55cm. So the distance between this and this predicted point, this and this predicted point, distance between this and this predicted point, this and this predicted point, this and this predicted point, this and this predicted point. If I do all this distance and this is basically the error, right? Because these are my true points and these are my predicted points. So right now just understand till here I will again explain in more detail. Right. So this distance should be if I do the summation of this distance it should be minimal. And that is the technique we basically use to select the best fit line. Now once we create this best fit line, whenever we get a new data

point, let's say with respect to this particular weight, how do we predict our output? What should be the height? All we do is that we try to just put this line on this particular see, and we just try to see where this point meets this particular line, which is my best fit line. This is basically my best fit line. Okay. And then we try to see where in the y axis it is basically meeting. Okay. So suppose if it is meeting at this particular point. So this specific is my output for this specific weight. This is what we actually do in simple linear regression okay. Over here the mean is that we just try to create a best fit line such in. Whenever we give our new weight we should be able to predict what should be the height. Okay. Now as we go ahead, uh, let's understand what is the mathematical equation of this particular best fit line. And when I spoke about this error, what is this specific error? We'll also try to understand and we'll try to understand in a very amazing way okay.

Simple linear regression.

We have understood, you know, what is our main aim, what we are actually trying to do with the help of simple linear regression, we are actually trying to create the best fit line. But before we go ahead and understand the maths behind it, we really need to understand some of the notation that I'm actually going to use while explaining this entire machine learning algorithm. So let me go ahead and let me again consider this. So let's say that in the x axis I have my weight. And, uh, in the y axis. I basically have my height, and I do make sure that I populate some of the points which are like this. Let's say I've just randomly created some points, and this points basically mention our data set right. Whatever data set we have. Right. And this points only we are going to train our specific model. You know, uh, and what we are planning to do is that we are planning to create a best fit line in this. Right? We are trying to create this best fit line. Now, in order to create this best fit line, we basically provide some equations to this best fit line. You know, you can basically use the equation of a straight line like y is equal to m x plus c. Or if you have seen some of the research paper they may also use something like y is equal to beta zero plus beta one multiplied by x. You know, um, and you may also have another equation, uh, which, uh, you know, Andrew Ng also uses this specific equation when he teaches it, you know, so he uses h of x, which is nothing but theta zero plus theta one into x. Right. So this equation so in this entire simple linear regression that I'm actually explaining again some amount of credits is definitely goes to Andrew. And he's quite amazing with respect to teaching any kind of topics. So I will also be using this specific notation which is basically nothing but h of x is equal to theta zero plus theta one into x. Okay. Now here if I talk about x, x basically means my independent feature. Now in this particular case it is weight okay. Now let's go ahead and let's try to understand what is theta zero and what is theta one. That is the most important thing that you really need to understand in this best fit line okay. And what is this h t of x. Okay. So over here let me write down the equation h theta of x is equal to theta zero plus theta one into x. Now first of all what exactly is theta zero. So theta zero. We can definitely say it as intercept. Why do we say it as intercept. In short, let's consider that if my x value is zero, then what will happen if my x value becomes zero? Then what happens? H state of x is nothing but theta zero. Now this basically indicates that when my x value in this particular, uh, you know, in this line that I'm actually drawn, right. Wait, is my x in the x axis. Right. So when weight is zero where does this particular line meet at the y axis okay. So this particular line you will be seeing it meets

somewhere here at the y axis. So this particular point is nothing. But this particular point is mentioned as intercept. So in short what does theta zero basically indicate. It basically indicates the intercept. That basically means when x is zero what is the value of theta zero okay. So theta zero wherever it is meeting the y axis that is basically called as intercept okay. So this is the first thing that you really need to know. Very very important. Now let's go and understand what is theta one. Theta one is nothing but it is called as slope or coefficient. Okay. Now whenever I talk about slope or coefficient, what does this basically indicate? This basically indicates that with the unit movement in x axis. Suppose if there is a unit movement in x axis. So if I try to plot this over here in the y axis. We are just going to try to see that. What is the what is the movement with respect to the y axis? So in short, what is the slope movement? Okay. So this is the movement that we are trying to see. So with the unit movement in the x axis, what is the movement with respect to the y axis that indicates something called as slope? Okay. So very simple. This is also called a slope or coefficient that is basically indicated by theta one. Okay. Suppose if I have many many independent features then this equation becomes something like this h theta of x is equal to theta zero plus theta 1X1 plus theta 2X2 plus theta 3X3. Like this up to theta and x n. So I can just write theta and x n. So this basically indicates that I have many many features. So definitely I'll be having that many number of slopes like theta one, theta two, theta three till theta n. But here since I just have one independent feature. So I'm just going to basically say that okay, I have just one slope okay. One slope. Now this is the basic understanding with respect to theta zero plus theta one x right. So x is nothing but my data points with respect to the x axis. Now one more very super important thing. Now as I said that once we create our best fit line, whenever I get a new data point, let's say this is my new data point with respect to weight, okay, with respect to weight, this is my new data point. Then how do I predict my height? All I do is that I try to project this over here, and I try to basically see what is my y value. Okay, so this is basically my y value. So this point I can say this is my predicted point right? Predicted point. So whatever points I'm actually predicting over here that is mentioned by h theta of x. So h theta of x is nothing. But this is basically my predicted point. Let me give one more notation. This notation is nothing but y hat okay. So y hat I'll be using to basically specify my predicted points. Okay. Predicted points. Very simple. Okay, now let's go and understand. What exactly is error. Can I say error? Over here is nothing but the difference between. See, this orange point is nothing by y point. Right y point. Why I'm saying because over here in my data. Right. This is my output. Let's consider this is X. This is Y right. So this basically is my output. Right. And the points that I probably get let's say this is my y point. And let's say with respect to this, my true point is some my predicted point is somewhere here. Right. So if I do this difference then with respect to this specific data point, this should be my error, right? This should be my error. So error over here. Nothing. What I'm doing I'm just subtracting y minus y hat right y minus y hat. So in short, I'm just trying to find out the difference between this two points. And that actually specifies my error. Again let me tell talk about like what is our main aim. Our main aim is to basically come up with a best fit line wherein when I try to calculate or do the summation of all this error, it should be minimal. If it is not minimal, suppose there is another best fit line that actually brings up this error or minimizes this error. I'm just going to select that specific best fit line. Okay, so I hope you have got the idea about error was instead of x is y what y hat is and what all uh with respect to theta zero what is theta one. And all right. So these were some of the notations that you really need to know when we start learning about the simple linear regression. Now as we go ahead we will try to understand how this best fit line is basically created. Because we cannot just go ahead and randomly create many best fit line, and then probably try to check each and every, you know, uh, what is the error if once I do the all the summation of the error, which is less, which is less, and then try to select a best fit line, no, this

is not efficient. So we'll try to find out an optimized way wherein first of all we will try to create one best fit line. And then we'll try to rotate this best fit line by changing the values of intercept and coefficient. Right. We are going to change this value of theta zero and theta one, theta zero and theta one. And then try to find out the best fit line with minimal error. Right. So this is what we are going to learn. And we'll try to see how this optimization technique will basically happen and how we are going to perform all these things.

We are going to continue our discussion with respect to the simple linear regression. And in our previous video, we have already seen what is our main aim. And we have understood about the notation. And we also found out that okay, let's try to find out an optimized way to actually get the best fit line. Okay. Now for this, uh, we will be using a cost function. Now here I'm just going to mention about the specific cost function. And this cost function is basically given by a notation. Let's consider this specific notation j theta zero comma theta one is nothing but one by two m summation of I is equal to one to m. H theta of x of I minus y of I whole square. Now what? This h theta of x of I minus y of I whole square. As I said that we have to make sure that we have to create a best fit line in such a way that when we do the summation of all the specific errors, it should be minimal. So that is the reason we are taking up this cost function in this specific way. H of x of I is nothing, but this are my predicted points, right? These are my predicted points and these are my truth points. So basically the true output right. And I'm going to do the summation is basically do this. Uh when we do the subtraction we basically get the error value. And the reason we are squaring it because uh, the technique, the cost function that we are using this, we basically say this cost function as mean squared error okay, mean squared error. And there are a lot of advantages of using mean squared error, which we will probably discuss as we go ahead. Now you may be thinking, Chris, is there different kind of cost function also used? Yes, absolutely. You also have something called as mean absolute error root mean squared error. We will all try to discuss about all this kind of cost function and what is the advantage and disadvantage with respect to that okay. But here, right now the reason I'm taking up this cost function, because we really need to minimize this cost function. And we need to minimize in such a way by changing the theta zero and theta one value. Theta zero basically tells us about the intercept. And theta one basically talks about the slope. Okay. So what is our final aim? I'm just going to write out finally what we need to solve. Okay. What we need to solve. This is super important. Then only here I basically used the cost function. So our main aim is basically to minimize the cost function j theta zero comma theta one, which is basically denoted by this specific equation. That is, one by two m summation of I is equal to one to m h theta of x of I minus y of I. Okay. Whole square. Okay. Uh, since we are dividing by m, so we are getting this as mean right mean squared error. We basically say this cost function and mean squared error. And how we are going to minimize this by changing the value of theta zero and theta one. So we really need to continuously change this theta zero and theta one and try to find out the best fit line wherein we do the summation of this particular error. And this should be minimal okay. Minimal as possible okay. So this is what is our final aim that we are looking at okay. And how how we are going to do this. Let's try to discuss about it now. I hope everybody knows what is the equation of my straight line. This is nothing but h theta of x theta zero plus theta one into x. So that basically means if I probably consider this okay, the best fit line, whatever straight line I'm actually getting, let's consider that I'm getting this specific straight fit

line. So this line is basically indicated by this equation. Now one important thing guys, since there are two variables like theta zero and theta one. Uh let's consider that theta zero is zero okay. Let's consider there is a reason why I'm saying because I really wanted to show you everything by drawing it in a 2D diagram. So let's consider theta zero is equal to zero. If I say theta zero is equal to zero, that basically means when my x value is zero, it is basically passing through the origin. Right. This this line is basically passing through the origin. So this intercept is exactly zero okay. So now my equation becomes a straight of x is equal to theta one. Divide by x. Let's consider that, uh what I'm doing over here. My intercept is basically passing through the origin. I'm just considering it so that I can draw a cool 2D diagrams to make you understand more in depth. So currently I'm going to just use this specific equation okay. Now let's consider some of the data points. Let's say this is my x and y. I'm just going to write 1122 and three three. Okay. Let's consider that this is my entire data set. And for this particular data set I really need to create a best fit line. Because this will give you a clear idea that what we are trying to do. So what we will do, we will just try to create this line like this. Okay, this is my x, this is my Y. So let me create it a little bit straight so that it looks cooler and good. So this is my X and this is my this is my x. And this is my y. Let's say these are my values like 123. And with respect to x y also I have one two and three okay. Now let's start okay. And obviously everybody knows what is the equation. It is nothing but a state of x is equal to theta one multiplied by x. Why. Because theta zero is equal to zero because we are considering it passes through the origin okay. So I have my data points. My data points are one comma one two comma two three comma three. So let's go and plot it okay. So here is one comma one. Then this is two comma two. And this is basically three comma three okay. Now if I have this particular data points I am going to use this equation of a straight line to make sure that I plot my best fit line. Okay. Now let's consider we need to initialize theta one right with something. So let's consider let theta one is equal to one because this is my slope right. And we have to initialize the slope. And later on we will keep on changing the slope to get the different different best fit line. So if theta one is equal to one then when x is one what will be my h theta of x. So my x theta of x will be actually one when my x is equal to one, right? I hope this is very much clear. Right. So uh, let me just put this points. Then this will be basically my predicted uh, points over here with respect to theta of x. Now when x is equal to two, what will be my theta h theta of x h theta of x will be. Since I've initialized theta one as one when x is equal to two, that basically means h of x will be two. So my second point will be somewhere here, right? The predicted point h of x okay. And similarly when my x is equal to three, when theta one is equal to one, that is, my slope is equal to one, then h street of x is equal to three because three multiplied by one is nothing but three. So my third point is somewhere here. Now if I try to create a best fit line. So this is going to pass through all these points right. All these points. So it is going to exactly pass through all these points. And obviously it will also pass through the origin okay. So in short these are my predicted points. And over here I also have my truth points right I also have my truth points. Now let's go ahead and apply this cost function J of theta uh theta one. This is this entire equation. So what is my cost function over here. My cost function is very simple. I will write j theta one because theta zero is zero right. It is passing through the origin. So I will write one by two m summation of I is equal to one to m. And here I will basically say h theta of x of I minus y of I whole square. Right. So how many points I have. Totally. I have three points right. In my data set I have three points one, two, three. So I'm just going to basically write one by two multiplied by three. And I'm just going to basically say summation of I is equal to 1 to 3 okay. So it will be the entire summation of 1 to 3 right. So no need to write 1 to 3. So I will start expanding this. So first thing is what what is s of x of I when I is equal to one. So over here you can see when I is equal to one h theta of x of I. The predicted point is also one right. So I'm just going to basically write one minus what is y of I at that point of

time. So y of I is also one. And here also we have plotted it. So one minus one whole square okay. Then plus will go to I is equal to two I is equal to two. Over here you can see when the x value is two instead of x of I is also two and y is also two. Right. So over here I'm going to write two minus two whole square. And similarly this is three minus three whole square okay. So overall if I try to find out it is zero. So what does this basically indicate. My cost function is zero. This is absolutely right. Right. Because obviously there is no error. Exactly. The best fit line passes through all the points. Right now this is one way okay. Now let's go ahead and consider and let's change our theta one value. Let's say that our theta one value is. Now let's consider that our theta one value. Let's say our theta one value is 0.5. Now if it is 0.5 then what will be h theta of x when if x is equal. One, so it will be nothing but theta one is .5.5 multiplied by one, right? So this is going to be point five, right. So when x is one. So my h theta of x will be point five. So let me just denote this particular point over here it is be point five okay. Now similarly when uh theta one is point and if x is equal to two then here you will be seeing h theta of x will be one. Because here we are just multiplying theta one. Theta one is nothing but one okay. One multiplied by two sorry one multiplied by sorry. Theta one is point five. So in this equation x t of x theta one x right point five multiplied by two is nothing but one. Then h of x is equal to. When x is equal to three, this is going to be 1.5 okay. So when your x is two I'm actually getting one. So I'm going to basically note this particular point. And when x is equal to three h of x is nothing but 1.5 which is basically mentioned at this particular point. Okay, now here you will be able to see that I am just going to draw this next best fit point. So I'll make it a little bit better so that the point is basically drawn. So this is my best fit line for the next one. Then. Sorry. Uh. So. Three. Five. Okay, now, this time, I should be getting a best fit line. Okay, so this is my best fit line. Okay, now, here, you should be able to understand that these are my predicted points. This this this are my predicted points. Okay. Now with respect to this again let's go ahead and calculate j theta okay j theta one. So here I'm basically going to write what is j theta of one j. Theta of one is nothing but one by two m. Again I'll apply the same formula. So this will be one multiplied by two one divided by two by three. And now let's go ahead and try to find out. So when my theta one is 0.5 let me do write it. Let me write it down below okay. So that you will not miss anything okay. So the for the first case uh uh, let's consider I have also drawn this specific point and let me draw one more, uh, amazing graph okay. And this graph will be mentioning something very important. So let's create one more graph over here. And we will continue. We'll continue with respect to different different values. This graph is between theta theta one and j of theta one. Okay. And let me denote it some points like this point five. This is basically 11.52.02.5. Right. And similarly this is .51.0. Okay. And this is 1.5. And this is 2.0 okay. So this points I'm just creating along with this a different graph over here. And this graph basically indicates uh the graph between uh theta one and J of theta one. The relationship between theta one and J of theta one. Now in the first case when my theta one was one, what was my j of theta one. In this particular case it is zero. So I'm just going to basically plot this particular point okay. When my theta one was one. And here you can basically see my j of theta was zero. So can I plot this particular point somewhere here. So I'm plotting it somewhere here okay. So this is my first point that I've got now with respect to theta one is point five. Let's go and compute my J of theta one. So j of theta one is nothing. But I will start writing it down over here. And you also make sure that you write this particular equation very much easily. So it will be nothing but one by two multiplied by three. And the first case what is h of x when uh theta one is 0.5. So over here you can see when if x is equal to one. So here you can see that h of x is 0.5. So here I'm going to subtract right point five minus one whole square okay one how I'm getting my true value one is over here right. Still my real point is over here. So 0.5 minus one. So this is the error that I'm actually getting right. Similarly this is the error I'll be getting here. And this is the error that I'll be getting here. And we

really need to write that specific error. So point five minus one whole square. Then I have one minus two whole square plus 1.5 minus three whole square. So once I try to find it out. So this will be when my theta one is equal to point five. Right. When my theta one is equal to point five, the j of theta one. Just try to compute it. You will approximately be getting somewhere around 0.58 okay once you do the calculation. So over here my my theta one is 0.5. I'm just going to get J theta one as 0.58. So .58 will come somewhere here. So I'm just going to make this point somewhere here okay. Not here. Sorry. Uh, some area with respect to theta one is 0.5. I'm going to get the point somewhere here. Okay. And this is them. Okay. Now let me consider one more thing. Let's consider what will happen if I consider my theta one is equal to zero. If I consider theta one is equal to zero. That basically means when x is equal to one, that basically means what theta one is zero zero multiplied by one will be zero. So my first point will be here. My second point will be here. My third point will be here. Right. So this will basically be my next best fit line. And obviously the error will still be higher. Then obviously I consider that when I probably try to, uh, calculate j of theta one, J of theta one, if my theta one is equal to zero. In this particular case, what will happen j of theta one. Just try to compute it. Uh, it will be very simple. Uh, here I will write one divided by two by two multiplied by three. And theta one is first zero zero minus one whole square, then zero minus two whole square, then zero minus three whole square. Right. So I'm just going to write like this zero minus one whole square plus zero minus two whole square plus zero minus three whole square. Now obviously this is not the best fit line. Obviously I am going to get my error quite high. So it will be somewhere around 2.3. So J theta one and theta one is equal to zero. It is nothing but 2.3. Now here if I try to plot this when my theta one is zero and I'm getting somewhere around 2.3, which is somewhere over here right at this particular point. Now, if I keep on continuing doing with different different theta one value, after some points, you will be seeing that I will be getting this kind of curve. Okay. This kind of curve. And always remember when we are trying to get this kind of curve, okay. The main point that I got over here at this particular point, you know, that the best fit line was this line. Right at this point of time, my error was very, very less. So I can definitely say this is the point. This is the global minima. This this point is basically called as global minima. And we really need to achieve this particular point, or at least near to this point, to say that my error has definitely minimized my error has been minimized. Why? Because at this particular point, I am actually getting my best fit line. So this entire curve is called as something called as gradient descent. Okay. So this curve is something called as gradient descent. And this gradient descent is super important even in deep learning techniques. Okay. We have to come to this specific point by changing our theta one or theta zero values okay. So unless and until we get this particular point that basically indicates that we are going to get our best fit line. So here you can clearly see that when I had this specific point. And over here the cost function was zero. So that basically indicated that I did not have to move much weight over here. And this basically indicated my global minimum okay. Global minima. So we our aim is that we should definitely try to come to this particular point by moving along this particular gradient descent. Okay. And that is how we should try to change our theta one and theta zero values. Right now. Since theta zero is just passing through the origin, we don't have to change it much. But yes, definitely. You got an idea about gradient descent and our aim. What were we actually planning to do? We are planning to minimize this J theta zero comma theta one sin theta zero is zero. So I have not written theta zero specifically over here. But we had to minimize this by changing the value of theta zero comma theta one. In this particular case theta zero was already zero. So we changed theta one value and we minimize this value right. So here you could see that when we minimized because see when my theta one over here in this particular case my g of theta one was zero. The cost function was zero when when my theta one was one. Right. So at this specific instance I was

getting my cost function as zero. And this was the point where I got my best fit line. Okay. So in this way we should definitely change this things. But again I cannot keep on just randomly selecting different different theta one value. Right. That is not at all possible. So we should apply some kind of convergence algorithm. Now let's in my next video I will try to make you understand what exactly is the convergence algorithm. Because we cannot randomly initialize theta one value, we should select one theta one value, and we should try to find out a mechanism of changing those theta one value. Okay. So yes, uh, in the next video we're just going to discuss about the convergence algorithm. So I hope you are liking it. I hope you are able to understand it. If you have any queries, please make sure that you write down in the comment section. Thank you.

