Audio file

Your Recording 22.wav

Transcript

00:00:02 Speaker 2

But let's consider now a data set that's slightly more complex.

00:00:06 Speaker 2

Let's use our car example that we started our linear regression example with. We know that age has a distinct effect on the car sale price. We know it's quite a big impact as well. Newer cars will achieve a higher price than older cars.

00:00:18 Speaker 2

But let's introduce more input features, more attributes of the car. So things like color, things like whether or not we have a sunroof, the mileage, how far that car has been driven, whether or not the car has an alarm.

00:00:28 Speaker 2

We're starting to get more features, so now I've got 5 input features. I'm thinking in five dimensions. Again, it's quite hard for me to imagine beyond 3 dimensions, but mathematically we can absolutely express that.

00:00:38 Speaker 2

But when I think about this, I can think that certain 1 certain input features will have a bigger impact than others. For example age. We know that a very new car would achieve a much greater price than a very old one.

00:00:49 Speaker 2

But that might not be as obvious with something like color. Color will have an impact to the price. Something that's maybe bright pink with a green stripe, it might not be as desirable as a car that's just a regular color like red.

00:00:59 Speaker 2

But there will be a pattern that determines price, that it just might not be as big an impact. It might not move the needle, as they say, as much as one of the other features. So these different features, OK, those different input features will have different weightings in terms of determining what price that car will achieve. How important is sunroof? How important is mileage? Important is alarm to achieve a particular price. Now, before we can let the model do its work, ultimately the model is going to be doing some numerical calculation. We've seen linear equations like F of X equals.

00:01:29 Speaker 2

WX plus B. That's math. Math works with numbers. The color blue, the color red, it's not a number. Whether or not a car has a sunroof or not is not a number.

00:01:37 Speaker 2

So if we want our model to really understand the relationships better so it can create more accurate predictions, then we need to have a way of turning categorical data into numerical data. Now we are going to see how to encode later on and this is where we could take something like color and just assign a numerical value for red or blue or black and we substitute that into our data set instead of the word blue, red, black, yellow, white and so forth. That way it can form part of the equation that the model is building. Same for sunroof, we could literally have a value of 0 and one to represent having a sunroof or not.

00:02:07 Speaker 2

And this is one of the subtle ways that we need to prepare our data for training rather than just taking a raw CSV data set and training on that. We need to have a little understanding as to how a model works so that we can make the adjustments to the source data so we will get the best possible results out of the model. Now as we know, certain features will carry greater weighting on the ultimate price. Mileage will have a far bigger impact on the target price than say colour. So something the 100,000 miles would achieve 1 price, something on 20,000 miles would achieve a far greater price. What's the impact of it being a Violet colour versus orange?

00:02:37 Speaker 2

Minor, maybe it made two $300 difference, we don't know, but we know mileage would have a far greater impact. So what I want us to think about is that each one of those input features age, color, sunroof, mileage, alarm. Let's give each one of those features a label X1X2X3X4 and X5. So age we'll call X1. Color we'll call X2. All the way through to alarm, we'll call X5.

00:02:58 Speaker 2

No. Why am I doing that? I'm doing that because for each feature.

00:03:02 Speaker 2

I'm going to have a coefficient, I'm going to have a waiting that will determine what impact that feature has on the ultimate target variable, the target price. So weight 1 will be the coefficient of the multiplier for age. Weight 2 will be the coefficient or multiplier for color. So we would then think that for age, that would be a larger number, larger emphasis or weighting that we would apply to age compared to color, which might be quite a small value because it doesn't contribute as much to the price of the vehicle. Sunroof weight 3, mileage, weight 4, alarm weight 5. So I'd expect weight 1.

00:03:32 Speaker 2

To be bigger than weight 2, maybe. Weight 3 might be bigger than weight 2, maybe weight 4 might be the biggest. Maybe mileage is going to be our biggest contributor to final target price. So for each input feature, X1 through X5 will have a corresponding weight that determines its importance in its contribution to the target price.

