In this video, we are going to discuss about a new machine learning algorithm, which is called as Support Vector Machine. Now, this particular machine learning algorithm helps us to solve both classification problem and regression problem. If we are solving a classification problem, we basically say this as support vector classifier. That is, as we see in the case of regression, we basically say support vector regression. So I'm just going to mention over here we're going to discuss about this two algorithms. First will try to finish up support vector classifier. And then we will try to go ahead and see which is called as support vector regression. Okay. Now if I talk about support Vector machine, first of all, it is very, very important that you really need to understand logistic regression. So if you have skipped logistic regression, if you have come over here directly, I again suggest you please go through the logistic regression, maths, intuition and everything. So to begin with, uh, again let's consider that I have two axis. And here let's see that this is a uh in in logistic regression. What we do suppose if we have some two categories of point, let's say it is a binary classification. And similarly if I have another category over here in logistic regression our main aim is basically to create a. Best fit line, that is, a line between these two particular points, such that this line is able to clearly separate them. And this is what we basically do with respect to a logistic regression, right? We try to create a straight line in the case of a 2D plane. Straight line. In the case of a 2D plane, let's say if I have a 3D plane. So suppose if I have coordinates like X1X2. X3. Right? Let's say these are my three features. And let's say these are some of my data points, which is over here. And my other data points are somewhere here. Now in this particular case, what we used to do in with the help of logistic regression is that we used to create a plane. Right. So this is basically a 3D plane. And with the help of this 3D plane, we were trying to categorize this particular data points. And whenever a new test data. Suppose if it comes over here, obviously we know that it is below this particular plane. So we would use to categorize this into this category. That is what we were trying to do in uh, logistic regression. Again, I'm just going to write it in logistic regression. Similarly with respect to n dimensions we basically have to create a hyper plane. Okay. Now with the help of support vector machine. Support vector machine. Now let's just discuss about the support vector machine. Now in Support Vector machine what we do is that. Specifically with respect to support vector classifier. Let's consider first of all classifier. And here I'm going to take an example with respect to as we see. Now in this, let's say that I have this two axis. That is x1 and x2. We are just trying to understand the geometric intuition, what we are planning to do with support vector machines. Let's say this is my x and y coordinate, and let's say I have some other data points over here. And I have some more data points, which is over here. Now in Support Vector Machine. What we actually try to do is that again, we also create a plane or a best fit line along with this, along with this super important suppose let's say that, uh, this is my best fit line in the case of, uh, two dimension. In the case of three dimension, it becomes a plane 3D plane, and in the case of n dimension it becomes a hyper plane. Now, what we try to do over here in Support Vector Machine along with this best fit line, or let's say in the three dimension, I can basically call this as 3D plane. I also try to create two more lines, and this lines are basically called as marginal plane. Okay. These are called as marginal plains. So along with the best fit line we also some create something called as marginal plains. So this is also my marginal plain. This is also my marginal plain. And this marginal plane. First of all, let's say that this distance combined is something like D this both should be equidistant, and this marginal plane should be created in such a way that the distance between this marginal plane should be maximum, okay. Should be maximum. So let me again repeat with respect to support vector machine. If I'm just considering as we see, what we are trying to do is that we are trying to create a best fit line and in this best fit line in the case of 2D plane. So if I'm just creating a two dimension in the in the case of two dimension categorization, right, I really need to create this best fit line. Along with this, I have to create a two marginal plane, and this marginal plane distance should be maximum. So suppose let's say the same points if I try to populate it over here. Okay if I try to populate it over here? Let's say I have this points over here also. Okay. And I'm just going to draw the same points like this. I hope I'm drawing it right because there may be some mistakes, but approximately it will be equal. I'm just drawing the same coordinates. Okay. Now suppose, let's say now my best fit line looks something like this. And along with this, I try to create my marginal plane, which will pass through the nearest point from here, and similarly it will pass to the nearest point from here. Right. And this is