We are going to continue the discussion with respect to simple linear regression. In our previous video, we had discussed about cost function and how we could actually find out different, different cost function by changing our theta one value. And we also saw that we could form a gradient descent curve, which is this kind of curve. And our main aim is basically to come up to this global minima. Because in this global minima, when come near this global minima, that basically means we are very much near to the best fit line because our cost function is quite minimal. But when we are trying to find out our cost function, we were randomly changing our theta one value, which is our slope. First time I took one, then we took point five and then I selected at zero. This is not an efficient technique, so I told you that we will be discussing about the convergence algorithm, which will actually help you to initialize one theta one value and automatically, you know, based on this gradient descent, it should be able to, you know, increase the theta one value or decrease the theta one value. So let's understand the convergence algorithm. So over here I'm just going to mention the convergence algorithm which is super important for the optimized technique okay. So this is the convergence algorithm. And this convergence algorithm main aim is to optimize. The changes of theta one value, which is my. Theta one value, okay. Which is my slope. Okay. In this particular problem statement. So what does convergence algorithm basically say. Convergence algorithm basically say repeat. Until convergence. Okay until convergence basically means until we reach the global minima, the specific global minima point okay. This specific global minima point. So here you can see that when we reach this global minima point we got the best fit line. Or at least if we come near this we will probably get the best fit line okay. So repeat until convergence. And here I'm just going to write a simple equation which is like theta of j. Okay. Theta of J is equal to theta of J minus. Okay. Alpha. Alpha. I'll talk about this. What exactly is Alpha? And then we are going to basically write derivative of theta j with respect to. The J theta one okay. Or here I'm just going to basically write theta j. Now what exactly this is okay. Now try to understand okay. Uh this theta j. Initially we'll initialize something and then we'll try to find out the cost function. And obviously you know that with respect to the cost function how does the cost function basically is getting created. So here if I initialize and create a graph with respect to theta j and j of theta okay. So this is what we created this graph right over here. You could see this right. You know that we usually get this kind of gradient descent right. Let's say with respect to and this equation will definitely work okay. Will this equation is definitely going to work. That basically means this equation actually helps us to change our theta value. Specifically in our problem statement, the theta one value much more efficiently, much more efficiently. So we are going to change this theta

one value much more efficiently. How? Okay. So let's consider my theta j value is somewhere at this specific point. So j of theta will be coming somewhere here. Okay let's consider that this is the point that I'm going to probably get. Okay. Now you need to understand what exactly this is okay. Derivative of theta j. Derivative of theta j with respect to j of theta j okay with respect to the. In short, we are just trying to find out the derivative of this specific point. Now whenever we say derivative in short we are going to calculate the slope okay. Now let's say we have initialized one theta one value randomly. At this point we got our j of theta at this particular point. Now our main aim is that we need to come to this specific point okay. Because why why this specific point. We really need to come. Let me just draw it much more broadly. So let's let's draw this curve like this. Our main aim is to come to this point okay. This point is basically my global minima or near to this point okay. This point if I come which is my global minima, then we are basically getting our best fit line. Okay. Now initially my theta value got initialized. At this particular point I got j of theta. But I have to make sure that I come to this specific point. My j of theta should come to this point so that my cost will basically minimize, right? Now my theta of J. How do I make sure that? How do I make the changes such that my j of theta will come over here that we are going to look at? Now, as soon as it comes over here, this derivative of uh, derivative of j of theta with respect to theta j is nothing. But we are trying to calculate the slope at this particular point. That is what derivative specifies okay. Now in this particular case how do we understand what is the derivative of this particular value. Okay. In order to find out the derivative, one important thing that you really need to find out is that whether this is a positive slope or negative slope, because derivative is nothing, but we are actually trying to calculate the slope okay. We are trying to find out the slope okay. So for this particular point, what we do in order to calculate the slope we create a tangent line like this okay let's say that we are creating this tangent line. So this tangent line will probably get created like this okay. Now when we are creating this tangent line, first of all you need to find out that see the right side of the line okay. Because one important thing that we need to find out over here that whether it is a positive slope or a negative slope, because if we are able to find out whether it is a positive slope or negative slope, then we will be able to understand whether we have to reduce our theta j value or whether we have to increase our theta j value. Okay. Now, in this particular case, at this particular point, if I really need to understand whether this is a positive or negative slope, just see the right part of the line. The right of the part of the line is facing downwards. Okay. If it is facing downwards we will be seeing this as negative slope. Now, when we say negative slope, that basically means whenever we try to find out the derivative, this value is going to be negative. Okay. This value is going to be negative. So what we are going to do is that we will just suppose I get a negative slope. So in short I will write theta j is equal to theta j minus some learning rate alpha I'll talk about why alpha is used and this will be my negative value okay. This will be my negative value. So in short this basically indicates that. My theta J. I have to probably add with some positive value, right. Because negative into negative will be positive. So I have to add the specific positive value. And it is true over here because I need to basically increase my theta j value. So if I do this iterative process again and again wherein I'm continuously getting a negative slope, that basically means I have to increase my theta j value so that I reach my global minima. So this equation is going to add some value to my theta j. How much value? We don't know. Since this is an iterative process, our main aim is to basically optimize or optimize this process of selecting different different theta one value. Right. So in this way what will happen? We are incrementing or we are increasing our theta j value. Then in the next theta j whatever will be my cost function. Again that specific point will come over here. Let's say it is coming over here. Then again I will try to go and find out this slope again. If it is a negative slope we keep on adding so slowly, slowly you will be seeing that we will be able to converge it towards the global

minima. Okay. This is one process. What about, what about now? In this particular case I got negative slope. What about if I get another point over here. And this particular point is I actually a positive slope. How do I get a positive slope. Because if I try to create a best, if I try to create a tangent line, just see the right part of the line. Right part of the line is pointing upwards. So this becomes a positive slope. Now in the case of positive slope I really need to decrease my theta j value, right? Let's say that initially my theta one was over here and I got my J of theta one over here. So here I have to decrease it okay I have to decrease it. Then only I'll be able to reach the global minima. So does this satisfy. Yes it does satisfy because if I right now theta j is equal to theta j minus alpha. And now this is my positive slope right. So here with respect to this, whenever I try to find out the derivative it is a positive slope. Now whenever it is a positive slope all I am actually doing I'm subtracting with some value some positive value. When I subtract with some positive value, that basically means I have to basically decrease my theta j value. And if I continuously perform this process, after some time, you will be able to see that I am going to basically come at this specific point. Okay. So obviously this convergence algorithm will obviously work because what I am actually doing is that we are trying to calculate the slope in such a way that where if it is a negative slope, the theta j value, it is going to increase. If it is a positive slope, the theta j value, it will decrease. Okay. So this is what is a convergence algorithm all about okay. Now there is one important parameter that we did not discuss about is something called as alpha. So what exactly is this alpha. Alpha is nothing. But this is specifically a learning rate. Learning rate usually is a smaller value that is basically initialized. Let's consider the learning rate that may be initialized for. My problem statement is .001. Now what is the importance of this learning rate? Learning rate controls the speed at which the convergence should happen. Okay, if it is a very, very small value, then it will probably take more time to converge. If it is a very big value, then it may happen in such a way that it may continuously jump here and there, and it may never converge. So it you always have to make sure that you try to select a smaller value, but it should not be a very, very smaller value. It should be like 0.001, which is a very good practice that we should try to select. Okay. In simple linear regression, only, uh, alpha is equal to .001. In a sklearn library that we use is basically getting selected. So if anybody asks you in an interview like what is the importance of learning rate, you can definitely say that it actually the controls, the convergence rate how slow the convergence should happen. Okay. So this is what it basically says, right. So I hope you have understood about the convergence algorithm. And this process continues. Unless and until we don't come near this global minima. And we don't basically converge over here as soon as we converge at this particular point, you will be seeing that we will try to get the best fit line, which is this specific line with respect to any kind of data points. So this is the most important optimized technique that we should use with respect to the convergence algorithm. Okay. And uh, this basically gets applied in an amusing way. And most important interview questions when you're learning machine learning algorithms. So yes, in the next video we are going to also continue and understand about the performance metrics. Thank you.

Finally, you know, uh, in our previous video, we have seen till here, we have understood what is the convergence algorithm and how we can actually update our theta value, you know, which is

my coefficient. Uh, you know, based on the positive and negative slope. And this was the, uh, theta or coefficient updation formula here. Alpha. Also we used it and this specific alpha was nothing but learning rate. Now I really want to make one final conclusion and probably write some some more equations. You know, and uh, we can just consider that till now, you know, uh, we saw that, uh, we have actually covered something called as gradient descent. Right. And gradient descent whenever we talk about gradient descent. So in short, we are talking about that, uh. Curve. Okay. So in short we are talking about this specific curve. Uh, and this curve is basically, uh, we get it from when we try to plot with respect to theta and g of theta. Right. So over here you'll be seeing that this is the curve that we usually get. And our main aim is basically to come to this global minima. Right. So this global minima. And we try to update our theta based on this. Right. Basically we try to find out whether the slope is positive or negative. And based on that either I need to update the weights, increase increase this specific coefficient or decrease the specific coefficient which have actually explained over here. Now for our example, you know, we had actually considered you know, we had actually considered what that our intercept is zero. That basically means our best fit line passes from this to this. And uh, since it passes through this intercept. So we consider theta zero is equal to zero. And we were changing theta one value okay. So this was basically changeable right. We did not focus more on theta zero. The reason why I did this is that because I wanted to show you the gradient descent in the 2D diagram. So over here when I'm drawing this this is basically in the 2D diagram. Let's consider that theta zero is not zero that basically let's say that if it is not passing through the origin then this 3D diagram may look something like this. Right. So it let's say this this entire diagram will look something like this. I am just creating for an example. So it may look something like this okay. It may look something like this. Let's say consider that it is coming. Uh, it looks like this. So one theta 1st May be somewhere here. Okay. Theta zero may be somewhere here. And our main aim is basically to, you know, merge towards this global minima. Right. So our aim is basically come to this global minima. So this will also get updated like this. And this will also get updated like this. And finally we reach the global minima so that we usually do in a 3D diagram. And obviously I cannot draw a 4D diagram but just understand it's like a inverted mountain. And from the bottom, uh, from the top you are basically coming to the bottom right. So bottom at one single point. So this particular point will be the global minima. Now if I consider theta zero and theta one let me go ahead and write my convergence algorithm okay. So what was my convergence algorithm again. So I'm just going to update my convergence algorithm okay. And the convergence algorithm was pretty much simple here I will say that repeat. Until convergence. The same thing. What I have written on top okay. Repeat until convergence. But here my theta j will get updated something like this. Right. So theta j r is see this symbol that I'm using right. This is an continuous iterative process. So that is the reason I'm writing this colon and equal to. So it will just get updated with respect to this theta j minus alpha which is my learning rate. And then derivative of theta j okay. And one point I have to use is nothing but j of theta zero comma theta one. Right now since my theta zero is not zero. So I have to basically find out the derivative of both these things now. And you know that what is j of theta zero comma theta one. Okay. We already explained it. So j of theta zero comma theta one is nothing but our cost function. So if I know my cost function it is nothing but one by two m summation of I is equal to one to m. M basically means how many number of data points I have, and then I have h theta of x of. I minus y of I whole square. Right. So this is the cost function that I specifically used for j theta zero comma theta one. Now my main aim is basically to go ahead and calculate this right derivative. Now I'll try to just show you it's okay if you don't know derivatives. Also guys but understand our main aim is to basically converge to the global minima. Right. So here I will just going to write something like this. I will just try to calculate this.

And over here you know that since you have theta zero and theta one your j value will be zero and one, right zero and one. So let me just go ahead and write it over here. So derivative of theta j. Okay with respect to G. With respect to j theta zero comma theta one is equal to derivative of theta j. Okay. And I have over here one by two and I'm just replacing j t dot comma one with the cost function. Right. Whatever cost function I'm writing. So one by two m summation of I is equal to one to m okay. And then I have over here h theta of x of I minus y of I whole square. Right. So this is my entire derivative of j theta zero comma theta one okay. And uh you know let's go ahead and try to find out the derivative of this. Now guys it is simple right. If I take an example, if probably in a high school you have learnt about derivative. Derivative basically means you're trying to find out the slope. Suppose if I really try to find out the derivative of x squared right? So what I will be getting, I will be getting with respect to c. See over here I'm trying to find out the derivative of x square with respect to x. So this will be nothing but two x right? It's just like an uh x to the power of n derivative is nothing but n multiplied by x to the power of n minus one. So this is what is the formula. Right. So similarly I'm going to apply this derivative for this particular equation. But when I'm finding the derivative I have to find it for j is equal to zero and j is equal to one. Right. So I'm just going to write over here if. J is equal to zero, right? So if j is equal to zero this entire equation what it will become is this entire equation, this entire equation, this top equation, this entire equation will be becoming derivative of derivative of theta zero j of theta zero comma theta one, which is nothing but derivative of. Partita zero one by two M and here I'm basically going to use. Summation of I is equal to one to M, and this will basically be I'm just writing the same right as t of x of I. X tilde of x of I minus y of I whole square. Right. So this will be my entire thing. Now here you can basically see that. Uh, fine. I'm getting this. Exactly. Uh, I'll just go and find out the derivative with respect to theta zero. So if you know this equation, always remember this is just like your x squared. This. Let's consider this is my entire x squared right. So x squared will be nothing but 2X2 minus one right. Something like this. So whenever we have this x square, whenever I try to find out the derivative with respect to this it will be 2X2 minus one. Right. This is what is the basic formula okay. So these are simple derivatives guys if you don't know don't worry okay. But just I'm trying to write out uh the convergence algorithm with respect to theta zero and theta one. Because now we have a 3D graph 3D curve which will try to converge in one single point. Okay. So so finally, uh, here I can basically write, uh, uh, after this, you know, uh, in short, if I really want to write. So this entire value will get converted to something like this, so it will be nothing but one by M, because two will be there, this two will come down, so two and two will get cancelled. And here you can basically write summation of I is equal to one to m h theta of x of I minus y of I whole square. Okay. So here, uh, sorry, not whole square because two will go down. Right. So this is my entire derivative and h of x. You know what is a street of x h theta of x is nothing but theta zero plus theta one x. Right. So if I try to find out the derivative with respect to theta zero so this will be one. And this is actually a constant right. This entire thing is basically a constant right. So this will basically become zero right. So theta of zero will be one. So it will be multiplied by one. So in short we will just be getting this specific term okay. So I hope you are able to understand it. What we will do is that while finding derivative right we'll replace this h of x with this equation. Right. And since we are trying to find out the derivative with respect to theta zero, first of all it will be two x okay. And then we'll try to find out the derivative of this particular equation okay. And here we know that if I try to find out derivative of theta zero with respect to this zero. So here I'm going to get one. And this is a constant. This will become zero. That is how derivative actually work. Now similarly with respect to g is equal to one I'm just going to find out the derivative. And this will now change as derivative of theta one j of theta zero comma theta one. Okay. So here you can see derivative of uh theta one. Okay. And then I'm going to just apply for this specific

equation. And it will be same thing. Summation of I is equal to one to m. And let me just equate this as theta of x with theta zero plus theta one into x okay. Theta zero plus theta one into x. I could have replaced it over here but it's okay. This is what I will be getting right now. We're just going to find out the derivative with respect to this specific equation. Now in order to find out the derivative it's very simple. Uh, again I'm going to take this entire two down. So this will be one by M. And uh here I'll be getting summation of I is equal to one to m uh h theta of x which is nothing but uh, theta zero plus theta one multiplied by x. Okay. Then it will be plus or minus um y of I. Okay. So this will be my entire thing. Then this entire thing will be going to the next step. Okay. And from this you know that now theta one is not a constant. Right. Because we are trying to find out the derivative with respect to theta one. So just just imagine like if I'm trying to find out the derivative of theta one with respect to theta zero plus theta one x. So what will happen. So over here theta zero will become constant. Theta one will be basically getting a derivative. So here in short if I try to find out the derivative this will be x. So I'm going to just multiply by x over here. Right. So this in short will be actually helping us to find out the derivative with respect to theta one. Right. So this is my entire derivative with respect to j is equal to one okay. I have just taken this and found out the derivative it is x. So I have multiplied with x okay. So these are the two things that we really need to write. So finally I can quickly write something called as repeat until converge. This will be my final statement or final conclusion with respect to simple linear regression. Repeat until convergence. Okay. And here we will be using two equations two important equations. First of all we need to keep on updating theta zero. So theta zero will be nothing but theta zero minus learning rate one by n right. So what is theta zero c with respect to theta zero. This is what we need to replace derivative of theta zero with respect to j of theta zero comma theta one. What is the derivative of j of theta zero. Theta one with respect to j is equal to zero. This is my entire equation right. So I'm just going to write this entire equation over here. So it will be nothing but one. By m summation of I is equal to one to m. And here I will be seeing h theta of x of I. Minus y of I. Okay. And then theta one will get updated as theta one minus learning rate and derivative of j theta zero comma theta one. What is the derivative we got over here. This particular equation right. So here I'm going to basically write one by m summation of I is equal to one to m. And again I'll be having h theta of x of I minus y of I. And here I'm going to use x of I. Okay. So this is my entire equation that I specifically get. And this convergence will be happening unless and until we get the best fit line or we get the uh, we basically get the best fit line or we converge to the global minima. So this is the entire thing that you really need to know. I know there are a lot of equations that has been used, but again, understand the phenomena, what we are actually doing, we are converging. We are converging in this gradient descent. Right. If I use theta zero and theta one then it becomes a 3D curve, right? 3D curve. And at the end of the day theta zero, wherever it is we have to come to the global minimum. So, uh, I hope you have understood it. If you find any difficulty with respect to the equations, don't worry. Okay. Uh, because implementation of some amount of theoretical knowledge is required if you don't know the equation. Also, implementation will be quite easy with the help of a scalar. Right? So yes. Uh, this was uh, with respect to simple linear regression. And in the upcoming videos we'll see some kind of practical implementation. Thank you.

In our previous video, we have actually seen the convergence algorithm of the simple linear regression. In this video, we are going to talk about multiple linear regression. And we're going to

just understand what is the basic difference between simple linear regression and multiple linear regression. In our previous example, uh, when I consider our data set right, we have taken the example that my input feature is probably, you know, weight. So I really need to predict height okay. And for this you know this is specifically one input feature over here one input or independent feature. And this is specifically my output feature. So what we did with the help of simple linear regression is that we tried to find out the best fit line and we understood the equation, the cost functions, the convergence algorithm of updating the slope. Now in this particular case, since I just have one input feature. So my if I talk about the uh h of x, which is the equation of the straight line which we have actually discussed, we are indicating something like theta zero plus theta one multiplied by x right now over here this x basically indicates my input feature right input or independent feature. Independent feature. Right. And obviously, you know what is theta zero. Theta zero is basically intercept. And theta one is nothing but slope. Right now, what we were doing is over here, is that we were trying to change theta zero. And theta one will not change x x. Whatever is there in this training data set that only thinks we are going to take and probably train our model, right. So in this particular case what we will do will just change theta zero and theta one and will try to find out the best fit line. Now what if in your data set you have multiple input feature. So let's consider one simple data set. I call this particular data set which is called as house pricing data set. Okay. Now in this house pricing data set. And probably I'll show you this all kind of data set when we do the practical right. Let's say I have multiple features like this number of rooms in the house. Size of the house. Okay. And let's say, what is the location of the house? And finally I have something called as price of the house. So over here in this particular problem statement here, you can see that I have four four features over here, four features in total. And this first three features are basically my input features. Right. This first three features are basically my input features or independent features. So I'm just going to write this as independent features. Now in this particular case you can see that I have more than one input feature independent feature. And this is my output feature. Now in this particular case when we try to find out a best fit line, how does the equation of the best fit line look like. So now I'm going to basically use the same equation x t of x which is nothing but theta zero. Theta zero is always an intercept. They will always be one intercept for any kind of regression problem. Or if I'm basically using a simple linear regression or multiple linear regression, they'll always be one intercept that we really need to use it. Along with this, based on the number of features right. Let's consider number of rooms is my x one feature. So for this we will be having another slope which we which I'm mentioning over here as theta one. Then I will be having for my second feature theta two. So theta two is the slope for size of house or coefficient of the size of house okay. And similarly location will be theta 3X3X3 basically indicates location right. And theta three is basically the coefficient. So this equation basically indicates my multiple linear regression okay. So in the case of multiple linear regression, one important thing that you really need to understand is that we will definitely have many features okay. Here in this particular case I've taken three features. So with respect to this three features I will be having three coefficients. So over here theta one theta two. And theta three are basically my three slopes or coefficients. Okay. So as the number of input features increases, that many number of coefficient will increase. But your theta zero will always be the intercept and you will have only one intercept all the time. Now, in this particular case, if I probably talk about a gradient descent or gradient descent, uh, you know, let's say, uh, since there are three input features, you know, in this particular case, when I draw with respect to theta theta zero, not theta zero. So let's let's say theta zero is here. Then I have theta one over here. And probably we cannot draw 4D diagram. So let's consider just two features here. I'm just considering theta zero and theta one okay. And with respect to this suppose I have J of theta over here right. So we

usually get a different types of diagram which will be looking like a inverted mountain okay. So so let's say that this is my gradient descent curve okay. And here you will be seeing something like this okay. And our aim is to basically come over here. So theta zero can start from this point. Theta one can start from this point. And let's say if you are using theta two or theta three it can probably start from this point. But everybody's aim is basically to merge or converge to this global minima. Okay. So this is how we basically solve multiple linear regression. Obviously I cannot, uh, show you some kind of simulation and all. Let me see. Uh, you know, if I get any simulation tool in this particular course to put up. But just understand that in this particular case, we have many, many coefficients and we have to keep on changing theta zero, theta one, theta two, theta three, you know, and try to reduce our cost function j of theta. Now over here I can basically write j of theta as something like this. So I can have j of theta zero theta one. Now in this particular case since I've taken theta zero and theta one. So I'm just going to mention it like this okay. So this is what multiple linear regression is all about. In multiple linear regression we are focusing on right. Like our input features will be more not just only one. If we just have one input feature, we basically say it as simple linear regression. If we have more than one input feature, we basically say multiple linear regression. Okay. So this was the basic difference. I hope you have understood this. Uh, in the upcoming videos, we are also going to understand some amazing assumptions about linear regression. And this assumptions are very important to understand. So let's go towards the next video. Thank you.