00:03:48 Speaker 2

So that means I could start to express this as a mathematical formula. F of X equals weight 1 * X one, X1 being age, weighting 2 * X two, color, weighting 3 * X to be the sunroof, weighting 4 * X four, the mileage and weighing

5 * X five. The presence or absence of an alarm plus a bias. Again bias thinking about our offset, how high or low we are in the graph, just to again adjust this to be a best fit.

00:04:08 Speaker 2

Ah, we now have the concept that a model is building a mathematical equation.

00:04:13 Speaker 2

And it's going to adjust these values to come up with that line in a multi dimensional space of best fit.

00:04:19 Speaker 2

So during training, those weights W one through W 5 and the bias are being optimized with each iteration during the training process. Remember, the training process is iterating over your training data set, ingesting into the algorithm, adjusting these parameters, measuring the loss function sum of the squared residuals, and figuring out, well, wait, one seemed to be right, but weight 2, no, that's not quite right. That's producing a higher loss function. Let's adjust that again. So it adjusts these values and whether to increase them or decrease them. It's using this method called gradient descent internally to figure out which direction to go in, in terms of tuning those.

00:04:49 Speaker 2

Parameters.

00:04:50 Speaker 2

So gradient descent is a very clever system. And the way I was once described to me was imagine that you were up on a hillside and you're trying to find the point, the lowest point on the hill. You can start stepping in One Direction and going, am I higher or lower? If you're higher, then change direction and go in the other direction. Am I lower? Yeah, I'm lower than I am. Keep stepping in that direction until I get to the lowest point. And what you're doing there is you're using a methodical approach to adjust the weightings on those parameters that get us to the lowest loss function.

00:05:15 Speaker 2

So at the end of this, what we're trying to do is get a line of best foot in a multi dimensional space. So finding the optimal values for weighting one through weighting 5 and B in order that when we present input features to that F of X function, it is able to calculate a value output. And that value output is your predicted host price or your predicted car sales price.

00:05:34 Speaker 2

So we can see already, ML is just math. And that's why whenever you're learning ML, you will always, you have to learn the ML math. And that is true to an extent. But you don't need to learn all the ML math. But we do need to understand what's happening so that we can tune the training process so it goes in the right direction.

00:05:51 Speaker 2

So our motto when it makes predictions is going to use that F of X formula.

00:05:56 Speaker 2

With the weights of being calculated during the training process to apply to the input data that it's not seen before. And that way it will generate your target variable output. In other words, your predicted price. Now during the training process, how does it know it's getting this right? Well, remember that during the training process, you have all the answers, your input data, including the target value where you know what price was achieved. Therefore, during training that we can come up with an F of X formula value with a set of weights and biases. And then we could take some samples that we know that we've got the answer for.

00:06:24 Speaker 2

And supply our model candidate with the input values and compare them to what the actual value was in our training data set.

00:06:30 Speaker 2

And say how far are we close? Are we pretty good? Did we? Did we get it by owner or within a percentage? And that's how the training process is able to iterate and improve upon itself. How close was it in terms of the prediction it created compared to the actual truth value that is in the training data? Now we do describe that as a function F of X compared to the actual Y value and squared. But again, it's just a mathematical way of expressing how far off are we from the actual predictive value.

00:06:55 Speaker 2

Now with that gradient descent method that the algorithm is using, it can then decide, well, we're quite far off. Therefore I need to adjust the weights for my different input features or adjust the bias. And it will make the determination as to which weights need to be adjusted and in which direction increase or decrease. And then it will try again and it will repeat calculating F of X with those new weights. And then it will come up with prediction and it will compare that to the actual. And that iterative process will happen over and over until it reaches the level of accuracy that you determine is required by your model.

00:07:23 Speaker 2

So to express it another way.

00:07:24 Speaker 2

For all of the weights that correspond to each of our input features and bias, the algorithm will start out with some initial parameters and maybe randomized or they may be default values for that particular algorithm. We'll start out with initial parameters. Then it will calculate what the F of X value would be based upon input data and those weights and biases. Then it would compare that to the actual target variable going far off where we not great, let's adjust again in the right direction and it would make changes. So here it's decided to increase weight one for feature X1. It decided to decrease the waiting for feature.

00:07:55 Speaker 2

It's decided to leave weight 3, weight 4 and weight 5 as is, and it's a decided to adjust the bias ever so slightly.