my marginal plane. Now out of this two. Right, which is the marginal plane that we should select, or which is the best fit line that we should select along with the marginal plane. Obviously you'll be seeing this. Why. Because over here the distance between this marginal plane is less. Let's consider this d dash. And this is d. We know that from this d is greater than d dash. So what we do is that we select this best fit line along with this marginal plane for doing the classification. Okay classification. Now this is the first thing. The second thing that you really need to understand over here is that with the marginal plane passes through one of the nearest point from this particular plane, let's say this is my nearest point. And similarly over here, this is my nearest point. This points are basically called as support vectors okay. This point are basically called as support vectors okay. So same with respect to this specific point. This both the point are basically called as support vectors. Now. In short, what is the aim of the support vector machine is that we really need to create a best fit line. Along with that, we need to create a marginal planes so that we'll be able to categorize the point very clearly, or classify the point very clearly. What happens is that suppose if I get a new test data point, you'll be seeing that this belongs to this particular category. So this will get assigned to this particular category. Suppose if I get any new point over here, this will get assigned to this particular category okay. Now this is a super important thing that you really need to understand. Now, one more thing that I really want to conclude from here is that suppose let's say if I have a 3D, if I have a 3D points, let's say I have a three and a three dimension, it's X1X2X3. And let's say and this is also used in multi-class classification. Also let's say I have this two categorization of coordinates okay. Now here with respect to the 3D coordinates. Also, we have to create a 3D plane. Along with that, one more plane will be basically by marginal plane. It will be created like this. And another marginal plane which will be below this. It will be created like this. Okay, so this is how what is the differences between the 2D and the 3D plane? In this particular case this will be my marginal plane. This will also be my marginal plane. I have to make sure that this distance should be maximum. So this is what is the problem statement that we are trying to solve. And uh, yes. Uh, in my next video I'm going to, uh, explain you about what is soft margin and what is hard margin and support vector machine, a super important topic. And after that we are going to understand the maths behind it, like how we are going to create this best fit line along with the marginal planes and try to solve this particular problem statement. So yes, I will see you all in the next video. Thank you.

Now you have understood what is the main nerve of the support vector machine, specifically for a classification problem where we are using support vector classifier along with the best fit line, we also try to find out a marginal plane in such a way that this particular distance should be maximum. Okay. Now let's discuss about this important topic, which is called as soft margin and hard margin. Now usually see in this particular best fit line that we are using and marginal plane we are using, it is clearly separating all the points. But in the real world scenario we will not have this much clearly separable points. You know, there will be lot of overlapping. You know, suppose if I have some points in this two dimension, there will also be some overlapping. Right now in this kind of overlapping case, I cannot clearly separate, you know, just by using a best fit line and marginal plane. In short, there will definitely be some amount of errors. Now when I say some errors, this points may be coming somewhere here. This points may be somewhere coming over here right now. Obviously we cannot get a clearly perfectly best fit line along with the marginal plane to split the points. In this particular case, we will have some kind of errors. So this situation is basically called as soft margin. Okay. Now whenever I say soft margin that basically means some amount of errors will always be there. Because obviously in a real world scenario we may get lot of overlap points. Similarly, in this particular scenario here you can see that we are clearly separating this. Right. So this specific thing is basically called as here since we are clearly separating all the points and we don't have any errors. This scenario is basically called as hard margin. In hard margin we don't get errors in soft margin. We definitely get some amount of errors because there is a lot of overlapping. Now in the next video, we are going to discuss about what is the math's intuition. How do we draw this best fit line? How do we make sure that we come up with this marginal plane, or what equation is basically used for this particular line, you know, so we will again be revising all the, uh, you know, the equation of a straight line, the distance from a point to the plane, all these things again will be revising in our next video, and then we'll try to understand the maths in depth intuition. So yes, I will see you all in the next video. Thank you.