In our previous video, we have understood about the mathematical intuition behind linear regression and multiple linear regression. Now we are going to understand some of the performance metrics that is used to determine whether the model is good or not for a specific problem statement. And the metrics that will basically be used is something called as r squared and adjusted r squared. Now let us understand what exactly r squared is. And you know what is adjusted r square. And what is the exact difference between both of them. Now first of all let's go ahead and understand about R square. And R square is basically given by the formula which looks something like this. So this is r squared is equal to one minus sum of square residual. I'll talk about what is this residual and all. And this is sum of square total okay. So here a simple definition that you can probably see is that r squared is nothing but one minus ss residual. That is sum of squared residual divided by sum of squared total okay. Now let's understand what is the sum of squared residual. Suppose if I say this is my problem statement let's say this is my x axis y axis okay. And I have some other data points which looks like this okay. Which looks like this. Okay, I'm just drawing some data points over here. Our main aim is that we need to find out the best fit line. So if you are trying to find out the best fit line with respect to this orange points, these are my true output, right? I specifically say all these points belongs to y of I right y of I are basically my true outputs, right? Now similarly, with respect to this particular point, the predicted point is nothing but this part. Right? So this I can basically say it is y of I hat right y of I hat y of I hat are my predicted points right. So this is my predicted point. This is my predicted point. This is my predicted point. This is my predicted point. This is my predicted point. So I'm just drawing all my predicted points. Now whenever I try to find out the difference between them, then in short, I am basically saying that as this is basically my residual. Okay, residual or error. So if I probably try to find out this particular distance, this particular distance, this distance, this distance, this distance and this distance, this is basically the residual okay. Now

understand the definition over here. I am basically writing one minus sum of square residual. So I can basically write this as with this notation that is summation of y of I minus y of I hat whole square right. So in short, what I'm doing, I'm basically trying to find out the difference between y of I and y of I hat. Okay. And I'm doing all the summation of all this particular residual or error. I can also say this as errors. Okay. Now this will be divided by sum of square total. Now what is this sum of square total? Now with respect to this particular point I hope everybody found out okay, this is my best fit line. But with respect to the y axis and with respect to all these y points, if I try to calculate the average okay. So if I try to calculate the average then you may probably seeing that y of I average will come with respect to this specific point. Okay. Or let's say I'm just trying to calculate all the y of I points. I'm just taking all the y of I points, and I'm just calculating the average nine with respect to this kind of distribution. Let's let's assume that my y of I is basically getting passed over here. This y of I hat I will say okay. So this basically indicates this is nothing but average of the true points. That is y of I okay, average of all the y of y. Now this sum of square total basically means if I really want to give it in a mathematical notation, this is nothing but summation of y of I minus y hat y of I hat whole square. So this basically means that we are trying to find out the difference between the true point and this specific point, which is basically indicated by y of I hat y of I hat is nothing, but it is the average point. Okay, so from here to here I'll calculate a difference. Then from here to here I'll calculate the difference. Similarly here to here I'll calculate the difference here to here I'll calculate the difference here to here I'll calculate the difference here to here and here to here okay. So this orange line that you see this is basically the mean of all the y of I points okay. So this basically will be giving my r squared value okay. Now here you can clearly see that obviously just by uh simple understanding. Right. You can definitely see that y of I minus y of I hat whole square this residual this this difference of the residual. Because my best fit line passes through almost near to all the true points. Right. So this particular difference will be smaller number when compared to this particular difference. Right. Because this is with respect to the average this specific line. And obviously the distance will be quite big. So here I can definitely write something like this. I will write that it is y minus. Let's consider this is a small number when compared to the denominator right. So the denominator will be a bigger number right. Small number divided by bigger number. So now when we do this kind of division obviously we are going to get a smaller number itself a smaller number itself. Right. So let's consider if this numerator is a smaller number. And denominator is a bigger number. Obviously when we try to find out the division of this two, obviously we are going to get a smaller number. So one minus smaller number will be, you know, very much near to one itself. Right. It will be approximately near to one. So let's consider that. Suppose if I'm getting .70 right. So this basically indicates that I am actually having 70% accuracy. Right. Suppose if I get this specific difference is somewhere around .85. This basically indicates that I'm getting a 85% accuracy. Suppose if I say it is .90, I'm actually getting a 90% accuracy, right? So with the help of R squared, we are able to find out how our model is performing. Right. And more this value understand more this value. It is towards one right more accurate. My model is more accurate. My model is. More accurate. My model is my linear regression model is okay, but there is something called as overfitting okay, which we will probably discuss as we go ahead. Uh, we are going to discuss about overfitting and underfitting just in some time. We'll discuss about it, how we need to identify whether the model is basically getting overfitting or not. For that, we really need to understand what is train data test data, validation data. Just give me some time. We'll discuss about that as we go ahead. Now let's go ahead and understand about the second performance metrics, which is called as adjusted R square. Now with respect to adjusted R square okay. How it is basically different that also we are going to discuss with respect to the R square. So here I'm just going to write my adjusted R square okay R square.

Now, guys, if I consider this specific example, let's say I have in my data set, okay, I have in my data set one feature which is called as let's say one of the feature is like size of the house. Okay. Size of the house. Suppose this is my data set. I need to really predict what is the price of the house. Okay, so I need to predict this specific thing. Let's say I have one of the feature which is called as size of the house. So I'm definitely going to write price over here. Now suppose if I have size of the house and the price of the house, as we know that as the size of the house, as the size of the house increases. Increases. My price is also going to increase. So there is a direct correlation between size of the house and price of the house. And this is usually happens in any any state or any region that you're probably seeing. Staying right the size of the house as it is increases, the price also increases. So and similarly, when the size of the house decreases, the price also decreases. So here we can definitely find something called as a positive correlation right. Positive correlation a good positive correlation. So when we have in this particular case a good positive correlation uh then what happens. My r squared value probably increases. Right. Let's say this value basically increases. Now let's say in this particular case when I'm using this two features, my uh, and let's say that I have actually computed my r squared, my r squared value that was coming was somewhere around, uh, you know, 75%, let's say 75% is nothing but 0.75 with the help of this same formula. Okay. Let's say with the help of this same formula that we have used over here, we got 0.75. Now what happened is that let's say I included one more feature after the size of the house. I said number of bedrooms. Now you know that as the number of bedrooms increases, right. The price is also going to increase. So from this I can definitely say as my number of bedroom. And this is just a domain knowledge, right. When the number of bedrooms increases my price also is going to increase. So here also I can definitely see something called as positive correlation. Right. So when I'm actually seeing this positive correlation now if I go and probably find out my r squared value again after I create my model and probably plot or get my best fit line and then apply this entire formula. Okay, I am definitely going to see some increase. Let's say here, I'm just assuming that I got 80% accuracy. So this is nothing but .80. This is also perfectly fine. Now similarly, when I try to add one more feature with let's say this particular feature is location. And we know that with respect to location also, it is more positively correlated with respect to price. Okay, this two are more positively correlated. So there are again chances that my r squared value will definitely increase. Okay. My r square value will become 85%. Let's say I'm just assuming it has now increased to 85%. Now let's say that okay this many features I have okay. This many perfect so many independent features I have and uh, and this is my all my independent features. And this is my dependent feature right now let's say that I have added one more feature. Now, this specific feature basically says that which gender are going to stay in this house. Okay. Whether they're I'm just adding one feature. Okay. That does not make any sense okay. And obviously, you know that if I consider gender with price, obviously just by common sense, there will be no correlation with respect to price, right? Anyhow, no correlation. Whoever gender stays over there, obviously price is not going to get affected. Right. But if we add this particular feature, which is not at all correlated with price, and again, if we try to probably solve this problem, which again in the future, I'll be showing you in with the help of practical example, if I try to see the r squared value, even though this two variables, this input feature and this output feature are not at all correlated, then also, since we are applying this particular r squared formula, there will be chances that this particular value will still increase, but it will not increase by a bigger value. So here you can see that I'm actually getting somewhere around 87%. Let's say okay, I'm just assuming two more percent. It got increased because the formula it works in that specific way of R squared is that as more number of features you keep on adding, somehow the r square value is going to increase. Okay. And this all things I'll show you in practical also guys. So let's say this becomes 0.87. But now there is

some kind of problem. Even though this particular feature is not at all correlated with price here you can definitely see that my R square value is increasing. Okay, I do not prove it practically yet, because I'll do the practical and show you, uh, with respect to this particular problem statement also. Right. So in this case the R square, you can see that even though this and this feature are not at all correlated, still it is basically increasing. Right. So this is the problem with respect to R square right. So this is the problem with respect to r square problem of r square. Even though you don't have a feature that is directly correlated with the output feature, then also this particular value increases since uh the formula is in that specific way. We are just trying to find out the difference between residual and, uh, difference between the mean of which we basically say it as SS of total. Now, in order to prevent this, we basically use something called as adjusted R square. Now in adjusted R square, what it does is that it penalizes with respect to every feature that are not correlated with the output feature. Okay. So this is what amazing thing basically happens with the adjusted R square. Now in order to define. The adjusted R square. We basically use a very simple formula. So I'm basically using R square or the adjusted R square. So I'll let me write down the formula over here. And again you do not know how the formula is getting derived. Guys that is not at all important. But instead we should directly use this kind of performance metrics. So it is nothing but one minus one minus R square. And I'll talk about each and every term multiplied by n minus one divided by n minus p minus one. Okay. Now let me talk about what is this and what is this P. And all right. So n is nothing but. We basically see a number of data points. Okay. Number of data points. That basically means in our data set how many data points are actually there. Okay. And P basically indicates number of. Independent features. Number of independent features. That we are using now, guys, whatever formula that we have written over here with respect to the adjusted R square. One amazing thing that you will be finding in this and later on when we do the practical implementation, that thing will be clearly visible to you. So let me just note it down over here and in the later stages. In the practical part, I will try to show it to you. Now n basically is number of data points. P basically means number of independent feature. Let's say initially my p value is something like two. So this basically indicates that let's say my number of independent features are two. In this case let's say my R square is 90%. I'm just assuming I'm just giving you a hypothetical scenario. Right. If my R square value is 90, then if I try to apply the same formula with respect to R square adjusted, then here you will be seeing that I will be getting less than this particular r square value. So here let's say I'm getting 86% okay. Now as soon as I increase p value then obviously you know that we have seen that the R square value will increase. Okay. Let's say the r square value has got 92%. And let's say that additional one feature that we have added, the independent feature is not at all correlated, okay. Not at all correlated, or it does not have a direct impact on that specific output feature. At that point of time, you will be seeing that the R square adjusted value will get reduced. It will be somewhere around 8283. It will be less than the previous one. That is less than the 86% here. I'm just giving you a hypothetical number. Okay. This kind of scenarios, you will be able to see with the help of adjusted R square. Suppose if the additional independent feature that I have added is directly proportional to the output feature, one thing that you will be able to find out that it will increase and it will be greater than this 86%. If it is not at all correlated, then it is going to decrease. This is the assumptions I'm making it right now. Okay, later on, when I try to show you with respect to the practical, uh, you know, you will be able to find it out. You know, I may, I guess, like you may be thinking, okay, how this, this specific thing happen. Just understand, guys, when I am probably increasing the p value over here with respect to the denominator and when I start, uh, multiply or divide with this specific value, this kind of relationship will automatically be formed. So whatever problem that we had in the R square will basically get solved with the adjusted R square. Okay. So for right

now, just, uh, make this hypothetical note, uh, that I've actually made in front of you. Later on, I'll try to show you with the help of practicals also. So this was about the performance metrics. Most common performance metrics that we use is something called as R square and adjusted R square. And the same thing I will show you in the practical. And when I'm showing you in the practical, there are also some kind of assumptions that we make to make like over here I spoken about correlation. Right. Uh, and here also I showed you about like how r square adjusted is basically getting decreased. This is the formula that gets applied instead of this. So because of that that specific value is getting decreased okay. So let's continue uh the video. And in the next session we will discuss about more practical implementation. Thank you.

In this video we are going to discuss about mean squared error, mean absolute error and root mean squared error. Already in our previous video we have seen about R square and adjusted R square. Now you know that we use R square and adjusted R square to check how the model performance is. But with respect to each and every data points, if you really need to focus more on the error, we can definitely use mean squared error, mean absolute error and root mean squared error. But again I will be talking the advantages by using mean squared error, mean absolute error and root mean squared error. Remember advantages and disadvantages both I'm going to cover. Remember in the intuition that we have actually explained right. So let's consider that I have a data set. And this particular data set is basically let's say in the x axis I have years of experience. And in the y axis I basically have salary right now in this specific data set. Let's say I have some other data points over here. And let's say here I create my best fit line. Right. And as you remember, you know, in in in in the linear regression, I have also intuitively explained you about a concept of gradient descent and this gradient descent. The main aim is and this was my cost function theta zero comma theta one. And this was my uh theta zero coefficients in short. Our main aim was to basically come to this particular global minima. So this is basically my global minima. And I have to come over here. Right. And how I like to come have to continuously change this theta value, find out the derivative and try to come over here right now. Let's go ahead and see that if I have this specific use case over here, you know that let's say this is my data set experience and salary. Right? I will obviously be having some values right now over here. Now this salary is nothing but my y variable. That is my truth variable which is denoted by this white color and this best fit line which is basically created. This will actually help us to find out the predicted data points. Right. So here with respect to this particular data point, this is my predicted data point with respect to this. This is my predicted data point. It is this is my predicted data point and this is my predicted data point. So what we do we basically calculate the error like this. And our main aim is to reduce this specific error. And in order to reduce this specific error we use different different cost function. The first cost function that we used is basically called as mean squared error. Now let me go ahead and write this formula. Mean squared error is nothing but summation of I is equal to one to n y minus y hat y hat is the predicted output whole square divided by n, right, n is the total number of data points, and this is the same cost function that we used to basically create this gradient descent. Now what is so important about this cost function. And our main aim is to reduce this cost function. Because this is specifically talking about the error. And our best fit line should be created in such a way

that our cost function should keep on reducing. Okay. So we can also use this as a parameter to check how our model is basically performing. Okay. Now in mean squared error, here we are just trying to calculate the error right. So y y y minus y hat. So we are going to add up this particular error. Let's say I'm mentioning it as e1 e2 e3 e4. Right. I'll add up all this particular error. Now let's talk about the advantages of using MSE and disadvantages of using MSE. Now first of all, from our previous intuition, we have already seen that this y minus y hat whole square. It is basically a quadratic equation. So if you don't know what is a quadratic equation, I would suggest just open a browser. Let me just show you a quadratic equation how it looks like. So let's say over here I'm just writing it out. Plot. Quadratic equation. Okay, so when I plot this quadratic equation, if I probably go and see the images here, you'll be able to see we will be getting this kind of image right. And if you are getting this kind of image you obviously know that this is basically a gradient descent. That basically means we will be able to reach our global minima with the help of this quadratic equation. Now why I say uh. Now why I say this as a quadratic equation? Because understand this is in the format of a x plus b y plus c, right? Because if I take this and if I write a minus b whole square, it is nothing but a square minus 2AB plus b square. Right. So here it is basically a quadratic equation. And whenever you have a quadratic equation and if you plot this quadratic equation, in short you will be getting a gradient descent. And remember one important point in gradient descent that you will be having a global minima. And our aim is basically to come to this global minima or near to this global minima because here the error will be less right. So this is the major, major advantage. So let me go ahead and write the advantages of using RMSE. Sorry mean squared error. So first advantage is that uh if I probably write this as advantage let me go ahead. So advantage and disadvantage. The first. And the foremost advantage is that. And whenever we have this kind of gradient descent, one thing you need to know that it is differentiable at all point. And we also saw this right. It is differentiable at all point. So first advantage is that it is differentiable. Why it is differentiable? Because it forms this kind of gradient descent curve. So at every point we will be able to calculate the slope. And based on that we can reduce the we can reduce the. Coefficient and probably reach the global minima. So most a major advantage is that it is differentiable coming to. The next advantage is that over here this kind of curve it has. One local and one global minima. This is super important point which everybody needs to focus on, right? Let's say this particular gradient descent that I see is basically a convex function. Okay. If you see some of the curves, let's say that I have one curve like this. And this gradient descent curve looks something like this. Okay, now over here you can see that this is not a convex function. Obviously here you have one global minima. And you'll also be seeing that you'll be having some more minima. And this minima is basically called as local minima. If we use different types of equation in those equation, you'll be seeing that you'll be able to get. Or if you plot this equations, you'll be there'll be chances that you'll be getting both global minima and local minima. But since you're using this quadratic equation, you're plotting it. You will just be having one local or one global minima, right? Not multiple. Because what will happen if you have some local minima then your convergence will get stuck over here, right? Your convergence will get stuck over here. Because why here the slope is actually equal to zero. This is a super important point to understand. What is the differences between local minima and global minima? Okay. If you have a local minima, that basically means your convergence. Let's say your convergence is happening like this. It will go ahead and it will stop over here because here your slope is zero, right. But in the real you really need to reach to the global minima. Right. So if you use any different kind of equation specifically not quadratic equation, there are also some equation which is basically called as non convex equations or non convex functions. In this particular case you will be seeing that they will be having both local minima and global minima. But currently we are using this

quadratic equation that is y minus y hat whole square. In this particular case since it is differentiable, it will just be having one local minima or one global minima. Now coming to the third point, which is super important, this equation or this curve, let's say if I use this particular equation it converges faster. Right. And we have already seen how the convergence will basically happen. Right. So these are the three main advantages with respect to uh, by using this uh mean squared error. Right now coming to the disadvantage, which is super important. Again okay. Now in disadvantage, you will be seeing that the first and the foremost disadvantage of using this is that this equation is not robust. It is not robust to outliers. Now why I'm saying like this. It is not robust to outliers. Let's say I have a I have a. I have outlier like this right now. What will happen if I have an outlier? Obviously here, the error that you are trying to compute later on you will be squaring it. Right. Because y minus y hat whole square. Right. So here you're penalizing this. This process is basically called as penalizing the outlier. Penalizing the outlier. Now because of this, what will happen? Error will increase. Now let's consider that I have some data points which looks like this. Initially my data points was like this, let's say. So based on this I created a best fit line. But later on let's say that I included a new data point, which is a kind of outlier like this. Now what will happen? Your this line, right. As soon as it sees an outlier, obviously it will go and compute the error. Now because of this particular error, obviously this error will lead to the increase in the MSE. Right? The MSE will get increased. Now when MSE is getting increased we really need to optimize this MSE. Then what will happen is that because of this error, this line will move towards that particular outlier like this. But this is absolutely wrong, right. Because most of your data points are over here. And because of this outlier, you can see that there is a huge shift off the best fit line from here to there. Right? So this is what is an impact of an outlier. So this is not robust. There is definitely because of the outlier. The line is moving from here to there by a huge difference right. Because obviously we need to reduce the MSE. But currently the MSE is increasing in this particular case right. So we definitely say that it is not robust to outliers okay. So this is the major disadvantage with respect to using MSE. Now coming to the next point with respect to MSE okay here when I try to find out the difference between y y y of I and y hat whole square, you'll be able to see that whatever MSE that I get right, this is basically one by n summation. Or let me just use this particular equation again you know that. What is MSC? MSC is nothing but uh, one by n summation of I is equal to one to n y of I minus y of I hat whole square. Right. So this is the equation over here. Right now in this specific equation. What is happening is that you're trying to find out the differences between the truth and the predicted value, right over here. Just focus on this one, okay. Here you will be able to see that y of I minus y hat whole square. When you're doing whole square just focus on the units. Let's say my y of I which is which is the salary variable. Okay. Let's say my x axis is basically experience y axis is salary. And if I do y of I minus y of I hat whole square. The salary will be in some units. Let's say this is in units like lakhs. Now. Right now y of I is basically in lakhs. But if I do y of I minus y of I hat whole square that. Basically I'm changing the units. Now this is basically becoming lakhs whole square. So at the end of the day, you know what is happening with respect to MSE. Is that the y of I and y of I hat whenever we are trying to compute the MSE, you know the units is changing when compared to the output variable. So this is also a major disadvantage. So it is no longer in the same unit. Longer and same unit. Now, why I'm putting up this point, because it basically becomes very difficult in terms of, uh, understanding that specific information, because let's say if my error is coming as let's say it is coming as 2.5 lakhs, right? Or whatever error that I'm getting with respect to y of I minus y of I hat whole square is let's say, let's say that it is 2.5. Now here, you know that this particular unit that it is specified it is with lacks square. So it becomes difficult in understanding when you're comparing it with the truth value. Okay. So this is the major major advantage the major major disadvantage. So again