00:08:01 Speaker 2

Then it will calculate F of X again for an input sample and it will compare what the output predicted target variable is and it will compare that to the known truth from the training data. How are we more accurate or less accurate than we were? We go over and over and over into achieved a level of accuracy that we are happy with for our model.

00:08:19 Speaker 2

So now we have our model, we have our complex formula. So in this case here F of X equals the the adjusted waiting for feature X1, the adjusted waiting for feature X2 and so forth all the way through the remaining features.

00:08:30 Speaker 2

And that's it. It's math. So when you're training, what you're doing is algebra or under the hood, what's happening is gradient descent is adjusting the parameters or weights that correspond to your input features to come up with the best weighting so that the target variable can be predicted with an accuracy that meets your requirement. So that was a long one, but what did we see? We saw that we can train a model by using an off the shelf algorithm like Xgboost or Linear Learner, PCA or KNN. And we start with a data set where we have the input variables and we have.

00:09:00 Speaker 2

Market variable.

00:09:01 Speaker 2

Remember the target variable? Simply what we ultimately want to predict from our model. We have to host the model it's on some compute platform in order to be able to generate a prediction.

00:09:10 Speaker 2

That could be a virtual machine, could be a container physical server, or it could be Sagemaker itself. And that's the approach that we will be taking.

00:09:17 Speaker 2

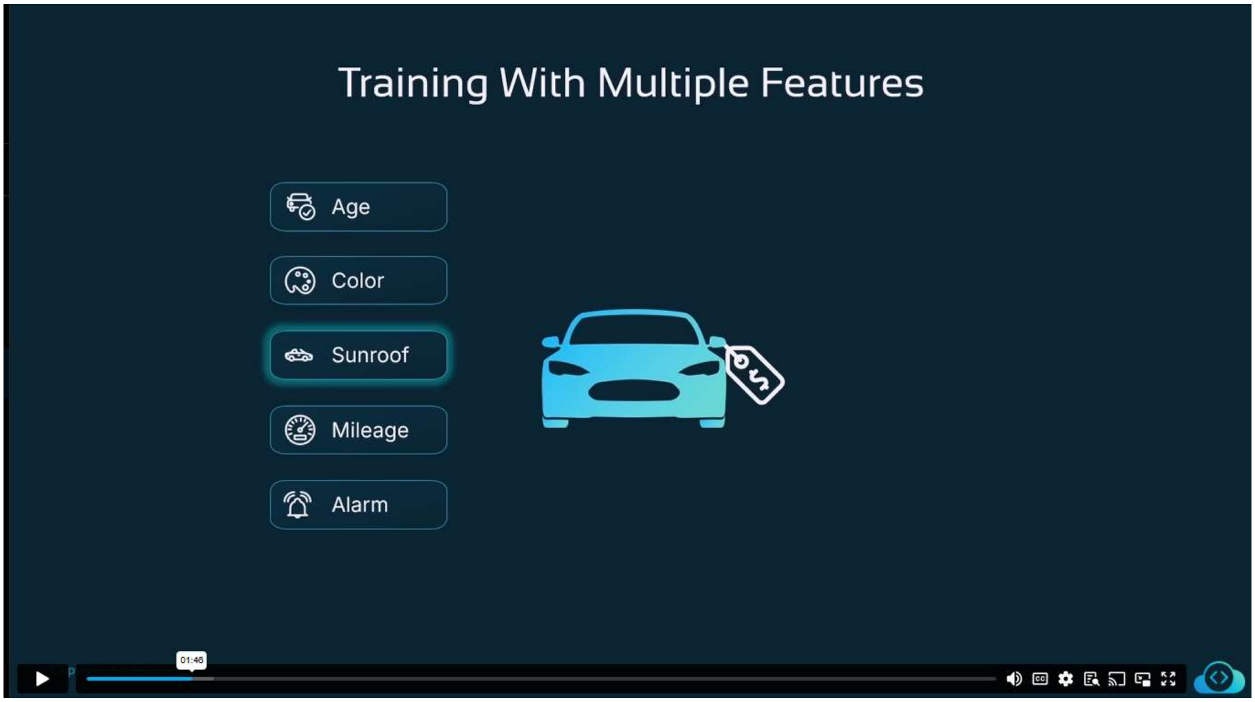
We have seen that when we present new data to our hosted model, we are providing the data in the same format that we provided the model during training, except we're only providing the input features or not providing the target value because we now want the model to predict the target value.