In this video we are going to understand the support vector machines in depth maths intuition. In our previous video we have already seen what is the main aim. We have to probably create the best fit line along with the marginal plates. Now let's consider I have two coordinates x and y. And obviously with respect to this 2D plane, I need to create a best fit line. Now let's consider this is my best fit line. And this is basically given by the equation w transpose x plus b is equal to zero. How this came earlier in my previous video, I've already explained that the equation of a straight line is given by x plus b, y plus c is equal to zero. If I can also write this as W1X1 plus W2X2 plus c or b is equal to zero. This is in turn nothing but w plus x is equal to zero w transpose x plus b is equal to zero. And if my line passes through origin then b will be equal to zero. So finally we can basically write w transpose x is equal to zero right. But here my b is not passing through the origin. So let's consider this is the equation of this particular straight line. Now one more thing. What w is over here w will be basically perpendicular to this particular line right. So w is a vector that is perpendicular to this particular line. That basically means it has a 90 degree angle. And already we have also seen one very very important thing is that suppose let's say that we have a specific coordinate here. And let's consider I'm going to take a coordinate like minus four comma zero okay. Now in this particular coordinate if I take this particular coordinate and if I try to find out the distance between this two coordinate right now when I try to find out the distance between this two coordinate, let's say this is my d dash. And let's consider one more coordinate. Let's say this is four comma four comma three. Right. And this is my another coordinate. Let's say this is s this is S dash. And if I really want to find out the distance between this to this hyperplane, what things we specifically used to do, we first of all used to calculate the angle between this particular vector that is w and s. And obviously here you can see that the angle is greater than 90 degree. So we have already discussed this previously. Whenever this angle between this particular vector w and s which is given by this coordinates minus four comma zero is greater than 90. Then with respect to this particular point and the distance between this point to this hyperplane will be negative okay will be negative. So any points that are there below this plane right below this best fit line. And if we try to calculate the distance it is always going to be negative okay. This is super super important to understand okay. Similarly with respect to the coordinates that are above this particular point right now in this particular coordinate, if I try to see or if I try to plot this vectors w and x here, you will be able to see that I'll be getting an angle that is between 0 to 90 degree, right. Whenever we have this kind of scenario, the distance between s dash to this particular coordinate or to this particular plane, or with respect to any points that are above this plane, the distance will always be positive, right? So the distance will always be negative below the plane. Right. And distance is going to be positive above the plane for all the points. Right. So this is the pure understanding that you really need to find out that every points that are below this, it is going to be negative. If I try to calculate the distance between this two plane and similarly all the points that are above the plane, uh, the distance between those specific point and the plane is going to be positive. So this we have already discussed, uh, in finding the distance of a point to a plane. Right. So please make sure that if you are not able to understand this, check out that specific video. Okay. Now till here. I hope everybody has understood what we are actually trying to do. Okay. Uh, we are just trying to calculate the distance and we are trying to find out. Now let's quickly go ahead and, uh, do one very important thing. Let's say I'm just going to draw this specific point again. So let's say. I've clearly made you understand that. Suppose if this is my best fit line, wherein my Y is basically perpendicular to this plane, and this is given by the equation w transpose x plus b is equal to zero. So I said that any point below this plane. And if you're trying to calculate the distance from here to probably this, it is going to be negative, right. So here what we can do is that we can create one more plane. So let's say this is my plane. And this plane is basically called as marginal plane okay. And this plane passes through one of the coordinates over here. Similarly on the right hand side, what we can do is that we can create another marginal plane. Okay, another marginal player wherein my other points that are the other coordinates and obviously from this particular coordinate to the distance between the marginal plane, it is always going to be positive, right? Positive. So let's say if this is my nearest point from this plane. So these are my support vectors. These are my support vectors. Right. And this is what we have to basically do. And obviously you have understood any points that are below this plane are going to be negative with respect to distance from this point to the marginal plane and sorry to the hyper plane. And similarly from this particular points to the to the hyper plane, I'm always going to get a positive, uh, positive distance value, right? Positive and negative. This is super clear. Now can we provide some equation to this specific line. So here I'm just going to write w transpose x plus b is equal to plus one. The reason why I'm writing plus one because over here you can see that from this particular plane. Any time I go up right it is going to be positive. It is going to be plus one plus two plus three plus four plus n plus some values. Right. We can also replace this with plus k. But in many research paper they usually use plus k or plus one. So here I'm going to basically write it as plus one okay. Similarly this particular line that you see below the plane I can also mention this as w plus