let me revise with respect to advantage and disadvantage. First the advantage is that obviously whenever you use MSE it is going to create this kind of curve. Because y minus y hat whole square is nothing, but it is a quadratic equation. And whenever you have a quadratic equation, at the end of the day you are going to get this kind of curve. Uh, so the major advantage with respect to this curve, which is called as gradient descent curve, it is first of all differentiable. That which I have specified, it has one local and one global minima. It does not have multiple local or Ma or more than one global minima like like this kind of situation. So obviously we know that where we really need to reach that is over here. And when we reach near this, the error will be less. The third thing is that it converges faster because we don't have any local minima over here. The disadvantage is that it is not robust to outliers, and it is not it is not longer in the same way it it that basically it is not in the same unit. Okay. I'm just going to mention this again. It is not in the same unit okay. So this is about MSE. Now let's go ahead and discuss about mean squared error. Uh sorry mean absolute error. So here uh I've discussed about MSE. The second one that I'm going to discuss is about mean. Absolute error. Now mean absolute error. What happens if we take this cost function or loss function when we are creating our gradient descent, right. So mean absolute error is basically given by the formula one by n summation of I is equal to one to n, and this is y minus y hat whole, not whole square. It will be the absolute value of y minus y hat. Now here let's talk about the advantages and disadvantages. First of all let's go ahead towards the advantage. And let's go towards the disadvantage. Now I think just by seeing the equation you'll be able to understand the advantage. First of all this will definitely be robust to outliers. Now why it is robust to outliers because let's consider these are my data points. And let's say I have some other data points like this. And let's say I have created a best fit line. Let's say after again while training my data points, you know, I get an outlier, which is over here. Now, what will happen is that since my formula is not squaring or not penalizing this specific error, you know that m a e will get increased, but it will not get increased like how MSE used to get increased, right? Error. We are trying to square it over there. But here the error. What it is coming, it will just increase me by some amount. Okay, so because of this, what will happen? My line will not. My best fit line will not divert that much, but it will just divert in a simpler way like this. Okay, so this movement is just this much. Whereas in the case of uh, MSE, you could see there is a huge movement in the best fit line. Okay. And obviously we need to, uh, reduce this MSE also. That is also the main major thing. Okay. The second advantage is that when I'm trying to calculate y minus y hat, it will be in the same unit, right. It will be in the same unit. So. Same unit. So understanding the data over here will be simpler because your y hat and y minus y hat will be in the same units. That is like lacs. If I compare with respect to salary it is not becoming lacs square. Right? So this is the second advantage. And obviously because of this only this thing is getting there. Right. But coming to the disadvantage which is super important point okay. Obviously uh means mean absolute error is used to uh, you know, basically, uh, because of the outlier that is present in mean squared error, because of the impact of the outlier that is present in the mean squared error, we are trying to solve it with the help of mean absolute error. But here if I talk about differentiability, right. In the case of mean absolute error, if I use this equation, usually what happens I will get this kind of curve. Now this curve you will be seeing. It looks like gradient descent. No. Understand guys, here you are passing through zero here. At this point we cannot differentiate any value at zero. We cannot differentiate any value at zero. Right. So what we do is that we create something called as sub gradients. Okay sub gradients. Now sub gradient basically means what I will be taking a part off this part of this. And then probably finding out the, uh, slope and then trying to compute and trying to come to a global minima point. Okay. So this is a major disadvantage with respect to this. The major disadvantage is that convergence usually take more time, usually takes more

time. Take more time because y y y y. The convergence will take more time because. Understand the optimization. The optimization is a complex task. Now, what is the optimization over here that we really need to come to a global minima. But right now here you basically have this kind of gradient descent. And you know that at this point of time, if you probably go at this at this particular point, that basically means it is not at all differentiable at zero, and that is what derivative derivative will not happen at the point of zero, right? You cannot differentiate anything that is at zero. Okay. So what you do is that you try to divide that into sub gradient and then you try to calculate the slope okay. Then only you will be able to calculate the slope. Now. Optimization is a complex task over here. Definitely. Uh, optimization is not that specific. Easy, but one very important thing that you will be able to see that, uh, you know, uh, in this it is definitely robust to outliers. Right? This is the major task now here. It will definitely be time consuming. Okay. So these are some of the advantages and disadvantages with respect to Ma. Ma that is mean absolute error. Now let's go ahead with respect to the third one which is called as. Later. So here I'm just going to write it as root mean squared error. Now here we can definitely write our MSC as root of MSC. So if I say root of MSC what it is it is nothing but root of one. By n summation of I is equal to one to n y of I minus y of I hat whole square. Okay, so this is the entire formula with respect to our MSC. Again uh, in this uh, what will be the benefit? One benefit that I can definitely compare, uh, since I'm doing the square root. Right. So initially in MSC it was not in the same unit. Right. So first advantage that I would like to call about is that. First advantage here. The values that you are getting is in the same unit. Second thing, this is obviously differentiable. Okay, since here also you will be anyhow getting a gradient descent. But the major disadvantage if I try to talk about is that since we are again doing squaring here, it is not robust to outliers. Okay, not robust to outliers. Now we have learned about this. All three techniques at the end of the day. You may be considering Krish. Which one should we use? Should we use Ma, MSE or RMSE? My answer will be that guys, as a starter we should try to look on all three of them, right? Our MSE, all three of them. So whenever you'll also be seeing that in further. Whenever I try to create any linear regression model I will also be showing you the errors that is going to come because see, once we probably create a best fit line. Obviously let's say if there are some data points. First of all, I will try to create a best fit line. Then what I'll do, I will try to calculate the MSE, the mean squared error, the mean absolute error and MSE. And after this, I will also try to check the performance of this model with the help of R square and adjusted R square. So as a starter, you should try to focus on all these things and try to see the performance perm, uh, performance uh metrics that you are able to get. But all this in combined totally will basically become my performance metrics that I am specifically going to use. Okay. Similarly, in classification problem you have different performance metrics. But as mentioned over here you have to focus both on you have to focus on all the metrics like performance uh, mean squared error, mean absolute error or MSE, r square and adjusted R square. But if someone asks you in an interview question, why should you use MSE when compared to Ma, then you know that you have to tell the answer that it is very differentiable because the curve that you are going to get because of this quadratic equation, uh, it has one and local and one global minima, it converges faster. Suppose if they say that, what about Ma? Then you should say that Ma. Most advantage thing is that it is robust to outliers. Right. And the convergence usually take more time. This is the disadvantage. Right. And then why should we use our MSE. So this is the specific question. So the question that may come is that when should you use MSE versus when should you use Ma versus when should you MSE. And the other question is that when should you use r square versus adjusted R square. So these are the two important questions that may come up in the interview. Right. So I hope you have

understood all the performance metrics till now. Uh in this video we have covered MSE, Ma and MSE. So yes, uh, I will see you all in the next video.

In this video, we are going to understand about some of the terminologies that are basically used when you are training your model practically right with the help of Python and sklearn, which we are going to use probably after a couple of sessions there, you'll be seeing this kind of words like training data set. What is test data set. What is validation data set. When do you say when your model is overfitting or underfitting? And with respect to that what is bias and variance. So in this video I'm going to cover all the specific topics okay. Let's consider uh I have a data set okay. Let's say that okay I basically am starting my training. So what I'm going to do is that I will probably have my data set right. Let's say in this particular data set I have somewhere around 1000 data points, okay, 1000 data points, irrespective of how many number of features are there. Right. And initially, let's say this is my entire data set that I have. And I really need to train some model in this. Okay. Train some model. And this model will be able to predict something. Let's say this is house pricing data set. So I will be having features like size of house okay. Price uh size of house, number of bedrooms. Okay. And, uh, one more feature will be price of the house. Let's say I have this many data points. 1000 data points with respect to our data right now, uh, this specific data points initially, whenever you start training your model, the first thing that you really need to do is that try to split this data into two. Important part one is the training data set. One is the something called as training data set, and the other one is something called as. Test data set. Okay, so what we do over here is that we are trying to split this entire data points between two parts. One is training data set and one is test data set. And let's say that I have taken some ratio like 70 to 30% ratio 70% to 30% ratio. That basically means my 700 records are over here and my 300 records are over here. Now why we are specifically doing this specific step. You know why we are dividing our data into the training data set and test data set. And from this also we will be able to understand whether our model is overfitting or not. Okay. So this training data set I will probably use this data for my model to train. Okay. And this test data set what I will be doing I'll be using this to test my model. Okay. So for testing the model I will use this specific data. And this data. Once I do the splitting I'm never going to show this data to my model. Okay. While training the my model will never know about this specific data points. Okay now let's take up this training data set. And what is the next step that we do in our training data set. We do further splitting of this. Now what kind of splitting we do. We basically divide this into again two parts okay. The first part is something called as train. And the second part is something called as validation. Now there is a minute difference between train and validation okay. Training train. Train part is basically used to train the model. Okay, train the model and in the future, whatever practical session we'll be doing will be following the same steps so that this will give you an idea that how we can identify whether our model is overfitted or not. Similarly, with respect to the validation, you know what we why do we use this validation data? We use this for hyperparameter tuning our model okay. Hyper parameter tuning your model. So if I really want to tune my model I will basically be using this validation data set. So in short I will combine train and validation data in order to make our model performance better. Now what kind of when do we face overfitting? When do we face underfitting that I'm trying to

discuss over here. Okay, now let me go away and let me give some of the scenarios. Obviously you have seen that I have my train data and my test data. Let's say my model gives very good accuracy. Very good accuracy. To the training data. Okay. Let's say this is my model that I'm trying to create. It gives a very good accuracy with respect to the training data and very good accuracy with respect to the test data, because later on, when we, uh, take this test data and predict it, we can basically compare whether it is giving us very good accuracy or not. And this is the model that I specifically want. And this model is basically called as generalized model because it is working well with both train and test data. So this is the generalized model that I specifically want. But let's say that I have a scenario wherein uh, wherein with respect to the training data, I get a very good accuracy. But with respect to the test data, the new test data, I get a bad accuracy. Bad accuracy over here. Let's say that the very good accuracy is somewhere around. I get somewhere around 90% when the. In case of bad accuracy, let's say I'm getting somewhere on 50%. Now in this particular scenario, since our model is giving a very good accuracy on the train data and bad accuracy with respect to the new on the test data, we say this condition as our model is overfitting. Our model is overfitting. So when does when do we say our model is overfitting? When my model has been trained well with the training data set, but with respect to the new data points, it is not able to predict. So because of that I'm getting a bad accuracy. Okay. So this is a scenario of model is overfitting. Now whenever I get a good accuracy we basically say or use a term which is called as low bias. Okay. And since I get a bad accuracy with respect to my test data, I usually use this term, which is called as high variance. Okay. High variance. So this is one terminology that we specifically use. Now let's consider one more scenario okay. Now in this scenario let's say with respect to my training data and with respect to my test data okay. Let's say here my model accuracy is low. With respect to the training data. My model accuracy is low, and obviously if my training accuracy is also low, then this model accuracy for the test data will also be low. Right now in this scenario, this particular problem statement, I can definitely say my model is overfitting. Sorry, my model is underfitting. Under fitting. So in this basically way we have, we are just trying to convey that my model is not even trained well with the training data. So if it is not well trained with the training data, obviously it is going not going to perform with respect to the test data. So this scenario, we say that my model is underfitting. So in this scenario I basically use this as high bias. And high variance. Okay. We usually use this notation. Okay. And coming to the first situation where I get very good accuracy with respect to the training data and very good accuracy with respect to the test data. I usually say this as. With a simple name low bias. And low variance. And always our aim should be that we should try to always find out this specific generalized model where my training and test accuracy should be good. Let's say if I get my training, accuracy is 90%, and if I get my test accuracy is 85%, then I can definitely say that, okay, I'm getting almost similar kind of accuracy with respect to my train and test data itself. Right? So this is the basic difference between overfitting and underfitting and this entire thing when I'm using bias and variance here I am specifically saying that okay, this is this is also generally called as a bias variance trade off. Okay. Now, uh, if I talk with respect to regression, right. If I try to just draw this points. Okay. Let's say, uh, this is my, um, this this all points that you probably see are my training data points. Okay. Now, if I try to find out a best fit line. Okay. Now, with respect to this particular best fit line, let's say my new test data is somewhere here. Then obviously you can see that I will be getting almost similar accuracy. So this best fit line will serve or will provide you this specific condition. It will provide you a kind of generalized model. Okay. So this this best fit line that I have actually formed, this will basically be giving me good accuracy for the training data. Also for the test data also. So this is the generalized model that I specifically want. But if I have some other scenario wherein let's say this is my model okay. And let's say, you know, I have uh, the new test data,

which looks something like this, let's say over here, over here, over here. And suppose I probably try to create a best fit line. So this is my best fit line. And let's say my new data points is somewhere here. Right. So in this particular case, I know that C with respect to this, obviously for the training data set, the accuracy will be high because it has properly fitted the specific model. Right. But with respect to the test data, it is not being able to provide a good solution because obviously the error is very, very high with respect to the test data. Now in this particular case this will actually lead to overfitting. Okay. So one assignment that I really want to give you is that just try to find out how underfitting it will be. Right. Obviously for the train and the test the error will be quite high. So whenever I have a generalized model I can definitely say this as low bias and low variance. And with respect to the overfitting, I can definitely say this is low bias and high variance. Okay, so this is the scenario with respect to overfitting and generalized model. And obviously you can guess about the underfitting part. So this specific thing you really need to understand because as we go ahead all the practical part will be implemented in this specific way. And uh, you know, we'll try to use the R square and adjusted R square in order to find out whether it is overfitting or underfitting. So, yes, I hope you liked it. Uh, we'll continue the practical part in the next session. Thank you.

Hello guys. In this video we are going to see how we can implement linear regression using OLS. That is ordinary least square. Okay, now in our previous video we have already seen how to implement linear regression using gradient descent. But now let's go ahead and try to see with the help of OLS. Now the major aim is that guys over here you knew that like in the gradient descent, right? Whenever I have some kind of data points, my best thing my my main aim using linear regression, using gradient descent is that I need to find out this best fit line. And finding out this best fit line was done using an optimized algorithm, right? Some optimization was actually done right. And uh, this optimization usually happened with the help of loss function we have basically used. And for that, if you remember we use something called as a gradient descent. Right. And at every point we try to calculate the derivative. And based on that we try to come near to the global minima. So this particular point was something like global minima. Now what does OLS? What does OLS technique basically say? OLS technique basically say that. Suppose if I have a specific data points, can I apply some formula and calculate. My beta zero. That is my intercept and my coefficient that is beta one. So that is what ordinary least square basically tells us. Okay. Now we will try to see, with the help of some formula how we can basically calculate beta zero and beta one. But before I go ahead and directly tell you the formula, first of all, we'll try to understand or we'll try to derive this formula also by understanding the aim of OLS again as usual. The aim of ordinary least square is that we should try to reduce this error, right? We should try to reduce this error that is between the predicted and the truth point. Right. So this is basically the error with respect to any best fit line that we create. Right. So here what you can do is that again let's say that if I have another line, if I have some line like this. So for this the intercept will be different. The coefficient will be different. For this the intercept will be different. The coefficient will be different. But we should try to select that line which gives us a less error. So now let's go ahead and let's try to derive the formula for OLS. And uh again guys uh OLS uh here I'm just going to give a notation. Let's say this is my

ordinary. Least square. Okay. You know what is the aim of ordinary least squares? I'm going to give you a notation with s. Let's say this is beta zero and beta one. Let's consider that this best fit line is given by s theta of x is equal to beta zero plus beta one into x one. I'm just considering this as a simple linear regression. And as said in the ordinary least square, I'm not going to use any loss function like mean squared error, but I'm just trying to say that, okay, I have a scenario wherein I need to reduce my difference between y of I and s theta of x. What is s of x? H theta of x is nothing but beta zero plus beta one multiplied by x one. And this is nothing but my predicted value y hat. So I'm just going to use this particular equation like this. And probably right over here this minus will get subtracted to this okay. So it will become minus beta zero plus ma. And it will be minus beta one into x one. And this should be whole square. So here what I'm doing is that I'm just trying to reduce the error okay. Uh, my my aim is that this difference that I'm actually getting, this is basically the error okay. And I'm just going to reduce this specific error. Uh, and obviously this this basically looks like a mean squared error formula itself. Right. So OLS by default I think it uses mean squared error. Okay. And uh, if I try to find out the average it will be divided by n okay. So one by n uh multiplied by this. So here let me write it by white pen only. Otherwise you'll get a get confused. So this will be one by n y of I minus beta zero plus beta one into x one. Now in ordinary least square, the aim is we need to calculate beta zero and beta one. And obviously we'll try to derive the formula through which beta zero and beta one can be calculated. So before I go ahead and uh, talk about this, you know, in gradient descent, also we try to find out derivative, right, with respect to every coefficient. And we try to come to the global minima. Okay. Now, what we are trying to do is that we are trying to find out what should be the formula of beta zero and beta one. In order to find it, first of all, what I will do is that I will try to calculate the derivative. So derivative of dz of d beta zero and let's say beta zero comma beta one is over here. Here I can specifically write this as okay. Now if I try to find out the derivative of this value what it will happen. See guys, if I have an x square value and if I have y is equal to x square. And if I write derivative of y with respect to derivative of x, then this what will happen? It will say 2X2 minus one multiplied by x. Right. Sorry. Multiplied by derivative of x. Right. This is what exactly happens. Right? And then finally I get two x because derivative of x with respect to x is one right. So if I try to if I have equation y is equal to x square. And if I try to find out the derivative of y with respect to x, I'm going to get two x. Now similarly over here you will be able to see that if I try to find out the derivative what will happen. This will be two by n okay. Summation of. One to N, and then I will write this entire equation y of I minus beta zero minus beta 1X1. Now here it will be two minus one right two minus one. So I'm not going to write one separately over here. And. In the next one. I'm just going to find out the derivative of this entire value, right with respect to beta zero. This becomes a constant. So obviously this will be zero minus beta zero is there. Then this will become minus one sorry one because derivative of beta zero with respect to beta zero is one. And then here you can see that this will again be a constant. So I'm going to write -0. So at all if I try to find out derivative of s with respect to derivative of beta zero with respect to beta zero comma beta one, I'm going to get two by n minus two by n summation of I is equal to one to n y of I minus beta zero minus beta 1X1. Right. This is what I'm actually going to get. So this I'll be assigning it to zero. So this will become my first equation. Let's say okay this will become my first equation because I need to derive beta zero and beta one. Similarly, if I write derivative of s with respect to derivative of beta one with beta zero comma beta one. Here the equation that I'm actually going to use again, what I'm going to do, I'm going to find out the derivative with respect to beta one. Right. Then again it will be what two by n okay. And this will be summation of I is equal to one to n. And here I will write y of I minus beta zero minus beta 1X1. Now this entire thing I have to find out the derivative. Now this will become a constant. This will become a constant. And finally I will be getting minus beta one. Sorry minus x one right. Why

x one. Because derivative of this value with respect to beta one. This beta one will become one and x one will get remained right. So now I will assign this it to zero. So this will basically become my second equation okay. And this equation again I can basically write it over here because I'm trying to prove you what is the formula. Okay I is equal to one to n. Beta zero. Beta 1X1 multiplied by x one, which is equal to zero. Okay. So this is my entire equation. And this becomes my second equation. Now with the help of this two equation what I'm actually going to do I'm going to basically simplify this equation and try to find out what is beta zero and beta one. And we'll try to calculate. Now let's go ahead and take the equation one okay. Now equation one is what two by n right. Or minus two by n summation of I is equal to one to n y of I minus beta zero minus beta 1X1. Right. So here I'm going to take minus two by n summation of I is equal to one to n. This is y of I minus beta zero minus beta one into x of I is equal to zero. Okay. Now let's consider this two I'm just taking this equation. This is my equation one. And we'll try to calculate what beta zero beta one will be. Okay. Now in this equation when I take this two below. And if I put it to the denominator this will be zero by two. Then obviously I'm going to get zero. And if I take it up it is obviously going to be zero. So here in this equation I can also write something like this. I will now write summation of I is equal to one to n and this will be minus over here y of I minus beta zero minus beta one x of I which will be equal to zero. Now let's go ahead and multiply. This summation of I is equal to one to n along with y of I. Right? So now here you'll be able to see I will be getting minus summation of I is equal to one to n y of I. This will be plus, since I don't have any variable that is ranging between I one to n. So here I can directly multiply n multiplied by beta zero minus uh. And again this minus into minus will be plus. And here you know that I can write beta. One summation of I is equal to one to n x of I is equal to zero okay. So this is what we have actually done over here. Now all I have to do is that, uh, I can basically consider n multiplied by beta zero plus beta one multiplied by summation of I is equal to one to n x of I is equal to. I'll move this entire equation over here. So this will become. Summation of I is equal to one to n y of I. Okay, now again my aim is basically to calculate beta zero. Let's say from this I'll be writing beta zero is equal to. Now here you have summation of I is equal to one to n y of I minus. If I take this value over here, then it will be beta. One summation of I is equal to one to n x of I. Okay, and here you already know that I have n now n what I'm actually going to do over here, I'm going to move it in the below denominator. So let's say this is also divided by n and this is also divided by n okay. So finally I can prove that my beta zero is nothing. But if what is happening over here see this is nothing. But this is the summation of y of I divided by n. So this can I say this is a average. So definitely I can write this as y of y hat. Right. So this becomes an average. And similarly over here x of I summation of I is equal to one to n x of I divided by n. This is nothing but beta one multiplied by x bar. So beta zero in this entire equation can be calculated by using this particular equation. Right? So my intercept if I really want to calculate my intercept my intercept can be calculated by using this formula okay. So just imagine we have taken the first equation from here after doing the derivative. And we have computed how to calculate beta zero. Now let's take the second equation and let's do the same thing right now in the second equation you know that it is minus two by n. So here what I'm going to do. Let's consider equation two. Let's consider equation two. Now since we have got beta zero over here obviously now it will become easy to get the beta one right. And beta zero is nothing, but it is the intercept. Now equation two I'm just going to write minus two by n. So let's go ahead and write over here minus two by n. And here I'm just going to write it down for my notes I is equal to one to n y of I minus beta zero minus beta one x of I multiplied by x of I right is equal to zero right. So this is the equation that I actually have. Now this two will go to the denominator n will go to the numerator. So everything will become zero. So now I can write this equation as again same thing here I'm just going to write summation of I is equal to one to n y of I minus beta zero beta one x of I