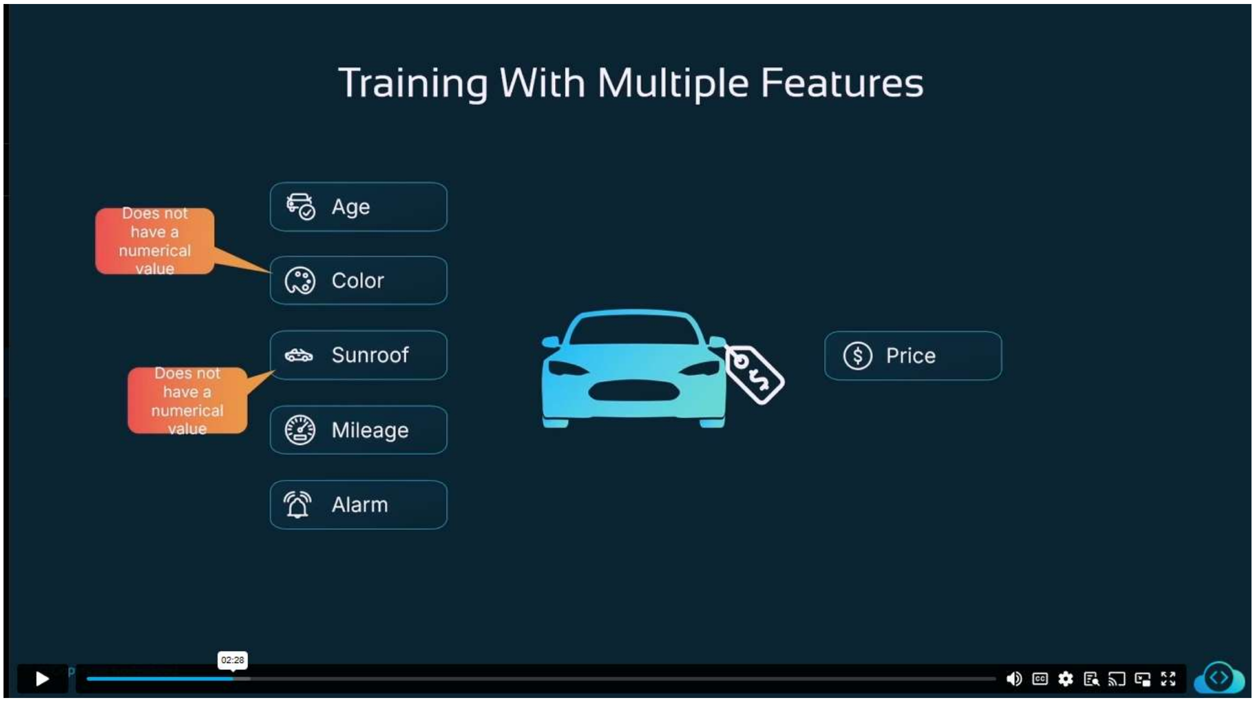
00:09:33 Speaker 2

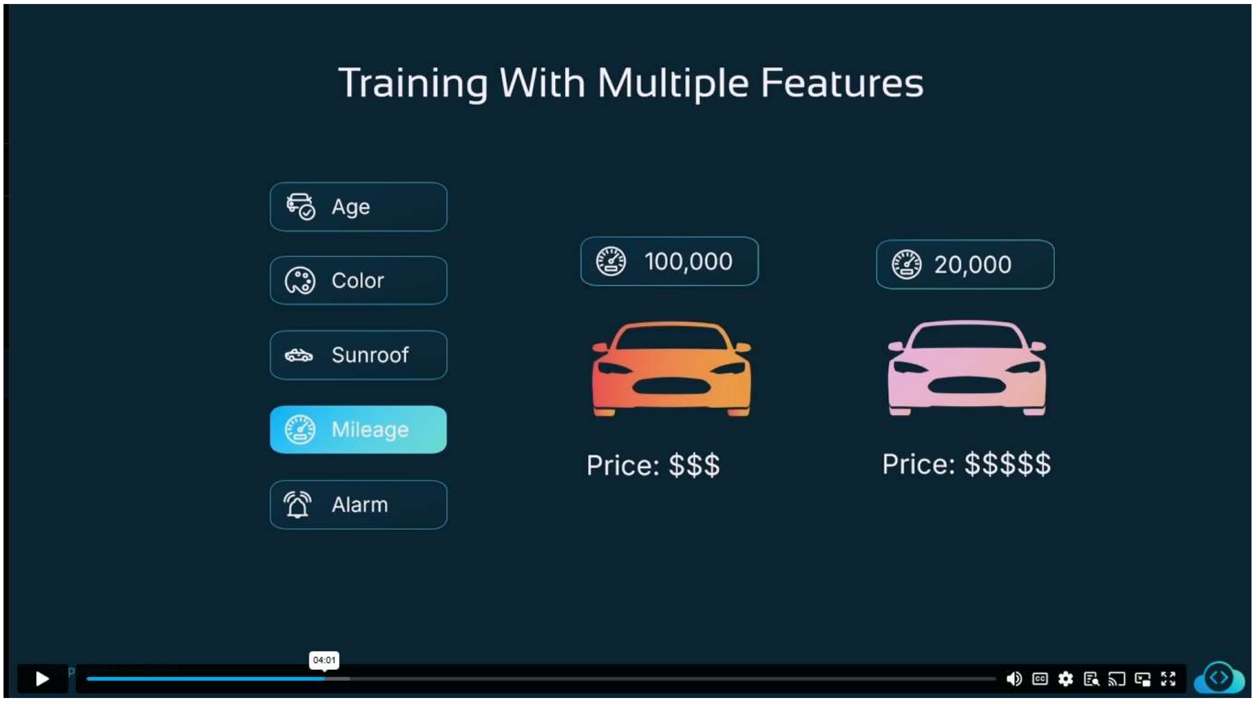
And we've seen that the way that the model actually generates those predictions is by using this essentially linear algebra built in that it is adjusted during the training process to apply different weightings to different input features. That's it, it's math. OK, the more we understand about the Mlmath, the better we can tune our models during training to do what we want. But you can see you don't need to be a linear algebra wizard. We just need to understand that there are weights and biases and they are being auto-adjusted during training. So if that wraps up this lesson, we should now have a better understanding of how ML works under the hood.