w transpose x plus b is equal to minus one. So obviously our main aim is basically to make sure that this distance is maximum. That is the distance between this marginal plane and this marginal plane. So in order to compute the distance, what I am actually going to do is that we are just going to subtract this two values. So let's say this is uh this is probably uh x two and x one. Right. So in short what I'm actually going to do, I'm just going to basically write w transpose x one plus b. And similarly I'm going to write w transpose x two plus b. Here I'm going to use plus one. And here I'm going to use minus one. Because this plus and this minus is there. And now if I really need to compute the distance between them and that distance needs to be maximum with respect to all the other best fit line or marginal plane that we are creating. So here if we try to solve this linear equation this will be minus minus this will get cancelled and this will be w transpose x one minus x two which will be equal to plus two right. So which will be equal to plus two. Now you really need to understand one very important concept which is called as unit vector. If I really want to calculate the unit vector of any vector itself right unit vector basically means where the magnitude of that vector is one magnitude of the vector is one. Right. Now, if I really want to show you that how this magnitude of the vector will be one, it's very simple. We just divide w transpose with the magnitude of w. Okay. So once we do this on both the side here we are basically going to get it as unit vector. And it is always a good idea that we need to come up with a unit vector, because all the points will be normalized within that value between 0 to 1. So guys, this arches that I have specifically got that is two by magnitude of W, you know. So this is basically going to be my cost function. Right? Because if I take this two by magnitude of W, I really need to maximize this value. Why we need to maximize this value. Because this is basically the distance between this two marginal planes. And it needs to be maximum. Now this maximization will be doing by changing the values with respect to w comma b okay. And on top of it I would definitely like to write another constraint okay. And this is nothing but distance between marginal planes. Distance between marginal planes. Along with this I'm going to write a small constraint and let me write that particular constraint. And this constraint is with respect to all the truth values. That is y of I. My true output y of I will be plus one. When w transpose x plus b is greater than or equal to one, right? Similarly, y of I will be minus one when w transpose x plus b is less than or equal to minus one, right? These are for all correctly classified points okay. Correctly classified points. Now whenever I say correctly classified points what does this basically mean? This basically means that whenever all the points that are above this w transpose x plus b is greater than one, I am going to basically categorize this as plus one and this all points I am going to categorize as minus one. So this will basically say about my truth points okay. About my true points. Whenever in our data set, whatever output feature I specifically have that all true points will be there. So w transpose x plus b greater than one will basically be talking about the positive points. And w transpose B is less than or equal to minus one. Whenever I find out with respect to any points, we are going to label it as minus one. Okay. Now let me finally write one important thing for all correct points. We know that if I multiply. So here I'm just going to add one more constraint. If I multiply w of I multiplied by w transpose x plus b, you will always be getting this as greater than or equal to one for all the correctly predicted points. Right? Because see if you are getting w transpose x plus b is greater than equal to one if you multiply. If it is greater than equal to one, that basically means our output is plus one. So if I multiply both these two, I'll get positive okay. Similarly, if my w transpose x plus b is less than or equal to one if I multiply with minus one for the correctly points, I'm also again going to get it as greater than or equal to one. Okay, so this is my entire cost function. And I really need to maximize this specific cost function by changing w comma b. This is the constraint with respect to like for all the points that are above the plane, I'm mentioning it as plus one and all the points that are below the plane. I'm mentioning it as minus one. Okay. And when I probably multiply plus one with w transpose x plus b, how always I'm going to get greater than or equal to one for all the correct points. Now let me fine tune the cost function a little bit. And this part I will try to show you in my next video. Thank you.

Guys. Till now we have understood the cost function with respect to a support vector machine, specifically with respect to support vector classifier. Here our main task is to basically maximize w maximize two by magnitude of w by changing w comma b. And this will basically increase our distance between the marginal plane. And I've also mentioned two constraint. One constraint is for the truth values. Whenever w transpose x plus b is greater than or equal to one, we are going to label it with plus one. And whenever w transpose x plus b is less than or equal to one, we are going to label it as minus one. And one more constraint is that for all the correct points, whenever we multiply w of I with w transpose x, we are going to get always greater than equal to one. See. In this case, also, when we multiply minus into minus we are going to get this. Now what I am actually going to do I'm just going to little bit fine tune the cost function. Now whenever we write something like this. See whenever I write my cost function which is like maximize w comma b and this is my cost function, I can also write this as minimize. By changing w comma b magnitude of w by two. Right. And both these things are almost same. Right. Both these things are almost uh, I'll say exactly same because, uh, in whenever we talk about machine learning, most of the time we try to