multiplied by x of I, which is equal to zero. Okay. And the next thing that we are probably going to do over here is that, uh, let's multiply x of I. And summation of I is equal to one to n inside this okay. So after we multiply this here you'll be able to see I'll be getting summation of I is equal to one to n x of I multiplied by y of I. So this will be x I y I, and then I will get minus beta zero. Summation of I is equal to one to n and this x of I will get multiplied over here and minus. Uh, here you'll be seeing I'll be getting beta one. So. Omission of I is equal to one to n, and this will be x of I whole square, right. So this is the equation that will happen after doing all this things. Now one common thing that you see is summation of I is equal to one to n. So what I can do, I can take it as summation of I is equal to one to n, and this will become x of I, y of I, and this will be beta zero of x of I, and this will be beta one into x of I uh, whole square. Right. And this will be let me write like this beta one x of I whole square and this will be basically equal to zero. Right now you know that beta zero value you already know it is y hat minus beta one into x bar. So what I'm going to do I'm going to replace beta zero. With y hat minus y hat minus beta one into x. So if I replace this into this beta one into x bar right. Then what will now happen. Now let me go ahead and write in different colors. So this will be summation of I is equal to one to n x of I, y of I minus beta zero is nothing but y hat minus beta one into x bar, and then you have x of I minus beta 1X1X of I whole square. And this is basically equal to zero. So guys, now once we replace beta0 by y bar minus beta one into x bar here you'll be able to see this. Now in the next step what I will do I will just write summation off. I is equal to one to n and I will be multiplying x of I inside this. Okay. So this will be x of I y of I minus x of I y bar. Then this minus into minus will be plus. Then here I can write beta one x bar x of I minus beta one x of I whole square. Now you will be knowing why we are doing all this things, because at the end of the day will be getting an equation to find out. Beta one okay. Now the next step after this, what we basically do is that here in all this equation, the most common thing that you will be finding is x of I. So let me remove x of I outside. So here we will basically have y of I minus y bar plus beta one into x bar minus beta one into x of I. And this x of I will come outside okay. Now what we do you can remove this x of I by just putting this in the denominator. So obviously this will be entirely zero. Only this side. And this x of I can also be lost. Now the next thing that we are going to do I will write summation of I is equal to one to n. And here I will write y of I minus y bar. Okay. So this is my first equation. And then I will also say beta one. And what I will do here I will write x bar minus x of I is equal to zero okay. Finally. Now the next step that we are going to do. And obviously I can also take this as a separate group. Uh y of I minus y bar together. Right. We can also take this like this together. Right. Because, uh, understand I'm just grouping this and I'm taking a common of beta one, and I'm basically writing beta one x bar minus x. Now what I'm going to do here, I'll write beta one. Let's say that, uh, this is there outside in my complete bracket. Now, if I try to multiply this summation of I one, then it will be summation of I is equal to one y of I minus y bar. Okay. Plus summation of I is equal to one to n beta one x bar minus x of I. Okay, so this is what I'm actually get going to get with respect to zero okay. So I've just made it into two groups okay. Now uh after this I can definitely write something like this. Summation of I is equal to one to n beta one. X bar minus x of I. And here I'm just going to take this as minus. And this will be summation of I is equal to one to n y of I minus y bar. I'm just taking this particular equation at this side. Then finally I will be getting beta one which will nothing be. It will be minus summation of I is equal to one to n okay y of I minus y bar okay y of I minus y bar divided by divided by summation of I is equal to one to n, and this will be x bar minus x of I. Now see over here, if I try to bring this x of I forward then I have to probably take a negative value. So here finally you will be seeing that beta one. I can write it as minus I is equal to one to n y of I minus y bar. And if I take a minus value outside, then I can also write this as I is equal to one to n and x of I minus x bar. Write this minus n minus will get cancelled. And finally see my equation of

beta one will be. Summation of I is equal to one to n y of I minus y hat divided by summation of I is equal to one to n x of I minus. Minus. X bar. Right. So this is my entire equation of calculating beta one. And if you remember beta one is nothing but it is the intercept. So sorry it is the coefficient. So finally, I have actually proved you that with the help of OLS, if you really want to calculate. The beta zero, then your formula is basically given over here, which I've actually proved it. Y hat minus beta one into x bar. So here you'll be seeing y hat minus uh beta. One, I think beta one. Right. Or let me write a beta one into x bar. Right. X bar. This equation is basically to calculate the intercept. This equation is basically to calculate the intercept. And in order to calculate the coefficient. We use this specific equation that is. Y y of I minus y bar divided by x of I minus x bar. So let's say I have some data points x comma y. And here you will be having some values okay. From this obviously you'll be able to calculate x bar and y bar. And all you have to do in order to calculate beta zero. Uh, you just have to equate this particular equation y bar minus beta one into x. First of all, if you really want to calculate beta one, first of all we'll go ahead and calculate beta one over here. And beta one you know y of I y hat. So you'll go and compute this component. You'll go and compute this component and you'll divide this both. You'll get beta one. Once you get beta one you can equate this is over here right. So first thing let's see over here you will calculate y of I minus y bar. Right. With respect to every point you know whatever y bar is there you'll get all these values right. And then you'll calculate x I minus x bar. Here you'll be getting all the values. Then you divide in both of them. Once you divide in both of them you will basically be getting beta one. And after that you can equate in this particular equation beta zero equal to y hat minus beta one into x bar. Then you'll be able to get your intercept. So this is basically your coefficient. And this is basically your intercept. Now, this is with respect to a simple linear regression in multiple linear regression. Again this equation will change, but at the end of the day here you'll be having more other components. Okay. But just to give you an idea like how ordinary least square is basically calculating here I have derived you some kind of formulas. Okay, so I hope you have understood this. Uh, and uh, in the upcoming videos we'll try to do both practical implementation using linear regression using sklearn and the OLS method. And we'll try to compare the results. If you compare the results, uh, the OLS method will approximately be giving the same value when compared to a normal linear regression using sklearn. Okay this is for sure. So I will show you an example as we go ahead by doing some practical implementation. So yes, I will see you all in the next video.

Hello guys! So I'm quite excited that after so much of theoretical intuition understanding both linear regression using gradient descent and linear regression using OLS, now we are going to implement it practically. And this is what we are going to do. We are going to cover both sklearn implementation and OLS implementation. In this specific video we will start with a simple project. And for this particular project we are just going to take a data set which has two features. One is weight and height. And here you can see based on some weight there is some different heights. Right. And this is my entire data and I've just taken 24 records. In short I've just created this data set by my own. And what we are going to do is that on top of this particular data set, we'll try to apply a simple linear regression. Right. And why do I say simple linear regression? Because I just have one independent feature and one dependent feature. So here

weight is basically my independent feature. And since I need to predict height this is my dependent feature. Now what I'm actually going to do over here is that, uh, I'm going to take up this specific data set, and we will try to implement it with using a scalar. So let's go ahead without wasting any time. So here it is. Uh, I'll just open my file. So the first thing is that we will go and implement or import some of the libraries. Right. So first of all I'm going to import pandas as PD pandas is a library which will actually help you to uh, read the data set from different data sources like Excel CSV sheet or. Right. And then I'm also going to import matplotlib. Dot pyplot as plt. This is for visualization purpose. Apart from this I'm also going to import numpy as np. Okay. And guys, please uh, remember one thing that if I probably make any mistakes right, I will not, uh, you know, remove that part of the video. The reason is that I really want to keep all the things in this particular video should look like raw, because whatever issues you are going to face, it is better that we fix it in front of you, because tomorrow you may be also facing that particular issue. So just a reminder. Okay. So next thing what I'll do I'll do matplotlib. Uh, and then I will also say inline so that I can display all my visualization graph within this Jupyter notebook. So let me just go ahead and execute. So here it says uh first error no module named matplotlib inline right. So here if you see what did I import import matplotlib.pyplot as plt. Right. So let's see why this thing is not executing okay. Import matplotlib.pyplot as plt. Because here you can see clearly I have made one spelling mistake. So that is the reason why I'm saying you. Even though I face any kind of issue, I'm not going to remove any part of it. So I'll make it matplotlib. So this is the this is the far spelling mistake was there in this library. So you may also face it. So I'm just going to execute it now. Now the next thing is that uh what I'm actually going to do, uh, here is that I will be using I will be reading the data sets now to read the data set. What I will do, I will write DF dot and let me just open the data set over here. And here you can see the data set name is height uh dash weight dot csv. And this is present in the same folder location where I'm actually working. Okay. So let's go ahead and read this data set. So I'm going to write PD dot read underscore CSV. And here I will be reading my data set. And my data set is nothing but height weight dot. Height, dash weight, dot csv. So if it is already present in this folder location, if I probably go and click on file and open it right, you'll be able to see this particular file. Let me just show you. So right now this are all the files that are created inside this. And here you have height weight dot csv okay. So if you are starting this project, after you download the file, you know, please make sure that you keep that particular files on that same location. Okay. Now once I execute this here now I'm going to get something called as DF dot head. If I see DF dot head here, you'll be able to see. Here, you'll be able to see 2 in 1 inputs and one output that is weight and height. So these are my two features that I'm going to use. Uh, and I'm going to probably create a simple linear regression. Now first of all one thing that you really need to understand what is the relationship between weight and height. Whenever you are probably creating a simple linear regression, it is very much important to actually check how your independent and your dependent features are related, whether they are correlated properly or not. Okay, so one way is that I can use some scatter plots. Okay. And uh, here I've taken a simple data set. Guys, uh, you may be thinking, Krish, why you're not doing feature engineering. Is there any Nan values you have to check or not? Yes, obviously you need to check, but I know in my data set I don't have any other problem. And this is the first simple project. So I really want to go ahead and show you most of the important steps. So first of all I will go ahead with Scatter Plot. In Scatter plot I will be using plt dot c scatter. So scatter is a function which will actually help you to project your data points with respect to x and y axis. That is your weight and height. So I just want to see how what kind of relationship is there when we scatter the data points. So here I'm going to use DF of. Wait. And then I'm also going to use the off off height. Right. So height I think it is capital. So height. Now when I display this you can see that over here right in

this. If you see this specific plot here, you can see on the x axis I've taken the off off height right on the y axis I've taken the off off, uh, sorry. In the x axis, I've actually taken the off off weight in the y axis. I've actually taken the off off height. Now let me just put some axis also. So I'll write plt dot x label. Okay. Um, and here I'm basically going to put it as. Wait. Well, I'm just labeling the x and y label so that it'll look better for you. PLT dot y label. And here is specifically my height. Okay. Let's go ahead and plot it. Now. Here you can see that in the x axis you are able to see height weight. And here you are able to see height okay. Now this is perfect. Now from this particular data point you can obviously see that as the weight is increasing right. Your height is also increasing. So this is a kind of linear relationship. Right. And if you really want to find out whether this relation whether this relationship is positive or negative, you can also use correlation. Right. So what I will do I will just write over here of finding correlation. Right. So if I write df dot c o r r right. This basically means it will show you the correlation between your x and y coordinates. That is your weight and height. Now here you can see within weight to weight right. They will hardly be any variance right. If you try to compare. So this is 1.0. But if you see with respect to the relationship between weight and height it is 0.93. That basically means that it is highly positively correlated. Because inside this correlation, if I go and press shift tab, the method that is used is something called as Pearson correlation. Okay Pearson correlation. And this Pearson correlation basically finds out the relationship between your x and y coordinates. Okay. Now this is perfectly fine. You can also use one library which is called as seaborn for visualization. So let me just show you that first of all what I'll do I will import see. But as SNS. Right. And here you can write SNS dot pair plot. Okay, now this pair plot, what it will do is that in a diagrammatically way, it will try to show you. It will try to display this specific coordinates. Okay. So here I'm just going to write DF and just execute it. So here you will be able to see the same correlation is basically displayed over here right. And obviously height and weight is positively correlated. It is in a linear correlation. And this is also in a linear correlation. Now suppose if you have something in linear correlation you get a strong feeling that yes linear regression is going to work well. Yes, there may be other problems with respect to the independent feature. Like usually when we do multiple linear regression. Right. We also check this relationship between the independent feature. Right. But right now I just have one independent feature. Uh, we will will get to know that how to check multicollinearity and all multicollinearity is a scenario wherein if you have 2 to 3 independent feature and the correlation between those independent feature are high, then we can neglect, uh, those features which are highly correlated. Either of the feature we can basically take. Okay. So here uh, let's go ahead with this one right now because since in my independent feature I just have one in the output feature I have one. So this is perfectly fine. And here you can see that okay, it is linearly correlated. So you get a good feeling that simple linear regression will work well because at the end of the day I have to create a best fit line through it. Okay. Now the first step that we probably start, uh, in simple linear regression here I am just going to divide my, uh, features into independent and dependent feature. Super important uh, whenever we basically starts right features. Now if I probably go and go ahead and display DF dot head here, you know that how many features I have now see this error has come. So don't worry because I know what is the error I made. So here you have something like weight and height okay now this weight is a independent feature and height is your dependent feature. So in order to create your independent feature I will just name it as X, just a naming convention. It is a good practice that we should use this kind of naming conventions. Okay. So x is equal to df off. Now suppose if I want all my weight feature in the x axis so I can first of all write like this, right? If I probably write like this, right then the x that is basically getting created is a series, right? So obviously I get my weight value over here, right? But understand if I probably try to see the type of x here, you'll be

able to see that it will be a series, not a data frames right data frame series now. But if I probably try to see type of DF, it basically shows me that it is data frame. Now series basically means just a single record. You have to take care of one important thing guys, whenever you probably try to create independent feature, right. And if I use one additional bracket then what will happen is that just see this, okay. If I go and execute this like this. And if I write type off X, now you know that we will try to get a data frame right. And probably if I try to. He's here, you'll be able to see that I'm getting a data frame with column name. Always make sure that your independent feature right should be in the form of data frame or two dimensional array. Okay. Independent. Yeah I'm writing it over here because this will be super important when you're training your model. So your independent feature should be data frame. Or it can also be two dimensional array. When I say two dimensional array what does this basically mean. This basically means I can have something like this. See. Suppose if I write like this okay. And now probably let's say I'm going to write NP dot array. Okay. And I'll probably write X. Okay. Now if I write like this here you can see that it is a two dimensional array. Okay. Two dimensional array. And this basically indicates that, uh, how many rows I have I have 123. Let's, let's just see how many rows I have for seeing how many rows I can basically go and execute shape. So if I write dot shape here, you can see that it basically has 23 rows and one column. Okay. But in in data frame what was happening. Let's see in data frame what used to happen. Suppose if I go and execute like this and I remove this one bracket okay. And probably if I write NP dot array right I'm probably just go and see x and then write dot shape. So here you'll be seeing that this will be a one dimensional array. Here no information regarding the column is given. But here importance should be given to the column right. So always make sure that at the end of the day either it should be a data frame which has one column, or it can be a two dimensional array. Also with with uh number of rows and some specific columns. Right now in multilinear regression what will happen. This columns will increase okay. Never keep in this particular format because what will happen I'll tell you okay. Let me just say that I am just writing this as X underscore series. I will try to train with this. And also I'll try to train with that also. And let's see where we'll get an error okay. So over here now this is my x. Uh obviously I've taken this as a two dimensional. DataFrame, which has one column, and then if I probably go and write x dot head, you will be able to see I will be getting this specific value. Okay. So this is my x. Perfect. Now till here we have executed. And let's go ahead and execute this also. Uh, and you know that uh, probably if I also see uh, okay DF of weight uh. What happened? Uh, NP dot array. Oh, I have to check it in X underscore series. Okay. Now here you can see it is one dimension. Perfect. This is done. Now the next thing is that after creating an independent feature I have to create dependent feature. Now in the case of dependent feature I can have it as a series okay. Because dependent feature will only be one one column values, right? We don't have to worry much about that, but we do not have to make it two dimension. Okay, one dimension is properly fine in this because we will have only one dependent feature in a specific problem statement, and this is a regression problem statement. Why do we say this as a regression problem statement? Because here my output value is continuous okay. Super important point. Now here I'm going to basically write df of height. Now this is my y and this is my x. Let's go and execute. Now if you want to see your y here is my output feature right with respect to this. And always remember this needs to be in series or it can be in one dimension okay. So here I'm going to write it down over here. This is variable. Can be in series form or 1D array. Okay, 1D array basically means if I probably write NP dot array of y okay. Dot shape. Here you'll be able to see only one array. So obviously 23 since you have 23 inputs you have 23 outputs right. So this is super important. So and this is very much important guys. Otherwise at the last you will face some issues okay. So here I've actually done my independent dependent feature. The next step we specifically have to do something called as train test split.

And I have already explained why you need to do train test split. This is for the overfitting underfitting condition the train data only will be using for training purpose, and the test data will just be used for checking how the model is predicting for the new data point. So for this we will be using a library from sklearn dot. Production. So I'm going to import. Train test split. And here I'm going to use X train or X test. White rain. Why test? That basically means I'm creating a train data. And with respect to that particular train date, I see train and test. So training data will use the model for training that specific data for testing that model or for seeing that how the model is performing for the new data will only use test data. So that is the reason we are creating this four components. Initially we just had two components. One is x and y. But obviously I need to create x train x test, y train y test. With respect to the output of X train it will be present in Y train. Okay. So this four parameters will get called. When we use this train test split. And internally we'll give x comma y. Let's say I'm giving my test size as 25% okay. So in order to give test size is 25% I will write .25. Now how do you think I've got to know all these things. Just go and press shift tab okay. So here. This, then only you'll be able to see it. Okay, so here you'll be able to see. I have just imported this. And now if I press shift tab here, you'll be able to see or. The arrays are over here you can see test size I have given the arrays basically means x and y array train size. Also you can give and there is also something called as random state. But before this let me make some cells. Just a second I pressed F1. That should not happen. Now let me just go and make more cells. So I'll just copy this entire thing and I'll put it over here. Okay. Perfect. Now understand one thing over here guys. There is something called as random state. Now see, usually when we basically take this random state, let's say I'm going to take 42 the type of train test split. If you don't put this random state right for everybody, it will be a different kind of train test split. We can randomly put pick up the records 75% of the records and put that in the training data set. And we can put we can randomly pick up the 25 percentage of record and put in the test data set. Right. So that is what usually basically happens in this case. So if you write a random state as 42. So whatever way it is basically taking me like taking my data and putting in the training data set. Similarly for you, also the training data set will get selected. So once I execute it. So here you can see now I will be able to see my X train and I will just print the shape also. And obviously remember the shape should be two dimension okay two dimension okay x train dot shape. Sorry I don't have to use this. And since here you can see this is two dimension right? That basically means this one is because of I have one feature and these are 17 rows right. So 17 rows and one feature. Similarly if you have five feature then this will become five features. Okay. Like that it will happen okay. But at the end of the day you will be getting a two dimension. Perfect. Now let's go ahead and let's, uh, discuss what we have to do after this. Okay. Now we have to perform a very important step which is called as standardization. Standardization why it is basically required. Now just understand guys, I'm also going to write in front of you. Let's say I have two features x and y. Let's say x is calculated based on kilogram units. Y is calculated based on some other units, let's say over here. In this particular case, since this is height I'm going to use centimeter okay. Now understand one thing. Whatever values you get over here right. These are of different units. And you know that in linear regression when we use a scalar we use something called as gradient descent. So obviously based on this unit if this value is higher to come to the global minima. Right. But it is going to take time, right? Because I have a bigger value and all the maths equation that is getting applied, it will get applied to a big value, right? Which is again not a suitable solution. Okay. And obviously for the optimization problem will take long time to reach the global minimum. Okay. So what we do is that a very good approach is that we take up each feature, not the output feature. We take up each independent feature and we apply a simple formula which is called as z score. And this formula will basically be converting all your values with mean is equal to zero and standard

deviation equal to one. Okay. That basically means if I take up all these data points, I will write x of I minus mu divided by standard deviation only. This formula gets applied to all data points. So at the end of the day what will happen is that my mean all these data points will get converted with mean is equal to zero and standard deviation equal to one. Super super important okay. So please make sure that you remember this. Uh, and this is a very good step with respect to standardization in the upcoming future, I will also talk about which all algorithm you not apply feature. Uh, you you need not apply standardization. Okay. That part also I will try to discuss. Okay. Now what I'm actually going to do, I'm rub this and let's, uh, apply standardization now for standardization again we will be using an sklearn library. See you guys. Every library that we are going to use you will be able to get it in sklearn. Okay. So uh, if I probably show you one case, right. Let's say I have used train test split. So if you go and probably search for sklearn train test split here, you will be able to see that this is the algorithm that is specifically used. So here you can see all the parameters what parameter what it returns. And you can also see some of the examples right. So all the time suppose if you don't understand where this library exists all you can do is just search the Google for that specific library. So that is the reason I'm able to write from sklearn dot model selection, import train test split. Similarly here you will be able to see sklearn standardization is there which we call it as standard scalar. Okay. Now if I probably search this and if I probably see this here, you will be able to see that, uh, you are getting all the values over here. Right. So perfect. Uh, we are very much in sync and we are able to understand. Okay, now the next thing that we are going to do over here is perform standardization. So from standardization what is the what is the library from sklearn dot pre-processing okay. So here I'm basically going to write from sklearn dot pre-processing okay. Import. Standard scalar. Okay. So once we do this let me just execute it. And here now we are going to standardize my input feature not the output feature. We do not have to standardize the output feature because our gradient descent gets applied into the independent features. Right. So uh, what I'm going to do over here is that I'll just say I'll initialize standardize. So let's say I'm using standard scalar. So this will initialize with that formula that I have written z is equal to x of I minus mu divided by standard deviation okay. Then after this it is just going to write scalar dot fit underscore transform. So let me give this formula again. Scalar dot fit. Underscore transform. And here I'm just going to give my extreme. Okay. Now what will happen for all my extreme data, this transformation will get applied now. Here you can see all the data points over here mean will be zero and standard deviation will go to one. Then that basically means all the data points will be transformed in this scale. And this I will assign it to extreme. Now the same thing I need to do for test data also. Right. But always remember in test data we do not apply fit underscore transform. See the difference between fit underscore transform and transform is that initially whatever data that I have, let's say whatever data I have in train, I'm going to use the same mean. Let's say the formula is like this x of I minus mu divided by standard deviation. I'm going to use this main mean and standard deviation only of this training data set. And I'll apply the same formula in the test data set right in the test data set. So for this particular case what I do in training I say fit underscore transform. And in the test I just say transform. When I say transform, that basically means I'm going to use the training data set mean and standard deviation and use it over here okay. And apply the same formula like x of y minus mu divided by standard deviation. For all these data points I'll be using this mean and standard deviation only if I use transform. If I use fit transform then again mean or standard deviation will get calculated for this data. I don't want that. And why do we do this? Because there is a important concept which is called as data leakage. I don't want my training data to know any info about test data. Okay? I don't want that thing to happen. You know my my model that is created. This should be the truth value okay? This should be the information that should only be known. Right? And when my