00:10:03 Speaker 2

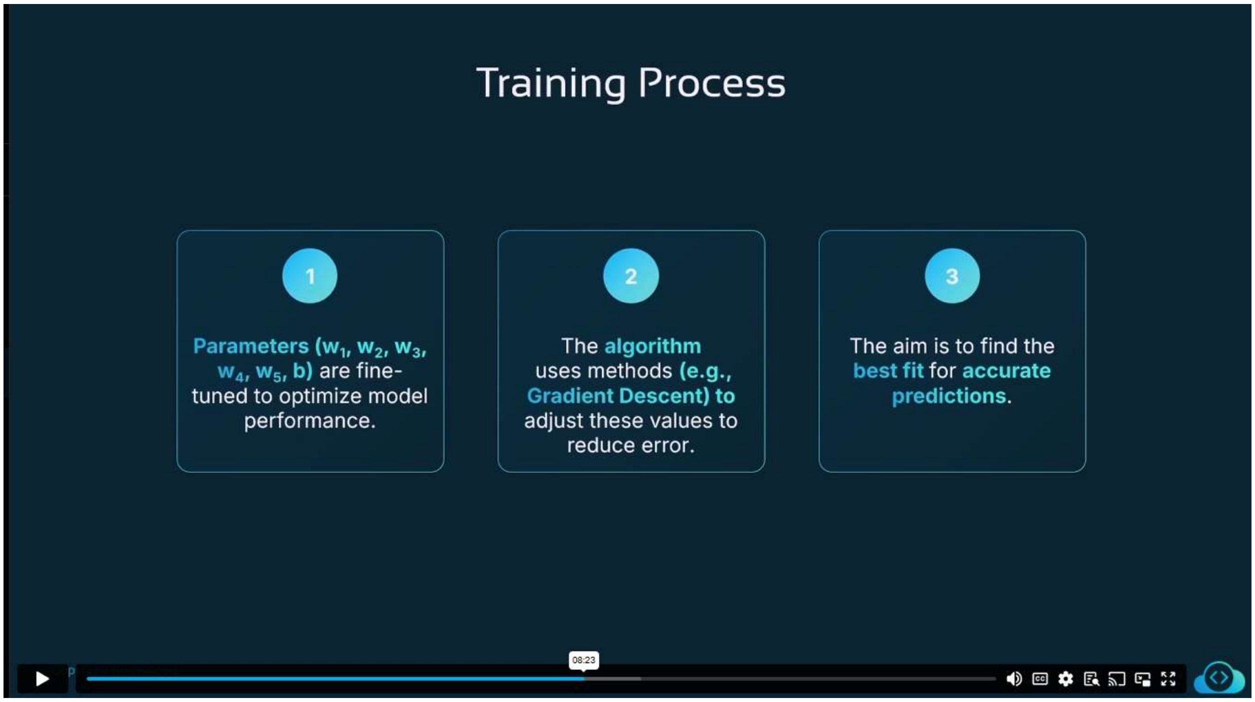
In our next lesson, we're going to look at something called the ML pipeline and understand the sequence of activities that go from the beginning of ideation of a machine learning problem right the way through to performing inference requests to a hosted model.