minimize the cost function, right. So finally, I can definitely write my cost function of SVM specifically related to SBC support vector machine. And here I'm just going to write it in bracket SBC. I'm basically going to write cost function is equal to. I have to minimize by changing w comma b wherein my magnitude of w by two. And this is my cost function obviously. But in the real world scenario guys, see I can definitely use this as my cost function if all my points are clearly separable, right? Let's say if all my points are clearly separable, then I can definitely create a best fit line. And along with this, I can create a marginal plane which can pass like this and which can pass like this. But in a real world scenario, this kind of scenarios will not exist. There'll be a lot of overlapping. You know, some points may be here, some points may be here, and some points may also overlap somewhere here. So we should add some more amazing hyper parameter on top of this particular cost function. So the first hyper parameter that we are basically going to add over here is something called as c j. Okay I will mention what exactly c j is c j is a kind of hyper parameter. And along with this I'm going to also make sure that I'm going to add this summation of I is equal to one to n, and this will basically be a symbol which is called as eta. Okay. So eta of I and this we are just going to find out the summation of ETA off I. Okay. And this specific cost function, this, this hyper parameter that I'm trying to add is something called as hinge loss. Hinge loss okay. Now let's understand this component. What is C of j or C of I. Okay I'll not write it as C of j because I'm not using j anyway. So this is basically c of I. So c of I basically says that how many how many points. We want to avoid misclassification. Super important guys. See, this is a kind of hyper parameter. And this basically indicates this is a kind of hyper parameter. And this is basically indicating that how many points we can probably consider to avoid misclassification. Let's say I may say that my C of I is five. That basically mean means that I can basically ignore five misclassified points okay, five misclassified points. Because obviously it is not true that I can get a exact best fit line along with the marginal plane. There will be a lot of overlapping, but I'll say, okay, if there are five misclassified points, it's okay. We can consider that. And let's consider and let's go ahead with that kind of hyper parameter. And then also we'll try to find out the best fit line. Now in this particular case let's say over here I'll be seeing that one misclassified point. Is there two. Are there three. Are there four. Are there five, six seven. So within this I'm actually saying that okay, fine. If my C of I value is six I'm also going to consider okay fine. It's fine that I have six misclassified points. But I am again going to consider this as my marginal plane, this as my marginal plane, and this as my best fit line. It's okay, let's six errors be over there okay. Now let's go ahead and understand what is this ETA? ETA basically says it is the summation off. Because here I'm doing the summation off right from I is equal to one to n. So here I'll say summation off the distance. Of the incorrect data points. Incorrect data points from the marginal plane. From the marginal plain. So here, in short, what I'm doing is to basically computes the distance between this point to this marginal plane, this white point to the above marginal plane, this yellow point. So this are two hyper parameter. One is that how many number of misclassified points I can have. And the second parameter is that okay this much distance is allowed. If I probably do the summation of all this distance, that much distance is basically allowed. Okay. So this is what a cost function of SVM looks like. And this is not related to hard margin. But this is the cost function that is related to soft margin okay. So with the help of this particular cost function I will try to minimize this. Along with this, we will be setting up parameter with respect to C of I and eat off I. And this specific hyper parameter is called as a hinge loss like how we have in logistic log loss. Similarly we have over here as hinge loss okay. So I hope you have understood about how does an SVC basically work. And with the help of this, we will be able to create our best fit line along with the marginal planes. So this was the entire in-depth maths intuition behind Support Vector Machine which is specifically Support Vector Classifier. In the upcoming videos we are going to understand about support vector regression. So yes. Ah, I will see you all in the next video. Thank you and keep on revising this mathematical concepts. Thank you guys.

In this video we are going to discuss about the support vector regression machine learning algorithm. In our previous video, we have already seen support vector classifier and we have seen what is the cost function, what is the constraint. And we have also learnt about a new loss function which is called as hinge loss. Now in this specific hinge loss you know that we have two parameters, that is C of I and E to fight. Right. And our main aim is basically to reduce this cost function by changing w comma b. Right. And uh, if we discuss about the regression problem statement, the regression problem statement will also be working like this. So first of all we really need to find out a best fit line. Along with that we need to find out marginal plane. And we have to make sure that this distance is maximum. Now here I'm going to take a simple problem statement. Over here you can see the uh okay. Let me just change this parameter a bit. So first of all I'm going to basically write in the y axis. This is the price of the house. And this is the size of the house. And we really need to create an SVM or SVR model that is support vector regression model to basically predict based on the size of the house, what is the price of the house? Okay. And over here why? This is a regression problem statement because all your output, your output feature that is your price is a continuous value okay. So this is a regression problem statement.