model sees this, it should always see this as a new data. So that then only we'll be able to understand whether our model is performing well or not. Okay. Super important data leakage fit underscore transform and transform. So this is a very important interview question okay. Why do we use fit underscore transform. And this we will also be doing in PCA. That is principal component analysis okay. Now let's go ahead and let's continue. So here what I'm actually going to do here for the test data I'm writing scikit scaler dot transform. Okay. Scalar dot transform. Okay. And then I will basically write this on my x test data. Let's see whether we'll get any error. So here again this is perfectly done. Now if you probably go and see your X test you are going to get this. So this is my test data and this is how it has got transformed. This is perfect. Till here. Now the next step that we are probably going to do is that apply machine learning algorithm. That is linear regression. Apply linear regression okay. And this is super important I will again be using this this particular scalar value okay. This this will be again used for the new data. Also any specific new data that has come will again transform them okay. Now I will apply simple linear regression. Now in order to apply will again. Killer. And in this killer, I will search for simple linear regression. So here you'll be able to see something called as linear regression. So this is the model that we are going to apply. So let's go ahead and let's write form sklearn dot linear model import. Linear regression. Okay. Now here you'll be able to see the first thing that I will do again initialize the object whenever we are using any any libraries from sklearn. First of all we need to initialize that object okay. So here I'm going to initialize linear regression. By default the parameters what all things are there. So there is a first parameter which is called as fit underscore intercept. Now this fit underscore intercept is nothing. But it will say that whether to calculate the intercept for this model or not. Obviously I want to calculate this. So by default whatever true it is I'm going to keep it as true. Now this normalize function is there. This parameter is ignored when fit underscore intercept uh intercept is set to false. If true the regressor s will be normalized. And this process we have already done this right. We have normalized or we have standardize this by using standard scaler. We have already done that. So whatever will be the default value will. If you are not that done that step normally you could have kept this as true. Okay then you have an underscore jobs. This n underscore jobs is a very important parameter because if you set this as minus one it is going to use all the processors that are there in your system with the with the help of them, it will try to converge and it will try to do the training. Okay. And uh, positive basically says that when set to true forces the coefficient to be positive. So I don't want this specific thing to happen because some of my coefficients can be negative. Then in return, what all things you'll be getting if you apply this algorithm here, you'll be getting coefficients rank rank of matrix singular singular value of x uh, intercept. But out of all these things, right, the most important thing is coefficient and intercept. And we'll try to see that, uh, how many, uh how many, what is the coefficient intercept and what is the relationship with respect to the final output? We'll see that there is also a feature that will get returned that is called as feature underscore names. Underscore in name of feature seen during fit that also it will show we'll try to see this okay. So here I have applied linear regression. Let me just go and execute with all the default parameters. And I'll say regression dot fit. Now I want to fit this regression model to our training data. Right. So all we will do is that we will go ahead and assign this and we'll assign it to X underscore train. Now understand guys, let's say with before giving this X underscore train, let's say I created one more variable right. And that specific variable was called as x underscore series. Right. And you know that this is one dimension. Now instead of x underscore train if I give this what will happen. See. Let's say if I'm giving this and I'm giving. Why? Because this had all the data. Now see what will happen. It will give us an error. Why? Because it is saying that accept a 2D array. Right. Got one d array instead. Right. And this was a 1D array right. So always remember any algorithm that you'll be seeing in the fit. And whatever

training data I am giving it should always have a 2D array in the output. If you have 1D array that is fine, not a problem. But here you need to give it as a 2D array. Okay, so here what I'm going to give now x train comma y train. Now we will go and do the fit. Now once we go and do the fit. Now let's see over here if I press shift tab. Sorry. Over here. If I press shift tab here, you'll be able to see one parameter, which is called as an underscore jobs. Let's say it is none right now. Right. I'm just going to keep it as minus one. So what it will do. How many processors are there in my. So how many processors are there. It will try to run it on of them. So once I execute here you can see that n underscore jobs is equal to minus one. Yes. You cannot see that because it is happening within the system right now. If I probably go and see my regression now it is basically showing linear regression. Now all I have to do is that in order to predict or in order to predict my test data, how do I do it? So there will be a function which is called as predict. But before this I already told you that how many different types of parameter it is running coefficient. Coefficient basically means slope, uh, intercept and underscore features. Rank and singular will not focus much onto it. So let's go ahead and write to write regression dot. Here I will say coefficient. Now if I go and execute this here I'm going to get the coefficient. Now this coefficient is the slope. And why only one. Because I just have one independent feature. Now if you remember the entire equation if I have one independent feature. So my equation will become y is equal to or y hat is equal to beta zero plus beta one into x one. Right. This beta one is nothing but this specific value okay. And now I can actually also get this value because I have already calculated the intercept. And these are my data points. In this particular case this data points are nothing but weight right. So this data points are weight this. We have calculated beta one intercept we will be able to get when we are writing a regression dot intercept underscore okay. So here let me remove this quickly and let me again show you it to you. The reason why I'm using this. So that I write anything in front of you so you'll be able to understand okay. So here now this is my coefficient. So let me print my coefficient. So here I'm just going to say. Coefficients or slope is nothing but is this specific value? Comma. Regression coefficient. Okay, so here I have just printed it okay. Similarly I can also print intercept. So I'm just going to copy this and write print over here. Intercept. And I'll talk about this. What is the meaning of this? Also intercept is equal to. So this will nothing be I will just write intercept underscore. So this is my intercept and this is my slope. Now what does this be. Uh, with respect to this. Okay. Now, super, super important. That basically means one unit movement in the weight value. Let's say one unit movement in the weight value. That leads to 17.29 unit movements in the height value. That is what it basically says this coefficient. Again understand guys one unit movement okay. Let me just draw it again in front of you. Suppose this is my x and y. This is my weight and this is my height right. This is my height. It basically says that one unit movement with respect to the y axis it leads to 17.29 unit movement. So from here to here basically this is 17.29 unit movement okay. With respect to the y axis. So this is what this probably will look like. My slope okay. So this is what we are doing. And the next point is that what is this. 156 156 says that when your weight value is zero, when at what point your line is meeting in the y axis. So this particular point is nothing but 156 .47. So I hope you have got an idea about it okay. The importance of coefficient and slope okay. And uh, let me do one thing apart from this. Uh, now you have got your coefficient and intercept. Now you may be thinking, Krish. Okay, fine. You have created this. Can we see some kind of line? So suppose if this are my data points, can we see some kind of line which I can say it as a best fit. The answer is yes and I'll show you that. So let's go ahead and plot this particular points right. So here I'm going to plot the training with respect to training data I'm going to plot. Plot. Best fit line. Okay. So here let's go ahead and plot it. So first of all I will say PLT dot scatter. And here I'm going to scatter my x train and y train data. Okay. So what will happen if I plot this here. You'll be able to see this kind of plot okay let me make some more fields. Okay, so here you can definitely

see some plotting happen right on top of it. Now you want to see how our best fit line is created. Then what do I do? I will go and write plt dot yellow t. Plot. This plot is also used. And here I will be taking x train. And we know that how to find out our predicted value right now in order to find out the predicted value, all I will be doing is that I'll take this model. Okay, this model is regression. So here I'll write regression dot predict okay. And here I'm specifically giving what value over here. See if I write regression dot predict. Here I'm going to predict for x train. So this will basically give my predicted value along with the straight line. Now if I go ahead and execute this now see this is the best fit line that has got created in this specific data set. Right. And this best fit line will have an coefficient of 17.29 and intercept of 156. Because see if I probably, um, if I probably try to, uh, you know, um, make sure that or extend this when my value is zero, then you are actually going to get an intercept of 156. Okay. So this is what it basically says with respect to this. Okay. So now here you can see I've got my best fit line. And with respect to this particular best fit line we are able to see that. Okay. Uh, the the error also looks very less right here. Yeah, you can definitely see the error is also very very less. Now let's go ahead and let's go to the next step. Now we need to do the prediction for the test data now in order to do the prediction for the test data. Prediction. For test data. How do I do for the prediction for test data? So here you will be able to see I will be using plot dot scatter or no need to use plot dot scatter also. So my model is something called as regression dot predict on my x test. Now here when I do the prediction this is my predicted value. So this will basically become my y underscore bread okay so let me just execute it. Perfect. Now in order to see how the result is you know how the result is, we will try to uh now check with some of the performance metrics. But before that, if whenever I do prediction for test data. Right I will just go and copy this thing. Okay. Just let's see. Uh, I hope everybody will understand it. Let's say I'm creating a markdown over here. Here you can see the predicted height output will be intercept plus coefficient multiplied by weights. Right. So right now what is my intercept. My intercept is nothing but 156 okay. Let's say this is my value okay. So I'm just going to copy it over here. And my coefficient is nothing but 17.29. So I'm just going to copy this and paste it over here. Okay. So this is what is basically happening in order to find out all my y pride. Okay. So here I'm just going to execute it. So that will give you an idea. Okay. So I hope everybody is able to understand this already. This understand data is already transformed also okay. If you don't know. Now let's go ahead and do some more things in this. Fine. We have done the prediction. Now we'll focus more on the performance now which all performance metrics we have already discussed that we are going to see. So in this I'm just going to write the performance metrics that I have actually discussed. I'll write from sklearn. Dot matrix. I'm going to import. Uh, the library that I'm going to import is something called as mean squared error. So here you have mean absolute error. Mean mean squared error okay. And there are also many, many other performance metrics which can also use right. For right now I'll just use this because we have learned about this okay. And then what I'll do I will basically calculate my MSE mean squared error. So here I'm going to write mean squared error. And here I'm going to give my why underscore test comma y underscore pred. So this will basically give my MSE. Similarly m a e is nothing but mean absolute error. And this will basically give my y underscore test comma y underscore pred. And finally root mean squared error, which is nothing but NP dot square root right off MSE. Then I'm just going to print. MSC. Print m a e. And then print r.m.c. As I said that we really need to check all the parameters over here. Now here you can definitely see that 114 .84 uh, is your mean squared error? Definitely. It has penalized a lot of outliers, you know, whereas in this case here you can see it's almost near by by using mean absolute error and all. Okay. So here, uh, this is what we have actually got with respect to our performance metrics. Now the next thing that we are going to use is something called as uh r square also. So r square, uh, if you remember, uh, the formula for r square, how

we are going to use the R square. So here I'm just going to copy and paste the entire formula for you. We have already discussed about this. It is nothing but one minus sum of square residual divided by sum of square total. Right. So here this is the formula that will apply in order to apply. Also we can also use this from my library. So I'll write from sklearn dot matrix import r2 score. R2 score. Right. So it should be metrics. Okay. So this also will look good because now overall we'll be able to get how our model is performing. So finally my score will be equal to R2 underscore score. Is nothing but y underscore test comma y underscore pred. Right. And if I probably print my score, this will be my R2 score. And here I'm getting 73%. And obviously you cannot get a perfect just by linear line. We'll be seeing how we can improve this by using some of the algorithms like polynomial, linear regression, decision trees and all. Many algorithms are going to come up in the further stages, which will try to improve this particular accuracy. But right now I'm getting 0.73. And obviously let's go and find out. One more. That is called adjusted R square. And as you know adjusted R square formula is this which again I have discussed in my theory part right. It is nothing but one minus one minus r square multiplied by n minus one divided by n minus k minus one. Right. R2 is nothing but R2 score of the model number of observation n number of predictor variables. Right? Right. Now I just have one predictor variable. So I don't think so. It will be such a big issue or very big change in the score. So if I probably go and see it is like somewhere around 0.50. Okay. So I have just applied this formula where I have written one minus score length of y, test minus one y. This is basically giving you the number of observation and all. And this is your X test. And this is your other thing. Right. So yes, uh, I think I have covered all the most of the things over here. Uh, uh, we have also seen that. Okay. I've got 0.73, which is good. I've also seen all the performance metrics like mean squared error, mean absolute error, root mean squared error. This is absolutely fine. We have done some kind of visualization analysis and all. Now let's try to see one thing okay. Let's try to do the same thing with OLS technique OLS technique of linear regression. So can we get the same. Because that formula that I derived right. Can we get the same output uh that we are looking for. You know, now in order to implement OLS linear regression, I will say import Statsmodels dot API as SM okay. And I'm going to execute it. And now I'll just use my model is equal to this model. Will uh, this API that I'm actually, uh, using. Right. Or this uh, library that I'm using, this has a library which is called as OLS. And, you know, in OLS, all we need to give is with respect to your training data. So if you probably go and see over here, OLS, here, you have something called as endog and xog. Right. Endog basically means 1D indigenous response variable. That is my output variable. Xog basically means my true input variable. Right. So here I'm basically going to give my X train. And then I'm just going to do dot fit on this like how we did for linear regression. Once I do this I will get my model. And all I have to do for prediction I will do model dot predict on. X underscore test. And this will basically give my prediction value right. This will give my prediction value. So if I probably execute this print prediction you will be able to see the prediction also. Perfect. So these are my predictions. Now let's do one thing okay I'm just going to print my model summary over here okay. In OLS and see whether I'm getting the same thing. What I did with the help of linear regression using gradient descent. So, uh, again, I'm printing this now here, I'll just zoom out here. You can see that there is one feature X1, and it is having coefficient of 7.29. Now is this matching with this or not. We'll try to see. So if you go over here here also I'm getting coefficient as 7.298 which is quite amazing. Now you can see the proof that with the help of OLS also I'm getting the same coefficient. And yes there are some more parameters over here. Right. They are like r squared value adjusted r squared value right. F statistics value, and this is specifically for uncentered centered data. It is not there. Right. So all these things here you can basically see that. But at the end of the day we are getting a coefficient which is 17.29 right. Which is approximately equal to this particular coefficient that

we got with linear regression. Okay. So this is the most important point with respect to this. That basically means with the help of OLS also we are able to get some good values. Right. Good, good. Uh, because almost all OLS and this sklearn works in the same strategy. Okay. Now till here we have discussed about so many things right now. What how to do the prediction for the new data. So this is super super important prediction for new data. Right now let's say that I get a new weight okay. Let's say my weight value over here. And in order to do the prediction I will write regression dot predict. And my new weight value. My new weight value. Always make sure this you are providing in two dimension again right. So it should be let's say I want to predict it for 72 weight. What should be my output. So here you will be able to see the output. And now guys here you can see that after prediction it is showing 1401. Oh my god. This basically means that the person who is having a weight of 72 is getting a height of 14 zero one centimeters. So what mistake did we do? You know why it is basically giving this kind of output? The answer is very simple guys. We missed a very important step which is called as standardization. See we did this standardization right. And right now my data points are not, you know, standardized down or normalized in this particular, uh, range. Right. So what do we do. We will again use scalar dot transform. So here I will just copy this and I will write over here simple scalar dot transform. And I will give this value inside this. Now once I give this value inside. Then you will be able to see I will be getting a good value. So now this is the correct output with respect to this particular data right now. Just imagine if you did not remember that scaling part you had worried, right? Krish? What happened? Why? I'm getting such a high value. Right. So always make sure that whenever you are doing some kind of predictions, okay? You have to make sure that this problem, this scaling transformation needs to be always done. Okay. So this is super, super important right now. This was all about this. You know the entire linear regression model, simple linear regression model where we took a simple data set and we tried to see much and much more of information. And we tried to do the prediction. So I hope you have understood this. In our next video we are going to see a problem statement where we have multiple features and we'll try to implement multiple linear regression. And we'll also try to see with this OLS whether we will be able to get the same output or not. So yes, I hope you like this video. I will see you all in the next video. Thank.

Hello guys. So this is our second project and in this project we are going to solve a use case using multiple linear regression. Now whenever I say multiple linear regression that basically means I will have at least more than one independent features. Okay. So for this I will just increase the complexity more when compared to the previous one. And the data set that we are going to use is something called as economic underscore index dot CSV. And again this particular data set I have actually prepared by uh seeing some of the stats information. I've not taken the complete data set, but at least 25 records from there. And we'll understand how we can implement multiple linear regression on this particular data set. Let me talk more about this particular data set. This data set is all about index price. Now first of all we'll try to understand what is index price okay. So what is index price. Now here you'll be able to see. Index price is a

measure of relative price changes, consisting of a series of numbers arranged so that a comparison between the value for any two periods. So to calculate this index price and government uses this specific index price over here we require two important things. That is interest rate of the bank and the unemployment rate. And based on this, by applying some formula we basically calculate the index price. Now I have all this details based on month right. All 12 months. You can see two years data is there. That is the reason you can actually find out 24 records. So 12 years data is there. Now we need to create a model which will be taking this two important features. That is interest rate and unemployment rate. And it should be able to give us the predicted output index underscore price. And that is what we are going to do. Why we say this as a multiple linear regression. Because here I specifically have two features. Okay. Uh, in the previous example we just had one independent feature and one output feature. But here we have two independent, uh, independent features and one output feature. Okay. So uh, let's go ahead and let's try to solve this problem. Now, first of all, as usual, what I'm actually going to do over here is that, uh, import pandas as PD. It's the same step import matplotlib dot pyplot as plt, then import numpy as np because this all things has been already done. So I'm just going to quickly, uh, complete this so that I hope everybody knows how to do this. Now we will try to read the data set. So here I'm just taking the specific data set and I'll say DF underscore index okay. And uh I'm just going to read this data set. Let's say I'm going to write read underscore CSV. Read underscore CSV. And here I'm basically going to take sorry PD dot read underscore CSV. And here I'm going to basically take economic index dot CSV right. Now if I write df underscore index dot head. So I basically have my two years data. Very simple. Now over here the first thing that you will be able to see right. You have this extra column that is getting created. So first of all I will just drop this column in order to drop this unnecessary column. Let's drop unnecessary column. And here unnecessary column basically means I can also consider year and month because I don't require this right? Based on interest rate and employment rate, I need to be able to predict the index price and whatever year or whatever month. If this values are given, we should be able to predict the index underscore price okay. So let's drop unnecessary columns. So here first of all I'll use DF dot drop. And always make sure that whenever you're trying to drop a column you have to use one important parameter. Okay. So let me just go ahead and write df underscore index dot drop. And let me press shift tab here. You just need to provide the labels and your axis value. You can also provide index or columns okay. So here first of all let. Recent columns. So columns here, I can definitely provide something like let's say in the list it will be unnamed. UN named, or I'll just copy the same thing. Okay. Hmm. And here I will add access is equal to one. So if I execute this here you can see that this unnamed column will get dropped. Right. And this will be the remaining data that I'll have. But I also have to drop year and month. So what I'll do I in this list I will add here and I will also add my month column. So once I do this here you'll be able to see that. It is getting deleted. It's working perfectly fine. Now, when you are sure about this, then there will be one more feature inside this one more that will be in place is equal to right? So here when you do in place, in place is equal to true. That basically means you are actually dropping the column now. Right. So uh and once you drop the column it will get updated in DF underscore index. That is the main uh importance of in place underscore true. That basically means within this particular data set you are dropping it and you're updating it. So now if I go ahead and execute this this will be my DF underscore index dot head. Now here you can see I have two features independent features and one dependent feature. Perfect. Now first of all let's go and see whether we have any null value. So in order to check the null value I will write df dot is null dot sum. This is the common code to basically find out check null values okay. And right now I'm not taking it, but as we go ahead we'll have some amazing problem statements. So let me go ahead and write it as DF underscore index because that is the habit

right? Most of the time we will try to write df. DF. DF. And. In. In. If. Now after checking the null value here you can see you don't have any null values. So obviously this particular data set, what I'm actually first going to do is that let's do some visualization okay. And this visualization that I specifically do, I use a library which is called as import seaborn. As soon as. And then here I'm just going to write a SNS dot. Uh, for that specific library, uh, function that I'm going to use is something called a spare plot. And if I give my DF value here automatically, sorry, it should be DF underscore index. Again, I'm making this mistake again and again. Not a problem. Now, if I probably go and see this DF underscore index, here is what things you're actually getting. You can basically see with respect to in index price and interest rate you can see some some linear relationship is there with respect to index. Parade. You can see inverse relationship is there. Right. So again if you check employment rate and interest rate again here you are getting an inverse relationship. So if I probably go ahead and write df.df_index.co rr. So here what we are going to get is that we are going to get some negative values. Also because this interest rate and employment rate here you are having a negative correlation. Similarly with respect to this index price and unemployment rate here you are having a negative correlation. But if I see with respect to interest rate and index price you have a positive correlation. So that basically means as the interest rate is increasing, your index price will keep on increasing. If your unemployment rate is increasing, your index price will be decreasing. Now we are going to use this two specific feature and try to implement some amazing, uh, you know, uh, will basically create a machine learning algorithm using multiple linear regression. Okay. Now let's go ahead and let me first of all make some cells. I'd also like to make sure that I visualize this data points right. Unemployment rate interest rate properly right. So let's go ahead and visualize this more closely. Visualize the data points more closely okay. So here uh I'm going to just take DF underscore index. Or I can also use plt dot scatter plot. And here I will be using df off. Let's say I will go from interest rate. Interest underscore rate along with. I really also want to see the correlation between um, okay, here one more thing that we saw. Right. What is the correlation between unemployment and interest rate. So obviously here you can see that it is a inverse one. And with respect to index price also it is inverse one. So obviously both these features will play an important role in predicting the output. Okay. So now if I go ahead and probably uh see the scatter plot DF of interest rate and DF of unemployment rate, okay. So let's say unemployment underscore rate okay. And here I will probably use a color. Let's say color is equal to red points. Okay. So here it will be our. Now if I probably execute it. So here you can see DF underscore index. Again same mistake. But I don't want to edit this part now because it can happen with you okay. So here obviously you can see that it is having an inverse relationship okay. So obviously here I can also do plt plt dot x label. And here I'm just going to write it as interest rate. PLT dot y label. Here I'm going to basically have. Unemployment rate. Okay. And once I execute it here, you'll be able to see it obviously. Inverse relationship, the same diagram. We have got it in a bigger one okay. Now. Will do it for, uh, the next one. That is interest rate and index price. Obviously, you'll get a positive correlation. Fine. This is perfectly fine. Now let's go ahead and implement. First of all, what I'm going to do is that get my independent. And dependent feature, right? And when I get my independent and dependent features, first of all I'll write x is equal to df underscore index. I know what all column I basically want. So I'm just going to write my column name. So first one is interest rate. Let's see. Um this will be my interest rate. And this. The next column will basically be my. Unemployment rate. Okay. Uh, I can also do in this particular way. Let me show you. And another way how to how you can basically do it. Okay. So DF underscore index I can also use iloc okay. And here I loc first parameter says take all the rows comma from all the rows. Just remove the minus last row. And here you will be able to get your x value. That basically means from all the features, just from all this features that you have, just remove the last feature.