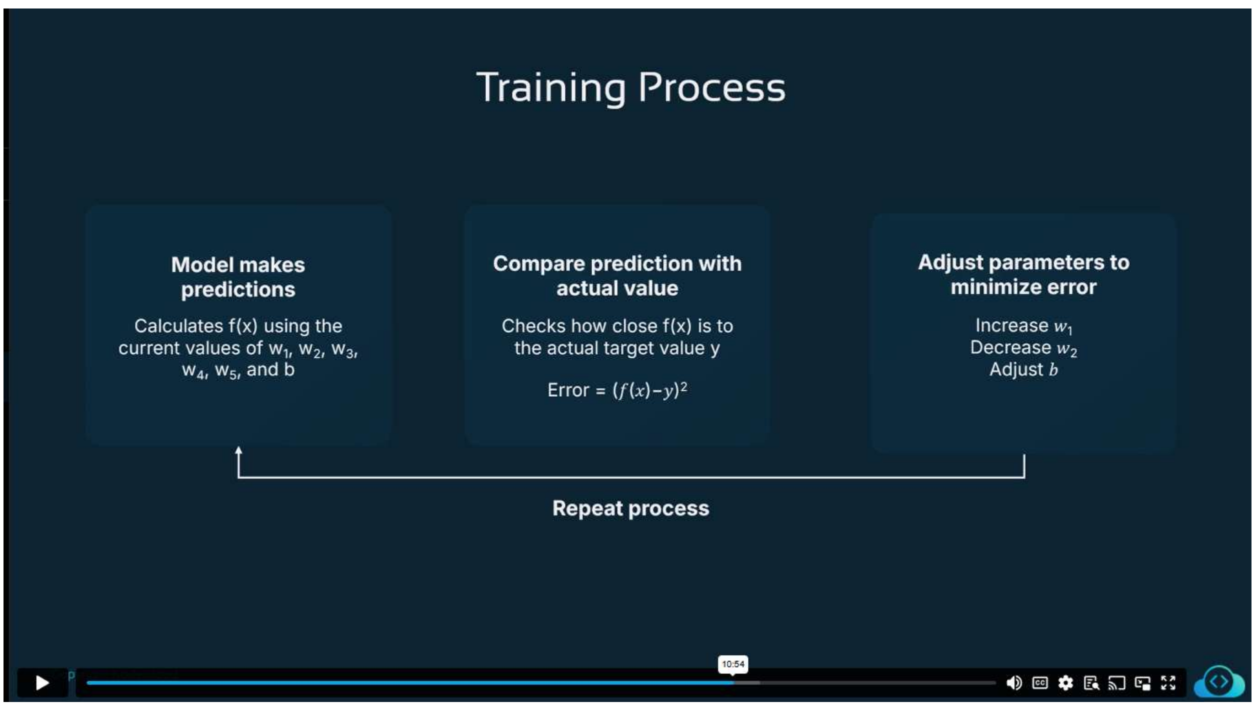

Training With Multiple Features
Age
Color
Sunroof
Mileage
Alarm
01:46

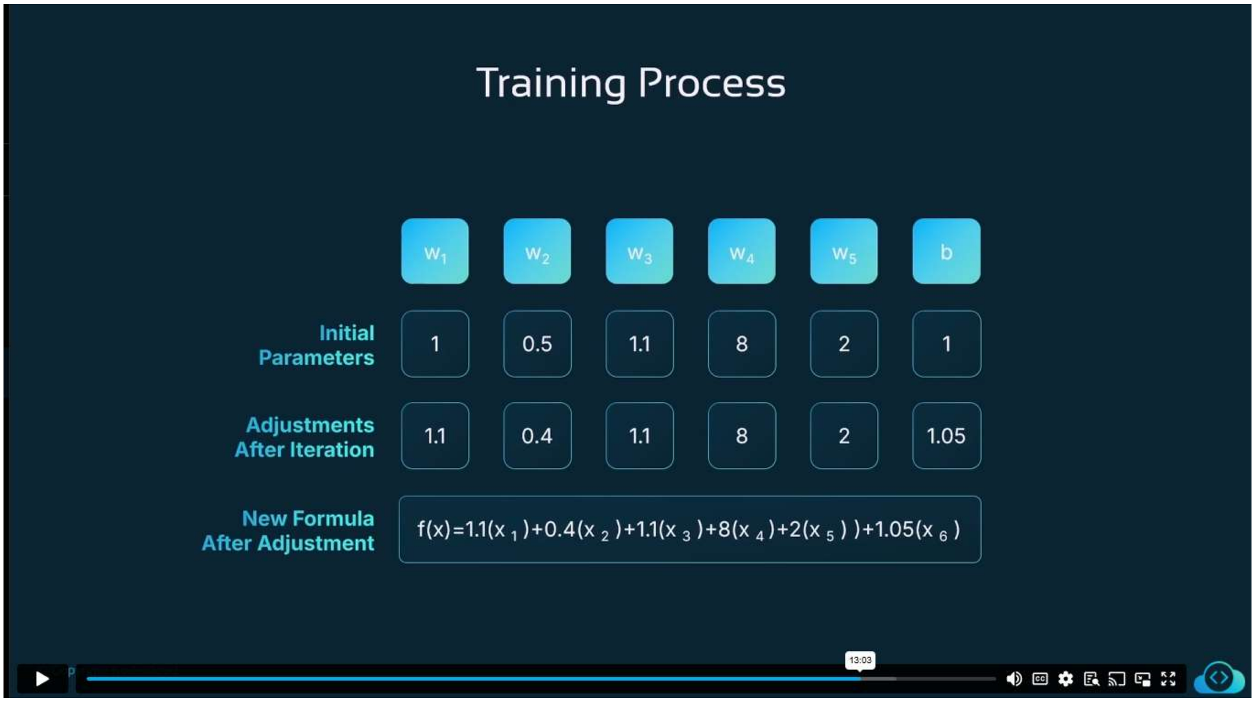

Training With Multiple Features
Does not have a numerical value
Age
Color
Does not have a numerical value
Sunroof
Mileage
Alarm
Price
02:28