Now to begin with whenever if I really want to construct this best fit line along with the marginal plane, I can definitely write my cost function which looks something like this. So the cost function will be. Main aim will be to minimize by changing w comma b, and this will be magnitude of w by two. And this from where do we get. We had actually derived this over here. You can see it okay. Now once we write this cost function let's go towards the as we see and see along with the cost function I had written some kind of constraint okay. And the constraint was that y of I multiplied by w transpose b uh plus b should be greater than or equal to one. This is super, super easy. Now if I see in the support vector regression see this is super important okay. Our main aim should be to basically create this best fit line along with the marginal plane in such a way that my, uh, distance between the predicted points and the real points. If I do the summation, it should be really less. Okay. So let's consider that this particular line I am denoting as w transpose x plus b okay. We denote this particular line with w transpose x plus b. What about this line that is this top marginal plane. Now here, if I probably compare from the best fit line to this marginal plane, we are in short adding some value right this value. Let me denote it as epsilon. And this below line we are subtracting a value which is epsilon. So here I can basically write a equation which is called as plus epsilon w transpose x plus b plus epsilon. And this will be minus epsilon. So if I say what is epsilon. This is nothing but my marginal error. Marginal error basically means marginal error or margin error basically means the distance between this point to this point. Similarly, the distance between this point to this point and both this distance will be almost same. Okay. It will not be changing over here. This distance will be almost same. So I am using this three equation to find out the best fit line. I'll be using w transpose x plus b. This will be w transpose x plus b plus epsilon and this will be w transpose x plus b minus epsilon. Now if I if I'm probably creating this for a regression problem statement, I have to make sure that most of my points data points over here that are present in this problem statement are covered within our marginal plane. So the first constraint that I'm actually going to write over here, that is the distance between y of I and w of I x of I w of I x of I is nothing. But this is my points with respect to the predicted points over here, and y of I is basically my truth point right this this truth point. So if I probably try to find out the difference right, this should be less than or equal to epsilon. If it is less than or equal to epsilon. That basically means our model is really, really performing well. So most of my data points over here also you can see it is less than this particular epsilon. That basically means it is falling within the marginal plane right within the marginal plane. So this should be one of my constraint whenever I am actually creating my best fit line and the marginal plane for a regression problem statement. But there is one simple catch. Okay, here you'll be seeing that okay, most of the points are falling within the marginal plane. But what about the other points that are falling away from the marginal plane. So here we will again define some deviation. Some deviation basically means that this deviation can probably used as an hyperparameter to construct our best fit line and the marginal plane. So here I'm going to make some changes with respect to our cost function. First of all I'm going to add a parameter which is called as c of I. So this will be summation of I is equal to one to n. And here I'm going to basically use as eta of I. Okay. Now this same thing I have also used over here right. I have used over here like this is my hyper parameter. We basically say this two parameters are nothing but the hyper parameter. And we can also say this as my hinge loss okay. Now let me explain what is eat of I in this particular case okay. Eat off I in this particular case. Now as I said that most of my points will be falling within this marginal plane. If it is falling within the marginal plane, that basically means the difference between the real point and the predicted point will always be less than or equal to epsilon. Okay, but what about this specific points that are outside this marginal plane. So here you can see I have some points over here, here here that are outside the marginal plane here. Also I have some points that are outside the marginal plane. So for this also if I really want to I should also be considering this points when we are constructing our marginal plane along with the best fit line. So what we do is that we define one more parameter which is called as eta of I okay, eta of. I basically says that okay, this will be a deviation from our top marginal plane and from our bottom marginal plane, that okay, if we are also getting this much distance, we should also consider this as a hyper parameter to construct our marginal plane along with the best fit line. So this value is basically my e of I. The distance here is also eat off I with respect to the distance. Okay, now this is super super easy till here. I hope everybody is able to understand. Now whenever I try to do the summation off eat off. I now see this is super super important. If I try to do the summation off, eat off I. That basically means I'm going to find out the summation or I'm going to find out the difference. Just just let me rub this quickly. So I'm I'm in short, finding out the distance between this point and this particular point. So this will basically be giving my ETA of I for this point. Similarly, if I try to do the summation of the distance between this point to this point, I'll be getting different ETA Phi. So I'm just going to do the summation off all these values. And we are going to also use the hyper parameter while probably constructing our marginal plane and our best fit line. So we are going to do the summation of all these things with respect to the ETA Phi. Okay. So this was it. So what I am actually going to do after this is that I'm also going to change my constraint. So this should be epsilon plus eta off uh epsilon plus eta right. So this constraint will now change like this. The reason is that now we are just not considering epsilon. Epsilon is this specific value the distance between this to this. But when I talk about ETA off I am basically trying to find out