Because index underscore price is your output feature. And this is obviously a continuous value. So regression problem we are trying to solve okay. Similarly for the y variable here you can write df. DF underscore index dot iloc. And here colon comma colon. Or minus one. I'm just saying that. Take the last column and put it in the Y value. Okay. So this two way also you can basically do it okay. So once I execute it here you'll be able to see x. So let's go and see x dot head. Here you see your interest rate and unemployment rate. And similarly if you probably go and see y here you'll be also able to see. Anyhow this will give you an n tuple. So I'll not suggest you to go and display your data like that here. It's perfect. Now the next step is very simple. We will do the train test split as already explained. Train test split. Okay. From sklearn dot model selection. I'm just going to import train test split okay. And then you have X train comma. X test. Comma y train comma y underscore test okay. And then here I'm just going to use my train test split. Uh, I'm just getting bored a little bit. So let me go ahead and copy some of the code from here. So this is my train test split. Let's say I'm doing 25%. So I'm just going to paste it over here okay. So here I've given x y test size 0.25 and random state underscore 42. So once I execute it here you will be able to see your X train Y train. And all right. Now, guys, I'm also going to introduce you all to a new kind of plot in Seaborn, which will basically be very, very helpful when you are actually working with this kind of data. Now let me go ahead and show it to you. First of all, let me import Seaborn. Okay. As SNS. Now in Seaborn you have something called as Reg plot. Okay. And if you probably want to see this plot, I'll just execute this. Okay. And let me copy this one over here and let me just write shift tab. Now here in this particular plot, this plot is all about this plot. Uh, it plots data and a linear regression model fit. Okay, so let's say if I'm using df of df of, uh, let's say in my x axis, I'm just going to put one of the feature which is like interest rate. And my another feature over here will be DF off. The index price. Okay. So if I just try to plot this. Okay. So it should be DF underscore index. Okay. My mistake. Again not a problem. You should do mistakes okay DF underscore index. So once you plot this here you'll be able to see with respect to this kind of data points. This is the regression line that will get created. So this gives you an amazing idea that yes the errors that probably are going to get is quite less okay. Similarly, let's say that I'm going to copy this same thing and I'm just going to plot this along with, um, the interest rate, let's say the interest rate and the unemployment rate. So if I probably write unemployment rate and if I plot it here also, you'll be able to see that. Yeah, you are able to get, uh, an amazing regression line, uh, within this particular points. Okay. And, uh, this way and this, this, uh, shaded region basically shows that, yes, you can hyperparameter tune this particular line to some level. Okay. So and similarly, if I probably try to plot this with index underscore price here also you'll be seeing an inverse manner. And from with respect to that here also you'll be able to see this kind of line. Okay. So uh, this was just an idea to give you every video. I'll try to teach you something new. So I wanted to include this regression plot also, which will actually give you a idea. But again you cannot do this for more than two features. You cannot basically create a 3D plot. It asks for only two features over here x and y. So with respect to that, you can definitely check it out. Okay, now since we have done our train test split, uh, the next thing that we are going to do is that obviously standard scaling right. So I'm writing going to write from sklearn dot. Now see I am not remembering where the standard scalar fall, so I can go and search for it as sklearn standard scalar. I'm actually you do not remember everything guys. So it is present in processing okay. Write from sklearn dot pre-processing. I'm going to import. Standard scalar. Okay. Standard scalar. I think that is only the spelling okay. Standard scalar. Okay, perfect. So let me just go ahead and import uh processing. This usually happens when you are doing the coding because I'm not seeing any materials as such right now because whatever comes in my mind, I'm basically, uh, following the best methodologies as a developer, you know? So this will also help me to learn things better. So now what I'm actually going to do over here is that I'm

going to create a scalar, and I'm going to basically say standard scalar. Um, standard scalar. I'm going to initialize it. Okay. Then what we do we write scalar dot fit underscore transform on my x underscore train data. Similarly this will go in my x underscore train data. And similarly here you will be able to see x underscore test is equal to scalar dot fit underscore transform on our X test data okay. So this is my Xtrain and xtest. Perfect. Uh we have got our X train. Now let's see. We don't have to do it for the Y value. Now see here you have got two values right. And when we do fit underscore transform it converts this into a two dimensional array. So initially I had a data frame. It gets converted into a two dimensional. So this becomes my input right now the next step what I'm actually going to do I'm just going to write from sklearn dot. Ah linear model. Okay I'm going to import something called as linear. Linear regression. Okay. And this will basically be my regression where I'm specifically writing linear regression. And this I've got executed. Okay. Then what I'm actually going to do regression dot fit on my x train comma y train data okay this is done. Now let me introduce you to something called as Crossvalscore also. Okay. Now this crossvalscore will also give you some good results. Okay. So let me just search for sklearn Crossvalscore. Okay. Now as I said every video I will try to teach you something new. Okay? Now Crossvalscore is basically a val a cross validation method. Okay. And this is where I really want to talk about cross validation. And here all you have to do is that give your model, give your X train, Y train. And this in turn will basically create a cross validation. Now what does this basically mean. Let's say I'm going to write over here okay. This is my regression right. It has fitted to the Xtrain and ytrain. Let's say I import cross validation. So if I say sklearn dot let's see whether cross validation is falling. It falls in model selection. And I'm going to import Crossvalscore okay. Now inside this crossvalscore what I'm going to say said okay fine, this will be my cross Val or cross or let me say that this is my validation score, okay. This in turn. First of all, we will give this crossvalscore over here the data set that we are going to give. Obviously, when I press shift tab, I have to give my estimator. Estimator basically means my model okay. So here I have got my model. And then in the next parameter, if I probably go and see, I have to give my x and y value. Okay. So here obviously I will give my x train comma y underscore train perfect. The next parameter that I will probably give over here if I see there is something called as scoring okay. Scoring parameter. Now in this particular case uh let's see what should be the scoring parameter okay. So if you probably go and see over here similar to cross validate but only a single metric is used. So if I probably see a scoring parameter in that, uh, let's say I'm just going to search for sklearn scoring metrics. Okay. Now inside this, if I go and check it out, how many different types of scoring parameter are there. So for classification you have this many. For clustering you have this many for regression you have this many. So let's say that I am just going to consider negative mean squared error okay. Negative mean squared error basically means I'm just going to get the mean squared error. But in a negative uh manner okay. So I'm just going to copy this and put it over here okay. This will be my scoring parameter okay. So here I'm just going to paste it. Perfect. Now this is done. Now my next parameter what I'm actually going to do. Uh if you go over here, the next parameter that is present right with respect to cross val score is cross validation. Now this is super important. If I say my CV is five okay. What does this basically mean. Well let's say I'm just going to make CV is equal to three. Let's say if I have 1000 records total records okay. Okay. Now, initially let's say I made this as train test split. Okay. Let's say in train I have 900 records. In test I have 100 records. This data, I'm never going to use it. Right. I will just be using this data now. This data will only be used when we need to test our model with the new data. But further to hyperparameter tune my model or to make my model better. You know I can split this train data into train and validation. Now when my when I split this data into train and validation, what will happen is that let's say I will split this based on some mechanism. What mechanism? I will say let's go ahead and take cross validation as five.

Now what will happen if I divide 900 by five? This basically indicates that how? What is the value that I am going to get? 515818 right. So this basically means in my first experiment when I am taking my entire data set, the first 180 records will be my test data. And remaining. All will be my train data. Then in the second experiment that is, cross validation is equal to two. The next 180 record will be my test data and remaining all will be the training data. Similarly, I'll go with respect to cross validation five the next study. Let me just do control Z. The next 180 record will be my test and remaining all will be the train. Similarly, I'll go like this. And at the end of the day we'll be able to see at cross validation five. The last 180 records will be my train test and remaining. All will be my train data. Okay, now as you know that with this cross validation score for every cross validation we will be getting a MSE. From where is this MSE come? This came from this particular scoring parameter. Okay. So here we are going to use negative mean squared error. That basically means whatever MSE I'm getting it will be a negative value okay. Negative value. So this is what we are going to do. And again our target is that it should be the more this value comes towards zero the more better this specific model is. So here with the help of cross validation I will be getting five different accuracies. This will be my MSE one, MSE two, MSE three, MSE four, MSE five. And then I can probably take the average of all this and probably print it over here. Okay. So this is what we are going to do over here. So let me just go and show it to you. Right now I've just taken cross validation is equal to five. That basically means I'm going to get three validation scores okay. So let me just execute it. And this has got executed. If I go and probably see my validation underscore score here I'm getting three MSE values mean squared error values. And now if I probably do the average of this how do I find out the average of this. You can basically do NP dot mean right NP dot mean. So by this way you will be able to get the mean MSE. Okay. So this is basically my mean validation score by performing all my uh, basically cross validation on this particular data set. Okay. But again, uh, this is just to check or to do additional step of cross validation. So here this is how you have to basically do the cross validation okay. With the model that we have created. But now let's go and and do the prediction. So now I will do the prediction. Okay. Now before we do another prediction guys, I also want to make some assumptions okay. And this assumption are super important. But before that let me do the prediction. So here I will be using the regression dot predict. And here I'm just going to give my x underscore test data. And this will basically be my y pred right. So y underscore pred. Perfect. Now after this uh all the things that are common. Right. So first thing, uh, I'm just going to go ahead with this entire code that is performance metrics. And I'm just going to copy it over here. And similarly I will go ahead and copy this where I'll be finding the mean squared error and all. And here I this is where I've actually printed. So this is my mean squared error with respect to the prediction data. And uh, this is my mean absolute error with respect to the prediction data. And this is fine right now let's start with some assumptions. Assumptions. But before this, let me also print the r square and adjusted r square. Right. So here is my code. See guys I'm just doing along with you. Please make sure that you also do it and you also practice. So this is my R square. And similarly I will go and copy this my print R square. And after that I will also go ahead and print this specific value. Right. So this is my adjusted R square right. So here. This part. Also, I'm going to print it. Now let's execute. Now here you can see 75%. I'm actually getting it as R square and 59% I'm getting as an adjusted R square. Okay. And obviously adjusted R square will be letter less when compared to this okay. And but this is with respect to the test data to just see that how my model is performing well or not okay. But again I just have small number of data points. So I'm able to get this. If you have more data points, if you have more dimensions at that point of time, you know, we also have to check how the data is performing well or not. Now let's go ahead and let's learn something new. These are the assumptions like, finally, I should be able to find out whether my model is performing well or

not. And by seeing these assumptions, we can definitely come to a conclusion that my model will definitely perform well. So the first assumptions are after we do the prediction right, the first assumption is that I will just go ahead and plot my scatter plot, and I will say here, I'm just going to give my y test comma y pred. Now, if I execute this here, you'll be able to see that they will be following some some relationship here between the whitest and vibrate. If there is a linear relationship, that basically means your model has performed well. Okay, now the next thing that I am going to do is that with respect to residuals, let's go ahead and calculate the residuals or the errors between y test and y pred. If I take up this residuals okay. Residuals here you'll be able to see I will be getting this errors right. These are my errors that I'm actually getting. Now if I plot this residuals. If I plot this residuals with the help of this plot, let's say how do we write this plot as SNS dot distplot? It is basically present in Seaborn. And if I probably write residuals over here, and the kind of plot that I'm going to take is something called as KDE plot, that is kernel density estimator. And if I execute it here, you'll be able to get something called as a linear regression a normal distribution. So if this kind of curve is basically coming that basically means yes, uh, the model that you have created is good. Okay. And one final thing, uh, uh, that we have to probably take with respect to assumptions is that let's create a scatter plot. With respect to. Predictions and residuals. So if I probably create this plot with residuals and predictions, you'll also be able to see some observation. Okay so what I'll write. I'll say plt dot scatter with. Um. Why underscore pred comma residuals. Here you'll be able to see. That. Follow any pattern. See, this data is uniformly distributed. Okay, now you may be thinking Chris. Oh, the data is very less. Know why you are saying it is uniformly distributed? First of all, let me remove this. Okay. Everything will make sense. Are uniformly distributed. Basically means all data is distributed everywhere. You know it does not follow any, uh, any patterns as such here. Also, you cannot see any patterns when will be taking huge data set uh, in going forward. You know, we'll be doing some projects which will be having many data points. You'll be seeing that this data sets points will be something like this. So this is uniformly distributed. It does not follow some patterns. If I say some patterns it may follow something like this okay. It may follow this kind of pattern. If it follows this kind of pattern that basically something is wrong okay something. But if it follows this uniform distribution, we can definitely come to a conclusion that there is no such problem as such. Okay, so uh, yes, most of the thing we have covered in this now final thing is that obviously we need to compute the OLS. That is ordinary least square. So again I'm going to copy this entire thing. Uh, copy the. Y train and X train part. And do the fit. And once I execute it and if I see model dot summary. Now our main thing is basically to check whether are we getting the coefficients almost similar. So coefficient over here is 88.2728 and 116.2572. So are we getting that similar kind of coefficients. Finger crossed. Right. So let's go ahead and print it. Uh, where did I print it? So where is the code? Oh, I did not even print it. Amazing. Okay, so let's print the coefficient, uh, for my model. So here also, if I probably print. Regression dot coefficient. Here you'll be able to see perfect 88.272 and 116.257. So yes, we have done a fabulous job. And with this formula also we are able to get this. Okay. Now in the next video, uh, what we'll do is that, uh, we'll take up a problem statement. And, you know, let's say that if I have many features and all of these features have a concept of multicollinearity, multicollinearity basically means that. Suppose if I have some input features. Okay. I have feature one. Feature two. Feature three. Feature four. Okay. And with respect to if this two are correlated. If this two are correlated, if this two are correlated, if this two are correlated then it is fine. Okay. But what happens if this two feature are correlated by 95% or 96%? Then what we can do is that since both this feature since if if this both the feature are correlated by this much percent, we can definitely say that almost both the feature are almost same. So we can remove one of the feature. But how to do this kind of elimination? We will try to have a look on to the next project that we are going to do. Okay, so

yes, this was it from my side, I hope. You like this particular session? And yes, I will see you all in the next session. Thank you.

In this video, we are going to discuss about a new machine learning algorithm, which is called as polynomial regression. Now guys, already we have covered simple linear regression, multiple linear regression and I hope everybody knows about the equation. Let's consider an example over here. Suppose if I have some kind of a single independent feature that is x and single dependent feature that is y. And let's say if I have some kind of data points, the main aim in simple linear regression was to create a best fit line. Right. So our main aim was to basically create a best fit line. And this was specifically to one input feature and one output feature. But if you have more than one input feature. So we basically have to select a plane right. A 3D plane, let's say if there is a, uh, three dimensional points, you know, with respect to X, Y and Z, at that point of time, if Z is my output feature, then I have to probably create a 3D plane and if there is more than three dimension, then we probably have to create a hyper plane, right? So that was our aim in this, right? In uh, in regression and obviously the simple linear regression. Why do we say it as simple? Because you just have one independent feature. If you say multiple linear regression, that basically means the independent feature are more than one. Okay. Now what if if my data set is something like this, which is not linear, which does not have a linear relationship, let's say over here, X is my input feature and y is my dependent feature. Right. And if I plot my points here you can see some kind of curve. Right. So this is basically my curve right. And here there is no linear relationship. So this is a non-linear relationship I really want to put this point okay. Non linear relationship okay. Now what happens if we try to create a best fit line. Let's say if this is the best fit line over here then obviously what will happen if we just try to use this best fit line. The error will be quite high right? The loss value that you are probably going to compute that is the value. Between the truth value and the difference between the truth value and the predicted value, that specific error is going to be quite high. Right. So obviously we cannot just use this linear regression over here or multiple linear regression in the case of, uh, if you have more than one feature, we cannot use this. So we need to find out a way wherein we should be able to catch all this particular non-linear relationship. Or, uh, when I say catch, it should be able to identify it. So at the end of the day, let's say I want a line which should be able to capture like this right now, if you are able to capture like this, that basically means for any input point, I will be able to do the prediction and this will definitely have error less right? Error will be less. And how do we do this? For this we will be using something called as polynomial regression okay. Now polynomial regression concept is very much simple. If you already know simple or simple linear regression and multiple linear regression, you will be definitely be able to understand polynomial regression. So here one thing that you really need to focus on is something called as polynomial degrees okay degrees. So if I probably write a simple polynomial equation just have a look how it will look like. Suppose if I say my polynomial equation with degree is where the degree is zero. Now when I write degree zero that basically my equation will look something like this. See over here, let's say I will start with simple polynomial regression okay. So here I'm just going to write it down as simple polynomial regression. I'm just going to write down the equation. Now let's consider and write our simple polynomial regression

with degrees equal to zero. So here I'm just going to write. The degree is equal to zero. So my equation will look like h of x. H theta of X is equal to nothing but beta zero plus. First of all, I'll just write the equation like this beta one x. Right now when I say x is equal to zero, that basically means we are not coming at this particular stage yet. Okay, so here when I say my polynomial degree is zero, understand one important thing. Okay. As we know that our simple linear regression is given by this particular equation okay. So if I write beta zero plus beta one into x here definitely we'll be having a number one. Right. And for this one we can also write this in this particular format that is multiplied by x to the power of zero. Right. So whenever we write a simple polynomial regression with polynomial degree as zero, we get this specific equation. And I even do not write this because here we are taking this degree as zero okay degree as zero. So if degree is zero, obviously the next term that we are going to get with respect to any input feature, it will not count. Okay. So usually this particular value, whenever we say we are trying to create a polynomial regression with degree as zero, this will be a constant value okay. Super important. Now see everything will make sense when I keep on increasing the degree value. Okay, so let's say now my polynomial degree. My polynomial degree is one. When it is one. I will go ahead and write my state of X. Now my state of X is nothing, but this will be beta zero multiplied by x of zero. Now, since my polynomial degree is one, now I'll go and write beta one into x to the power of one. Right now this is nothing, but this is my simple linear regression, right? Obviously this matches my simple linear regression. Very simple right. Beta zero plus beta one into x. Right. So this is nothing but what it is simple linear regression. Right. So you can basically say simple linear regression with only one independent feature here. You can say that it is nothing, but it is a with polynomial degree is equal to one. Now what happens when polynomial degree becomes two. Now what will be the changes in this specific equation? Now when the polynomial degree is equal to two at this point of time, my equation will basically change into h of x beta zero. Back to the power of zero beta one multiplied by x to the power of one plus beta two multiplied by x to the power of two. Right. So here you can basically see that. Now since my polynomial degree is two. Now the same input feature I took x to the power of one. And then I used beta two multiplied by x to the power of two. Right. And this is with respect to simple polynomial regression. When I say simple polynomial I just have one input feature and one output feature. Right. This is super important. Okay. So I hope you are able to understand. So if I have finally a polynomial degree. With respect to N, then you'll be seeing my h theta of x will be beta zero multiplied by x to the power of zero plus beta one multiplied by x to the power of one, then beta two multiplied by x to the power of three. And then I can still write beta three multiplied by x to the power of three. And finally, like this, I'll go to beta n multiplied by x to the power of n, right? So this is what is the final equation with respect to a simple polynomial regression and why we are doing this. Why we are doing this because as we keep on taking the polynomial values. So let's say if I plot this 2D plots right. And let us say this is my x and y axis. And here you'll be able to see some data points. Right now. What will happen is that suppose, let's say with degree is equal to one. I know what is the kind of line that I may be getting. I will be getting something like this. Right. So this is nothing but simple linear regression. But when I keep on increasing the degree, then what will happen is that let's say the next line will get created. Something like this. Okay. Let's say this is with degree one. Okay, since we are increasing the polynomial now, what will happen is that this specific h theta of x or best fit line that we create. As we keep on increasing the degree value, you will be able to see that it will be able to fit very much properly. Okay. Then it will finally be able to fit very much properly. Okay, now what will happen? Let's say if I take degree is equal to some 15, some higher value, then what will happen? My best fit line will start over fitting all these points like this. Right. So it is our task and this becomes an overfitting condition okay. It is our task that when we are creating a best fit line, we always need

to go ahead and play with this degree values. And we have to select a, we have to select a degree value which will be able to. You know, probably, uh, it should not overfit the model. It should not underfit the model, but it should generalize the specific model. Now, in this case, the white line that I've actually drawn, I feel that it is generalize the model. Right. This is super important. Right. So at the end of the day, with the help of polynomial degrees, you're trying to create a model for a non-linear relationship. Here. Specifically you're solving this non-linear relationship. Now, this was with respect to only one independent feature. Suppose if I have multiple independent feature, then how do I write with degrees equal to one? So here I have h theta of x is nothing but beta zero plus beta 1X1 plus beta 2X2. Let's say I'm just taking two independent feature and this will be with degree is equal to one. Right. If I say degree is equal to two, then how? My polynomial degree. Whenever I say degree, that basically means polynomial degree. Now my theta x will become beta zero plus beta one. X to the power of one, then beta one. Sorry, beta two. Beta two is already there, so I will keep on writing this one beta 2X2. Then I will write beta three x to the power of one whole square, then beta four x to the power of two. Sorry x two whole square. Right? So this will basically be my equation right now here I have taken with respect to this for every input feature, right. I have to also do the squaring of that. But basically since my degree is two. So I have to basically do the squaring part of what both x one and x two. And similarly when degree is equal to three forward, I will keep on writing beta five multiplied by x to the power of uh, sorry x three, sorry x one whole cube plus beta 6X2 whole cube, something like that. Right. And here you can see that we have two independent features. You have more independent feature will keep on basically using this kind of uh technique. Right. So I hope you got an idea about all these things. And I hope you are able to understand about polynomial regression. Now we'll go ahead and solve a problem with respect to this. And we'll try to see practically. Also try to implement and show it to you how we can solve this kind of problem. But again polynomial regression main thing is that we are trying to solve non-linear relationship data points with respect to our independent and dependent feature. So yes. Uh, I will see you all in the next video. Thank you.