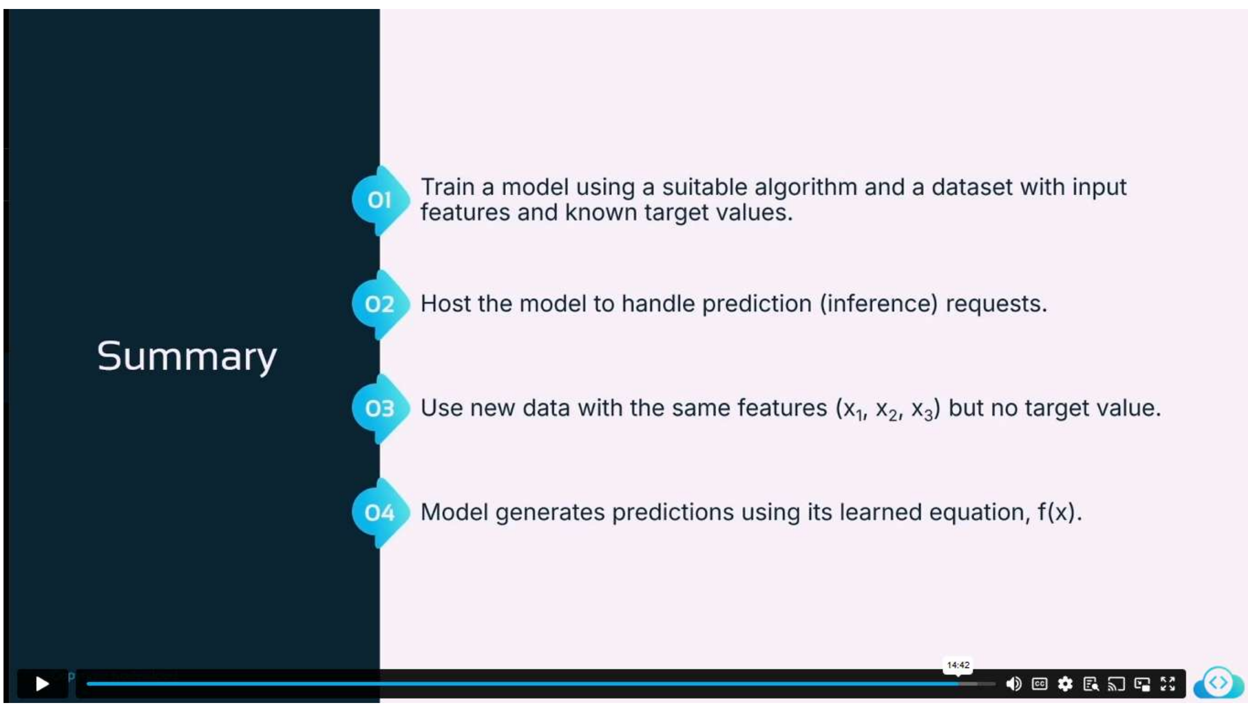

Training With Multiple Features
Age
Color
Sunroof
Mileage
Alarm
100,000
20,000
Price: $$$
Price: $$$$$
04:01


Training With Multiple Features
Age
Color
Sunroof
Mileage
Alarm
w_1
w_2
w_3
w_4
w_5
$f(x)=w_1x_1+w_2x_2+w_3x_3+w_4x_4+w_5x_5+b$
06:21


Training Process
1
Parameters (w1, w2, w3, w4, w5, b) are fine-tuned to optimize model performance.
2
The algorithm uses methods (e.g., Gradient Descent) to adjust these values to reduce error.
3
The aim is to find the best fit for accurate predictions.
08:23


Training Process
Model makes predictions
Calculates f(x) using the current values of w1, w2, w3, w4, w5, and b
Compare prediction with actual value
Checks how close f(x) is to the actual target value y
Error = (f(x)−y)²
Adjust parameters to minimize error
Increase w1
Decrease w2
Adjust b
Repeat process
10:54


Training Process
w_1
w_2
w_3
w_4
w_5
b
Initial Parameters
1
0.5
1.1
8
2
1
Adjustments After Iteration
1.1
0.4
1.1
8
2
1.05
New Formula After Adjustment
$f(x)=1.1(x_1)+0.4(x_2)+1.1(x_3)+8(x_4)+2(x_5))+1.05(x_6)$
13:03


Summary
01 Train a model using a suitable algorithm and a dataset with input features and known target values.
02 Host the model to handle prediction (inference) requests.
03 Use new data with the same features (x_1, x_2, x_3) but no target value.
04 Model generates predictions using its learned equation, f(x).
14:42