all the points, all the distance between the points that are above the marginal plane and this specific predicted points on the marginal plane. So this both will be specifically used in order to, uh, you know, fine tune our model with respect to the SVR. Okay. Super, super easy. Now what exactly C is. Okay. C is again, one kind of another hyper parameter. Okay. Hyper parameter. And one more point that I missed. Guys, this is obviously a kind of loss function right. It can be any loss function because here we are trying to find out the difference between the true point and the predicted point. Okay. Now with respect to C we'll just try to I'll just try to create a small diagram to make you understand the relationship between uh c and loss okay. See and loss function. Or let me just write this something like this. Now you know that when the C value will keep on increasing, the loss function will keep on decreasing. Okay. That is the relationship that we will be able to find it out, the relationship that we will be able to find it out that when the C value will keep on increasing, then here you will be able to see that my loss function will be decreasing. So just let me know how the curve will be. Take this as an assignment. But uh, if you know this, the curve will look something like this. Okay, so as the C value keeps on decreasing, the loss basically keeps on decreasing. Okay. Let me just create it in a good way so that you will be able to find it better okay. So this is what is the relationship between C and the loss function. But yes I hope you have understood with respect to uh regression problem statement. And this I will show you in practical like how you'll be able to find out the relationship between C and loss function. But, uh, here you can definitely see that in the case of a regression problem statement, you use something like epsilon. Here epsilon is nothing but your margin error from the best fit line margin error and eta eta of I is nothing but the error above the margin, error above the margin. And this both. The parameter can be used to basically create our best fit line along with the marginal planes. Right. So these are some of the hyper parameters with respect to the support vector regression. So I hope you have understood this particular video. If you have not, please make sure that you revise. Uh, if you have any queries, you can definitely put down in the comment section of this video. So yes, this was it from my side. Thank you. And I'll keep on rocking. Keep on learning. Thank you.

Hello guys. In this video we are going to discuss about support vector machine kernels. Right. So we also see it as SVM kernels. Already in our previous video we have seen support vector classifier, support vector regressor. We have understood about the hinge loss and many more things. But at the end of the day, what is the major aim of Support Vector Machine is that we really need to find out a best fit line. Along with that, we need to find out marginal planes. So these are nothing but marginal planes, and it should be in such a way that we should be able to efficiently separate both these points. Suppose in this case, which is a binary classification problem, right? Binary classification problem, and for this we specifically use SVC, that is support vector classifier. And since we have to create this best fit line along with marginal plane, we basically say this variant as linear SVC. Okay. So we also see it as linear SVC. Linear SVC basically means we basically create a straight line. Along with that we also create a marginal plane. Now what if you have some data points and those data points looks something like this. Okay. So let me just uh, show you some of the data points. Let's say I have two dimensions that is x one and x two. And let's say I have some of the features which looks like this okay. Now, in this particular case here you will be able to see my features. You know, it may be in this way. And the other feature which is my output category will be in this way. Now, in this particular scenario, if I really want to use this linear SVC and try to create a best fit line, let's say this is the best fit line. Along with that, let's say I create a marginal plane which is like this. Obviously for this kind of scenario, linear SVC will not work, right? Why? Linear ABC will not work because all these points are overlapped. It is not clearly separable. Right. So in during this scenario, my accuracy will definitely go low if I probably use a linear SVC uh model. Right. So in this particular case, obviously my accuracy will be low. My error will be high. Right. Just by seeing this you can see that efficiently. It will just be able to divide 50 percentage of the points. Right now during this scenario, can we find out a way to solve this. And that is where whenever we have this kind of data set or this kind of scenarios where my data points are not linearly separable, we can definitely use something called as SVM kernels. Okay. We will discuss about what are the different types of kernels later on in the upcoming video. But let's understand what this SVM kernel will do. Okay, so let me again draw this points again for you. So let's say if my data point is something like this. Okay. I have a lot of data points which is present over here like this. And obviously this is not linearly separable. This is just overlap points and distributed in this specific way. Now for this particular data points I'm just taking as an instance as an example okay. Let's say this is my x coordinate and this is my y coordinate. Now with the help of SVM kernel we are just going to apply some transformation on this data set. This is super important to understand some transformation. Once we apply this specific transformation then you will be seeing that the final outcome now because of this transformation, transformation basically means we will be applying some mathematical formula. Mathematical formula on this specific data set. Okay. It can be. We'll discuss about it. What are the different formulas we'll be using. There are different types of SVM kernels. Now what this will do is that this whatever transformation we are applying on this data set, this will create. Right now it is in two dimension, right? Let's say it will create a three dimension data point. And right now here you can see there are white points. There are red points. Right now this white points will look something like this. And this red points will probably look something like this. Okay. So what exactly has happened over here? Nothing.