In this video, I'm going to implement a simple machine learning project. With the help of polynomial regression machine learning algorithm. We will. What we will do is that we'll try to create our small data set, and we'll make sure that that specific data set will have a non-linear relationship. And with the help of polynomial regression, we'll try to solve that particular model. Or basically we'll create a model and we'll try to create some kind of best fit line so that it gives us, uh, good accuracy when compared to a simple linear regression model or a multiple linear regression model. So first of all, let's go ahead and let's do one thing over here. Uh, the first thing that I am actually going to do is that import some of the libraries okay. So let's go ahead and import some of the libraries. And obviously first step is importing libraries. Right. And everybody is familiar with that because you know from now onwards like whatever thing you know, what are libraries we basically install or import. So here import numpy as MP import. Pandas as PD and import. Matplotlib.pyplot as PLT. Right. So this is there. And then what I'm actually going to

do is that I see the type of data set that I really want to create. Let's imagine that I want to create a data set. Now whenever I say my data is non-linear, like if I probably create this kind of data with x and y. So here I can say, okay, it has some kind of linear relationship, right? But if I say if I have some data points which are in this shape, let's say it is in this specific shape. So here you can see that some type of curves are there right in this data set. And what exactly this is, this is nothing but this kind of curve in a data set. We get basically from quadratic equations. And what is the form of quadratic equation in this shape a x square plus b x plus y is equal to zero. So this is a quadratic equation right. And that is the reason why we use gradient descent in a specific. You know this gradient descent is also like whatever loss function we use right. Like MSE. This is also coming from this quadratic equation. So that is the reason we get this. Now what I'll do is that we'll try to create this data point okay. Like let's say this is my x and y. And for this we'll try to see what happens when we create a best fit line like this. And after that obviously here the error, the error that we are going to see over here will be quite high, right. Once we create a best fit line. So we really need to create something like this as a best fit line. And this will only be possible with the help of polynomial regression. Right. So with the help of polynomial regression, we'll try to see how we can create this kind of best fit line for for this specific data set. So uh, let's go ahead and let's quickly try to see. And anything that I probably have to write, I will be writing in front of you so that you will be able to understand and definitely make sure that you can practice all these things. Like you can also consider any kind of other data set which follows this specific pattern. So to begin with, what I'll do is that I'm just going to copy two lines of code from numpy, which will actually help me to create my data set. So here what I have done is that in the X which is my independent features here you can see I'm using NP dot random dot rand that basically randomly we are going to pick up 100 numbers okay. In this specific shape 100 comma one and minus three. That basically means I just want to add some kind of outliers to this x axis or x value by, you know, by subtracting with minus three. And this you can see this is a quadratic equation right. Ax squared plus bx plus c right plus NP dot random is again an outlier. We are specifically trying to add some kind of outlier okay. So here I can definitely write. This is my quadratic equation. Used. And here you have something like y is equal to two x to the power of sorry. It is .5.5 x to the power of two. Right. And then plus that is nothing but 1.5 x plus two plus outliers. Right. So this is the format that we have specifically used in this okay. And then let's go ahead and plot it. So I have already made sure that let's plot the uh matplotlib. Percentile matplotlib inline, so that we'll be able to display these images in this Jupyter notebook. Okay. So quickly done, right. Uh, mat plot plot. See the spelling? Okay. See the spelling over here? Your spelling should be absolutely fine. So done. Now if I say plt dot. Plot. PLT dot plot on my x axis, comma y axis. And let's say I'm going to use some green points just to plot it. Okay. And along with this I will say this will basically be my X labels. I'm just preparing my data set guys. Nothing. So uh, amazing in this because we should definitely try to know how to create our data set just for testing purpose for any algorithm. Okay. That will actually help us to see something. So here it says PLT dot plot. Again plot is not actually working. So what I'll do is that uh you can see the data points over here. Not need to use plot will use scatter. Let's see okay. Scatter in scatter. We don't give like this. Uh, we have to assign something like a color. And let's say here I'm giving red. Now here let's consider green one. Okay. Red is too dark. Now, the data points that you see over here, obviously it does not have a complete linear relationship right. Non-linear relationship is also there. You can see this kind of curve okay. Now we'll consider this particular data set. And we will try to probably implement uh a polynomial linear regression. And we'll try to see first of all with the help of simple linear regression, if we try to implement, you know, what is going to happen. And with the help of uh, uh, polynomial, how the accuracy is basically going to get improved. Okay. So after this, what I'm actually going to do, I'm

going to do a train test split. So I'll write from sklearn dot model selection because I really need to do a test train test split. So from sklearn dot model selection import. Train test split. Okay then here you have X train comma. X test. My train. Comma y. Test is equal to. Train test split. X comma y comma. Here I'm going to specify my test size. Let's say I'm just saying 0.2. And I will be considering some random state. Let's see. Random underscore state is equal to 42. Okay. Perfect. So let's see what is the error. I think somewhere some error may be there a random state. Perfect. Not a problem. If you are getting an error don't worry, you just have to fix it by seeing the description. Okay then. Now let's implement. Let's. Um, implement. Uh. Simple linear regression. Right. So here from again sklearn dot linear models, whatever we have actually done in the previous class import linear regression. And here I'm just going to say regression underscore one. This is my linear regression. So I will just say linear regression. And here I'll be initializing perfect. Then finally we will do regression dot fit on my x underscore train comma y underscore train data okay. And we'll try to execute over here. See you can also do scaling. But already you know the values that you will be seeing right of Xtrain and Ytrain. It is almost in the basic scale. But if you really want to do you can definitely do it C minus two one zero. So whatever initialization is basically done over there, right? It is based on normal distribution itself okay. So dot fit on Xtrain and Ytrain. And once we do this here you'll be able to see this thing. Now what I'll do from sklearn dot metrics I'm going to import the R2 score right. And finally I'll say score is equal to okay. Now we are going to use the R2 score just to calculate what is the score of the model. And here I'll say y underscore test comma. And then we can use the our regression one dot predict. And I will try to predict my x test data. Right. So that becomes my ypred. And finally I will just try to print the score. Now obviously you will be seeing that uh you'll be getting some, uh, error. So sorry. You'll be getting some less accuracy y less accuracy, obviously, because your data point is like this. If you try to create a best fit line, so obviously you're going to get a bad error, right? So let's visualize this image okay. Let's visualize this model and how it is uh created a best fit line okay. So in order to do this, what I will be using is that I'll be using a simple plot function. Uh, to just see that how the plotting is basically happened. So I can basically write PLT dot plot on x train. Comma. Uh, for the extent I really need to use my regression model dot predict on my X train data, because I'm just trying to see how the best fit line will come for the training data. Right? And here I can definitely use some color. Let's say the color is red. Now we can use a dark color. That will be good. You can actually see it. And along with that I'll also try to scatter all my Xtrain and ytrain data. Right. So now I'll just scatter this, uh xtrain. And ytrain data. Uh, any default color can be used. And apart from that, we will also be writing plt dot x label. So this will basically my x. Uh. Data set. And finally I'll just say plt dot label. And here this I'm just going to put it as y. Okay. And finally I'll just import it. Now see Matplotlib.pyplot has no attribute label y label I have to use y label. Now obviously when you get this kind of best fit line you're going to get a good amount of error right now let's see how do we solve this problem with the help of polynomial regression. Now first thing in polynomial regression is that what we do whenever we have a polynomial regression, right. Let's say I have some data. Right. Let's say this is my x data and y data right now x data is in degree one. Right. So I can probably take something like this. My h theta of x will be uh beta zero plus beta one into x one. Right now this is what my model is all about right. And obviously with this best fit line we are not able to get good model a good accuracy. So what I'll do I'll say h theta of x and I will increase this polynomial degree. And it will become something like this beta two multiplied by h two whole square. Sorry, it should be x one only, not x two because I just have one input features. So we will first of all try with this. If it does not work with respect to this, later on we'll try to create a generic formula wherein we just keep on increasing the degree of this x one feature. Right? Let's say I want to increase this again. This will be beta two. Then this will be

beta 3X1 of three. Right. So here if I probably just show you right. If I probably show you right over here. See. Okay. Uh, so here we will be using or will be increasing the polynomial degree. Okay. So that is what we are going to do it. Now with respect to all my features, I really need to increase the degree of the features and try to see whether my line will fit or not. Later on, we'll try to create a common function which will do all this specific task. Okay, now how do I do this? How do I probably create a degree for two, three, four? Right. So for that in sklearn we have something called as. So here I'm just going to say let's apply. So here let's apply polynomial transformation. Okay. And here what we are going to do is that we are just going to write from sklearn dot pre-processing. Import. Polynomial features. Right. And finally I will execute this. And here you will be also be seeing that I will write poly. Now here this polynomial features is used for the transformation like to increase the degree. Suppose if I have just x one variable. If I say I want to use a degree of two, that basically means x one square will also be created as a new feature, right? So here I will be writing polynomial features. And finally I will be using degree is equal to two. And here let me also say there is a parameter which is called as include bias. I'm also going to include the bias so that one will also get created. You know first the first variable right. So what do I mean by one. Let's say I'm just going to write this equation. Let's say this is my first equation. Or I'm going to use this I'm going to drop this okay. Let's say my h theta of x is nothing, but it is beta zero plus beta one into x one. Then beta two into x one whole square. So we if I say degree is equal to two that basically means x square is created. Include bias basically means here it will be multiplied by one. Okay. So we are going to take this one. We are going to take x one. And we are going to take x one square for our model training purpose. All these things we will try to show it to you okay. So as soon as I probably transform this you'll also be able to see that okay. Now the next thing is that what I will do after doing this I will say x underscore train underscore poly. And here I will be saying poly. Underscore fit. Fit. Underscore. Transform. And here I will be saying X underscore train. Okay. And similarly X underscore test underscore Polly will be nothing but Polly underscore fit underscore transform okay. On my X test data why do I write only fit underscore. Sorry this should be not fit underscore transform. It should be Polly dot. Fit underscore transform. And this should just be dot transform okay. So probably I've just written underscore. That should not happen. But again I've already discussed why do we difference the difference between fit underscore transform and transform. I just want to apply this techniques in my test data okay. I do not want to make sure that again I create all the features again. So that is the basic difference over here. Why this is done. Because I don't want to know the information with respect to my test data. My test data should be completely new. So whatever technique is getting applied over here, that same technique with that same value of X train will get applied to the X test. So in short, what will happen if I just transform this automatically for this particular feature? Degree two will also get created like x one square for this in the same mechanism. For this also it will get created, but there is a minute difference between fit underscore transform and transform fit. Underscore transform makes sure that the main data validation part from where we are actually trying to transform is my training data. No reference of the test data is basically taken. Okay, so this is super, super important for everyone. You really need to understand this. So let's go ahead and execute this. Now here you can see now if I probably try to execute my x underscore train underscore poly. Here is your data. So this became my one. This became my x one. And this became my x one square right. If you don't believe me open up the calculator. Let's see what I get from the calculator. Right. Suppose if I say -2.7488 okay. Whole square. So it is nothing but 7.5559. Right. Which is approximately equal to this if I probably take up all the data points. So this is my this is my bias. This is my x one feature. This is my x one square right. So degree polynomial degree as two okay. So this is what I have done. And similarly you can also check it for x underscore test or underscore poly. Now

once you have this particular data set all you have to do is that try to apply the same linear regression whatever we did. Right. So I'm just going to copy this because it is almost the same thing. And uh then we are going to probably see it. Now here you can see that since the batch parameter is already included, we have taken linear regression. I have used x underscore underscore poly y train. And then we did a prediction. And right now before the score was what my score was 0.62. Right now you see the score whether the score will improve or not. Yes I think it will improve. Now you're getting 0.9393, which is quite amazing. Now you may be excited. Krish how is the graph getting created? You can obviously see from this model, right? If you probably see one more thing. If I write regression dot coefficient here you'll be able to see three coefficient one is zero, one is 1.5, one is 0.5090. Why? Because I have three features now right before I just had one feature. Right. This is super important to understand, but with respect to intercept I will just be having one feature. So here I'm just going to write. Intercept. I will just be having one feature. Perfect. So this is quite amazing guys. We are able to get it. Now comes the main part. You can obviously do the prediction. You have also seen that with the help of prediction. Also when I'm seeing it is coming as 93%. Now let's take some new data and let's try to plot this or just before this, you know, let's plot this one. So I'll say plt dot scatter and let's plot my x train okay x train comma. The prediction that I did it for uh or I can basically write something like this regression. Regression. This regression is basically my polynomial regression okay dot predict okay. And I am going to predict my x underscore train underscore Polly okay. And once I execute this here you can see this right. See this curve. So this is my best fit line that is coming for all these data points. And if you also want to see the other data points all you have to do is I just write plt dot scatter okay. And here you can probably take or here you can probably take x underscore train. Why underscore train. Okay. Now once you plot this. Now see for all these data points, this is how your best fit line comes. So that is the reason why you're getting such an amazing accuracy okay. Such an amazing accuracy. But now the question arises Krish, what if I make this this degree value as three? Just let's consider okay. So let's see if I make this degree value as three then what will happen. You also need to think about that right. So if I make this degree three will my accuracy keep on increasing again I will copy this. I made it as three and then I will probably just try to see Xtrain first of all. Right. So if I write x train underscore poly what do you think we'll be having. And now I have four features right four features. Now this is nothing. But it is. Just imagine that here you are just trying to find out the cube of that right. This to the power of three. Right. So same thing here you are able to get. Now what I'll do I will apply the same thing and let's see whether the accuracy will increase or not. So here for degree polynomial I'm getting 0.9343 before I used to get how much. Okay just a second for this uh x test Pro x train poly. So here x train poly y train. This is my X train. Yeah I'm getting almost same accuracy somewhere around 0.9343. Right. So uh, before that, uh, what was the accuracy that I was getting over here after applying this? See, the accuracy is downwards here 0.9393. So just a minute difference. Right. But yes, I can definitely say that the previous one was working well. So I may probably just consider degree is equal to two. But sometimes you may be having a data set that will be having uh, too much different type of uh, non-linear relationship. Now let's go ahead and try to see the prediction for the new data okay. And this is super important guys. Prediction of new data. Okay. New data set. Now, first of all, what I'm actually going to do I'm going to probably take a data okay. Take a need data. So as soon as I get a new data, let's say I'm getting this new data with Linspace this, this. And I'm reshaping to 200 comma one, let's say this many data points. I'm actually coming as a new data point. First step I'll do poly dot transform on this new data. So here I will be getting a new polynomial feature. So in short if my degree is two okay I will be getting two features. That is 1X1X1 square. If I probably take three when I was actually creating my polynomial features, then it would have been how many degree of

polynomial would be three. Right now, in this particular case, if I probably go and execute my x underscore new underscore poly here, obviously what you will be able to see, you'll be able to see four features. Right. Because the degree of polynomial is three, one is the bias right. And remaining all you are able to see over here. Right. Very simple. Now what we are going to do we are going to just use regression dot predict. Now this is super super easy super super important. So I'm just going to copy and paste the code plot X new y new x train y train x test y test y new is the predicted for the new data points. And when I'm predicting this, I'm also plotting X train by train and test wider. Once I see this guys see this curve. Amazing, right? I'm actually getting this entire predictions along with the green points here and there and blue points here and there. Right. So this is how we are able to see this. Uh, so here I'm just going to write new uh, this is my training points here. Obviously a new prediction I'll write over here. Right. So this is how we are actually able to see. Right. This is for the new data set. Now see how my model has got trained for this kind of similar data. Okay. So this is uh so super important uh with respect to this. Again you can try with different different polynomial degrees, whichever gives you the best performance. You can basically take it up. Right. So I hope you liked this particular video. And yes this was it. In this video we have focused on implementing polynomial regression. Uh, yes. I will see you all in the next video. Thank you.

So guys, in our previous video we have already implemented the polynomial regression. And here you could see that we could have assigned different different degrees of polynomial degrees over here. And we tested that for which model it is being able to give a better result. Now in this video, what I'm actually going to do is that I'm going to introduce to something called as pipelining concept. Now in pipelining concept, all we have to do is that I will create a generic function in which whatever degree you assign, you will be able to see that what kind of curve it will be fitting to for a new data set. Okay, so for using the pipelining concept, and probably this is the first video with respect to pipelining concept. In the upcoming videos we are going to use lot of uh pipelining concepts as we go ahead. So here first of all, what we are going to do is that I'm going to create a poly regression function here. We initially take the data set. Uh I will not be using transforming over here okay. Then whatever. In this particular function we are creating a degree. Right. This degree is passed by the, uh passed by the front end. Right. So whatever degree you really want to play with the polynomial degree itself, you can actually pass it. So inside this polynomial features I'm going to use that specific degree and say include bias is equal to true. So automatically what it will do it will be able to transform our features with respect to that. And then I'm also initializing linear regression. So every time whatever degree I'm actually providing it will be trying to create a new features. And for that it will try to apply the linear regression. Now both this we need to be added in the form of pipeline one after the other. So first will happen poly underscore features. And the second will basically we are going to apply a linear regression model. And we are going to fit with respect to this particular data. So in order to do that whatever pipeline we have imported right we are going to combine this inside this pipeline. So here you can see the first thing is poly features. You can give this whatever

name you want, but make sure that you provide this in form of key value pairs. So over here should be the objects whatever objects you are initializing. So in this particular case poly underscore features is initialized over here. And linear underscore regression is initialized over here. Right. And then we combine this all we have to do this dot fit. So once we do a fit on Xtrain and ytrain what it will do, whatever degree it is basically done internally, it will transform this particular values based on this degree. Right? And then finally, I can use this particular model to do the prediction for our new data. So whenever I pass my new data, what it will do, the first step of this poly regression will be that it will be creating this particular features by using a specific degree. Right now in this particular case, the degree, whatever degree are assigning it over here with respect to that specific degree, it will get created. Right. And then finally, you can do the prediction on the new data with the help of linear regression, which is the next model in the pipeline. And finally here you can see I'm plotting the data based on my X new and Y parade, new and Xtrain and Ytrain x test and Y test. Okay. Now let's go ahead and execute this. And overall you'll be getting an idea. First let's say I'm going to try with degree one. So with degree one you can see that I'm getting a best fit line. Obviously the error will be high. Now if I do with degree two here you'll be able to see that yes it looks good. The curve is becoming better and the error will also get reduced. Then if I try with degree is equal to three here, you'll be able to see that. Okay I'm able to get a better one now. And similarly with degree is equal to four here. You will be able to see this now as you increase the degree. Let's say if I'm using degree ten it will start overfitting. Now here you can see that it is trying to overfit each of the data points. This green points are basically the green points are my test data points. The blue points are my training data points. Now, as you go ahead in increasing the polynomial degree, it will try to overfit each and every data points. So it is our responsibility to make sure that whatever degree we are selecting it does not overfit. Now, in this particular case, someone asked me should we use 15 or should we use 7 or 6? So here I will just again suggest that we should try to use six because it is not trying to overfit the model. Okay, so this was about the pipelining concept. So here in short what you have done is that we have combined two models that is polynomial features. And linear regression in a pipeline. Right. And initially whenever we take a data, first of all, with the help of this poly features, we will be getting all our features over here. So as soon as I write this polynomial regression, this pipe, this entire pipeline dot fit on Xtrain and Ytrain. In short, what it is going to do is that first step, it is going to create all the features by using this. And the second step is that it is going to do the linear regression fit with respect to the model. So finally here you can see when we do predict also with the help of predict. Also first of all it will try to create this polynomial features. Uh that is all my features. And then it will try to do the prediction okay. So this here you can basically say it is basically doing one thing is that it is the first step. It is basically creating your polynomial features and then fit of linear regression okay. So this is super, super important. We will probably be using this, uh, a lot as we go ahead. Okay. So yes, try to try it out. Try it with different different uh, parameters and degrees. Uh, and then try to see that which model is actually performing. Well fine. I will see you all in the next video with some more amazing examples.