It is very simple. Here you will be able to see that the data points that were present in 2D, that is two dimensions, has been now converted into three dimension. And how did it how it got converted to three dimension by using a simple transformation technique. Now what is this transformation? There are many which we will be learning, but just understand the basic idea over here. Now this will be x, this will be Y over here in this three dimension they will try to create an additional axis which is Z right. In this particular axis Z right. And now you will be able to see that this points because of this transformation is clearly separable. Right now. What you can do is that with the help of linear SVC, with the help of linear SVC, you can probably create a plane. And this is nothing but a 3D hyper plane. Along with that, you can also create a marginal plane. And then probably separate this all points. So this is what we are specifically doing with SVM kernels. We are making the data set. We are basically applying some transformation on this current data set, which is not at all linearly separable. And we are trying to convert this into some other dimension because of this transformation. And during this scenario here you can see that data points are clearly separable. So we can then create our best fit plane along with the marginal plane and probably separate this data points. And obviously during this particular scenario, if you are able to do this the accuracy will definitely go higher. Now what are the different different transformation techniques or what are the different SVM kernels? We will discuss about it. Okay. But let me just make you understand how this I definitely want to give you some basic example of some mathematical formula that we may be probably able to apply on a specific data set. Okay. Uh, so let's go ahead and see one basic example. And later on we'll try to discuss about all the types of kernels okay. So let's consider I have a data set. And this specific data set is just in one dimension okay. A basic example one dimension. Let's say this is my data points. Here is my zero in the x axis. So this is my x axis. And here I have a lot of data points let's say. Like this. I have data points and I have another categories of data points which looks like this. And then I have one more similar category of data points like this okay. Now here also you can see that if I really want to apply a SVC in this way, you can see that my data points is cleanly separable over here, over here also. But with the help of SVC, I can only create one line. One separation line, right? Now obviously if I separate it from here, still this many number of data points will be clearly misclassified. Because whatever is below this line, it will be considered as yellow points. And whatever is above this line, it will be basically considered as this orange points. So obviously we cannot solve this particular problem statement which is in one dimension, all the points in one line. We cannot solve it with the help of linear SVC. Now let's see how we can solve this okay. Now what it will do is that let's say I'm going to apply some transformation. Now this specific transformation is super important. Now what I'll do in this transformation is that I will create a new axis y. And here I'll just put y is equal to x square. Now when I apply this specific formula what will happen I will get a new axis y. This will be my x axis okay. And let's say this is my x and this is my y. Initially my points were something like this. Right. Let me draw this point again 1234. And then I will probably draw this points. Initially, my point was like this. Right now when I apply this transformation technique, then here I'll definitely have different different y values. So here you'll be seeing all this values, all these values. When we do x square you'll be seeing we'll be getting this kind of coordinate system okay. And then I will probably get yellow. And then probably I will get orange. Write something like this. Now, as soon as I apply the transformation here, you can see that this 1D got converted into two dimension, right? I just use y. I created a y axis. But what will be the advantage when we have this kind of data points? Now what is the advantage when we were able to convert this one dimensional to two dimension? Here you will be seeing one very simple thing is that now you can probably create a best fit line with the help of linear SVC. Now I can basically solve this with linear SVC and probably get my accuracy quite high. Now it is clearly separable, right? Because I was able to convert my data points from one dimension to a higher dimension. Now all I have to do is that just create a best fit line. Along with that, just try to create a marginal planes, right? And then you will be able to see that I'll be able to divide this particular point that is orange and yellow in a very easy way. Right. So this is the entire idea behind, uh, the transformation and obviously this kind of transformation in many of the scenarios we'll be using SVM kernel. Now, what are the different types of kernel we'll be discussing going forward? First of all, we'll be discussing about a kernel which is called as polynomial kernel. Then we'll also be discussing about RBF kernel. And there is also another kernel which is called as sigmoid kernel. The major thing that you really need to understand is that what is the transformation formula? It is basically applying in order to create a new dimension. Similar thing. Uh, like if you have a two dimensional point, we can also convert this into three dimension by just applying a transformation formula. Uh, and probably for that I, we may use RBF kernel. So different different techniques I'll be showing you with some different, different examples which we will be seeing. And uh, as we go ahead, uh, this kernels will be quite interesting because this is also asked in a lot of interviews. Okay. And it is quite powerful guys. Quite quite powerful. Okay. So I hope you have understood this. Uh, yes. I will see you all in the next video where I'll be discussing about different types of kernel. Thank you.